Dynamic Safety Under Uncertainty: A Control Barrier Function Approach

Thesis by
Ryan Kazuo Cosner

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy in Mechanical Engineering

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2025
Defended May 29, 2025

Ryan Kazuo Cosner
ORCID: 0000-0002-4035-1425

ACKNOWLEDGEMENTS

"What is control theory?" It's a question that I get asked a lot. I usually start by describing Newton's equations. Most people have heard of $\mathbf{F} = m\mathbf{a}$, so I'll explain that, in a sense, we try to guide or *control* $\mathbf{F}$. But that can sound overly domineering — as if we're forcing the robot, the system, or even the world to obey our orders. Someone once joked that the field must be full of "interesting personalities" based on the name alone. But that's not how I see it. To me, control theory isn't about fighting with the universe. Instead, it's the art of collaborating with it — listening to what it naturally wants to do, being one with the Force, mastering all four elements. The beauty of control theory, especially nonlinear control, lies not in overpowering the universe, but in working with it to guide our world's progression through time.

So my first acknowledgment is to the universe and everything in it: the laws of physics, the ground we stand on, the air we breathe. The universe flows through time in a beautiful dance, and it's amazing to be able to — for just a brief moment, in one small corner — take the lead.

Robotics, unlike control theory, is much easier to explain. Everyone has a clear image in their head of what a "robot" is. As long as the term "robot" has existed, people have dreamed of what robots might one day do. What's truly amazing is that we're achieving that dream with steady progress. Present day 2025 is starting to feel like science fiction: autonomous taxis can drive us home, flying robots rival firework shows, we can take robo-dogs on a walk, and humanoid robots are racing half-marathons. Being a roboticist keeps the spark of childhood curiosity burning strong. For me, the beauty of robotics is in building dreams.

So with that, I'd like to acknowledge all of the dreamers. The authors, screenwriters, and sci-fi fans who envision a better tomorrow, and the little kid in all of us who, whenever they see a robot, can't help but stop, watch, and imagine a new and exciting future.

Education has been core to my life — a perhaps unsurprising fact for someone who made it all the way to the 22$^{\text{nd}}$ grade. I believe in the deeply transformative power of education and a good mentor. Thank you to everyone along the way who has inspired me in my pursuit of knowledge and who has supported me along the way. Chris Miko, my 5$^{\text{th}}$ grade science teacher who helped me install my first Linux distro and inspired a deep sense of wonder and excitement. Jessica Bledsoe,

who coached my high school science olympiad team. My UC Berkeley mentors, George Anwar, Alice Agogino, Francesco Borelli, Shawn Shadden, Andy Packard, Alan Zhang, Drew Sabelhaus, and Douglas Hutchings. You all inspired me at every turn, supported me with compassion, and gave me the confidence to pursue a Ph.D. I wouldn't be the same person without each and every one of you. To the folks at Nvidia, Marco Pacone, Yuxiao Chen, Karen Leung, Shushant Veer, and Boris Ivanovic, thank you so much. The summer I spent working with you had an immeasurable impact on my research and career trajectory.

To my advisor, Dr. Aaron Ames, thank you so much for believing in my potential as a young graduate student and supporting me so far in my academic career. Thank you for dreaming big and giving me the freedom to pursue my academic interests; you made grad school a time where I was free to focus purely on research, and that is truly special.

To the rest of my Caltech mentors, thank you for everything you've done for me. To Joel Burdick, thanks for always being there as a stable rudder and guiding force. Your insights and perspectives on robotics, academia, and life in general were invaluable throughout my time as a grad student. To Günter Niemeyer, choosing to TA for your classes was among the most influential decisions of my life. Your passion, commitment, and care for your students is beyond inspiring. As a teacher, you've changed so many lives, including mine. I aspire to make the kind of educational impact that you've made on countless students. To Yisong Yue, thank you for all of your support. I really admire you as a scientist and mentor and it has been an honor to have your help throughout my Ph.D. To Katie Bouman, thank you so much for your mentorship especially at the beginning of my time at Caltech, I couldn't have done this without your guidance. To Preston Culbertson, thank you for all of your time and compassion. I'll truly value our months studying martingales in the basement of Gates-Thomas. Your support and mentorship have made me both a better researcher and a better person.

Also, thank you to the Caltech MCE staff for everything that you do, especially Mikaela Laite. None of this could've happened without you. Y también gracias a Leslie Linares; siempre disfrutaba nuestras conversaciones y apreciaba tu amistad y apoyo.

I couldn't have done any of this without the support of my family.

Caltech was the perfect place for me to pursue my Ph.D., not just because of the

renowned academics, small size, and amazing research facilities, but also because of the proximity to family. My parents met at the Caltech pool, my uncle and じいちゃん worked at a dental office just blocks away, and my cousin went to the community college across the street.

ばあちゃん, being able to spend time with you at Monte Vista was the highlight of my time in graduate school. Thank you for your undying support; I'll cherish our time together drinking tea and folding cranes forever.

Thank you to all of the Cosners. I feel proud to carry the torch as our family continues to contribute to the forefront of engineering, from Otto Ray with trains, Donald with rockets, Chris with satellites, and me with robots. To Jill and Jack McCaffrey, thank you so much for everything. Your support has meant the world to me and I can't wait to move to Boston to be closer to you. Dr. Jill, your empathy, understanding, and support helped keep me grounded and motivated through the rollercoaster of grad school and I can't thank you enough for that. And, Jill and Jack, thanks for being a second family to Bernadette in Boston. Your love means the world to us.

Thank you to all of the Itanos (板野) and Shimozonos (下園). Living at the family house in Altadena and hosting お正月 has been truly special.

To the Blashill-Muñoz family, thank you so much for being a second family to me and welcoming me in with open arms. Especially, Brett, Maria-Elena, Brianna, Juana Maria, and Ellie — I can't thank you enough for making me feel like part of your family.

Mom, Dad, and Ian, thanks for being my biggest cheerleaders and sources of stability through everything. Being able to drive across town to see you or step out of lab to grab lunch together has been so crucial to my time in grad school. You've kept me grounded and gave me the confidence to pursue my dreams. This is as much your achievement as it is mine.

Plus, a big thank you to Sky for being the goodest dog. And thanks to Roxy, my 1980 Mazda RX-7 that was lost in the Eaton Fire. Working on that car with my dad helped inspire my interest in engineering. Thanks for all the rides.

To my chosen family, thanks for making Caltech feel like home. To my labmates in the basement of Gates-Thomas thanks for being the best group of friends and roboticists that anyone could ask for. You all inspire me every day. A special thanks to my cohort of Noel Csomay-Shanklin, Amy Li, Wyatt Ubellacker, and Min Dai

— I truly couldn't have done it without you. To Andrew Taylor, I couldn't have asked for a better mentor. Your friendship and guidance means the world to me. Seriously, thank you for *everything*. Another special thanks to Gilbert Bahati and Ryan Bena. Gilbert, collaborating with you has truly been so fun. Your brilliance and curiosity never fail to inspire me. Ryan, I am consistently impressed by you in every dimension, your research and engineering prowess, but also your kindness and your mentorship. Working with you has been a great privilege.

To my other friends: My triathlon family, you made Caltech so special. Newton Nguyen, William Denman, Stephanie Breunig, Miles Chan, Angus Gruen, Matias Kagias, Leonie Schönbeck, Annette Böhme, Marianne Aellen, Carina Hausladen, Gullo Mastroserio, Rishav Mallick, Mattia Tagliavento, Lauren Conger, Marshall Yale, Nikhil Ranganathan, Han Zhang, Jessica Spake, Joy Shi, Apurva Badithela, Ethan Lin, Adreas Butler, Dylan King, Eitan Levin, and Chris Pukstza. Y'all rock. You've inspired me to do the craziest things (e.g., the lolipop of joy or waking up early every Friday to bike for two hours *before* work). Thanks for being the truly best group of people. My *Measure Zero* bandmates, Berthy Feng, Robbie Gray, and Piero Chiapina (and Mari Heimlich and Roo!), it's been an absolute pleasure performing with y'all. Thanks for all the great jams. To my Berkeley friends, Chris Elisondo, Joyce Luk, Miles Luhn, Jennifer Mathes, Gwen Gettle, and all the other Cal band folks, "Woah Bones!" I love you all and thanks for always being so supportive. To the Boot Troop, Kristyn Diamond Fudge, Clara de Guilhem de Lataillade, and Karson Yu. Y'all have been the best group of friends a person could ask for. Your friendship means the world to me. And to the other friends that I made throughout grad school, Angela Gao, Zach Singer, Roberto Treviño, Eric Ballouz, Tracy Lu, Stephanie O'Gara, Hardik Parwana, Devansh Agrawal — thanks for everything.

To Bernadette, thanks for being you <3. You're the love of my life, my sunshine on a cloudy day. Your insight, perspective, compassion, brilliance, and love inspire me every single day. *Amor de mis amores, tú eres mi cielo*.

ABSTRACT

Modern technological achievements in robotics, machine learning, and control promise an exciting future where autonomous robots are a useful part of everyday life, from automated manufacturing and driverless cars to robotic healthcare and autonomous delivery drones. However, as robots are deployed in increasingly complex, uncertain, and human-interactive environments, safety becomes paramount; we cannot deploy these systems at scale unless we are rigorously assured of their safety. Despite the capabilities of modern robotics, practical real-world safety is often achieved through conservative hardware designs, confining deployment regulations, or restrictive assumptions that severely limit a robot's capabilities.

The goal of this thesis is to develop methods for achieving dynamic safety: formal safety guarantees that preserve system performance and remain valid under uncertainty. To this end, this thesis advances the theory and practice of control barrier functions (CBFs), a leading framework for enforcing safety constraints on dynamical systems. While CBF-based methods offer strong theoretical guarantees, they do so by relying on several restrictive assumptions. Namely, they assume that the safety requirement and the system dynamics are compatible and that the dynamics model and state are perfectly known. These assumptions rarely hold in real-world settings and can result in false confidence and catastrophic safety failures when violated. This thesis addresses these gaps by systematically relaxing these assumptions and developing new theory to retain rigorous, deployable guarantees.

By leveraging structural properties of several relevant classes of system dynamics, Chapter 3 presents a myriad of constructive synthesis methods that make CBF design feasible for a wide range of robots. Chapter 4 then develops robust control methods that retain their rigorous safety guarantees in the presence of bounded dynamics and measurement uncertainty. However, despite the utility of these methods in guaranteeing safety, they often lead to highly conservative behavior that compromises system performance. Thus, to mitigate this conservatism, Chapter 5 integrates machine learning techniques to reduce uncertainty and determine desired levels of robustness. While this unification of machine learning techniques with safety-critical control may sacrifice formal guarantees, it enables safe and performant behavior. Moreover, the robust CBF framework developed in Chapter 4 provides a valuable degree of interpretability absent from typical end-to-end approaches.

Next, seeking a middle ground between conservative absolute guarantees and capable-but-heuristic methods, Chapter 6 adopts a probabilistic notion of safety that provides risk-based guarantees in the presence of unbounded disturbances. In particular, by illustrating the fundamental connection between DCBFs and supermartingales, it develops new theoretical guarantees and proposes several algorithms to achieve safety in the presence of stochastic uncertainty. Chapter 7 then deploys these methods on several complex systems experiencing significant uncertainty, including a quadrotor robot with a slung payload, a humanoid robot walking in unstructured environments, and multiple robots performing dynamic collision avoidance. To achieve this, we use generative modeling techniques to capture the necessary understanding of the uncertainty distribution. Here, we also forego the traditional CBF-based safety filter paradigm and show the performance and safety improvements that can be gained through the unification of CBFs and horizon-based methods such as model predictive control (MPC).

Together, the contributions of this thesis represent an advancement towards dynamic, safe, and capable robotic autonomy under uncertainty. The risk-aware, robust safety-critical control methods proposed here help close the gap between theoretical safety guarantees and the demands of real-world deployment.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] G. Bahati, R. K. Cosner, M. H. Cohen, R. M. Bena, and A. D. Ames, "Control barrier function synthesis for nonlinear systems with dual relative degree," *submitted to the 2025 IEEE 64th Conference on Decision and Control (CDC)*, 2025. [Online]. Available: https://arxiv.org/pdf/2504.00397,
RKC participated in the conception of the project, development of the mathematical framework, implementation of the hardware experiments, and writing of the manuscript.

[2] A. Capone, R. K. Cosner, A. D. Ames, and S. Hirche, "Safe online dynamics learning with initially unknown models and infeasible safety certificates," *to appear in the Proceedings of the 42nd International Conference on Machine Learning*, 2025. [Online]. Available: https://arxiv.org/abs/2311.02133,
RKC participated in the conception of the project, development of the mathematical framework, and implementation of the simulation examples.

[3] R. K. Cosner, R. M. Bena, and A. D. Ames, "Unified mpc+cbf control for performant safety: Mutual benefits and inherent robustness properties," *submitted to IEEE Transactions on Robotics*, 2025. [Online]. Available: http://www.rkcosner.com/assets/files/dodgeball_paper.pdf,
RKC led the conception, theoretical developments, and the writing of the manuscript. He collaborated in implementing the simulations and hardware experiments.

[4] L. Yang, B. Werner, R. K. Cosner, D. Fridovich-Keil, P. Culbertson, and A. Ames, "Shield: Safety on humanoids via cbfs in expectation on learned dynamics," *submitted to the 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025. [Online]. Available: https://arxiv.org/pdf/2505.11494,
RKC participated in the conception of the project, development of the probabilistic safety method, and writing of the manuscript.

[5] M. H. Cohen, R. K. Cosner, and A. D. Ames, "Constructive safety-critical control: Synthesizing control barrier functions for partially feedback linearizable systems," *IEEE Control Systems Letters*, pp. 2229–2234, 2024. DOI: 10.1109/LCSYS.2024.3412003,
RKC collaborated in developing the simulations, performed hardware experiments, and assisted in writing the manuscript.

[6] R. K. Cosner, P. Culbertson, and A. D. Ames, "Bounding stochastic safety: Leveraging freedman's inequality with discrete-time control barrier functions," *IEEE Control Systems Letters*, pp. 193 7 –1942, 2024. DOI: 10.1109/

LCSYS.2024.3409105,
RKC led the conception of the project, development of the theory, implementation of the experiments, and writing of the manuscript.

[7] R. K. Cosner, I. Sadalski, J. K. Woo, P. Culbertson, and A. D. Ames, "Generative modeling of residuals for real-time risk-sensitive safety with discrete-time control barrier functions," *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024. DOI: 10.1109/ICRA57147.2024.10611355,
RKC led the conception of the project and writing of the manuscript. He also participated in developing the mathematical framework, code, hardware platform, and in performing the experiments.

[8] P. Culbertson, R. K. Cosner, and A. D. Ames, "Input-to-state stability in probability," *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 5796–5803, 2023. DOI: 10.1109/CDC49753.2023.10383579,
RKC participated in the conception of the project, development of the main theory, and writing the manuscript.

[9] R. K. Cosner, Y. Chen, K. Leung, and M. Pavone, "Learning responsibility allocations for safe human-robot interaction with applications to autonomous driving," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9757–9763, 2023. DOI: 10.1109/ICRA48891.2023.10161112,
RKC participated in the conception of the project; developed the responsibility sharing paradigm; performed the machine learning, closed-loop and forensic tests; and led the writing of the manuscript.

[10] R. K. Cosner, P. Culbertson, A. J. Taylor, and A. D. Ames, "Robust safety under stochastic uncertainty with discrete-time control barrier functions," *Proceedings of Robotics: Science and Systems*, 2023. DOI: 10.15607/RSS.2023.XIX.084,
RKC participated in the conception of the project, development of the main algorithm, execution of the experiments, and writing of the manuscript.

[11] S. Veer, K. Leung, R. K. Cosner, Y. Chen, and M. Pavone, "Receding horizon planning with rule hierarchies for autonomous vehicles," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1507–1513, 2023. DOI: 10.1109/ICRA48891.2023.10160622,
RKC assisted in the development of the simulation experiments.

[12] D. R. Agrawal, H. Parwana, R. K. Cosner, U. Rosolia, A. D. Ames, and D. Panagou, "A constructive method for designing safe multirate controllers for differentially-flat systems," *IEEE Control Systems Letters*, vol. 6, pp. 2138–2143, 2022, ISSN: 2475-1456. DOI: 10.1109/LCSYS.2021.3136465,
RKC participated in the conception of the project, development of the mathematical framework, implementation of the quadruped experiments, and writing of the manuscript.

[13] K. Garg, R. K. Cosner, U. Rosolia, A. D. Ames, and D. Panagou, "Multi-rate control design under input constraints via fixed-time barrier functions," *IEEE Control Systems Letters*, vol. 6, pp. 608–613, 2022, ISSN: 2475-1456. DOI: 10.1109/LCSYS.2021.3084322,
RKC participated in the conception of the project, development of the mathematical theory, simulation implementation, and writing of the manuscript.

[14] T. G. Molnar, R. K. Cosner, A. W. Singletary, W. Ubellacker, and A. D. Ames, "Model-free safety-critical control for robotic systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 944–951, 2022, ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2021.3135569,
RKC participated in the conception of the project, development of the mathematical theory, and experimental implementation.

[15] R. K. Cosner, I. D. J. Rodriguez, T. G. Molnar, W. Ubellacker, Y. Yue, A. D. Ames, and K. L. Bouman, "Self-supervised online learning for safety-critical control using stereo vision," *2022 International Conference on Robotics and Automation (ICRA)*, pp. 11 487–11 493, 2022. DOI: 10.1109/ICRA46639.2022.9812183,
RKC participated in the conception of the project, development of the main algorithm, execution of the experiments, and writing of the manuscript.

[16] R. K. Cosner, M. Tucker, A. J. Taylor, K. Li, T. G. Molnar, W. Ubellacker, A. Alan, G. Orosz, Y. Yue, and A. D. Ames, "Safety-aware preference-based learning for safety-critical control," *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, Proceedings of Machine Learning Research, vol. 168, pp. 1020–1033, 2022. [Online]. Available: https://proceedings.mlr.press/v168/cosner22a.html,
RKC participated in the conception of the project, development of the robust safety theory, and implementation on the quadruped platform. He also developed the computer vision system and led the writing of the manuscript.

[17] R. K. Cosner, Y. Yue, and A. D. Ames, "End-to-end imitation learning with safety guarantees using control barrier functions," *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 5316–5322, 2022. DOI: 10.1109/CDC51059.2022.9993193,
RKC participated in the conception of the project, developed the mathematical framework, performed the experiments, and led the writing of the manuscript.

[18] A. J. Taylor, V. D. Dorobantu, R. K. Cosner, Y. Yue, and A. D. Ames, "Safety of sampled-data systems with control barrier functions via approximate discrete time models," *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 7127–7134, 2022. DOI: 10.1109/CDC51059.2022.9993226,
RKC participated in the conception of the project, development of the main theory and definitions, writing of the manuscript, and proved Theorem 4.

[19] N. Csomay-Shanklin, R. K. Cosner, M. Dai, A. J. Taylor, and A. D. Ames, "Episodic learning for safe bipedal locomotion with control barrier functions and projection-to-state safety," *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, vol. 144, pp. 1041–1053, 2021. [Online]. Available: https://proceedings.mlr.press/v144/csomay-shanklin21a.html,
R.K.C participated in the conception of the project, developed the learning method and associated code, aided in experiment execution, and participated in writing the manuscript.

[20] S. Dean, A. J. Taylor, R. K. Cosner, B. Recht, and A. D. Ames, "Guaranteeing safety of learned perception modules via measurement-robust control barrier functions," *Proceedings of the 2020 Conference on Robot Learning*, vol. 155, pp. 654–670, 2021. [Online]. Available: https://proceedings.mlr.press/v155/dean21a.html,
R.K.C. participated in the conception of the project, developed and performed the simulations, collaborated in the theoretical development of the main definition (Def. 3) and theorem (Thm. 2), and participated in writing the manuscript.

[21] R. K. Cosner, A. W. Singletary, A. J. Taylor, T. G. Molnar, K. L. Bouman, and A. D. Ames, "Measurement-robust control barrier functions: certainty in safety with uncertainty in state," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021. DOI: 10.1109/IROS51168.2021.9636584,
R.K.C. participated in the conception of the project, theory development, experimental implementation, and led the writing of the manuscript.

SUPPLEMENTAL VIDEO CONTENT

[1] Supplemental Video for "Guaranteeing safety of learned perception modules via measurement-robust control barrier functions." https://youtu.be/qMKKnhc6Je4.

[2] Supplemental Video for "Episodic learning for safe bipedal locomotion with control barrier functions and projection-to-state safety." https://youtu.be/NASRlnUIZ7U.

[3] Supplemental Video for "Measurement-robust control barrier functions: certainty in safety with uncertainty in state." https://youtu.be/xw4yy2XQEHw.

[4] Supplemental Video for "Model-free safety-critical control for robotic systems." https://youtu.be/_h8KTLsBGvw.

[5] Supplemental Video for "Safety-aware preference-based learning for safety-critical control." https://youtu.be/fEYkCY17xtY.

[6] Supplemental Video for "Self-supervised online learning for safety-critical control using stereo vision." https://youtu.be/k-9x6i_Z7fg.

[7] Supplemental Video for "Generative modeling of residuals for real-time risk-sensitive safety with discrete-time control barrier functions." https://youtu.be/QH6rO1KTXds.

[8] Supplemental Video for "Constructive safety-critical control: synthesizing control barrier functions for partially feedback linearizable systems." https://youtu.be/04Cu8ChoMAo.

[9] Supplemental Video for "Control barrier function synthesis for nonlinear systems with dual relative degree." https://youtu.be/sOU5oED-9Y4.

[10] Supplemental Video for "Unified MPC+CBF control for performant safety: mutual benefits and inherent robustness properties." https://youtu.be/VvBiguw0RMo.

[11] Supplemental Video for "SHIELD: Safety on humanoids via CBFs in expectation on learned dynamics." https://youtu.be/dJj4GBtH6Gw.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

Safety is a fundamental requirement for real-world robotic systems, spanning a wide array of modern application domains including autonomous vehicles and assistive devices to medical and industrial robotics [1]. As these technologies continue to advance, we stand at the brink of their widespread integration into society. Yet, despite their potential, the impact of robotics on everyday life remains limited, largely due to safety concerns.

Today, real-world safety is often achieved through conservative designs: kill-switches, mechanical limits, low-powered actuators, and strict environmental constraints like cordoned-off workspaces (e.g., robots in cages) [2]. While effective, these strategies constrain a robot's capabilities and limit its utility, rendering it unable to fluently engage with complex, dynamic, human-filled environments. Alternatively, a small but growing class of high performance systems, such as autonomous vehicles, are beginning to move beyond these limitations to achieve broad public impact by incorporating safety directly into their control algorithms. These systems aim to achieve *dynamic safety*: avoiding harm through real-time decision making without requiring mechanical or regulatory limitations on the robot or its environment. This goal of dynamic safety would allow the robot to retain its full actuation capabilities and operate freely in the real world.

In his fictional *Handbook of Robotics, 56th Edition, 2058 A.D.* [3], science fiction author Isaac Asimov envisioned a future where robots followed a hard-coded set of rules for dynamic safety. His "Three Laws of Robotics" offer a vision of this behavior-based understanding of safety:

Definition 1.0 ([3])**.** *Asimov's 3 Laws of Robotics are:*

1. *A robot may not injure a human being, or through inaction, allow a human being to come to harm.*
2. *A robot must obey the orders given to it by human beings except where such orders would conflict with the First Law.*

3. *A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.*

Unlike mechanical safety mechanisms or restrictive environmental limitations, Asimov's laws make the *robot* responsible for safety, requiring that it reason about the consequences of its actions in real time. His stories do not prescribe this safety framework as a panacea, but instead explore the deep philosophical complexities that arise when machines are expected to make ethical decisions. Crucially however, whatever the correct codified safety rules may be, Asimov makes the assumption that the robots can and will follow them.

This thesis takes the opposite approach. Rather than assuming compliance with a given set of laws, I ask: *if robots must satisfy a set of safety criteria, how can we endow them with the reasoning capabilities to actually do so?* I investigate how dynamic, algorithmic guarantees of safety can be rigorously achieved in practice, even under the significant uncertainty and complexity that come with real-world robot deployment. My goal is to develop the foundational tools that will allow robots to safely, performantly, and responsibily follow the safety regulations of the future.

Safety-Critical Control and Control Barrier Functions

To enforce this understanding of dynamic safety, this thesis will consider safety-critical approaches for synthesizing controllers that provide theoretical guarantees of safety.

In order to make these guarantees we must first define precisely what we mean by "safety." As is common in both control theory and robotics, we formalize our understanding of safety as the *forward invariance of a user-defined safe set*. That is, the user specifies a set of safety criteria (e.g., positions and configurations that avoid collisions) that define safe states, and we consider the closed-loop system "safe" as long as, starting in the safe set ensures that it stays there for all time.

Several frameworks have been developed to enforce this notion of forward invariance including Hamilton-Jacobi (HJ) backwards reachability [4], state constraints in model predictive control (MPC) [5], and control barrier functions (CBFs) [6]. This thesis will primarily focus on CBFs due to their relative computational simplicity and suitability real-time control. However, I will also explore connections to HJ and MPC approaches, especially with respect to how HJ methods study the important

problem of safe-set synthesis (a.k.a., viability kernel synthesis [7], [8]) which is often ignored in the context of CBFs and how MPC methods offer a principled framework for integrating performance objectives alongside safety constraints.

CBFs are a safety-critical control framework [9], [10] that were developed as a generalization of Lyapunov stability methods, particularly control Lyapunov functions (CLFs) [11], [12]. While Lyapunov methods study system stability through changes to a scalar quantity (e.g., the potential and kinetic energy of an inverted pendulum), CBFs analogously study system safety by encoding it as a scalar quantity that must remain non-negative. To guarantee this they then regulate how this safety value can change as a function of time, ultimately relying on Nagumo's theorem [13] which provides sufficient conditions for the forward invariance of sets.

Theoretically, CBFs are general and powerful tools for achieving guarantees of safety [6]. Furthermore, they display converse properties with forward invariance [14], where under certain conditions, the ability to keep a system safe (i.e., forward invariant) and the existence of a CBF are equivalent statements.

The practical utility of CBFs is, in part, enabled by their ability to create safety filters [15] which can be evaluated at rapid real-time speeds (e.g., hundreds to thousands of Hz). In particular, CBFs are often deployed in the form of a CBF quadratic program (CBF-QP) safety filter where a potentially unsafe action is filtered and minimally adjusted in a point-wise fashion to guarantee safe behavior, enabling their broad deployment as a modular "safety layer" in conjunction that adds theoretical guarantees to other control methods including CLFs [16] and learned controllers [17]. Furthermore, due to their underlying assumptions on the system dynamics, CBF methods generally yield computationally tractable convex optimization problems that often have closed-form solutions [18], facilitating their real-time deployment on robotic systems.

These advantages have led to successful hardware demonstrations across a diverse array of robotic platforms, including automobiles [19], robot swarms [20], aerial robots [21]–[23], robotic arms [24], quadrupedal robots [25], wheeled robots [26], and bipedal robots [27] among many others.

Robust Safety

Despite their demonstrated practical utility, CBF-based methods generally rely on mathematically convenient assumptions such as perfectly known dynamics, exact state measurements, and accurate perception models, that rarely hold in robotics.

As a result, their theoretical guarantees often fail to translate to systems effected by real-world uncertainty.

This thesis strives to bridge that gap, creating robust theoretical guarantees that remain valid in the presence of real-world uncertainty.

The robustness guarantees in this thesis can be divided into two main forms: (1) *worst-case robustness*, where safety is guaranteed in the presence of bounded uncertainties, and (2) *stochastic robustness*, where a probabilistic notion of safety is guaranteed in the presence of potentially unbounded uncertainties drawn from a probability distribution.

The first type of guarantee echos traditional control theoretic robustness frameworks [28]–[30] where safety is ensured for any possible disturbance up to a certain bound. In the context of CBFs this form of robustness is often encoded via the input-to-state safety (ISSf) property [31], [32], which describes how a bounded disturbance will cause a bounded expansion to the safe set. While these methods provide robust guarantees in the face of uncertainty, they often result in highly conservative behavior [15], [33], [34], limiting their practical utility. In contrast, the second framework embraces a more nuanced perspective that builds on stochastic stability theory [35] to handle a wider class of potential uncertainties and allow for tunable risk levels that can enable improved performance.

Performance Improvements

While safety filters and robustness guarantees can be useful in ensuring system safety, to successfully deploy robots at scale, we need them to be capable of simultaneously achieving safety and their performance goals. Unfortunately, robust safety-critical control methods often introduce significant conservatism to the system [32], which can assure system safety, but also render it ineffective. To overcome this, we turn to learning and horizon-based methods, both of which display significant capabilities in improving system performance.

Data-driven machine learning techniques have displayed immense capabilities in modern robotics [36]–[38]. In the context of CBF-based safety-critical control, learning methods have been used to improve upon safety-critical control techniques by reducing the dynamics uncertainty [39], [40], by generating safe sets from expert data [41], or coupling CBFs with reinforcement learning [42], [43], resulting in improved system performance alongside safety assurances. Although these methods generally compromise the theoretical safety guarantees, they can lead to high prac-

tical success, producing desirable closed-loop behaviors that safely achieve their performance goals.

Additionally, even without added robustifications, CBF-based safety filters can generate undesirable, stable equilibria (similar to artificial potential fields [44]) where there is deadlock between the safety constraint and the performance metric. This is caused by the continuity requirements of CBF-based controllers and the myopic, pointwise optimization performed by the safety filter [16]. Alternatively, horizon-based methods such as model predictive control [5] or reinforcement learning (via rollouts in the training process) [43] can produce significantly improved closed-loop behavior by optimizing for horizon-long performance given safety constraints. In particular, this thesis will focus on MPC methods due to its model-based interpretability. Prior work has combined MPC and CBFs in a layered architecture [25], [45] or by modifying the discrete-time constraint safety in the MPC [46], [47]. While these formulations exhibit improved performance when compared to standard CBF methods, limited work has been done to understand their robustness properties or the utility of the discrete-time CBF constraint in the MPC program.

1.1 Chapter Outline

The goal of this thesis is to fill these gaps in the literature and provide methods by which theoretical guarantees of safety can be made for robotic systems in a way that retains their performance and utility. To this end, I present the following results and their organization in this thesis.

Chapter 2 presents the necessary theoretical background required for the remainder of the thesis. This includes preliminary exposition regarding continuous time dynamical systems, stability, safety, Lyapunov functions, control barrier functions, and robustness frameworks like input-to-state stability and input-to-state safety.

Chapter 3 explores a critical assumption that the safety criteria for a system is compatible with its dynamics, i.e., that the safety constraints define a control invariant set. Since this assumption does not generally hold and is hard to verify in practice, I present methods for several classes of systems that enable the practical synthesis of CBFs and control invariant sets. This chapter includes contributions originally published in [48]–[50].

Chapter 4 introduces several robustifying methods that provide rigorous safety guarantees in the presence of bounded uncertainty that can stem from a variety of sources including measurement uncertainty, model mismatch, and actuation error.

This chapter includes contributions originally published in [51]–[54].

Chapter 5 presents several examples of how machine learning can be used to improve the performance capabilities beyond what is possible with the theoretical robustness guarantees of Chapter 4. In particular, I present methods for improving models in the case of uncertain dynamics and inaccurate sensor models and for learning intangible properties critical for safety like desired robustness levels and social responsibility. This chapter includes contributions originally published in [27], [53], [55], [56].

Chapter 6 adopts a different theoretical paradigm and reframes safety in the context of stochastic systems. I begin by providing the necessary theoretical background for this reframing and then present several methods for using CBFs to achieve probabilistic safety guarantees. This chapter includes contributions originally published in [57], [58].

Chapter 7 works to realize the probabilistic safety methods of the previous chapter on hardware. Generative modeling techniques and horizon-based optimization are employed to capture the real-world uncertainties and jointly optimize performance alongside safety. This chapter includes contributions originally published in [17], [59], [60].

Chapter 8 provides concluding remarks and proposes several exciting directions for future work.

Chapter 2

BACKGROUND

> "*A rose by any other name would smell as sweet*" - William Shakespeare, *Romeo and Juliet*
>
> "*Is a hotdog a sandwich? Is a calzone a dumpling?? Is a toaster a robot???*"

As Shakespeare points out, a name does not change the fundamental nature of a thing. And yet, we, as humans, love naming and categorizing things, from hotdogs to robots. In control theory, these names and definitions are critical in describing desirable properties and outlining entire subfields of study.

This chapter introduces the system models, properties, and mathematical definitions that form the foundation for this thesis. These names and concepts provide a conceptual lens through which we can understand the fundamental nature of safety and stability. Specifically, I present the theoretical background on continuous-time, deterministic nonlinear systems that underpins the contributions in the first half of this thesis in Chapters 3, 4, and 5. I begin with a review of nonlinear dynamics and a Lyapunov-based perspective on stability, control, and robustness, using control Lyapunov functions (CLFs) as tools to enforce convergence. I then build on this framework to introduce a control-theoretic notion of safety and present control barrier functions (CBFs) as a natural extension of CLFs that enable the synthesis of controllers with closed-loop safety guarantees.

2.1 Nonlinear Dynamics

We begin by considering control-affine dynamical systems of the form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$ is the system state and $\mathbf{u} \in \mathbb{R}^{n_u}$ is the control input[1]. The system dynamics depend affinely on $\mathbf{u}$, with *drift dynamics* $\mathbf{f} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ and *actuation*

[1] For ease of exposition, we present the concepts in this chapter while assuming the general state space $\mathbf{x} \in \mathbb{R}^{n_x}$ and unbounded inputs in $\mathbf{u} \in \mathbb{R}^{n_u}$. Future chapters will relax this assumption, including by considering a bounded input $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^{n_u}$.

matrix $\mathbf{g} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x \times n_u}$ that we assume are locally Lipschitz on $\mathbb{R}^{n_x}$. This control-affine structure is general and captures a wide range of real-world robotic systems; in particular, any system described by rigid-body Euler-Lagrange dynamics (3.5–3.6) can be written in this form.

Given a control-affine system in this form we seek to synthesize controllers $\mathbf{k} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ which render the following closed-loop dynamics safe:

Applying a locally Lipschitz continuous state-feedback controller $\mathbf{k} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ to the *open-loop* dynamics (2.1) creates the *closed-loop* system:

$$\dot{\mathbf{x}} = \mathbf{f}_{\text{cl}} \triangleq \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{k}(\mathbf{x}). \tag{2.2}$$

Since the functions $\mathbf{f}$, $\mathbf{g}$, and $\mathbf{k}$ are locally Lipschitz continuous on $\mathbb{R}^{n_x}$, the closed-loop dynamics $\mathbf{f}_{\text{cl}}$ are also locally Lipschitz. This then implies that, for any initial condition $\mathbf{x}_0 \in \mathbb{R}^{n_x}$, there exists a maximal time interval $I(\mathbf{x}_0) = [0, t_{\max}(\mathbf{x}_0))$ and a unique, continuously differentiable solution $\boldsymbol{\varphi} : I(\mathbf{x}_0) \to \mathbb{R}^{n_x}$ satisfying

$$\dot{\boldsymbol{\varphi}}(t) = \mathbf{f}(\boldsymbol{\varphi}(t)) + \mathbf{g}(\boldsymbol{\varphi}(t))\mathbf{k}(\boldsymbol{\varphi}(t)), \qquad \boldsymbol{\varphi}(0) = \mathbf{x}_0, \tag{2.3}$$

for all $t \in I(\mathbf{x}_0)$ [61]. Reflecting the behavior of physical robotic systems, we assume the system is *forward complete*[2], i.e., solutions exist for all $t \geq 0$ and $I(\mathbf{x}_0) = \mathbb{R}_{\geq 0}$ For notational convenience, we will often write $\mathbf{x}(t) = \boldsymbol{\varphi}(t)$ to denote the system trajectory.

2.2 Stability

Traditional work in nonlinear control has largely focused on synthesizing controllers that ensure the stability of these closed-loop systems. In this section, we provide a rigorous background on standard notions of stability and introduce Lyapunov-based methods for as tools for describing system stability and synthesizing stabilizing controllers. This background also will serve as the foundation for control barrier function (CBF)–based safety-critical control, which will later be introduced as a generalization of these Lyapunov-based stability methods.

Stability properties describe the regulation of the system to an equilibrium point $\mathbf{x}^*$, which is a point where, if the system starts there, it will remain there for all time:

Definition 2.1 (Equilibrium Point [61])**.** *A point* $\mathbf{x}^* \in \mathbb{R}^{n_x}$ *is an equilibrium point for the closed-loop system* (2.2) *if* $\dot{\mathbf{x}} = \mathbf{f}_{\text{cl}}(\mathbf{x}^*) = 0$.

[2]Forward completeness is not guaranteed for arbitrary nonlinear systems, but it typically holds for real-world robotic systems due to their inherently bounded inputs and well-behaved physical dynamics.

We can then define the exponential stability of an equilibrium point[3] $\mathbf{x}^*$ as:

Definition 2.2 (Exponential Stability [61]). *Let $\mathbf{x}^* = \mathbf{0} \in \mathbb{R}^{n_x}$ be an equilibrium point of the closed-loop system* (2.2)*. The closed-loop system* (2.2) *is said to be locally exponentially stable with respect to $\mathbf{x}^*$ if there exists positive constants $M, \lambda \in \mathbb{R}_{>0}$ and $\delta \in (0, \infty]$ such that:*

$$\|\mathbf{x}_0\| \leq \delta \implies \|\mathbf{x}(t)\| \leq M\|\mathbf{x}_0 - \mathbf{x}^*\|e^{-\lambda t} \tag{2.4}$$

for all $t \in \mathbb{R}_{\geq 0}$.

This definition captures the idea that the system state will converge, or "*stabilize*," to the equilibrium point over time.

While Definition 2.2 provides a concrete formulation of exponential stability, our goal is to adopt a more general framework that will serve as the foundation for our formal discussion of safety. To this end, we introduce class $\mathcal{K}$ and class $\mathcal{KL}$ comparison functions, which allow us to express stability and robustness properties in a more flexible and general form.

Definition 2.3 (Class $\mathcal{K}$ functions [61, Def. 4.2]). *A continuous function $\alpha : [0, a) \to [0, \infty)$ with $a \in \mathbb{R}_{>0}$ is said to belong to class $\mathcal{K}$ (denoted $\alpha \in \mathcal{K}$) if it is strictly increasing and $\alpha(0) = 0$. Additionally, it is said to belong to class $\mathcal{K}_\infty$ (denoted $\alpha \in \mathcal{K}_\infty$) if $a = \infty$ and $\alpha(r) \to \infty$ as $r \to \infty$.*

Class $\mathcal{K}$ functions display the useful properties of invertibility (the inverse of a Class $\mathcal{K}$ function exists and is also Class $\mathcal{K}$) and composability (the composition of two Class $\mathcal{K}$ functions is also Class $\mathcal{K}$), and a useful category of Class $\mathcal{K}_\infty$ functions is scaling by a positive constant, e.g., $\alpha(r) = \rho r$ for some $\rho > 0$.

We can now extend class $\mathcal{K}$ functions to include a notion of convergence, captured by class $\mathcal{KL}$ functions:

Definition 2.4 (Class $\mathcal{KL}$ functions [61, Def. 4.3]). *A continuous function $\beta : [0, a) \times [0, \infty) \to [0, \infty)$ with $a \in (0\infty)$ is said to belong to class $\mathcal{KL}$ (denoted $\beta \in \mathcal{KL}$) if, for each fixed $s \in [0, \infty)$, the mapping $r \mapsto \beta(r, s)$ belongs to class $\mathcal{K}$, and for each fixed r, the mapping $s \mapsto \beta(r, s)$ is decreasing and satisfies $\beta(r, s) \to 0$ as $s \to \infty$.*

[3] We can assume to be $\mathbf{x}^* = \mathbf{0}$ without loss of generality by translating the system.

These comparison functions allow us to define more general notions of stability:

Definition 2.5 (General Notions of Stability [61, Def. 4.5])**.** *The equilibrium point* $\mathbf{x}^* = \mathbf{0}$ *of the closed-loop system* (2.2) *is:*

Stable, if there exists $\alpha \in \mathcal{K}$ *and* $\delta \in (0, \infty]$ *such that:*

$$\|\mathbf{x}_0\| < \delta \implies \|\mathbf{x}(t)\| \leq \alpha(\|\mathbf{x}_0\|), \quad \forall t \geq 0, \tag{2.5}$$

Asymptotically stable, if there exists $\beta \in \mathcal{KL}$ *and* $\delta \in (0, \infty]$ *such that:*

$$\|\mathbf{x}_0\| < \delta \implies \|\mathbf{x}(t)\| \leq \beta(\|\mathbf{x}_0\|, t), \quad \forall t \geq 0, \tag{2.6}$$

Exponentially stable, if there exist constants $M, \lambda \in \mathbb{R}_{>0}$ *and* $\delta \in (0, \infty]$ *such that:*

$$\|\mathbf{x}_0\| < \delta \implies \|\mathbf{x}(t)\| \leq Me^{-\lambda t}\|\mathbf{x}_0\|, \quad \forall t \geq 0. \tag{2.7}$$

The exponential stability condition corresponds to the specific class $\mathcal{KL}$ function $\beta(r, s) = Me^{-\lambda s}r$ as in Definition 2.2.

These definitions formalize both the convergence of trajectories to equilibrium points and the invariance of bounded sets under the system dynamics. In the next section, we introduce Lyapunov methods as a tool for verifying and synthesizing controllers that achieve these forms of stability in the closed-loop system (2.2).

2.3 Lyapunov Stability

Next we introduce Lyapunov methods as a means of verifying these stability properties and synthesizing controllers that render the closed-loop system (2.2) stability.

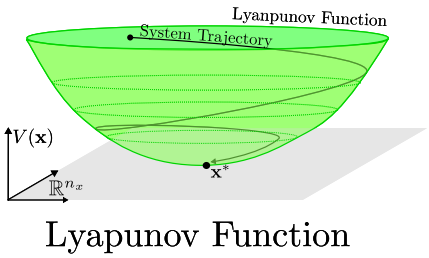


Figure 2.1. A visualization of a trajectory $\mathbf{x}(t)$ descending down the Lyapunov surface towards an equilibrium point.

A Lyapunov perspective on stability can be viewed as a generalization of classical energy-based arguments. In both cases, a scalar function is used to characterize the system's behavior over time. For example, an asymptotically stable mechanical system will dissipate potential and kinetic energy as it evolves. Analogously, an asymptotically stable system will admit a Lyapunov function that decreases over time along system trajectories.

Intuitively, convergence to an equilibrium point can be visualized as a system state "sliding down" with the Lyapunov function defining the height of the surface, as illustrated in Figure 2.1. While this energy-like metaphor provides useful intuition, we will now formalize the conditions under which such functions certify stability.

Theorem 2.6 (Lyapunov Stability [61, Thm.s 4.9 and 4.10])**.** *Let* $\mathbf{x}^* = \mathbf{0}$ *be an equilibrium point for the closed-loop system* (2.2) *and let* $V : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ *be a continuously differentiable function satisfying:*

$$\alpha_1(\|\mathbf{x}\|) \leq V(\mathbf{x}) \leq \alpha_2(\|\mathbf{x}\|), \tag{2.8}$$

$$\dot{V}(\mathbf{x}) = \underbrace{L_{\mathbf{f}}V(\mathbf{x})}_{\frac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x})} + \underbrace{L_{\mathbf{g}}V(\mathbf{x})}_{\frac{\partial V}{\partial \mathbf{x}}\mathbf{g}(\mathbf{x})}\mathbf{k}(\mathbf{x}) \leq -\alpha_3(\|\mathbf{x}\|) \tag{2.9}$$

for all $\mathbf{x} \in \mathbb{R}^{n_x}$ *where* $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{K}_\infty$*, then* $\mathbf{x}^*$ *is asymptotically stable, and if* $\alpha_i(\|\mathbf{x}\|) = k_i\|\mathbf{x}\|^c$ *for* $k_i, c > 0$ *and* $i \in \{1, 2, 3\}$*, then* $\mathbf{x}^*$ *is exponentially stable.*

Here we use the Lie derivative notation $L_{\mathbf{f}}V(\mathbf{x}) \triangleq \frac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x})$ and $L_{\mathbf{g}}V(\mathbf{x}) \triangleq \frac{\partial V}{\partial \mathbf{x}}\mathbf{g}(\mathbf{x})$ which capture the interaction of the system drift and actuator dynamics with the surface defined by V.

This stability theorem also comes with a converse result that establishes the existence of Lyapunov functions that reflect the appropriate stability properties of a system under relatively general regularity assumptions [61, Thms. 4.14 and 4.16].

Importantly, Lyapunov methods are not just a tool for analyzing the stability of closed-loop systems, but they can also be used for control synthesis. In particular, this takes the form of the *control Lyapunov function* (CLF) which were first introduced in [11]:

Definition 2.7 (Control Lyapunov Function[4] (CLF) [62])**.** *For the nonlinear, control-affine system* (2.1)*, a control Lyapunov function (CLF) is a continuously differentiable function* $V : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ *satisfying the following conditions:*

$$k_1\|\mathbf{x}\|^2 \leq V(\mathbf{x}) \leq k_2\|\mathbf{x}\|^2, \tag{2.10}$$

$$\inf_{\mathbf{u}\in\mathbb{R}^{n_u}} L_{\mathbf{f}}V(\mathbf{x}) + L_{\mathbf{g}}V(\mathbf{x})\mathbf{u} < -k_3\|\mathbf{x}\|^2 \tag{2.11}$$

for all $\mathbf{x} \in \mathbb{R}^{n_x} \setminus \{\mathbf{x}^*\}$ *with positive constants* $k_1, k_2, k_3 \in \mathbb{R}_{>0}$.

[4]We note that Def. 2.7 were originally called "exponentially stabilizing CLFs" due to the exponential stability that they enforce. While alternative constraints can be used to establish stability or asymptotic stability using CLFs, we will focus on the exponential case with $c = 2$ due to its simplicity and broad utility.

Unlike the previous stability notions which apply to the closed-loop system (2.2), Definition 2.7 applies to the *open-loop* system, enabling constructive controller design[5]. In this way, CLFs serve as functions that signify whether a system *could* be stabilized. In particular, the inequality in (2.11) defines a point-wise set of exponentially stabilizing control actions at each state $\mathbf{x} \in \mathbb{R}^{n_x}$ defined as:

$$\mathscr{K}_{\text{CLF}}(\mathbf{x}) \triangleq \{\mathbf{u} \in \mathbb{R}^{n_u} \mid L_{\mathbf{f}}V(\mathbf{x}) + L_{\mathbf{g}}V(\mathbf{x})\mathbf{u} \leq -k_3\|\mathbf{x}\|^2\}. \tag{2.12}$$

This set is nonempty at every state $\mathbf{x} \in \mathbb{R}^{n_x}$ if V is a valid CLF. We can check if this set is empty by verifying that the following implication holds:

$$\|L_{\mathbf{g}}V(\mathbf{x})\| = 0 \implies L_{\mathbf{f}}V(\mathbf{x}) \leq -k_3\|\mathbf{x}\|^2, \tag{2.13}$$

i.e., if the control input has no immediate influence on the Lyapunov function, then the drift dynamics must naturally satisfy the exponential decay condition.

We can now use the notion of CLFs to stabilize the open-loop system (2.1) by choosing locally Lipschitz controllers that select inputs from the stabilizing set $\mathcal{K}_{\text{CLF}}(\mathbf{x})$ at each $\mathbf{x} \in \mathbb{R}^{n_x}$:

Theorem 2.8 (CLF Stability [18])**.** *If there exists a CLF for the open-loop system* (2.1)*, then for every locally Lipschitz continuous feedback controller* $\mathbf{k} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ *that satisfies* $\mathbf{k}(\mathbf{x}) \in \mathscr{K}_{\text{CLF}}(\mathbf{x})$ *for all* $\mathbf{x} \in \mathbb{R}^{n_x}$*, the origin* $\mathbf{x}^*$ *of the closed-loop system* (2.2) *is exponentially stable.*

While CLFs signal the existence of a control action, they do not explicitly choose that action from $\mathscr{K}_{\text{CLF}}(\mathbf{x})$. One framework for choosing a stabilizing control input is the CLF-QP controller which selects inputs from $\mathscr{K}(\mathbf{x})$ that minimize the input norm in a point-wise fashion through the following constrained optimization problem called a CLF Quadratic Program (CLF-QP) with some $k_3 > 0$:

$$\mathbf{k}_{\text{CLF-QP}}(\mathbf{x}) = \underset{\mathbf{u}\in\mathbb{R}^{n_u}}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{u}\|_2^2 \tag{2.14}$$

$$\text{s.t.} \quad L_{\mathbf{f}}V(\mathbf{x}) + L_{\mathbf{g}}V(\mathbf{x})\mathbf{u} \leq -k_3V(\mathbf{x}). \tag{2.15}$$

Assuming unbounded inputs, this controller enforces the Lyapunov stability condition directly as a constraint and is itself is a convex quadratic program [64] with the

[5]Definition 2.7 also differs from the function V in Theorem 2.6 in that the second inequality (2.11) is now strict. This reflects the original definition in [11, Thm. 4.1] and more recent definitions as in [63, Def. 1]. This strict inequality ensures that controllers synthesized from the CLF can exhibit desired local Lipschitz continuity properties [63, Thm. 1].

following simple closed-form solution:

$$\mathbf{k}_{\text{CLF-QP}}(\mathbf{x}) = \begin{cases} 0, & L_{\mathbf{f}}V(\mathbf{x}) \leq -k_3\|\mathbf{x}\|^2, \\ -\frac{L_{\mathbf{g}}V(\mathbf{x})^\top}{\|L_{\mathbf{g}}V(\mathbf{x})\|^2}(L_{\mathbf{f}}V(\mathbf{x}) + k_3\|\mathbf{x}\|^2), & L_{\mathbf{f}}V(\mathbf{x}) > -k_3\|\mathbf{x}\|^2. \end{cases} \tag{2.16}$$

Robust Stability

Systems that are asymptotic or exponentially stable in the sense of Lyapunov also display favorable robustness properties. In particular, they are input-to-state stable (ISS), a property originally introduced in [65] that characterizes the how stability degrades in the presence of uncertainty.

To discuss robust stability we now consider systems with disturbances $\mathbf{d} : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n_d}$ where the disturbance enters the system in one of two forms:

$$\underline{\text{Matched Disturbance}}: \qquad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})(\mathbf{k}(\mathbf{x}) + \mathbf{d}(t)), \tag{2.17}$$

$$\underline{\text{Unmatched Disturbance}}: \qquad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{k}(\mathbf{x}) + \mathbf{d}(t). \tag{2.18}$$

A matched disturbance is one that *matches* the input insofar as it can be treated as an additive disturbance to the input and can be directly negated directly if it is known. Alternatively, unmatched disturbances are generally harder to overcome as the input is added directly to the drift dynamics and may not align with the actuatable directions of $\mathbf{g}(\mathbf{x})$.

Since $\mathbf{d}(t)$ is not known exactly, we assume that instead we know a worst-case upper bound on $\|\mathbf{d}\|$ that holds for all time. This is generally a conservative upper-bound, but can be used to make guarantees of robustness even when the disturbance behaves adversarially. We characterize the bound on the signal $\mathbf{d}$ using its essential supremum over timewhich we assume is bounded uniformly by some value $\overline{d} \in \mathbb{R}_{\geq 0}$:

$$\|\mathbf{d}\|_\infty \triangleq \operatorname*{ess\,sup}_{t \in \mathbb{R}_{\geq 0}} \|\mathbf{d}(t)\| \leq \overline{d}. \tag{2.19}$$

Given this bounded $\mathbf{d}$, it is generally impossible to choose a single input that ensures the convergence of the system to an equilibrium point regardless of which value $\mathbf{d}$ takes on. Instead we consider a notion of how this asymptotic or exponential stability can degrade as the bound $\overline{d}$ grows. This provides a very powerful paradigm for understanding how real-world uncertainties affect a closed-loop system.

Definition 2.9 (Input-to-State Stability (ISS) [29])**.** *Let* $\mathbf{x}^* = \mathbf{0}$ *be an equilibrium point of the undisturbed closed-loop system* (2.2)*. The closed-loop system with a*

matched (2.17) *or unmatched* (2.18) *disturbance is said to be Input-to-State Stable (ISS) with respect to* $\mathbf{x}^*$ *if there exists an* $a \in \mathbb{R}_{>0}$, $\beta \in \mathcal{KL}$ *and* $\gamma \in \mathcal{K}$ *such that:*

$$\|\mathbf{x}_0\| < a \implies \|\mathbf{x}(t)\| \leq \beta(\|\mathbf{x}_0\|, t) + \gamma\left(\overline{d}\right) \tag{2.20}$$

for all $t \geq 0$ *and disturbance signals* $\mathbf{d} : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n_d}$ *that are piecewise continuous on* $\mathbb{R}_{\geq 0}$ *and satisfy* $\|\mathbf{d}\|_\infty \leq \overline{d}$.

While ISS does not guarantee the stability of $\mathbf{x}^*$ in the present of disturbances, it does guarantee the attractivity and forward invariance of a region of the origin whose size grows monotonically with the disturbance bound.

We can also use the Lyapunov framework to prove that a system is ISS:

Theorem 2.10 (Lyapunov Characterization of ISS [65, Def. 2.2, Thm. 1])**.** *Let* $\mathbf{x}^* = \mathbf{0}$ *be an equilibrium point of the closed-loop system* (2.17) *(or* (2.18)*) with bounded uncertainty* $\|\mathbf{d}\|_\infty < \overline{d}$ *for some* $\overline{d} \in \mathbb{R}_{\geq 0}$. *If there exists a continuously differentiable function* $V : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ *that satisfies:*

$$\alpha_1(\|\mathbf{x}\|) \leq V(\mathbf{x}) \leq \alpha_2(\|\mathbf{x}\|), \tag{2.21}$$

$$\|\mathbf{x}\| \geq \alpha_4(\overline{d}) \implies \dot{V}(\mathbf{x}) \leq -\alpha_3(\|\mathbf{x}\|), \tag{2.22}$$

for all $\mathbf{x} \in \mathbb{R}^{n_x}$, *where* $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathcal{K}$, *then the closed-loop system* (2.17) *(or* (2.18)*) is ISS.*

This is a very useful property in capturing the natural robustness of stable systems. For example, exponentially stable systems exhibit the ISS property when exposed to unmatched disturbances:

Proposition 2.11. *If the undisturbed closed-loop system* (2.2) *is exponentially stable to* $\mathbf{x}^* = \mathbf{0}$ *and has continuously differentiable dynamics* $\mathbf{f}_{\text{cl}}$ *with a bounded Jacobian, then system* (2.18) *with an unmatched disturbance* $\mathbf{d} : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n_x}$ *where* $\|\mathbf{d}\|_\infty \leq \overline{d}$ *for some* $\overline{d} \in \mathbb{R}_{\geq 0}$ *is ISS.*

Proof. The exponential stability of (2.2) and the assumptions on $\mathbf{f}_{\text{cl}}$ ensure that an exponential Lyapunov function exists via the converse Lyapunov theorem [61, Thm. 4.14]. This Lyapunov function ensures the bounds:

$$\dot{V}(\mathbf{x}) = L_{\mathbf{f}}V(\mathbf{x}) + L_{\mathbf{g}}V(\mathbf{x})\mathbf{k}(\mathbf{x}) + \frac{\partial V}{\partial \mathbf{x}}\mathbf{d}(t) \leq -k_3\|\mathbf{x}\|^2 + \frac{\partial V}{\partial \mathbf{x}}\mathbf{d}(t), \tag{2.23}$$

$$\leq -k_3\|\mathbf{x}\|^2 + \left\|\frac{\partial V}{\partial \mathbf{x}}\mathbf{d}(t)\right\| \leq -k_3\|\mathbf{x}\|^2 + \left\|\frac{\partial V}{\partial \mathbf{x}}\right\| \|\mathbf{d}(t)\|, \tag{2.24}$$

$$\leq -k_3\|\mathbf{x}\|^2 + k_4\|\mathbf{x}\|\|\mathbf{d}(t)\| \leq -k_3\|\mathbf{x}\|^2 + k_4\|\mathbf{x}\|\overline{d}, \tag{2.25}$$

for $k_3, k_4 > 0$. Using this inequality, we find that: $\|\mathbf{x}\| \geq \frac{2k_4}{k_3}\overline{d} \implies \dot{V}(\mathbf{x}) \leq -\frac{k_3}{2}\|\mathbf{x}\|^2$ where Theorem 2.10 establishes that this system is ISS. □

In line (2.25) of the proof a portion of the exponentially stability of the nominal system $-k_3\|\mathbf{x}\|^2$ can be used to cancel the disturbance $k_4\|\mathbf{x}\|\overline{d}$. Thus, the naturally stability of the system provides robustness without requiring any knowledge of the disturbance itself.

Alternatively, for system systems with matched disturbances (2.17), we can be actively reduce the effect of the disturbance by tightening the CLF constraint as in the following theorem, similar to the use of ISS-CLFs for controller design in [66, Thm. 2]. In this paper we will consider the following notion of ISS-CLFs which reflects the notion of ISSf-CBFs to be defined in the next section.

Definition 2.12 (Input-to-State Stable Control Lyapunov Functions (ISS-CLFs))**.** *For the nonlinear, control affine system* (2.17) *where* $\mathbf{f}(\mathbf{0}) = \mathbf{0}$*, an input-to-state stable CLF (ISS-CLF) is a function* $V : \mathbb{R}^{n_x} \to \mathbb{R}$ *satisfying the following conditions:*

$$k_1\|\mathbf{x}\|^2 \leq V(\mathbf{x}) \leq k_2\|\mathbf{x}\|^2, \tag{2.26}$$

$$\inf_{\mathbf{u}\in\mathbb{R}^{n_u}} L_{\mathbf{f}}V(\mathbf{x}) + L_{\mathbf{g}}V(\mathbf{x})\mathbf{u} + \frac{1}{\epsilon}\|L_{\mathbf{g}}V(\mathbf{x})\|^2 < -k_3\|\mathbf{x}\|^2 \tag{2.27}$$

for all $\mathbf{x} \in \mathbb{R}^{n_x}$ *and positive constants* $k_1, k_2, k_3, \epsilon > 0$*.*

Here the standard CLF constraint (2.11) is tightened by $\frac{1}{\epsilon}\|L_{\mathbf{g}}V(\mathbf{x})\|^2$. We note that since $\mathbf{u}$ is unbounded in (2.11) and (2.27), any CLF must also be an ISSf-CLF.

The addition of this robustifying term $\frac{1}{\epsilon}\|L_{\mathbf{g}}V(\mathbf{x})\|^2$ results in the following property where the effect of the disturbance can be reduced via the choice of ϵ:

Theorem 2.13 (ISS-CLF [67, Thm. 13])**.** *If* V *is an ISS-CLF for system* (2.17) *with a matched disturbance, and* $\mathbf{k} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_m}$ *is a locally-Lipschitz state feedback controller satisfying the ISS-CLF constraint* (2.27)*, then the system satisfies* $\|\mathbf{x}(t)\| \leq M\|\mathbf{x}_0\|e^{-\lambda t} + \frac{\epsilon\overline{d}^2}{4}$.

These robustness properties expand a stable equilibrium point to a stable set and it is precisely this set stability/invariance from which we will construct our understanding of safety.

2.4 Forward Invariance and Safety

To define safety, we start with the human-understandable definition:

A system is safe if it does not cause harm to itself or its environment.

In order to encode this as a mathematical statement, we consider a region of the state space that satisfies this safety requirement, i.e.,

$$\mathbf{x} \in \mathcal{C}_0 \subset \mathbb{R}^{n_x} \implies \text{the system is not currently harming itself or its environment.} \tag{2.28}$$

Examples of safety criteria which can be used to define $\mathcal{C}_0$ include things like geo-fences for racing drones [22], foot-placement for walking robots traversing stepping stones [25], [27], or collision avoidance [17].

In general, the field of safety-critical control seeks to ensure that the system never exists this user-defined set $\mathcal{C}_0$. In other words, we seek to ensure the *forward-invariance* of $\mathcal{C}_0$.

Definition 2.14 (Forward-Invariance and Safety [6, Def. 1])**.** *A set $\mathcal{C}_0 \subset \mathbb{R}^{n_x}$ is forward-invariant for the closed-loop dynamics* (2.2) *if $\mathbf{x}_0 \in \mathcal{C}_0$ implies that $\mathbf{x}(t) \in \mathcal{C}_0$ for all $t \geq 0$. The system* (2.2) *is said to be "safe" with respect to the set $\mathcal{C}_0$ if the set $\mathcal{C}_0$ is forward invariant.*

Unfortunately, for general nonlinear systems it may be impossible to render $\mathcal{C}_0$ forward invariant since there may exist states in $\mathcal{C}_0$ from which the system will eventually become unsafe regardless of the control actuation (e.g., a "don't hit the ground" safety requirement for a rock thrown off a cliff). This is because the safety requirement may be incompatible with the system dynamics. Thus, we will instead look for subsets $\mathcal{C} \subseteq \mathcal{C}_0$ where there exist inputs that can render the smaller set $\mathcal{C}$ forward invariant and thus keep the system trajectory safely inside of $\mathcal{C}_0$.

We call a set that can be can be rendered forward invariant by some controller a *control invariant set*:

Definition 2.15 (Control Invariant Sets[6] [68])**.** *A set $\mathcal{C} \subset \mathbb{R}^{n_x}$ is control invariant if there exists a controller $\mathbf{k} : \mathcal{C} \to \mathbb{R}^{n_u}$ such that $\mathcal{C}$ is forward invariant.*

[6] Blanchini originally called these sets "controlled-invariant" [68]. We opt for the term "control invariant" as in [5] for flow.

The problem of synthesizing control invariant sets $\mathcal{C} \subset \mathcal{C}_0$ from open-loop dynamics (2.1) is studied in [68], [69] where the largest control invariant subset of a safety criteria set is called a *viability kernel*. Unfortunately, finding a viability kernel is generally a very computationally complex and requires solving a backwards reachability problem. This becomes particularly difficult with increasing nonlinearity of the dynamics and dimensions of the system [4].

Instead of solving for the viability kernel, Chapter 3 of this thesis is dedicated to synthesizing safe sets for special classes of systems using computationally tractable methods. For now, we assume that the safety requirement defines a control invariant set, i.e., $\mathcal{C} = \mathcal{C}_0$.

Control Barrier Functions

After we have obtained a control invariant subset of the safety requirement, we still need to find a controller which achieves that forward invariance. Control barrier functions (CBFs) are a useful tool for this controller synthesis process.

To synthesize these controllers, we connext the Lyapunov stability concepts presented in Sections 2.3 and 2.3 to our understanding of safety. Just as Lyapunov functions analyze the stability of a system using a scalar value $V(\mathbf{x})$, we will analyze safety using a scalar value.

To do this, consider a continuously differentiable function $h : \mathbb{R}^{n_x} \to \mathbb{R}$. In the case of a Lyapunov function, $V(\mathbf{x}) > 0$ indicates that the system is some distance away from the equilibrium and $V(\mathbf{x}) = 0$ indicates that the system has reached it. For safety, we extend this understanding by adopting the paradigm that the system is if $h(\mathbf{x})$ is nonnegative, on the boundary of the $\mathcal{C}$ if $h(\mathbf{x}) = 0$, and unsafe if $h(\mathbf{x})$ is negative. This relates to the safe set $\mathcal{C} \subseteq \mathcal{C}_0 \subset \mathcal{X}$ through the following structure:

$$\mathcal{C} \triangleq \{\, \mathbf{x} \in \mathbb{R}^{n_x} \mid h(\mathbf{x}) \geq 0 \,\}, \tag{2.29}$$

$$\partial\mathcal{C} \triangleq \{\, \mathbf{x} \in \mathbb{R}^{n_x} \mid h(\mathbf{x}) = 0 \,\}, \tag{2.30}$$

$$\text{Int}(\mathcal{C}) \triangleq \{\, \mathbf{x} \in \mathbb{R}^{n_x} \mid h(\mathbf{x}) > 0 \,\}, \tag{2.31}$$

where we assume that zero is a regular value[7] of h and that $\mathcal{C}$ is non-empty and has no isolated points, that is, $h(\mathbf{x}) = 0 \implies \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}) \neq 0$, $\text{Int}(\mathcal{C}) \neq \emptyset$, and $\overline{\text{Int}(\mathcal{C})} = \mathcal{C}$.

Since $h(\mathbf{x})$ can take on negative values where $V(\mathbf{x})$ could not, we must also extend our notion of class $\mathcal{K}$ functions to admit negative arguments:

[7] The value $c \in \mathbb{R}$ is a *regular value* of a continuously differentiable function $h : \mathbb{R}^{n_x} \to \mathbb{R}$ if $h(\mathbf{x}) = c \implies \frac{\partial h}{\partial \mathbf{x}} \neq 0$.

Definition 2.16 (Extended Class $\mathcal{K}$ Functions [6]). *Let $a, b \in \mathbb{R}_{>0}$. A function $\alpha : (-b, a) \to \mathbb{R}$ that is continuous on $(-b, a)$ is said to be an extended class $\mathcal{K}$ function (denoted $\alpha \in \mathcal{K}^e$) if α is strictly monotonically increasing over $(-b, a)$ and $\alpha(0) = 0$. This function α is an extended class $\mathcal{K}_\infty$ function (denoted $\alpha \in \mathcal{K}^e_\infty$) if $a = b = \infty$, $\lim_{r\to\infty} \alpha(r) = \infty$, and $\lim_{r\to-\infty} \alpha(r) \to -\infty$.*

Again, we find that the standard linear scaling $\alpha(r) = \rho r$ for some $\rho > 0$ is a class $\mathcal{K}^e_\infty$ function. While other class $\mathcal{K}^e_\infty$ could be used in theory, this thesis will predominately use these linear scalings.

Now we can introduce the preliminary notion of a barrier functions as a tool for *verifying* the safety of a closed loop system (2.2), similar to how the Lyapunov functions can be used to verify stability (Thm. 2.6):

Definition 2.17 (Barrier Function [6]). *Let $\mathcal{C} \subset \mathbb{R}^{n_x}$ be the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^{n_x} \to \mathbb{R}$ with 0 as a regular value. The function h is a Barrier Function (BF) for the closed-loop system* (2.2) *if there exists an $\alpha \in \mathcal{K}^e_\infty$ such that:*

$$\dot{h}(\mathbf{x}) = \underbrace{\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x})\mathbf{f}(\mathbf{x})}_{L_\mathbf{f} h(\mathbf{x})} + \underbrace{\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x})\mathbf{g}(\mathbf{x})}_{L_\mathbf{g} h(\mathbf{x})} \mathbf{k}(\mathbf{x}) \geq -\alpha(h(\mathbf{x})), \tag{2.32}$$

for all[8] $\mathbf{x} \in \mathbb{R}^{n_x}$.

The inequality in Definition 2.17 shows a direct connection to the Lyapunov convergence inequality in (2.6). In the Lyapunov case, inequality (2.9) ensures that the convergence towards the equilibrium point is upperbounded by a negative number, forcing the system down level sets of the Lyapunov function and towards the equilibrium point. However, in the case of safety, when $\mathbf{x} \in \mathcal{C}$, the converge rate down towards the boundary of the safe set ($h(\mathbf{x}) = 0$) is *lower* bounded by a negative number, thus the system can decay towards the boundary, but must slow down as it approaches.

The utility of barrier functions in verifying safety is formalized in the following theorem:

[8] Note that in order to achieve safety, the barrier function inequality (2.32) is only required to hold for all $\mathbf{x} \in \mathcal{C}$. The stronger assumption that it holds for all $\mathbf{x} \in \mathbb{R}^{n_x}$ is presented here for convenience and will be used in study of the ISSf robustness property (Def. 2.22)

Theorem 2.18 ([18]). *If $h : \mathbb{R}^{n_x} \to \mathbb{R}$ is a barrier function for the closed loop system* (2.2) *and the set $\mathcal{C} \subset \mathbb{R}^{n_x}$, then* (2.2) *is safe with respect to $\mathcal{C}$.*

This result was established in [18] where the safety guarantee was achieved by applying Nagumo's theorem [13] under the assumption that h has 0 as a regular value. Later [70][Thm. 1] proved that h does not need to have 0 as a regular value, as long as $\alpha \in \mathcal{K}_\infty^e$ and (2.32) holds for $\mathbf{x} \in \mathbb{R}^{n_x} \setminus \mathcal{C}$. A detailed discussion of this is provided in [70], however, for this work we will make the assumption that 0 is a regular value of h since it is not a significantly limiting assumption.

Similar to the converse results for Theorem 2.6, converse results have been proven for barrier functions under a variety of relatively general assumptions [14].

We can now generalize barrier functions to the setting of controller synthesis in a format similar to CLFs:

Definition 2.19 (Control Barrier Function (CBF)[6]). *Let $\mathcal{C} \subset \mathbb{R}^{n_x}$ be the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^{n_x} \to \mathbb{R}$ with 0 a regular value. The function h is a control barrier function (CBF)*[9] *for the open-loop system* (2.1) *if there exists an $\alpha \in \mathcal{K}_\infty^e$ such that for all $\mathbf{x} \in \mathbb{R}^{n_x}$:*

$$\sup_{\mathbf{u} \in \mathcal{U}} \dot{h}(\mathbf{x}) \triangleq \underbrace{\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x})\mathbf{f}(\mathbf{x})}_{L_\mathbf{f} h(\mathbf{x})} + \underbrace{\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x})\mathbf{g}(\mathbf{x})}_{L_\mathbf{g} h(\mathbf{x})} \mathbf{u} > -\alpha(h(\mathbf{x})). \tag{2.33}$$

Although this definition can be rescricted to $\mathcal{C}$ instead of $\mathcal{X}$, the feasibility of the CBF constraint (2.33) over the larger domain will be useful for providing ensure the robust safety and set atractivity properties of $\mathcal{C}$.

Intuitively, as with the barrier function constraint (2.32), the CBF constraint (2.33) requires that the system slow down as it approaches the boundary of $\mathcal{C}$.

As with the CLFs, the CBF inequality (2.33) defines the point-wise set of safe control actions:

$$\mathscr{K}_{\text{CBF}}(\mathbf{x}) = \left\{ \mathbf{u} \in \mathbb{R}^{n_u} \mid \dot{h}(\mathbf{x}, \mathbf{u}) \geq -\alpha(h(\mathbf{x})) \right\}, \tag{2.34}$$

and $\mathscr{K}_{\text{CBF}}(\mathbf{x})$ input set is non-empty at $\mathbf{x} \in \mathbb{R}^{n_x}$ if :

$$L_\mathbf{g} h(\mathbf{x}) = \mathbf{0} \implies L_\mathbf{f} h(\mathbf{x}) > -\alpha(h(\mathbf{x})) \tag{2.35}$$

[9] As in the CLF Definition (Def. 2.7), the strict inequality appears in (2.33) to enable the synthesis of locally Lipschitz controllers [63]; this strictness also implicitly requires that h has 0 as a regular value.

and h is a CBF if this implication holds for all $\mathbf{x} \in \mathbb{R}^{n_x}$.

A main result in [10], [33] relates CBFs to the forward invariance of $\mathcal{C}$ of the closed-loop system (2.2):

Theorem 2.20 (CBF Safety [6])**.** *Given a set $\mathcal{C} \subset \mathbb{R}^{n_x}$ defined as the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^{n_x} \to \mathbb{R}$, if h is a CBF for* (2.1)*, then any locally Lipschitz continuous controller $\mathbf{k} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$, such that $\mathbf{k}(\mathbf{x}) \in \mathscr{K}_{\text{CBF}}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{C}$, renders the closed-loop system* (2.2) *safe with respect to $\mathcal{C}$.*

Similar to CLFs, a common controller used to achieve this is the control barrier function quadratic program (CBF-QP) (2.36). This safety-filter is a point-wise optimal controller that minimally adjusts a desired (but not necessarily safe) locally Lipschitz controller $\mathbf{k}_{\text{desired}} : \mathbb{R}^n \to \mathbb{R}^m$ in order to satisfy the CBF inequality (2.33).

$$\begin{aligned} \mathbf{k}_{\text{CBF-QP}}(\mathbf{x}) = \underset{\mathbf{u} \in \mathbb{R}^{n_u}}{\operatorname{argmin}} \quad & \frac{1}{2}\|\mathbf{u} - \mathbf{k}_{\text{des}}(\mathbf{x})\|_2^2 \\ \text{s.t.} \quad & \underbrace{L_{\mathbf{f}} h(\mathbf{x}) + L_{\mathbf{g}} h(\mathbf{x})\mathbf{u}}_{\frac{d}{dt} h(\mathbf{x}, \mathbf{u})} \geq -\alpha(h(\mathbf{x})). \end{aligned} \tag{2.36}$$

Remark 2.21 (On the performance of CBF-based controllers)**.** *The performance of the CBF-QP safety-filter towards achieving a particular goal (such as stabilizing to a point) is indirectly achieved using the desired controller $\mathbf{k}_{\text{des}}$. An alternative method called the CLF-CBF-QP [6] enforces a slackened form of the CLF constraint alongside the CBF constraint with a cost function that minimizes the input norm and the slack variable. However, in both cases it is common for the point-wise minimization in both controllers to result in locally stable undesirable equilibria [16], resulting in a closed-loop system that is unable to achieve its performance goals. Section 7.4 will explore modification to this controller that remove these obstructions and better incorporate performance goals by for performance over a horizon, instead of through point-wise minimization.*

As with the CLF-QP, the CBF-QP with no input constraints admits a closed-form solution:

$$\mathbf{k}_{\text{CBF-QP}}(\mathbf{x}) = \mathbf{k}_{\text{des}}(\mathbf{x}) + \frac{-L_{\mathbf{g}} h(\mathbf{x})^\top}{\|L_{\mathbf{g}} h(\mathbf{x})\|_2^2} \begin{cases} \mathbf{0}, & \mathbf{a}(\mathbf{x}) \geq 0 \\ \mathbf{a}(\mathbf{x}), & \mathbf{a}(\mathbf{x}) < 0 \end{cases} \tag{2.37}$$

where $\mathbf{a}(\mathbf{x}) = L_{\mathbf{f}} h(\mathbf{x}) + L_{\mathbf{g}} h(\mathbf{x}) \mathbf{k}_{\text{des}}(\mathbf{x}) + \alpha(h(\mathbf{x}))$.

Since all of the component functions of this controller can be easily and analytically computed, the CBF-QP controller and similar CBF-based controllers can be implemented on hardware systems at real-time speeds with significant successes for the real-time control of robotic systems as will be seen throughout this thesis.

Before discussing robust safety, I would like to end with a warning that motivates a large portion of this thesis: *be very careful when assuming that h is a CBF a known α* this a very strong assumption, that is stronger than assuming that $\mathcal{C}$ is control invariant. This process of synthesizing CBFs, as studied in Chapter 3, is often considered the greatest challenge of CBF-based approaches.

Input-to-State Safety

To consider the problem of safety for real-world systems, we must additionally analyze the effect of uncertainty on our safety guarantees.

Although Theorem 2.20 generates rigorous safety guarantees for controllers like the CBF-QP, it makes several critical assumptions which are not generally valid in practice. One of which is perfect model knowledge: to effectively implement controllers which satisfy the constraints of Theorem 2.20, one is required to have an exact model of the open-loop system dynamics $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$. However, in reality we likely have a simplified model of the system that may ignore complexities like drag, motor viscosity, flexibility of internal components, and more. Although this simplified model may be very useful, it introduces error which can invalidate the safety guarantees of Theorem 2.20.

As with robust stability, we consider systems with matched (2.17) or unmatched (2.18) disturbances. In stability, we saw that the ISS property captured the idea that, in the presence of bounded disturbances, convergence to an equilibrium point became convergence to a set. We will see a similar property for systems with attractive safe sets. Essentially, convergence to a set becomes convergence to an expanded set in the face of bounded disturbances, with the size of the expansion depending on the disturbance bound. This property is captured in the Input-to-State Safety property of CBFs:

Definition 2.22 (Input-to-State Safety [31])**.** *The closed loop system with matched* (2.17) *or unmatched* (2.18) *disturbances is Input-to-State Safe (ISSf) with respect to $\mathcal{C}$ if there exists $\gamma \in \mathcal{K}_\infty$ such that for all $\overline{d} \in \mathbb{R}_{\geq 0}$ and disturbances $\mathbf{d} : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n_d}$*

satisfying $\|\mathbf{d}\|_\infty \leq \overline{d}$*, the set* $\mathcal{C}_\delta \subset \mathbb{R}^{n_x}$ *defined as:*

$$\mathcal{C}_\delta \triangleq \{\mathbf{x} \in \mathbb{R}^{n_x} \mid h(\mathbf{x}) \geq -\delta = -\gamma(\overline{d})\} \tag{2.38}$$

is forward invariant.

To obtain the ISSf property from barrier functions a small modification to (2.32) is needed in the general case as presented in [31, Def. 4]. However, in the simplified case where h has a bounded gradient and α is a linear class $\mathcal{K}_\infty^e$ function, we have the following relationship between barrier functions and ISSf systems with unmatched disturbances:

Proposition 2.23. *Let* $\mathcal{C} \subset \mathbb{R}^{n_x}$ *be the 0-superlevel set of a function* $h : \mathbb{R}^{n_x} \to \mathbb{R}$ *that is continuously differentiable on* $\mathbb{R}^{n_x}$*. If* h *is a barrier function for the nominal system* (2.2) *with a bounded gradient* $\left\|\frac{\partial h}{\partial \mathbf{x}}\right\| \leq b_h$ *for all* $\mathbf{x} \in \mathbb{R}^{n_x}$ *and some* $b_h > 0$*, and an* $\alpha \in \mathcal{K}_\infty^e$ *such that* $\alpha(r) = \rho r$ *for some* $\rho > 0$*, then the closed-loop system with an unmatched disturbance* (2.18) *is ISSf with respect to* $\mathcal{C}_\delta$ *for* $\delta = \frac{b_h \overline{d}}{\rho}$*.*

Proof. Since h is a barrier function for the closed-loop system (2.2) with a bounded gradient and $\alpha(r) = \rho r$, the following inequalities hold for the system with an unmatched disturbance (2.18):

$$\dot{h}(\mathbf{x}) = L_\mathbf{f} h(\mathbf{x}) + L_\mathbf{g} h(\mathbf{x})\mathbf{k}(\mathbf{x}) + \frac{\partial h}{\partial \mathbf{x}}\mathbf{d}(t) \geq -\rho h(\mathbf{x}) + \frac{\partial h}{\partial \mathbf{x}}\mathbf{d}(t), \tag{2.39}$$

$$\geq -\rho h(\mathbf{x}) - \left\|\frac{\partial h}{\partial \mathbf{x}}\right\| \|\mathbf{d}(t)\| \geq -\rho h(\mathbf{x}) - b_h \overline{d} = -\rho\left(h(\mathbf{x}) - \frac{b_h \overline{d}}{\rho}\right). \tag{2.40}$$

Since $\frac{d}{dt}\left(h(\mathbf{x}) - \frac{b_h \overline{d}}{\rho}\right) = \frac{dh}{dt}$, we have that $h_\delta \triangleq h(\mathbf{x}) - \frac{b_h \overline{d}}{\rho}$ is a barrier function for system with an unmatched disturbance (2.18) that guarantees the safety of the expanded set $\mathcal{C}_\delta$ for $\delta = \frac{b_h \overline{d}}{\rho}$. □

In this proof, we see that the set attractivity property of the barrier function, i.e., $-\rho h(\mathbf{x})$ when $h(\mathbf{x}) < 0$, is used to counteract the effect of the disturbance and form an expanded safe set in the presence of the unmatched disturbance. This reflects how the stability component of the Lyapunov function is used to cancel the disturbance in the proof of Proposition 2.11.

As with ISS-CLFs, we can robustify the CBF constraint to control this expansion in the presence of matched disturbances:

Definition 2.24 (ISSf-CBF [32, Def. 3])**.** *A continuously differentiable function* $h : \mathbb{R}^{n_x} \to \mathbb{R}$ *with 0 a regular value is said to be an input-to-state safe control barrier function (ISSf-CBF) for* (2.1) *on* $\mathcal{C}$ *as in* (2.29) *if there exists* $\alpha \in \mathcal{K}_\infty^e$ *and* $\epsilon > 0$ *such that for all* $\mathbf{x} \in \mathcal{X}$*:*

$$\sup_{\mathbf{u}\in\mathbb{R}^{n_u}} L_\mathbf{f} h(\mathbf{x}) + L_\mathbf{g} h(\mathbf{x})\mathbf{u} > -\alpha(h(\mathbf{x})) + \frac{1}{\epsilon}\|L_\mathbf{g} h(\mathbf{x})\|^2. \tag{2.41}$$

ISSf-CBFs include the robustifying term $\frac{1}{\epsilon}\|L_\mathbf{g} h(\mathbf{x})\|^2$ to mitigate the impact of disturbances while providing practical safety guarantees [32]. Note that if h satisfies (2.33), then it also satisfies (2.41) as robustness is only added when control actuation would immediately effect safety (i.e., when $\|L_\mathbf{g} h(\mathbf{x})\| \neq \mathbf{0}$). Thus, (2.41) increases the robustness of safety to disturbances while retaining feasibility.

More precisely, this results in the following ISSf property:

Theorem 2.25 (ISSf-CBF Safety [32, Thm. 2])**.** *Let* $\mathcal{C} \subset \mathbb{R}^{n_x}$ *be the 0-superlevel set of an ISSf-CBF* h *with* $\alpha \in \mathcal{K}_\infty^e$ *and* $\epsilon > 0$ *for the closed-loop system with a matched disturbance* (2.17)*. If the disturbance is bounded* $\|\mathbf{d}\| \leq \overline{d}$ *for some* $\overline{d} \geq 0$*, then for any Lipschitz continuous controller that satisfies the* (2.41) *for all* $\mathbf{x} \in \mathcal{X}$ *the closed-loop system* (2.17) *is safe with respect to* $\mathcal{C}_\delta$ *with* $\delta = \alpha^{-1}\left(-\frac{\epsilon\overline{d}^2}{4}\right)$*. Thus, the system is ISSf with respect to* $\mathcal{C}$*.*

Thus, the ISSf-CBF constraint ensures that the expanded set $\mathcal{C}_\delta$ is kept safe and the exact size of this expansion can be controlled by the tuning parameter ϵ, providing both a theoretical lens through which to view the graceful degradation of safety in the presence of uncertainty and also a practically useful control technique for mitigating these effects.

With this background understanding of nonlinear control, Lyapunov methods, and CBFs, we can now move on to the main contributions of this thesis that will tackle the problems of safe set synthesis, robustness to additional forms of uncertainty, and the problem of achieving high performance behavior alongside safety guarantees, with example hardware demonstrations provided on robotic systems.

Chapter 3

SAFE SET SYNTHESIS

"An experimentalist and a theoretician are stranded on a desert island with sealed cans of food. The experimentalist suggests smashing the cans open or heating them until they burst. Unimpressed, the theoretician says, 'Why don't we just assume that we have a can opener?'"

"I can prove that, if unicorns existed, they'd be nice."

These quotes capture a common failure of safety-critical control: it can often provide elegant theoretical results that rely on strong assumptions that are difficult to satisfy. While CBFs offer compelling infinite-horizon safety guarantees, they rely on the assumption that the user has already constructed a control invariant safe set, an assumption akin to assuming the theoretician has a can opener.

This chapter tackles that missing step by focusing on the construction and verification of control invariant safe sets, which are essential for deploying CBFs with practical guarantees.

Abstract

Control barrier functions (CBFs) have emerged as a powerful framework for enforcing safety guarantees, offering infinite-horizon guarantees and input-to-state safety robustness properties. However, these theoretical guarantees rely on strong assumptions that often fail to hold in practice. In particular, they assume the knowledge of functions $h(\mathbf{x})$ and α for which the CBF condition (2.33) holds. This requirement is *more restrictive* than assuming that the safe set $\mathcal{C}$ is control invariant and, when it is violated, can lead to systems that were thought to be safe to have catastrophic failures. Thus, verifying or constructing control invariant safe sets is essential for deploying CBFs with real-world meaningful guarantees.

Unfortunately, verifying control invariance or synthesizing such sets is often computationally intractable for complex, nonlinear, or high-dimensional systems. This chapter addresses that challenge by presenting constructive safe-set synthesis methods for a variety of system classes including hierarchical systems with tracking

controllers, feedback linearizable systems, dual relative degree systems, and systems with verified backup controllers. By relying on structural properties of the dynamics, these methods present computationally tractable solutions for generating CBFs and control invariant sets.

Published content: This chapter is adapted from previously published work in [48]–[50], [52].

3.1 Introduction

The provable guarantees of safety-critical control and CBFs are attractive for robotics, where safety is paramount in real-world deployment. However, one of the confounding factors that has limited the general application of CBFs lies in determining a "valid" CBF, i.e., a function h_0 that satisfies Definition 2.19. It may be easy to create a function h_0 whose 0-superlevel set describes the system's safety requirements, but it is generally very difficult to assure that that function defines a control invariant set or is actually a CBF as discussed in Chapter 2.

In general, the construction of control invariant sets is a difficult problem that is made computationally intractable for large systems by the "curse of dimensionality" [4], [71], [72]. Several methods have been developed, to overcome this computational difficulty and rigorously assure control invariance. For example, this is achieved in [8], [73] using reachibility methods for simple linear systems. It is achieved in [5] by computing a small control invariant set and then expanding it using horizon-based planning. In the CBF literature, it is common to construct valid CBFs by hand [74] although this method scales poorly with system complexity. Alternatively, methods like exponential [75] and higher-order [76]–[78] CBFs have attempted to solve this control invariance problem indirectly by instead solving the relative degree problem, where the effect of the input might not appear in the first derivative of $h(\mathbf{x})$. While these extension methods can improve feasibility of the CBF condition, they can still result in safety failures as noted in [79]. Other CBF synthesis methods include data-driven approaches like [41] which, while practically useful, require collection of expert data and generally do not enable the same rigorous safety guarantees as other CBF-based methods, and backstepping methods which require significant assumptions on the structure of the full-order system dynamics [80].

As an alternative to synthesizing controlled invariant sets and valid CBFs for the whole system, several works have simplified the problem by instead constructing CBFs for a reduced-order model (ROM) of their system, whose validity can be

more easily verified. This is inspired by the many robotics successes that rely on ROM-based controller synthesis [81], including single integrator models used for multi-robot applications [20], [82], [83] or quadrotor applications [84] and unicycle models for wheeled robots [85], [86]. While the naive application of this idea without consideration of the full-order system dynamics suffers from model-mismatch between the ROM and the true system [84], recent work has explored the use of hierarchical control methodologies to guarantee safety with respect to the full-order system [48], [87], [88]. This results in a simpler ROM-based CBF synthesis problem, but generally does so at the cost of system performance as any deviation between the ROM and the full-order system is considered a disturbance.

This chapter will address safe set and CBF synthesis from a variety of perspectives, offering tools for several classes of systems. In general these methods leverage an underlying structure of the system to guarantee convergence to a set, point, or trajectory. Using this convergence guarantee, we can then convert safe sets for the ROM to safe sets for the full-order system. These methods span a variety of system types including partially feedback linearizable, robotic, and dual relative degree systems. The remainder of this chapter introduces a new synthesis method in each section.

Section 3.2 leverages the connection between CBFs and CLFs to construct control invariant sets and valid CBFs from existing CLFs. Then, Section 3.3 introduces a model-free CBF approach where velocity tracking controllers are used to implicitly synthesize valid CBFs. Next, Section 3.4 presents a CBF synthesis method for partially feedback linearizable systems, inspired by [80], that explicitly synthesizes full-order CBFs for underactuated robotic systems. This method is then expanded on in Section 3.5 where we generalize beyond partially feedback linearizable systems to a new class of systems, termed dual relative degree, that have inputs appearing at two different relative degrees, similar to the translational and rotational inputs of unicycle and quadrotor systems. Finally, in Section 3.6 we consider the backup set CBF method, originally presented in [26], that leverages a finite horizon to expand a smaller, known "backup" safe set. These methods provide a compendium of tools for rapid CBF and/or control invariant set synthesis for a wide variety of relevant systems. We additionally provide hardware-based robot examples throughout to demonstrate the utility of these methods.

3.2 Synthesizing CBFs from CLFs

As presented in Chapter 2, CBFs can be viewed as a generalization of CLFs, extending Lyapunov theory from stabilization to set invariance. This relationship naturally motivates the construction of CBFs *using* CLFs.

In this case, since a CLF guarantees asymptotic stabilization, each of its sublevel sets is control invariant. Consequently, these sublevel sets can be used to define CBFs.

Theorem 3.1 (CBFs from CLFs)**.** *If $V : \mathbb{R}^{n_x} \to \mathbb{R}$ is a CLF (Def. 2.7) for the open-loop system* (2.1)*, then $h(\mathbf{x}) \triangleq C - V(\mathbf{x})$ is a CBF with $\alpha(r) = \frac{k_3}{k_2} r \in \mathcal{K}_\infty^e$ for system* (2.1) *for any $C > 0$.*

Proof. Since V is a CLF, it is a continuously differentiable function that satisfies the CLF inequality (2.11):

$$\sup_{\mathbf{u}\in\mathbb{R}^{n_u}} L_{\mathbf{f}}V(\mathbf{x}) + L_{\mathbf{g}}V(\mathbf{x})\mathbf{u} \leq -k_3\|\mathbf{x}\|^2 \leq -\frac{k_3}{k_2}V(\mathbf{x}). \tag{3.1}$$

With $h : \mathbb{R}^{n_x} \to \mathbb{R}$ defined as $h(\mathbf{x}) \triangleq C - V(\mathbf{x})$, this inequality can be equivalently rewritten as:

$$\sup_{\mathbf{u}\in\mathbb{R}^{n_u}} -L_{\mathbf{f}}h(\mathbf{x}) - L_{\mathbf{g}}h(\mathbf{x})\mathbf{u} \leq -\frac{k_3}{k_2}(C - h(\mathbf{x})), \tag{3.2}$$

$$\inf_{\mathbf{u}\in\mathbb{R}^{n_u}} L_{\mathbf{f}}h(\mathbf{x}) + L_{\mathbf{g}}h(\mathbf{x})\mathbf{u} \geq -\frac{k_3}{k_2}h(\mathbf{x}) + \frac{k_3}{k_2}C \geq -\frac{k_3}{k_2}h(\mathbf{x}). \tag{3.3}$$

Thus, h satisfies the CBF inequality (2.33) with $\alpha(r) = \frac{k_3}{k_1} r \in \mathcal{K}_\infty^e$. □

Line (3.3) reveals that the stabilization condition imposed by the CLF is stronger than the CBF condition required for forward invariance. In particular, the positive term $\frac{k_3}{k_1}C$ is dropped to yield the CBF inequality. Therefore, if a CLF is known and its sublevel sets lie within the safety requirement $\mathcal{C}_0$, this method produces valid CBFs for control-invariant subsets $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^{n_x} \mid C - V(\mathbf{x}) \geq 0\} \subseteq \mathcal{C}_0$. This process is visualized in Fig. 3.1.

However, this method has several limitations. The resulting sets $\mathcal{C}$ are constrained to be sublevel sets of V, which are typically compact and centered around an equilibrium point. More significantly, the method requires a pre-existing CLF. While CLF synthesis has been extensively studied, general methods such as sum-of-squares programming [89] remain computationally expensive for high-dimensional nonlinear systems.

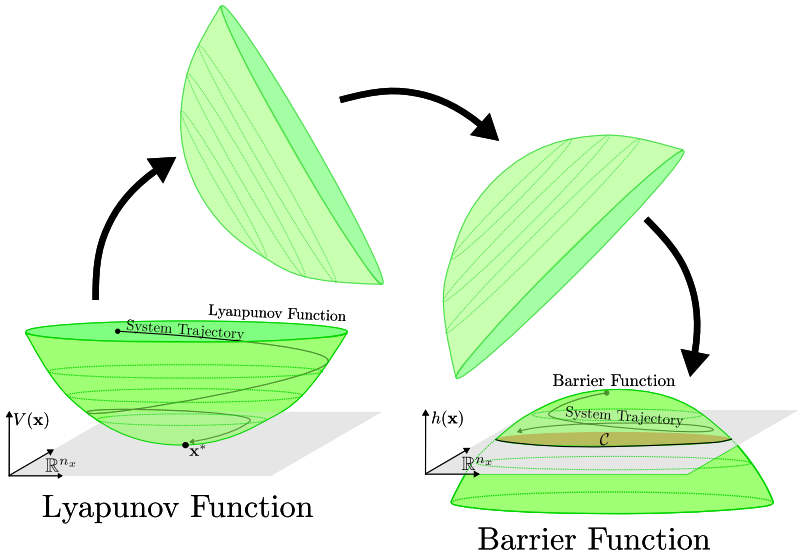


Figure 3.1. A visualization of the connection between Lyapunov and barrier functions that shows how a CBF can be constructed as a sub-level set of a Lyapunov function.

In contrast, synthesizing CLFs is straightforward for systems that are full-state feedback linearizable. This connection motivates the following definition and will serve as the basis for several generalizations later in this chapter.

Definition 3.2 (Full-State Feedback Linearizability[1] [61, Def. 13.1 and Sec. 13.3])**.** *The open-loop control system* (2.1) *is full-state feedback linearizable if there exist a diffeomorphism* $\boldsymbol{\Phi} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ *that enables the change of coordinates* $\boldsymbol{\eta} = \boldsymbol{\Phi}(\mathbf{x})$ *and a controller* $\mathbf{k} : \mathbb{R}^{n_u} \times \mathbb{R}^{n_u}$ *such that:*

$$\dot{\boldsymbol{\eta}} = D\boldsymbol{\Phi}(\mathbf{x})(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{k}(\mathbf{x}, \mathbf{v}))\big|_{\mathbf{x}=\boldsymbol{\Phi}^{-1}(\boldsymbol{\eta})} = \mathbf{A}\boldsymbol{\eta} + \boldsymbol{B}\mathbf{v} \tag{3.4}$$

where $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$ *and* $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$.

For example, if the actuation matrix $\mathbf{g}(\mathbf{x})$ is globally invertible, then the system is full-state feedback linearizable via the identity transformation $\boldsymbol{\eta} = \boldsymbol{\Phi}(\mathbf{x}) = \mathbf{x}$ and the controller $\mathbf{k}(\mathbf{x}, \mathbf{v}) = \mathbf{g}(\mathbf{x})^{-1}(-\mathbf{f}(\mathbf{x}) + \mathbf{v})$, where $\mathbf{A} = \mathbf{0}$ and $\mathbf{B} = \mathbf{I}$.

Once linearized, if the pair $(\mathbf{A}, \mathbf{B})$ is controllable, tools from linear systems theory can be used to design stabilizing feedback controllers for $\mathbf{v}$ [90]. The existence of such a controller guarantees the existence of a CLF via the converse Lyapunov theorem [61, Thm. 4.14], which can then be used to construct the CBF $h(\mathbf{x}) = C - V(\boldsymbol{\Phi}(\mathbf{x}))$, where V is a CLF in $\boldsymbol{\eta}$-coordinates and $\boldsymbol{\Phi}$ is the coordinate transformation.

[1] I make several simplifying changes regarding the locality of this definition and the structure of the linear system. Please see [61, Def. 13.1] for a more nuanced definition.

Then, if the equilibrium point of the CLF is in Int($\mathcal{C}_0$), we can choose $C > 0$ to find a control invariant set $\mathcal{C} \subset \mathcal{C}_0$.

Although this approach requires strong assumptions, including feedback linearizability and controllability, it produces a valid CBF satisfying Definition 2.19 with a known $\alpha \in \mathcal{K}_\infty^e$. The remainder of this chapter generalizes this method in several ways. In Section 3.3, we replace the constant C with an arbitrary safety function h_0 and relax the assumption of convergence to an equilibrium by considering Lyapunov stability with respect to safe trajectories. In Sections 3.4 and 3.5, we extend this approach to partially feedback linearizable systems and dual relative degree systems, respectively. Finally, in Section 3.6, instead of choosing C by hand, we consider an implicit method for expanding a known Lyapunov sub-level set.

3.3 Model-Free CBF Synthesis

In this section, I present a model-free CBF synthesis method that relies on a hierarchical control structure. Inspired by layered architectures that are common in robotics, we divide the system into two components: (1) the configuration space for which the safety requirement is defined and (2) higher-order terms that determine the ability to track configuration-space trajectories.

By dividing the system into these subcomponents, we enable CBF synthesis through a process similar to that in Section 3.2 where a CLF that establishes the exponential tracking of safe configuration trajectories is used to synthesize a valid CBF, this can be thought of as a formalization of the approach in [84] where safety was achieved in a practical setting by synthesizing and tracking safe velocities. Since the safety criteria relies only on the configuration space, this approach is agnostic to the application domain as long as an underlying tracking controller exists for the system. In general the existence of velocity tracking controllers is well-established in robotics and are available for many robot platforms [91]. Once sufficient tracking capabilities are established, enforcing safety does not require further consideration of the high-fidelity model, thus only the safety criteria and the tracking certificate are required. A pictorial description of this method can be seen in Figure 3.2.

The contributions of this section are as follows:

- A proof of model-free CBF synthesis and safety guarantees for complex robotic systems that use tracking controllers to execute model-free safe behaviors.

- Demonstrations of the broad applicability of this method on a wheeled robot in simulation and flying and legged robots on hardware.

The text for this section is adapted from:

> T. G. Molnar, R. K. Cosner, A. W. Singletary, W. Ubellacker, and A. D. Ames, "Model-free safety-critical control for robotic systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 944–951, 2022, ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2021.3135569,

A video for this section can be found at [48].

Model-Free Motivation and Problem Setting

For this section we consider robotic systems in generalized coordinates $\mathbf{q} \in \mathcal{Q} \subseteq \mathbb{R}^n$ with Euler-Lagrange dynamics given by:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}(\mathbf{q})\mathbf{u}. \tag{3.5}$$

Here, $\dot{\mathbf{q}} \in \mathbb{R}^{n_q}$ is the generalized velocity, $\mathbf{D}(\mathbf{q}) \in \mathbb{R}^{n_q \times n_q}$ denotes the positive definite and symmetric inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n_q \times n_q}$ denotes the Coriolis matrix, $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^{n_q}$ represents gravitational and other potential effects, $\mathbf{B}(\mathbf{q}) \in \mathbb{R}^{n_q \times n_u}$ is the actuation matrix, and $\mathbf{u} \in \mathbb{R}^{n_u}$ is the control input. This robotic system model can then rewritten as a control-affine system in the form of (2.1):

$$\underbrace{\frac{d}{dt}\begin{bmatrix}\mathbf{q}\\ \dot{\mathbf{q}}\end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix}\dot{\mathbf{q}}\\ -\mathbf{D}^{-1}(\mathbf{q})(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}))\end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix}\mathbf{0}\\ \mathbf{D}^{-1}(\mathbf{q})\mathbf{B}(\mathbf{q})\end{bmatrix}}_{\mathbf{g}(\mathbf{x})}\mathbf{u}. \tag{3.6}$$

Next, we assume that we have knowledge of a user-defined safety requirements represented by the set $\mathbf{q} \in \mathcal{C}_0 \subset \mathbb{R}^{n_q}$ defined on the configuration coordinates, that we would like our system to stay within.

Assumption 3.3. *The safe criteria set $\mathcal{C}_0 \subset \mathbb{R}^{n_q}$ is defined as the 0-superlevel set of a continuously differentiable function $h_0 : \mathcal{Q} \to \mathbb{R}$:*

$$\mathcal{C}_0 = \{\mathbf{q} \in \mathcal{Q} : h_0(\mathbf{q}) \geq 0\}, \tag{3.7}$$

where the gradient of h_0 is bounded, i.e., there exists $b_{h_0} \in \mathbb{R}_{>0}$ such that $\left\| \frac{\partial h_0}{\partial \mathbf{x}} \right\| \leq b_{h_0}$ for all $\mathbf{q} \in \mathcal{C}_0$. That is, safety depends on the configuration $\mathbf{q}$ only and h_0 is independent of $\dot{\mathbf{q}}$.

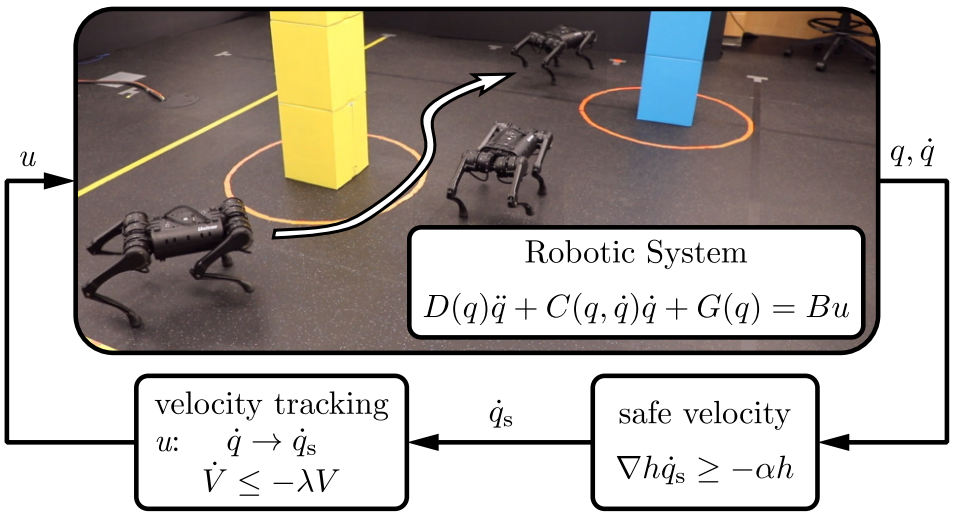


Figure 3.2. The model-free control method in Section 3.3 and its execution on hardware. While the safety-critical controller does not rely on the full dynamical model of the robot, it controls the motion in a provably safe manner.

The rest of this section will then focus on solving the following problem:

Problem 3.4. *For the robotic system (3.6), design a controller* $\mathbf{k} : \mathcal{Q} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ *that establishes the forward invariance of* $\mathcal{C}_0$ *in (3.7) under certain conditions on* $\dot{\mathbf{q}}_0$. *Furthermore, construct a CBF for this system that defines a control invariant set* $\mathcal{C} \subseteq \mathcal{C}_0$.

Following [84], we will solve this problem by generating safe trajectories in configuration space and tracking them with an underlying velocity tracking controller. This reduces the complexity of safety-critical control significantly by allowing us to directly use the safety requirement h_0 in conjunction with widely available velocity tracking controllers [91].

In particular, we synthesize a *safe velocity* $\dot{\mathbf{q}}_s \in \mathbb{R}^{n_q}$ that satisfies:

$$\frac{\partial h_0}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{q}}_s \geq -\alpha h_0(\mathbf{q}), \tag{3.8}$$

which is the CBF safety condition (2.33) for a single integrator system, for some $\alpha \in \mathbb{R}_{>0}$ to be selected later. The safe velocity $\dot{\mathbf{q}}_s$ depends on the configuration $\mathbf{q}$. Note that (3.8) is a kinematic condition that does not depend on the full dynamics (3.6). We consider this inequality to be "model-free" since it is a single integrator description of safety.

To track these safe velocity, we define the tracking error:

$$\dot{\mathbf{e}} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_s. \tag{3.9}$$

and assume the existence of a velocity tracking controller $\mathbf{k}(\mathbf{q}, \dot{\mathbf{q}})$ that is able to drive the error $\dot{\mathbf{e}}$ to zero exponentially[2].

Assumption 3.5. *The velocity tracking controller* $\mathbf{k}(\mathbf{q}, \dot{\mathbf{q}})$ *achieves exponentially stable tracking:* $\|\dot{\mathbf{e}}(t)\| \leq M\|\dot{\mathbf{e}}_0\|e^{-\lambda t}$ *for some* $M, \lambda \in \mathbb{R}_{>0}$*. That is, if* $\dot{\mathbf{e}}$ *is differentiable (i.e.,* $\ddot{\mathbf{e}}$*,* $\ddot{\mathbf{q}}_\text{s}$ *exist*[3] *), there exists a continuously differentiable Lyapunov function* $V : \mathcal{Q} \times \mathbb{R}^{n_q} \to \mathbb{R}_{\geq 0}$ *such that* $\forall (\mathbf{q}, \dot{\mathbf{e}}) \in \mathcal{Q} \times \mathbb{R}^{n_q}$*:*

$$k_1\|\dot{\mathbf{e}}\| \leq V(\mathbf{q}, \dot{\mathbf{e}}) \leq k_2\|\dot{\mathbf{e}}\|, \tag{3.10}$$

for some $k_1, k_2 \in \mathbb{R}_{>0}$*, and there exists* $\lambda \in \mathbb{R}_{>0}$ *such that* $\forall (\mathbf{q}, \dot{\mathbf{e}}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_\text{s}) \in \mathcal{Q} \times \mathbb{R}^{n_q} \times \mathbb{R}^{n_q} \times \mathbb{R}^{n_q}$ *the closed-loop system* (2.2) *with the state-feedback controller* $\mathbf{k}(\mathbf{q}, \dot{\mathbf{q}})$ *satisfies the Lyapunov stability condition:*

$$\dot{V}(\mathbf{q}, \dot{\mathbf{e}}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_\text{s}, \mathbf{u}) \leq -\lambda V(\mathbf{q}, \dot{\mathbf{e}}). \tag{3.11}$$

For exposition's sake, below we assume $\ddot{\mathbf{q}}_\text{s}$ *exists.*

Model-Free Safety Guarantees and CBF Synthesis

In what follows, the main result of this section proves that tracking a safe velocity achieves safety for the full dynamics if parameter α is selected to be small enough and we show how this method generates an implicit CBF and control invariant set for the system. Specifically, for tracking controllers satisfying Assumption 3.5 stability translates into safety for the full system (3.6) if $\lambda > \alpha$. As this result is agnostic to the application domain given this assumption[4] we refer to this method as *model-free safety-critical control*.

The following theorem summarizes the safety guarantees provided by tracking the safe velocity and produces an implicit control invariant set $\mathcal{C} \subset \mathcal{C}_0$ and an implicit CBF h.

Theorem 3.6 (Model-Free Safety)**.** *Consider system (3.6), safe set (3.7), safe velocity satisfying (3.8), and velocity tracking controller satisfying (3.11). If* $\lambda > \alpha$*,*

[2] In this thesis, we will focus on scenarios with exponential tracking convergence. The case where the tracking controller may be unable to achieve complete exponential converge of the tracking error is further explored in [48, Sec. III.C].

[3] The error $\dot{\mathbf{e}}$ is assumed to be differentiable in Assumption 3.5 only for exposition's sake. Theorem 3.6 can be extended to non-differentiable signals satisfying $\|\dot{\mathbf{e}}(t)\| \leq M\|\dot{\mathbf{e}}_0\|\text{e}^{-\lambda t}$. The proof relies on the fact that $\dot{h}(\mathbf{q}, \dot{\mathbf{q}}) \geq -\alpha h(\mathbf{q}) - b_{h_0} M\|\dot{\mathbf{e}}_0\|\text{e}^{-\lambda t}$ holds, and by the comparison lemma with $\dot{y}(t) = -\alpha y(t) - b_{h_0} M\|\dot{\mathbf{e}}_0\|\text{e}^{-\lambda t}$, $y(0) = h(\mathbf{q}_0)$ one can show that $h(\mathbf{q}(t)) \geq y(t) \geq 0$.

[4] We recognize that constructing velocity tracking controllers is, however, very model-dependent and provide a compendium of these controllers for a variety of systems in [48, Sec. III.D].

safety is achieved such that $(\mathbf{q}_0, \dot{\mathbf{e}}_0) \in \mathcal{C} \Rightarrow \mathbf{q}(t) \in \mathcal{C}_0$*,* $\forall t \geq 0$*, where:*

$$\begin{aligned} &\mathcal{C} \triangleq \{(\mathbf{q}, \dot{\mathbf{e}}) \in Q \times \mathbb{R}^{n_x} : h(\mathbf{q}, \dot{\mathbf{e}}) \geq 0\}, \\ &h(\mathbf{q}, \dot{\mathbf{e}}) \triangleq -V(\mathbf{q}, \dot{\mathbf{e}}) + \alpha_e h_0(\mathbf{q}), \end{aligned} \tag{3.12}$$

with $\alpha_e = (\lambda - \alpha)k_1/b_{h_0} > 0$ *and* b_{h_0}*,* k_1 *defined at (3.7, 3.10). Additionally, this* $h(\mathbf{q}, \dot{\mathbf{e}}_0)$ *is a CBF for (3.6).*

Proof. Since $V(\mathbf{q}, \dot{\mathbf{e}}) \geq 0$, the implication $h(\mathbf{q}, \dot{\mathbf{e}}) \geq 0 \Rightarrow h_0(\mathbf{q}) \geq 0$ holds. Thus, $h(\mathbf{q}(t), \dot{\mathbf{e}}(t)) \geq 0, \forall t \geq 0$ is sufficient to prove that $h_0(\mathbf{q}(t)) \geq 0, \forall t \geq 0$. We prove this by noticing that the initial conditions satisfy $h(\mathbf{q}_0, \dot{\mathbf{e}}_0) \geq 0$ and we also have:

$$\dot{h}(\mathbf{q}, \dot{\mathbf{e}}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_\mathrm{s}, \mathbf{u}) = -\dot{V}(\mathbf{q}, \dot{\mathbf{e}}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_\mathrm{s}, \mathbf{u}) + \alpha_\mathrm{e} \frac{\partial h_0}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{q}}, \tag{3.13}$$

$$\geq \lambda V(\mathbf{q}, \dot{\mathbf{e}}) + \alpha_\mathrm{e} \frac{\partial h_0}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{q}}_\mathrm{s} + \alpha_\mathrm{e} \frac{\partial h_0}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{e}} \geq \lambda V(\mathbf{q}, \dot{\mathbf{e}}) - \alpha_\mathrm{e}\alpha h(\mathbf{q}) + \alpha_\mathrm{e} \frac{\partial h_0}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{e}},$$

$$\geq (\lambda - \alpha)V(\mathbf{q}, \dot{\mathbf{e}}) - \alpha_\mathrm{e} \left\| \frac{\partial h_0}{\partial \mathbf{q}}(\mathbf{q}) \right\| \|\dot{\mathbf{e}}\| - \alpha h(\mathbf{q}, \dot{\mathbf{e}}), \tag{3.14}$$

$$\geq (\lambda - \alpha)k_1 \|\dot{\mathbf{e}}\| - \alpha_\mathrm{e} b_{h_0} \|\dot{\mathbf{e}}\| - \alpha h(\mathbf{q}, \dot{\mathbf{e}}) \geq -\alpha h(\mathbf{q}, \dot{\mathbf{e}}). \tag{3.15}$$

Here we used the following properties in the six steps of the inequality: (1) definition (3.12) of h, (2) stability condition (3.11) and definition (3.9) of $\dot{\mathbf{e}}$, (3) condition (3.8) on the safe velocity, (4) definition (3.12) of h and the Cauchy-Schwartz inequality, (5) lower bound of V in (3.10) and upper bound b_{h_0} of $\left\| \frac{\partial h_0}{\partial \mathbf{q}}(\mathbf{q}) \right\|$, (6) definition of α_e. This guarantees $h(\mathbf{q}(t), \dot{\mathbf{e}}(t)) \geq 0, \forall t \geq 0$ by Theorem 2.20. Additionally, this chain of inequalities verifies that h satisfies CBF inequality in Definition 2.19. □

Intuitively, condition $\lambda > \alpha$ means the controller tracks the safe velocity faster than the speed at which safety is allowed to decay. In practice, one can simply pick a small enough α for a given velocity tracking controller, for example, by gradually increasing α from 0. Note that there is a trade-off: for smaller α the system generally results in more conservative closed-loop behavior.

Theorem 3.6 requires initial conditions to satisfy $(\mathbf{q}_0, \dot{\mathbf{e}}_0) \in \mathcal{C} \iff h_0(\mathbf{q}_0) \geq V(\mathbf{q}_0, \dot{\mathbf{e}}_0)/\alpha_\mathrm{e}$. This is a stricter condition than $\mathbf{q}_0 \in \mathcal{C}_0 \iff h_0(\mathbf{q}_0) \geq 0$ since we must also take into account the higher-order states and how they affect the system's ability to regulate $h_0(\mathbf{q})$. The additional conservatism that is introduced by this restriction is reduced when the initial tracking error $\dot{\mathbf{e}}_0$ is smaller (since $V(\mathbf{q}_0, \dot{\mathbf{e}}_0)$ is smaller) and when the tracking is faster, i.e., $\lambda - \alpha$ is larger (since α_e is larger).

Applications to flying and legged robots

Next, we executed the obstacle avoidance task on two fundamentally different hardware platforms: a quadrotor drone and quadruped; see Fig. 3.3. The obstacle locations were known *a priori*, sensing measurements from were used to determine the robots' position only. We performed two classes of experiments by synthesizing safe velocities based on the single integrator $\frac{d}{dt}\mathbf{q} = \dot{\mathbf{q}}_s$ and the unicycle model:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi & 0 \\ \sin\psi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_s \\ \omega_s \end{bmatrix}, \tag{3.16}$$

respectively. For the multiple obstacles in these experiments, we considered the closest one at each time. This results in a nonsmooth CBF which has been analyzed in [92]. A video of the experiments can be found at [93].

Example 3.7 (Quadrotor and Quadruped Safety using a Single-Integrator Model)**.** *First, we considered the single integrator model and we tracked the associated safe velocity with the quadrotor and quadrupedal robots by platform-specific tracking controllers. We used the CBF:*

$$h_0(\mathbf{q}) = \|\mathbf{q} - \mathbf{q}_{\text{obs}}\| - r, \tag{3.17}$$

where $r \in \mathbb{R}_{>0}$ is the combined radius of the obstacle and the robot, and safe velocity generated by the CBF-QP:

$$\begin{aligned} \underset{\dot{\mathbf{q}}_\text{s}\in\mathbb{R}^2}{\operatorname{argmin}} \quad & \|\dot{\mathbf{q}}_\text{s} - \dot{\mathbf{q}}_\text{d}\|^2 \\ \text{s.t.} \quad & \frac{(\mathbf{q} - \mathbf{q}_{\text{obs}})^\top}{\|\mathbf{q} - \mathbf{q}_{\text{obs}}\|} \dot{\mathbf{q}}_\text{s} \geq -\alpha(h_0(\mathbf{q})). \end{aligned} \tag{3.18}$$

The desired velocity was $\dot{\mathbf{q}}_\text{d} = -K_\text{P}(\mathbf{q} - \mathbf{q}_\text{goal})$ with saturation for the goal location $\mathbf{q}_\text{goal} \in \mathcal{Q}$.

The quadrotor drone was a custom-built robot [22], shown in Fig. 3.3(a). It has 6 degrees of freedom and 4 actuators. The state of the robot (position, orientation and corresponding velocities) were measured by IMU and an OptiTrack motion capture system. State estimation and control action computation ran at 400 Hz. The safe velocity was commanded to the drone wirelessly from a desktop computer, while velocity tracking was done using an on-board betaflight flight controller and the geometric control scheme presented in [94]. The safe velocity was calculated with $K_\text{P} = 0.7\,s^{-1}$ and $\alpha = 0.2\,s^{-1}$. Fig. 3.3(a) shows the quadrotor reaching the goal

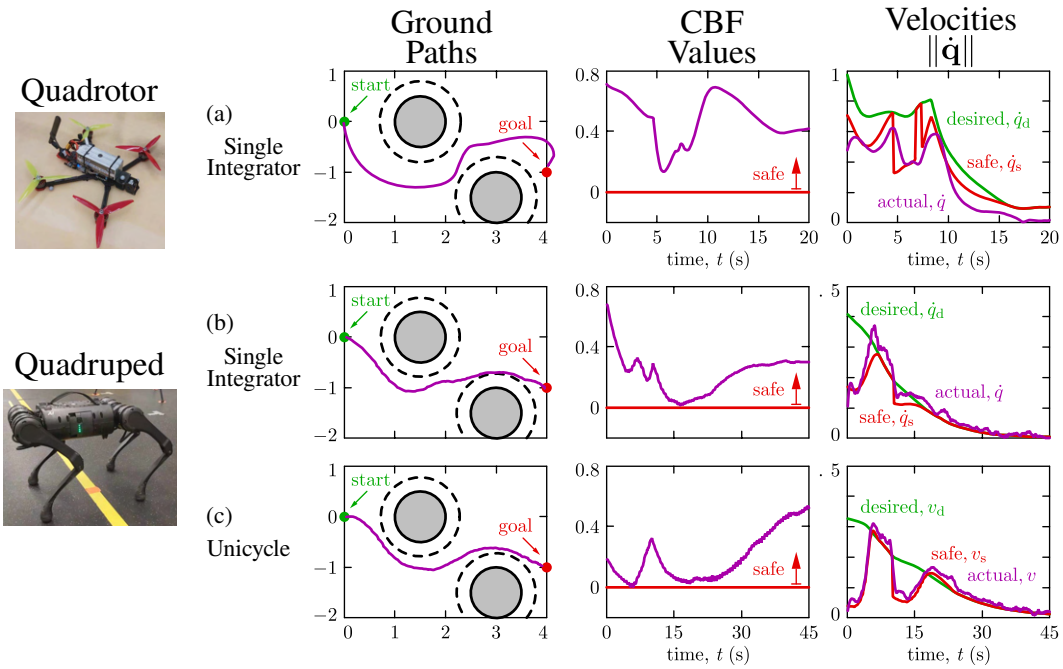


Figure 3.3. Hardware experiments using the proposed model-free safety-critical control method. An obstacle avoidance task is accomplished by two fundamentally different robots: a custom-made quadrotor (top) and a Unitree A1 quadruped (bottom). (a) The quadrotor is tracking a safe velocity determined based on single integrator model. (b) The quadruped is tracking a safe velocity based on single integrator model via side-stepping and (c) based on unicycle model via turning. Both robots executed the task with guaranteed safety. A video of the experiments can be found at [93].

safely, as guaranteed by Theorem 3.6 since α was selected small enough for the available tracking performance. The value of α was chosen based on the simulated response of the single integrator.

The quadruped was a Unitree A1 robot, shown in Fig. 3.3(b), which has 18 degrees of freedom and 12 actuators. Its position was measured based on odometry assuming the feet do not slip, while joint states were available via built-in encoders. An ID-QP walking controller was realized at 1 kHz loop rate on this robot to track a stable walking gait with prescribed forward and lateral velocities and yaw rate, designed using the concepts in [95]. Individual commands were tracked via the motion primitive framework described in [96]. In the single integrator experiments, the yaw rate was set to zero, while the safe velocity (3.18) with $K_P = 0.1\,s^{-1}$ and $\alpha = 0.2\,s^{-1}$ was tracked by forward- and side-stepping. The quadruped executed the task safely similar to the quadrotor (see Fig. 3.3(b)), although it has fundamentally different full-order dynamic behavior. This indicates the application-agnostic nature of our model-free approach.

Finally, we used the unicycle model (3.16) to achieve safety on the quadruped:

Example 3.8 (Quadruped Safety using a Unicycle Model)**.** *The safety requirements for this system were described as:*

$$h(\mathbf{q}) = \|(x_{\text{obs}} - x, y_{\text{obs}} - y)\| - r - \delta\cos(\psi - \theta), \tag{3.19}$$

where $\theta = \arctan((y_{\text{obs}} - y)/(x_{\text{obs}} - x))$ *is the angle towards the obstacle,* r *is the combined radius of the robot and the obstacle, and* $\delta \in \mathbb{R}_{>0}$ *is a tunable parameter.*

We again use a CBF-QP framework to generate the safe velocities and set the desired forward velocity and yaw rate $\mu_{\text{d}} = (v_{\text{d}}, \omega_{\text{d}})$ *based on the distance* $d_{\text{goal}} = \|(x_{\text{goal}} - x, y_{\text{goal}} - y)\|$ *to the goal as* $v_{\text{d}} = K_v d_{\text{goal}}$ *and* $\omega_{\text{d}} = -K_\omega(\sin\psi - (y_{\text{goal}} - y)/d_{\text{goal}})$.

The safe forward velocity and yaw rate were tracked by the same ID-QP walking controller. Fig. 3.3(c) shows the quadruped traversing the obstacle course with $K_v = 0.08\,s^{-1}$, $K_\omega = 0.4\,s^{-1}$, $\alpha = 0.2\,s^{-1}$, $\delta = 0.5\,m$ *and* $R = 0.5\,m$*. While safety is maintained, the quadruped performs the task with different behavior than in the previous experiment: it walks forward and turns instead of forward- and side-stepping. Still, safety is provably guaranteed in a model-free fashion.*

Conclusion

This section considered a model-free safety-critical control paradigm with wide application to a variety of robots. Safety and CBF synthesis is achieved in Theorem 3.6 and deployed in the provided examples through the use of a hierarchical approach that generates safe velocities and tracks them using a platform-specific tracking controller. Here the exact CBF h is constructed implicitly using the safety requirement h_0 and a Lyapunov function guaranteeing the exponential velocity tracking of the system. In the application of this method, the exact CBF h is not used directly and serves only as an implicit, theoretical tool for guaranteeing safety. Instead, the safety criteria is used to directly synthesize safe velocities in a model-free fashion.

While this method proves to be incredibly useful, it presents a conservative, implicit method for safe-set synthesis as we rely indirectly on a tracking controller to attain safety instead of directly modifying the low-level control actions. The following two sections will explore how the proof method for Theorem 3.6 can instead be used to generate explicit CBFs for the full-order system.

3.4 Synthesizing CBFs for Partially Feedback Linearizable Systems

This section considers methods for explicitly constructing CBFs when the system exhibits a partially feedback linearizable structure. This generalizes the results of

Section 3.2 by extending from fully to partially feedback linearizable systems. It also differs from the approach in Section 3.3 by directly leveraging the structure of the output tracking dynamics to construct a closed-form CBF for the full-order system.

We demonstrate how feedback linearization techniques [97] facilitate CBF synthesis, with particular attention to applications in underactuated robotics. Unlike previous work that either uses a Lyapunov function directly (as in Sections 3.2 and 3.3) or treats the safety constraint as an output function [74], [76], we instead define outputs based on the states relevant to safety, and then use output feedback linearization to construct tracking controllers that guarantee invariance of the corresponding safe set similar to [80]. Specifically, we show that if a system is input-output linearizable with respect to a smooth output function, then under mild regularity conditions, any smooth safety criteria (expressed as an inequality constraint) on the output can be extended to be a valid CBF for the full-order system.

The contributions of this section are as follows:

- A constructive framework for synthesizing CBFs for high-dimensional and underactuated systems, with explicit characterization of the required system properties. In particular, we establish the existence of the smooth controller required for the initial step in the CBF backstepping procedure.
- Numerical examples demonstrating the design of CBFs for a variety of underactuated robotic systems, including the first hardware demonstration of CBF backstepping.

The text for this section was adapted from:

> M. H. Cohen, R. K. Cosner, and A. D. Ames, "Constructive safety-critical control: Synthesizing control barrier functions for partially feedback linearizable systems," *IEEE Control Systems Letters*, pp. 2229–2234, 2024. DOI: 10.1109/LCSYS.2024.3412003,

A video for this section can be found at [98].

Contructing CBFs for Feedback Linearizable Systems

The objective of this section is to systematically construct CBFs using methods from feedback linearization [97]. Central to our approach is the notion of relative degree:

Definition 3.9 (Relative Degree [97]). *A smooth function* $\mathbf{y}_{\text{out}} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ *is said to have* relative degree[5] $r \in \mathbb{N}$ *with respect to* (2.1) *on an open set* $\mathcal{E} \subseteq \mathbb{R}^{n_x}$ *if for all* $\mathbf{x} \in \mathcal{E}$*:*

$$\textit{i)} \quad L_{\mathbf{g}} L_{\mathbf{f}}^{i} \mathbf{y}_{\text{out}}(\mathbf{x}) = \mathbf{0}, \quad \forall i \in \{0, \dots, r-2\}, \tag{3.20}$$

$$\textit{ii)} \quad \text{rank}(L_{\mathbf{g}} L_{\mathbf{f}}^{r-1} \mathbf{y}_{\text{out}}(\mathbf{x})) = n_y. \tag{3.21}$$

Given a smooth output $\mathbf{y}_{\text{out}} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ that has relative degree $r \in \mathbb{N}$ on $\mathcal{E} \subseteq \mathbb{R}^{n_x}$, we can define:

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \vdots \\ \boldsymbol{\eta}_r \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{y}_{\text{out}}(\mathbf{x}) \\ \vdots \\ L_{\mathbf{f}}^{r-1} \mathbf{y}_{\text{out}}(\mathbf{x}) \end{bmatrix} \in \mathbb{R}^{n_y r}, \tag{3.22}$$

noting that the output dynamics are then given by:

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \dot{\boldsymbol{\eta}}_1 \\ \vdots \\ \dot{\boldsymbol{\eta}}_{r-1} \\ \dot{\boldsymbol{\eta}}_r \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta}_2 \\ \vdots \\ L_{\mathbf{f}}^{r-1} \mathbf{y}_{\text{out}}(\mathbf{x}) \\ L_{\mathbf{f}}^{r} \mathbf{y}_{\text{out}}(\mathbf{x}) + L_{\mathbf{g}} L_{\mathbf{f}}^{r-1} \mathbf{y}_{\text{out}}(\mathbf{x}) \mathbf{u} \end{bmatrix}. \tag{3.23}$$

The problem that we tackle in this section for this system with outputs of relative degree r is:

Problem 3.10. *Consider a safety requirement defined using a smooth function* $\psi : \mathbb{R}^{n_y} \to \mathbb{R}$ *on the outputs* $\mathbf{y}_{\text{out}}$ *as*

$$\mathcal{C}_0 \triangleq \{\mathbf{x} \in \mathbb{R}^{n_x} : \psi(\mathbf{y}_{\text{out}}(\mathbf{x})) = h_0(\mathbf{x}) \geq 0\}, \tag{3.24}$$

where $h_0(\mathbf{x})$ *is then defined as the composition of* $\psi(\cdot)$ *and* $\mathbf{y}_{\text{out}}(\cdot)$*. Since* $\mathcal{C}_0$ *is not necessarily control invariant and* h_0 *is not necessarily a CBF, our goal is to construct a CBF* h *that defines a safe subset* $\mathcal{C} \subseteq \mathcal{C}_0$ *so that enforcing forward invariance of* $\mathcal{C}$ *leads to satisfaction of the safety requirement.*

To accomplish this we first establish that, when the open-loop dynamics (2.1) are partially feedback linearizable with respect to a $\mathbf{y}_{\text{out}}$ then, under mild regularity conditions, one may construct the desired CBF to solve Problem 3.10. The following lemma outlines the regularity conditions that ψ must satisfy.

[5] A vector-valued output may have different relative degrees for each of its components. For simplicity of notation, we focus on outputs whose components share the same relative degree

Lemma 3.11. *Let $\psi : \mathbb{R}^{n_y} \to \mathbb{R}$ be a smooth function defining a set $\mathcal{C}_{\mathbf{y}} \subset \mathbb{R}^{n_y}$ as:*

$$\mathcal{C}_{\mathbf{y}} \triangleq \{\mathbf{y}_{\text{out}} \in \mathbb{R}^{n_y} : \psi(\mathbf{y}_{\text{out}}) \geq 0\}. \tag{3.25}$$

Let $\mathcal{D}_{\mathbf{y}} \supset \mathcal{C}_{\mathbf{y}}$ be an open set and suppose that ψ satisfies the regularity condition:

$$\frac{\partial \psi}{\partial \mathbf{y}_{\text{out}}}(\mathbf{y}_{\text{out}}) \neq \mathbf{0}, \quad \forall \mathbf{y}_{\text{out}} \in \mathcal{D}_{\mathbf{y}} \setminus \text{Int}(\mathcal{C}_{\mathbf{y}}). \tag{3.26}$$

Then, for any smooth $\alpha \in \mathcal{K}_{\infty}^{e}$ there exists a smooth $\mathbf{k}_y : \mathcal{D}_{\mathbf{y}} \to \mathbb{R}^{n_y}$ such that for all $\mathbf{y}_{\text{out}} \in \mathcal{D}_{\mathbf{y}}$:

$$\frac{\partial \psi}{\partial \mathbf{y}_{\text{out}}}(\mathbf{y}_{\text{out}})\mathbf{k}_y(\mathbf{y}_{\text{out}}) > -\alpha(\psi(\mathbf{y}_{\text{out}})). \tag{3.27}$$

The full proof of this Lemma can be found in [49, Lemma 1].

The conditions in Lemma 3.11 are equivalent to the statement that ψ is a CBF for a single integrator $\dot{\mathbf{y}}_{\text{out}} = \mathbf{u}$, a very mild requirement similar to that required for the model-free results of the previous section. Notably, unlike [76] this does not require $\mathbf{x} \mapsto \psi(\mathbf{y}_{\text{out}}(\mathbf{x}))$, i.e., $h_0(\mathbf{x})$), to have a uniform relative degree on $\mathcal{C}_0$, which would be overly restrictive[6]. Instead, we require that the output $\mathbf{y}_{\text{out}}$ have a relative degree, which is less restrictive[7]. More intuitively, the *outputs* defining safety need to have a relative degree, not the value of safety itself.

Next consider the output dynamics (3.23) which are in *strict feedback form* and are thus amenable to backstepping-based designs. Following the backstepping procedure in [80], we propose the CBF candidate:

$$\begin{aligned} h(\mathbf{x}) \triangleq& \psi(\mathbf{y}_{\text{out}}(\mathbf{x})) - \sum_{i=1}^{r-1} \tfrac{1}{2\mu_i} \|L_{\mathbf{f}}^{i}\mathbf{y}_{\text{out}}(\mathbf{x}) - \mathbf{k}_i(\boldsymbol{\zeta}_i(\mathbf{x}))\|^2, \\ =& \psi(\boldsymbol{\eta}_1) - \sum_{i=1}^{r-1} \tfrac{1}{2\mu_i} \|\boldsymbol{\eta}_{i+1} - \mathbf{k}_i(\boldsymbol{\zeta}_i)\|^2, \end{aligned} \tag{3.28}$$

where ψ defines $\mathcal{C}_{\mathbf{y}} \subset \mathbb{R}^{n_y}$ as in (3.25), $\mu_i \in \mathbb{R}_{>0}$ for $i \in \{1, \ldots, r-1\}$, $\boldsymbol{\zeta}_j \triangleq (\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \ldots, \boldsymbol{\eta}_j) \in \mathbb{R}^{n_y j}$, $\mathbf{k}_y : \mathcal{D}_{\mathbf{y}} \to \mathbb{R}^{n_y}$ is any smooth function satisfying (3.27)

[6]Indeed, the gradient of relevant safety constraints often vanish at points in Int($\mathcal{C}$) [78], [87], [99].

[7]That is, $\mathbf{y}_{\text{out}}(\mathbf{x})$ may have a relative degree even when $\psi(\mathbf{y}_{\text{out}}(\mathbf{x}))$ does not. A simple example illustrating this point is the double integrator with state $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$, output $\mathbf{y}_{\text{out}}(\mathbf{x}) = x_1$, and constraint $\psi(\mathbf{y}_{\text{out}}(\mathbf{x})) = 1 - x_1^2$. This phenomenon is also present in the following examples and may arise when $\mathcal{C}_{\mathbf{y}}$ from (3.25) is a compact set (cf. [87, Footnote 4]).

for all $\boldsymbol{\eta}_1 \in \mathcal{D}_{\mathbf{y}} \supset \mathcal{C}_{\mathbf{y}}$ for a smooth globally Lipschitz $\alpha \in \mathcal{K}_\infty^e$, and:

$$\begin{aligned}\mathbf{k}_2(\boldsymbol{\zeta}_2) \triangleq& \dot{\mathbf{k}}_1(\boldsymbol{\zeta}_2) + \mu_1 \frac{\partial \psi}{\partial \boldsymbol{\eta}_1}(\boldsymbol{\eta}_1)^\top - \frac{\lambda_1}{2}(\boldsymbol{\eta}_2 - \mathbf{k}_1(\boldsymbol{\eta}_1)) \\ \mathbf{k}_{i+1}(\boldsymbol{\zeta}_{i+1}) \triangleq& \dot{\mathbf{k}}_i(\boldsymbol{\zeta}_{i+1}) - \mu_i(\boldsymbol{\eta}_i - \mathbf{k}_{i-1}(\boldsymbol{\zeta}_{i-1})) - \frac{\lambda_i}{2}(\boldsymbol{\eta}_{i+1} - \mathbf{k}_i(\boldsymbol{\zeta}_i)),\end{aligned} \tag{3.29}$$

for each $i \in \{2, \dots, r-2\}$ and where $\lambda_i > 0$ for $i \in \{1, \dots, r-2\}$. The CBF candidate in (3.28) defines a set $\mathcal{C}$ as in (2.29), which satisfies $\mathcal{C} \subset \mathcal{C}_0$. Before proceeding, it will be useful to define $\mathcal{D}_{\mathbf{x}} \triangleq \{\mathbf{x} \in \mathbb{R}^{n_x} : \mathbf{y}_{\text{out}}(\mathbf{x}) \in \mathcal{D}_{\mathbf{y}}\}$, where $\mathcal{D}_{\mathbf{y}} \subset \mathbb{R}^{n_y}$ is defined as in Lemma 3.11. We now illustrate that when $\mathbf{y}_{\text{out}}$ has a relative degree on $\mathcal{C}$ and ψ satisfies (3.26), then (3.28) is a CBF for system (2.1).

Theorem 3.12. *Consider the open-loop system* (2.1) *with smooth output* $\mathbf{y}_{\text{out}} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$*, the output constraint* $\psi : \mathbb{R}^{n_y} \to \mathbb{R}$ *defining a constraint set* $\mathcal{C}_0 \subset \mathbb{R}^{n_x}$ *as in* (3.24)*, and the CBF candidate* $h : \mathbb{R}^{n_x} \to \mathbb{R}$ *from* (3.28) *defining a set* $\mathcal{C} \subset \mathcal{C}_0$ *as in* (2.29)*.* ***If*** ψ *satisfies* (3.26) *on a set* $\mathcal{D}_{\mathbf{y}} \supset \mathcal{C}_{\mathbf{y}}$*, with* $\mathcal{C}_{\mathbf{y}} \subset \mathbb{R}^{n_y}$ *as in* (3.25)*,* $\mathbf{y}_{\text{out}}$ *has relative degree* r *on a set* $\mathcal{E} \supset \mathcal{C}$ *satisfying* $\mathcal{E} \subseteq \mathcal{D}_{\mathbf{x}}$*, and* $\lambda_i \geq \mathfrak{L}_\alpha$ *for each* $i \in \{1, \dots, r-2\}$*, where* $\mathfrak{L}_\alpha$ *is a Lipschitz constant of* $\alpha \in \mathcal{K}_\infty^e$ *from* (3.27)*,* ***then*** h *is a CBF for* (2.1) *on* $\mathcal{C}$*. Moreover, any locally Lipschitz controller* $\mathbf{k} : \mathcal{E} \to \mathbb{R}^{n_u}$ *that renders* $\mathcal{C}$ *forward invariant for the closed-loop system* (2.1) *ensures that* $\mathbf{x}(t) \in \mathcal{C}_0$ *for all* $t \in I(\mathbf{x}_0)$*.*

Proof. The proof follows a similar argument to that of [80, Thm. 5]. Since $\mathbf{y}_{\text{out}}$ has relative degree r on $\mathcal{E}$, the matrix $L_{\mathbf{g}} L_{\mathbf{f}}^{r-1} \mathbf{y}_{\text{out}}(\mathbf{x}) \in \mathbb{R}^{n_y \times n_u}$ has rank n_y and is thus right pseudo-invertible for each $\mathbf{x} \in \mathcal{E}$. Now, note that since $\mathcal{C} \subset \mathcal{E} \subseteq \mathcal{D}$ and $\boldsymbol{\eta}_1 \mapsto \mathbf{k}_y(\boldsymbol{\eta}_1)$ satisfies (3.27) for all $\boldsymbol{\eta}_1 \in \mathcal{D}_{\mathbf{y}}$, $\mathbf{x} \mapsto \mathbf{k}_y(\mathbf{y}_{\text{out}}(\mathbf{x}))$ satisfies (3.27) for all $\mathbf{x} \in \mathcal{E} \subseteq \mathcal{D}$, where $\mathbf{k}_y$ exists since ψ satisfies the conditions of Lemma 3.11. It then follows that since $\lambda_i \geq \mathfrak{L}_\alpha$ for each $i \in \{1, \dots, r-2\}$, each $\mathbf{k}_i$ satisfies the same conditions as those in the proof of [80, Thm. 5], which implies that the CBF candidate h in (3.28) satisfies the same conditions as those in [80, Sec. IV]. Hence, by following the same steps as in the proof of [80, Thm. 5], one may show that the smooth feedback controller:

$$\begin{aligned}\mathbf{k}(\mathbf{x}) \triangleq L_{\mathbf{g}} L_{\mathbf{f}}^{r-1} \mathbf{y}_{\text{out}}(\mathbf{x})^\dagger \Big[\quad & \dot{\mathbf{k}}_{r-1}(\boldsymbol{\eta}(\mathbf{x})) - L_{\mathbf{f}}^r \mathbf{y}_{\text{out}}(\mathbf{x}) \\ & - \mu_{r-1}\Big(\boldsymbol{\eta}_{r-1}(\mathbf{x}) - \mathbf{k}_{r-2}(\boldsymbol{\zeta}_{r-2}(\mathbf{x}))\Big) \\ & - \frac{\lambda_{r-1}}{2}\Big(\boldsymbol{\eta}_r(\mathbf{x}) - \mathbf{k}_{r-1}(\boldsymbol{\zeta}_{r-1}(\mathbf{x}))\Big) \quad \Big],\end{aligned} \tag{3.30}$$

where $(\cdot)^\dagger$ denotes the right psuedo-inverse and $\lambda_{r-1} \geq \mathfrak{L}_\alpha$, satisfies $\dot{h}(\mathbf{x}, \mathbf{k}(\mathbf{x})) > -\alpha(h(\mathbf{x}))$ for all $\mathbf{x} \in \mathcal{E}$, where α is from (3.27). Thus, for all $\mathbf{x} \in \mathcal{E}$, the CBF inequality (2.33) holds:

$$\sup_{\mathbf{u} \in \mathbb{R}^{n_u}} \dot{h}(\mathbf{x}, \mathbf{u}) \geq \dot{h}(\mathbf{x}, \mathbf{k}(\mathbf{x})) > -\alpha(h(\mathbf{x})),$$

therefore h is a CBF for (2.1) on $\mathcal{C} \subset \mathcal{E}$. Since $\mathcal{C} \subset \mathcal{C}_0$ any locally Lipschitz controller enforcing the forward invariance of $\mathcal{C}$ ensures that $\mathbf{x}(t) \in \mathcal{C}_0$ for all $t \in I(\mathbf{x}_0)$ as well. □

Theorem 3.12 highlights the interplay between the output $\mathbf{y}_{\text{out}}$, the output safety criteria ψ, the system's actuation capabilities, and the ability to construct CBFs. By ensuring that $\mathbf{y}_{\text{out}}$ has a relative degree on $\mathcal{E} \supset \mathcal{C}$, (2.1) may be partially transformed into a strict feedback system (3.23) on $\mathcal{E}$, enabling the application of backstepping [80] to construct a CBF. Theorem 3.12 characterizes the requirements on ψ, $\mathcal{C}_0$, and $\mathbf{y}_{\text{out}}$ for such techniques to be applicable to general control affine systems (2.1), complimenting the ideas introduced in [80]. Theorem 3.12 is, to our knowledge, the first to make the explicit connection between more general outputs and the constructions of CBFs. This connection has important practical implications as it enables the application of such ideas to a broader class of systems than those originally considered in [80]. Moreover, by not treating ψ as an output directly this construction overcomes the restrictive uniform relative degree requirements on ψ present in most high relative degree CBF techniques.

Constructing CBFs for Underactuated Robotic Systems

Next we apply this feedback-linearization-based method for CBF synthesis to underactuated robotic systems with the structured Euler-Lagrange rigid body dynamics as introduced in (3.5) and (3.6).

First, consider a twice continuously differentiable output $\mathbf{y}_{\text{out}} : \mathcal{Q} \to \mathbb{R}^{n_y}$, which is used to define an output constraint $\psi : \mathbb{R}^{n_y} \to \mathbb{R}$ and safety criteria set:

$$\mathcal{C}_0 \triangleq \{\mathbf{q} \in \mathcal{Q} : \psi(\mathbf{y}_{\text{out}}(\mathbf{q})) \geq 0\}, \tag{3.31}$$

defined in the configuration space $\mathcal{Q}$ of (3.5). Differentiating the output $\mathbf{y}_{\text{out}}$ twice leads to $\ddot{\mathbf{y}}_{\text{out}} = L_{\mathbf{f}}^2 \mathbf{y}_{\text{out}}(\mathbf{q}, \dot{\mathbf{q}}) + \frac{\partial \mathbf{y}_{\text{out}}}{\partial \mathbf{q}}(\mathbf{q}) \mathbf{D}(\mathbf{q})^{-1} \mathbf{B}(\mathbf{q}) \mathbf{u}$. Importantly, we see that the $n_y \times n_u$ "decoupling" matrix:

$$\mathbf{A}(\mathbf{q}) \triangleq L_{\mathbf{g}} L_{\mathbf{f}} \mathbf{y}_{\text{out}}(\mathbf{q}) = \frac{\partial \mathbf{y}_{\text{out}}}{\partial \mathbf{q}}(\mathbf{q}) \mathbf{D}(\mathbf{q})^{-1} \mathbf{B}(\mathbf{q}), \tag{3.32}$$

depends only on the configuration $\mathbf{q}$, implying that the relative degree depends only on the configuration. When $\mathbf{y}_{\text{out}}$ has relative degree 2, as is often the case for robotic systems, the CBF candidate from (3.28) simplifies to:

$$h(\mathbf{x}) = \psi(\mathbf{y}_{\text{out}}(\mathbf{q})) - \frac{1}{2\mu}\left\| \frac{\partial \mathbf{y}_{\text{out}}}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{q}} - \mathbf{k}_\psi(\mathbf{y}_{\text{out}}(\mathbf{q}))\right\|^2, \tag{3.33}$$

where $\mu > 0$ and $\mathbf{k}_\psi : \mathcal{D}_{\mathbf{y}} \to \mathbb{R}^{n_y}$ is any continuously differentiable function satisfying (3.27) for all $\mathbf{y}_{\text{out}}(\mathbf{q}) \in \mathcal{D}_{\mathbf{y}} \supset \mathcal{C}_{\mathbf{y}}$. The following corollary illustrates that (3.33) is a CBF for (3.6) provided ψ satisfies (3.26) and (3.32) has full row rank on a set containing $\mathcal{C}$.

Corollary 3.13. *Consider the robotic system* (3.6) *with twice continuously differentiable output* $\mathbf{y}_{\text{out}} : \mathcal{Q} \to \mathbb{R}^{n_y}$*, the configuration safety criteria* $\psi : \mathbb{R}^{n_y} \to \mathbb{R}$ *defining a set* $\mathcal{C}_0 \subset \mathcal{Q}$ *as in* (3.31)*, and the CBF candidate* $h : \mathbb{R}^{n_x} \to \mathbb{R}$ *as in* (3.33) *defining a set* $\mathcal{C} \subset \mathcal{C}_0 \times \mathbb{R}^{n_x}$ *as in* (2.29)*. Provided that* ψ *satisfies* (3.26) *on a set* $\mathcal{D}_{\mathbf{y}} \supset \mathcal{C}_{\mathbf{y}}$*, with* $\mathcal{C}_{\mathbf{y}} \subset \mathbb{R}^{n_y}$ *as in* (3.25)*,* $\text{rank}(\mathbf{A}(\mathbf{q})) = n_y$ *for all* $\mathbf{q} \in \mathcal{E}_1 \supset \mathcal{C}_0$ *with* $\mathcal{E} \triangleq \mathcal{E}_1 \times \mathbb{R}^{n_x} \subseteq \mathcal{D}$*, then* h *is a CBF for* (3.6)*. Moreover, any locally Lipschitz controller* $\mathbf{k} : \mathcal{E} \to \mathbb{R}^{n_u}$ *that renders* $\mathcal{C}$ *forward invariant for the closed-loop system* (2.2) *ensures that* $\mathbf{q}(t) \in \mathcal{C}_0$ *for all* $t \in I(\mathbf{x}_0)$ *when* $h(\mathbf{x}_0) \geq 0$*.*

Proof. As $\text{rank}(\mathbf{A}(\mathbf{q})) = n_y$ for all $\mathbf{q} \in \mathcal{E}_1$, $\mathbf{y}_{\text{out}}$ has relative degree 2 on $\mathcal{E}$ and since $\mathcal{C} \subset \mathcal{C}_0 \times \mathbb{R}^{n_x}$ and $\mathcal{C}_0 \subset \mathcal{E}_1$, we have $\mathcal{C} \subset \mathcal{E}$. Finally, since $\mathcal{E} \subseteq \mathcal{D}$ the conditions of Theorem 3.12 hold, implying that h as in (3.33) is a CBF for (3.6) on $\mathcal{C}$. □

Finally, we demonstrate our method on a real-world quadrotor robot that, to the best of our knowledge, constitutes the first hardware demonstration of CBF backstepping techniques.

Example 3.14 (Quadrotor Hardware Experiments)**.** *The quadrotor hardware platform is described in [59] and is modeled as a control affine system* (2.1) *with state* $\mathbf{x} = (\mathbf{p}, q, \mathbf{v}) \in \mathbb{R}^3 \times S^3 \times \mathbb{R}^3$ *representing the position* $\mathbf{p}$*, orientation* q *(represented as a quaternion), and velocity* $\mathbf{v}$*, and control input* $\mathbf{u} = (\boldsymbol{\omega}, \tau) \in \mathbb{R}^3 \times \mathbb{R}$*, where* $\boldsymbol{\omega}$ *is the angular rate and* τ *is the thrust. A full expression of the dynamics can be found in [59]. Our control objective is to keep the quadrotor's height above* $z_{\min}$*, where* $\mathbf{p} = (x, y, z)$ *and* z *denotes the quadrotor's height. To this end, we*

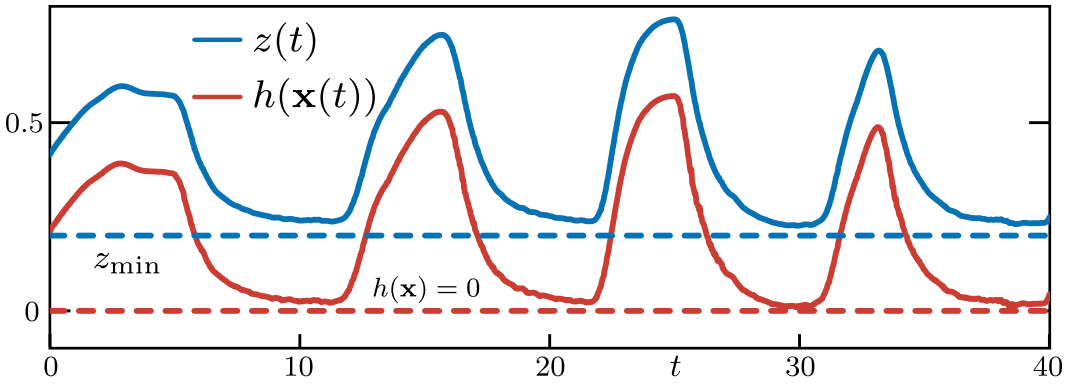


Figure 3.4. Experimental results for Example 3.14 illustrating the evolution of the quadrotor's height (blue) and CBF (red).

choose our output[8] *as* $\mathbf{y}_{\text{out}}(\mathbf{x}) = (z, q_x, q_y)$*, where* q_x *and* q_y *are components of the quaternion such that* $q = q_w + q_x i + q_y j + q_z k$*. Given this output, we define* $\psi(\mathbf{y}_{\text{out}}(\mathbf{x})) = z - z_{\min} - \lambda(2q_x^2 + 2q_y^2)$ *where* $\lambda > 0$*. This constraint ensures that* $\psi(\mathbf{y}_{\text{out}}(\mathbf{x})) \geq 0 \implies z \geq z_{\min}$ *and requires the quadrotor's orientation to remain level when* $z = z_{\min}$*. Leveraging the constructions in Theorem 3.12, this leads to the CBF candidate:*

$$h(\mathbf{x}) = \psi(\mathbf{y}_{\text{out}}(\mathbf{x})) - \tfrac{1}{2}\|L_{\mathbf{f}}\mathbf{y}_{\text{out}}(\mathbf{x}) - \mathbf{k}_1(\mathbf{y}_{\text{out}}(\mathbf{x}))\|^2,$$

where $\mathbf{k}_y : \mathbb{R}^3 \to \mathbb{R}^3$ *is defined using Sontag's "universal" controller [12] (see [49] for additional details). This CBF is used to construct a safety filter as in* (2.36)*, where* $\mathbf{k}_{\text{des}}$ *corresponds to commands given via joystick that lift the quadrotor up before lowering it to the ground. Applying this safety filter to the system produces the results in Fig. 3.4, where* z *remains above* $z_{\min}$ *and* h *remains positive for all time.*

A video of these experiments can be found in [98].

Conclusion

This section proposed a method for CBF construction that generalizes the feedback-linearization and Lyapunov-based methods of Section 3.2 by considering partial feedback linearizability, system outputs, and relative degree, resulting in a constructive method where we build a CBF (3.33) using the safety requirement alongside

[8] For the model described in [59], the first component of $\mathbf{y}_{\text{out}}$ has relative degree two whereas the second and third have relative degree one. Theorem 3.12 can be modified to account for such a situation at the expense of additional notation by transforming the output dynamics into a mixed relative degree cascaded system or a dual relative degree system as will be discussed in Section 3.5., but a formal presentation of such results is omitted here in the interest of space.

a negative definite function that characterizes stability, similar to the methods in Section 3.2 and 3.3. However, the results of this section differ substantially in that they incorporate the safety criteria directly and provide an explicit construction of the full-order CBF which can be directly used to modify the low-level control input.

3.5 Synthesizing CBFs for Dual Relative Degree Systems

In many robotics applications, safety is specified with respect to position outputs in Euclidean space [48], [84]. Yet, these outputs often fail to capture the full state dependencies relevant to safety, particularly in systems with orientation-dependent dynamics. For example, in the example in the previous section (Ex. 3.14), the safety requirement set $\mathcal{C}_0$ must be carefully constructed to account for system orientation. This limitation is common when safety constraints are expressed in terms of positional outputs, but the underlying system requires coordination between position and orientation for safe execution.

To address this issue, this section introduces a new method for synthesizing CBFs for a special but highly relevant class of systems that exhibit *dual relative degree*—that is, systems in which different components of the input affect the output at different orders of differentiation. This structure appears in many underactuated robotic systems, including unicycles and quadrotors. We extend the explicit CBF synthesis approach from the previous section to this class of systems by leveraging tracking controllers similar to those used for differentially flat systems. These controllers provide certificates of the system's ability to stabilize its orientation, playing a role analogous to the velocity-tracking guarantees used in Section 3.3.

The contributions of this section are as follows:

- A definition of the *dual relative degree* property.
- A constructive framework for synthesizing CBFs for these systems.
- In-depth case studies demonstrating the utility of this CBF synthesis method, including hardware demonstrations on quadrupeds and quadrotors.

The text for this section is adapted from:

> G. Bahati, R. K. Cosner, M. H. Cohen, R. M. Bena, and A. D. Ames, "Control barrier function synthesis for nonlinear systems with dual relative degree," *submitted to the 2025 IEEE 64th Conference*

on Decision and Control (CDC), 2025. [Online]. Available: `https://arxiv.org/pdf/2504.00397`,

A video for this section can be found at [100].

Dual Relative Degree Systems

This section will leverage the same relative degree properties (Def. 3.9) and output dynamics (3.23) as in Section 3.4. However, we now consider systems that do not have a valid relative degree and instead satisfy a dual relative degree property that captures the situation in which inputs can influence the outputs at two different orders of differentiation.

Before defining dual relative degree we note that, given an output $\mathbf{y}_{\text{out}} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ with relative degree r as in Definition 3.9, the output dynamics can be written as:

$$\frac{\mathrm{d}}{\Delta_t}\boldsymbol{\eta}(\mathbf{x}) = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I}_{n_y(r-1)} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{A}} \boldsymbol{\eta}(\mathbf{x}) + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{n_y} \end{bmatrix}}_{\mathbf{B}} \mathbf{v} \tag{3.34}$$

$$\mathbf{v} \triangleq L_{\mathbf{f}}^{r}\mathbf{y}_{\text{out}}(\mathbf{x}) + L_{\mathbf{g}}L_{\mathbf{f}}^{r-1}\mathbf{y}_{\text{out}}(\mathbf{x})\mathbf{u}, \tag{3.35}$$

where (3.35) is viewed as an input to (3.34). With $\mathbf{v}$ as the input, the output dynamics in (3.34) are a chain of integrators and techniques such as [101] may be employed to construct CBFs. Importantly, when $\mathbf{y}_{\text{out}}$ has relative degree r, any controller $\mathbf{v} = \hat{\mathbf{k}}(\boldsymbol{\eta})$ designed for (3.34) may be transferred back to (2.1) via:

$$\mathbf{u} = L_{\mathbf{g}}L_{\mathbf{f}}^{r-1}\mathbf{y}_{\text{out}}(\mathbf{x})^{\dagger}\left[\hat{\mathbf{k}}(\boldsymbol{\eta}(\mathbf{x})) - L_{\mathbf{f}}^{r}\mathbf{y}_{\text{out}}(\mathbf{x})\right], \tag{3.36}$$

where the right psuedo-inverse $L_{\mathbf{g}}L_{\mathbf{f}}^{r-1}\mathbf{y}_{\text{out}}(\mathbf{x})^{\dagger}$ exists given (3.21). When the output coordinates $\boldsymbol{\eta}$ are physically relevant to the original safety specification for the open-loop system (2.1), the method in Section 3.4 can be employed to synthesize CBFs for this system. In general, however, the outputs relevant to the safety specification for (2.1) may not have a valid relative degree, precluding the ability to directly transfer inputs from the output integrator system (3.34) back to the nonlinear system (2.1) via the controller transformation (3.36). This motivates our analysis of dual relative degree systems that provides a framework for relating inputs of the output integrator system (3.34) to those of the nonlinear system (2.1) under weaker conditions than relative degree (Def. 3.9) and which allows us to synthesize of CBFs for practically relevant systems.

To facilitate our approach, we assume that the open-loop system dynamics (2.1) has multiple control inputs, i.e., $n_u \geq 2$, and thus may be written as:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \underbrace{\mathbf{g}_1(\mathbf{x})\mathbf{u}_1 + \mathbf{g}_2(\mathbf{x})\mathbf{u}_2}_{\mathbf{g}(\mathbf{x})\mathbf{u}}, \tag{3.37}$$

where $\mathbf{u}_1 \in \mathbb{R}^{n_{u2}}$, $\mathbf{u}_2 \in \mathbb{R}^{n_{u2}}$ such that $n_u = n_{u1} + n_{u2}$ with $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2)$, while $\mathbf{g}_1 : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x \times n_{u1}}$ and $\mathbf{g}_2 : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x \times n_{u2}}$ decompose $\mathbf{g}$ as $\mathbf{g}(\mathbf{x}) = \begin{bmatrix} \mathbf{g}_1(\mathbf{x}) & \mathbf{g}_2(\mathbf{x}) \end{bmatrix}$. Given these dynamics and an output $\mathbf{y}_{\text{out}} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ for (3.37), the inputs affect these outputs via:

$$\underbrace{L_{\mathbf{g}} L_{\mathbf{f}}^i \mathbf{y}_{\text{out}}(\mathbf{x})}_{n_y \times n_u} = \begin{bmatrix} \underbrace{L_{\mathbf{g}_1} L_{\mathbf{f}}^i \mathbf{y}_{\text{out}}(\mathbf{x})}_{p \times n_{u1}} & \underbrace{L_{\mathbf{g}_2} L_{\mathbf{f}}^i \mathbf{y}_{\text{out}}(\mathbf{x})}_{p \times n_{u2}} \end{bmatrix}. \tag{3.38}$$

Rather than requiring $\mathbf{y}_{\text{out}}$ to have a relative degree, we will require it to have a *dual relative degree*, defined as follows:

Definition 3.15. *(Dual Relative Degree) A multi-input system* (3.37) *with smooth output* $\mathbf{y}_{\text{out}} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ *is said to have dual relative degree* $(r, q) \in \mathbb{N} \times \mathbb{N}$ *if* (3.20) *holds and for all* $\mathbf{x} \in \mathbb{R}^{n_x}$*:*

$$L_{\mathbf{g}_2} L_{\mathbf{f}}^{r-1} \mathbf{y}_{\text{out}}(\mathbf{x}) = \mathbf{0}, \tag{3.39}$$

$$\text{rank}(L_{\mathbf{g}_1} L_{\mathbf{f}}^{r-1} \mathbf{y}_{\text{out}}(\mathbf{x})) = n_{u1}, \tag{3.40}$$

$$\text{rank}(L_{\mathbf{g}_2} L_{\mathbf{f}}^{q-1} L_{\mathbf{g}_1} L_{\mathbf{f}}^{r-1} \mathbf{y}_{\text{out}}(\mathbf{x})) = n_{u2}. \tag{3.41}$$

Dual relative degree systems characterize those whose inputs influence the output at two different levels of differentiation[9], and capture systems such as unicycles and quadrotors.

When $\mathbf{y}_{\text{out}}$ has relative degree r, the controller (3.36) can be used to apply the controller $\hat{\mathbf{k}} : \mathbb{R}^{n_y r} \to \mathbb{R}^{n_y}$ to the linearized output dynamics. However, when $\mathbf{y}_{\text{out}}$ *does not* have a relative degree, there does not exist a one-to-one correspondence between inputs of (3.36) and (3.37). Despite this, if (3.37) has a dual relative degree, then given the desired controller $\hat{\mathbf{k}} : \mathbb{R}^{n_y r} \to \mathbb{R}^{n_y}$ for the linearized output dynamics, we can find the input $\mathbf{u}_1$ which actuates the outputs in the manner closest to that of $\hat{\mathbf{k}}$ via least-squares minimization:

$$\begin{aligned} \mathbf{k}_1(\mathbf{x}) :=& \underset{\mathbf{u}_1 \in \mathbb{R}^{n_{u1}}}{\text{argmin}} \left\| L_{\mathbf{f}}^r \mathbf{y}_{\text{out}}(\mathbf{x}) + L_{\mathbf{g}_1} L_{\mathbf{f}}^{r-1} \mathbf{y}_{\text{out}}(\mathbf{x}) \mathbf{u}_1 - \hat{\mathbf{k}}(\boldsymbol{\eta}(\mathbf{x})) \right\|^2 \\ =& L_{\mathbf{g}_1} L_{\mathbf{f}}^{r-1} \mathbf{y}_{\text{out}}(\mathbf{x})^\dagger \left[\hat{\mathbf{k}}(\boldsymbol{\eta}(\mathbf{x})) - L_{\mathbf{f}}^r \mathbf{y}_{\text{out}}(\mathbf{x}) \right], \end{aligned} \tag{3.42}$$

[9]While we will not explicitly leverage (3.41), it is often implicit in our other assumptions (e.g., on the existence of a tracking control Lyapunov function in Def. 3.16) and is thus included to better characterize the systems to which our approach applies.

where $L_{\mathbf{g}_1} L_{\mathbf{f}}^{r-1} \mathbf{y}_{\text{out}}(\mathbf{x})^\dagger$ is the left pseudo-inverse, which exists under the rank assumption (3.40) from Def. 3.15. Taking $\mathbf{u}_1 = \mathbf{k}_1(\mathbf{x})$ produces the partial closed-loop system dynamics:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}_1(\mathbf{x})\mathbf{k}_1(\mathbf{x}) + \mathbf{g}_2(\mathbf{x})\mathbf{u}_2 =: \mathbf{f}_1(\mathbf{x}) + \mathbf{g}_2(\mathbf{x})\mathbf{u}_2. \tag{3.43}$$

Although $\mathbf{k}_1(\mathbf{x})$ produces inputs closest to $\hat{\mathbf{k}}(\boldsymbol{\eta}(\mathbf{x}))$, it may not be able to completely eliminate the error between the output actuation $\mathbf{v}$ in (3.35) and the desired linear actuation $\hat{\mathbf{k}}(\boldsymbol{\eta}(\mathbf{x}))$. We write this error explicitly as:

$$\begin{aligned} \mathbf{e}(\mathbf{x}) &:= L_{\mathbf{f}}^r \mathbf{y}_{\text{out}}(\mathbf{x}) + L_{\mathbf{g}_1} L_{\mathbf{f}}^{r-1} \mathbf{y}_{\text{out}}(\mathbf{x}) \mathbf{k}_1(\mathbf{x}) - \hat{\mathbf{k}}(\boldsymbol{\eta}(\mathbf{x})), \\ &= \left(L_{\mathbf{g}_1} L_{\mathbf{f}}^{r-1} \mathbf{y}_{\text{out}}(\mathbf{x}) L_{\mathbf{g}_1} L_{\mathbf{f}}^{r-1} \mathbf{y}_{\text{out}}(\mathbf{x})^\dagger - \mathbf{I} \right) \left(\hat{\mathbf{k}}(\boldsymbol{\eta}(\mathbf{x})) - L_{\mathbf{f}}^r \mathbf{y}_{\text{out}}(\mathbf{x}) \right) \end{aligned} \tag{3.44}$$

which we will compensate for using Lyapunov-based techniques.

Definition 3.16 (Tracking Control Lyapunov Function)**.** *A continuously differentiable function $V : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ is a tracking control Lyapunov function (CLF) for a control affine system* (2.1) *with respect to error function $\mathbf{e} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_{u1}}$ if there exists $\beta, \lambda > 0$ such that for all $\mathbf{x} \in \mathbb{R}^{n_x}$:*

$$V(\mathbf{x}) \geq \beta \|\mathbf{e}(\mathbf{x})\|^2 \text{ and} \tag{3.45}$$

$$\inf_{\mathbf{u} \in \mathbb{R}^{n_u}} L_{\mathbf{f}} V(\mathbf{x}) + L_{\mathbf{g}} V(\mathbf{x}) \mathbf{u} \leq -\lambda V(\mathbf{x}). \tag{3.46}$$

We will use this notion of a tracking CLF to ensure convergence of our error $\mathbf{e}(\mathbf{x})$ to zero for the partial closed-loop system (3.43).

CBF Synthesis for Dual Relative Degree Systems

We now demonstrate how we can synthesize safety-critical controllers for dual relative degree systems. For this, we consider a dual relative degree system of the form (3.37) with an output $\mathbf{y}_{\text{out}} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ and output dynamics (3.34,3.35). We then consider a desired safe set on the output coordinates $\boldsymbol{\eta}$:

$$\mathcal{C}_0 := \{\mathbf{x} \in \mathbb{R}^{n_x} \mid h_0(\mathbf{x}) = h_\eta(\boldsymbol{\eta}(\mathbf{x})) \geq 0\}, \tag{3.47}$$

and suppose that $h_\eta : \mathbb{R}^{n_y r} \to \mathbb{R}$ is a CBF for the linear system (3.34) with $\mathbf{v}$ viewed as a "virtual" input, similar to the ψ function of previous section and Lemma 3.11. We further assume the existence of a smooth[10] controller $\hat{\mathbf{k}} : \mathbb{R}^{n_y r} \to \mathbb{R}^{n_y}$

[10] As discussed in [87], the existence of CBF (or ISSf-CBF) satisfying (2.33) with a *strict* inequality actually guarantees the existence of a controller, as smooth as the dynamics and CBF, satisfying the corresponding barrier condition. Thus, if h_η is a CBF for (3.34), we may, without loss of generality, construct a smooth feedback controller $\hat{\mathbf{k}}$ satisfying (3.48), with examples of such controllers available in [87].

enforcing the ISSf-CBF condition [79]:

$$\frac{\partial h_\eta}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta}) \left[\mathbf{A}\boldsymbol{\eta} + \mathbf{B}\hat{\mathbf{k}}(\boldsymbol{\eta})\right] > -\gamma h_\eta(\boldsymbol{\eta}) + \frac{1}{\epsilon}\left\|\frac{\partial h_\eta}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta})\mathbf{B}\right\|^2, \tag{3.48}$$

for all $\boldsymbol{\eta} \in \mathbb{R}^{n_y r}$ for some $\gamma, \epsilon > 0$. While h_η is a CBF for (3.34) with relative degree r, it is not necessarily a CBF for (3.37) with dual relative degree (r, q), and it may be impossible to apply $\hat{\mathbf{k}}$ to (3.37) directly.

In a method similar to the hierarchical approach of Section 3.3, we synthesize a CBF for the system with dual relative degree (3.37) by augmenting h_η with a scaled tracking CLF, $\frac{-1}{\mu}V(\mathbf{x})$ for some $\mu > 0$, to account for the error $\mathbf{e}(\mathbf{x})$ between $\mathbf{k}_1$ and $\hat{\mathbf{k}}$. We formally define this construction as:

Definition 3.17 (Dual Relative Degree CBF (DRD-CBF))**.** *Consider system* (3.37) *with dual relative degree* (r, q)*. If* $h_\eta : \mathbb{R}^{n_x} \to \mathbb{R}$ *is a CBF for the linear system* (3.34) *with degree* r*,* $\hat{\mathbf{k}} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ *is a continuously differentiable function satisfying* (3.48) *for some* $\gamma, \epsilon > 0$*, and* $V : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ *is a tracking control Lyapunov function for* (3.43) *with respect to error function* (3.44) *for some* $\beta, \lambda > 0$*, then the function:*

$$h(\mathbf{x}) := h_\eta(\boldsymbol{\eta}(\mathbf{x})) - \frac{1}{\mu}V(\mathbf{x}) \tag{3.49}$$

$$\text{with } \mu > 0 \text{ such that } \lambda \geq \gamma + \frac{\epsilon\mu}{4\beta}, \tag{3.50}$$

is a dual relative degree CBF (DRD-CBF) for (3.37)*.*

The condition in (3.50) dictates the relationship between the convergence rate λ of the tracking CLF, the safety of h_η (determined by $\hat{\mathbf{k}}$) via γ and the ISSf constant ϵ, and the scaling parameters μ and β. Intuitively, the condition (3.50) can be satisfied by increasing the error tracking speed of V by increasing λ, increasing the conservatism of $\hat{\mathbf{k}}$ by decreasing γ and ϵ, or by balancing the scaling of h_η and V via μ or balancing the scaling V and $\mathbf{e}$ via β.

Next, in Theorem 3.18, we prove that all DRD-CBFs are valid CBFs for system (3.37) by showing that the existence of control actions derived from $\hat{\mathbf{k}}$ and the tracking CLF certifies that (3.49) satisfies the CBF constraint (2.33). Thus, we show that DRD-CBFs are a special class of CBFs for dual relative degree systems that can be directly synthesized using a CBF, h_η, for a linear integrator system (3.34) and a tracking CLF, V.

Theorem 3.18. *Consider a system of the form* (3.37) *with dual relative degree* (r, q)*. If* $h : \mathbb{R}^{n_x} \to \mathbb{R}$ *is a DRD-CBF for* (3.37) *as in* (3.49)*, then it is also a CBF and any locally Lipschitz continuous controller satisfying* (2.33) *for* h *renders* $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^{n_x} \mid h(\mathbf{x}) \geq 0\} \subset \mathcal{C}_0$ *safe.*

Proof. Computing the time-derivative of h_η and bounding (omitting dependencies on $\mathbf{x}$ for brevity) we obtain:

$$\dot{h}_\eta = \frac{\partial h_\eta}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta})\left[\mathbf{A}\boldsymbol{\eta} + \mathbf{B}\mathbf{v}\right] = \frac{\partial h_\eta}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta})\left[\mathbf{A}\boldsymbol{\eta} + \mathbf{B}\hat{\mathbf{k}}(\boldsymbol{\eta})\right] + \frac{\partial h_\eta}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta})\mathbf{B}\left[\mathbf{v} - \hat{\mathbf{k}}(\boldsymbol{\eta})\right], \tag{3.51}$$

$$> -\gamma h_\eta(\boldsymbol{\eta}) + \frac{1}{\epsilon}\left\|\frac{\partial h_\eta}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta})\mathbf{B}\right\|^2 - \left\|\frac{\partial h_\eta}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta})\mathbf{B}\right\|\left\|\mathbf{v} - \hat{\mathbf{k}}(\boldsymbol{\eta})\right\|, \tag{3.52}$$

$$\geq -\gamma h_\eta(\boldsymbol{\eta}) - \frac{\epsilon}{4}\left\|\mathbf{v} - \hat{\mathbf{k}}(\boldsymbol{\eta})\right\|^2, \tag{3.53}$$

$$= -\gamma h_\eta(\boldsymbol{\eta}) - \frac{\epsilon}{4}\left\|L_{\mathbf{f}}^r \mathbf{y}_{\text{out}} + L_{\mathbf{g}_1}L_{\mathbf{f}}^{r-1}\mathbf{k}_1 - \hat{\mathbf{k}}(\boldsymbol{\eta})\right\|^2, \tag{3.54}$$

$$\geq -\gamma h_\eta(\boldsymbol{\eta}) - \frac{\epsilon}{4\beta}V = -\gamma h - \left(\frac{\gamma}{\mu} + \frac{\epsilon}{4\beta}\right)V. \tag{3.55}$$

In the above expression, (3.51) follows directly from the linear output dynamics (3.34) and then by adding zero. Next, (3.52) is obtained by using the assumption that $\hat{\mathbf{k}}$ enforces the ISSf-CBF inequality (3.48) for (3.34) and then by applying the Cauchy-Schwartz inequality[11]. We then complete the square to achieve (3.53), and then use the definition of $\mathbf{v}$ in (3.35) to rewrite (3.53) as (3.54). Next, we select $\mathbf{k}_1$ provided in (3.42) and use (3.45) to bound (3.54) using V. Finally, we use (3.49) to express h_η in terms of h and V to yield (3.55).

Since V is a tracking CLF for (3.43), for each $\mathbf{x} \in \mathbb{R}^{n_x}$ there exists a $\mathbf{u}_2 \in \mathbb{R}^{n_{u2}}$ s.t.:

$$L_{\mathbf{f}_1}V(\mathbf{x}) + L_{\mathbf{g}_2}V(\mathbf{x})\mathbf{u}_2 \leq -\lambda V(\mathbf{x}). \tag{3.56}$$

Now, computing the time derivative of h with $\mathbf{u}_1 = \mathbf{k}_1(\mathbf{x})$ from (3.42) and bounding at each $\mathbf{x} \in \mathbb{R}^{n_x}$ using the above expression, we obtain:

$$\dot{h} = \dot{h}_0 - \frac{1}{\mu}\dot{V} = \dot{h}_0 - \frac{1}{\mu}\frac{\partial V}{\partial \mathbf{x}}\left[\mathbf{f} + \mathbf{g}_1\mathbf{k}_1 + \mathbf{g}_2\mathbf{u}_2\right] \tag{3.57}$$

$$= \dot{h}_0 - \frac{1}{\mu}\frac{\partial V}{\partial \mathbf{x}}\left[\mathbf{f}_1 + \mathbf{g}_2\mathbf{u}_2\right], \geq \dot{h}_0 + \frac{\lambda}{\mu}V > -\gamma h + \frac{1}{\mu}\left(\lambda - \gamma - \frac{\epsilon\mu}{4\beta}\right)V \geq -\gamma h, \tag{3.58}$$

[11] Given $|\mathbf{a}^\top\mathbf{b}| \leq \|\mathbf{a}\|\|\mathbf{b}\|$ for all $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{n_x}$, $\|\cdot\| := \|\cdot\|_2$. Setting $\mathbf{b} = -\mathbf{c}$ gives $-\mathbf{a}^\top\mathbf{c} \leq |-\mathbf{a}^\top\mathbf{c}| \leq \|\mathbf{a}\|\|-\mathbf{c}\| = \|\mathbf{a}\|\|\mathbf{c}\| \implies \mathbf{a}^\top\mathbf{c} \geq -\|\mathbf{a}\|\|\mathbf{c}\|$.

where we used the partial closed-loop dynamics (3.43) to rewrite $\dot{h}$ in (3.57). We then select $\mathbf{u}_2$ that satisfies (3.56) and substitute the bound obtained in (3.55) for $\dot{h}_\eta$ to obtain the first two bounds in (3.58). Finally, applying the inequality (3.50) for λ yields the last bound in (3.58). Given that this choice of $\mathbf{u}$ guarantees $\dot{h}(\mathbf{x}, \mathbf{u}) > -\gamma(h(\mathbf{x}))$ for all $\mathbf{x} \in \mathbb{R}^{n_x}$, h is a valid CBF[12] for (3.37). Furthermore, since h is a CBF for (3.37), any Lipschitz continuous controller that satisfies (2.33) renders $\mathcal{C}$ safe [18, Cor. 2], and since $V(\mathbf{x}) \geq 0$, the safe set $\mathcal{C}$ is contained in the desired safe set $\mathcal{C}_0$, $\mathcal{C} \subset \mathcal{C}_0$, so trajectories that are safe with respect to $\mathcal{C}$ also remain in the desired safe set $\mathcal{C}_0$. □

The preceding result requires the existence of a global CLF. Due to various factors (e.g., topological obstructions to continuous stabilization [102, Ch. 4]), such a CLF may not exist for a given system of interest (e.g., that with states evolving on a differentiable manifold), and (3.46) may only hold on a set $\mathcal{D} \subset \mathbb{R}^{n_x}$. While global stabilization in such a situation may not be possible, enforcing safety is still possible, as demonstrated in the following result:

Corollary 3.19. *(Global Safety) Let the conditions of Theorem 3.18 hold, but suppose that* (3.46) *only holds on a set $\mathcal{D} \subset \mathbb{R}^{n_x}$. Define $\mathcal{E} := \mathbb{R}^{n_x} \setminus \mathcal{D}$. Provided that for all $\mathbf{x} \in \mathcal{E}$:*

$$\left\{ L_{\mathbf{g}_1} h_\eta(\boldsymbol{\eta}(\mathbf{x})) = \frac{1}{\mu} L_{\mathbf{g}_1} V(\mathbf{x}) \right\} \implies \left\{ L_{\mathbf{f}} h_\eta(\boldsymbol{\eta}(\mathbf{x})) - \frac{1}{\mu} L_{\mathbf{f}} V(\mathbf{x}) \geq -\gamma h(\mathbf{x}) \right\}, \tag{3.59}$$

then h is a CBF for (3.37)*.*

Please see [50, Cor. 1] for the proof which mirrors the implication in (2.35).

Given h in (3.49), Theorem 3.18 and the above Corollary allow for synthesizing an optimization-based controller as in (2.36) for any given $\gamma \in \mathbb{R}_{>0}$ and nominal controller $\mathbf{k}_{\text{nom}}$.

Hardware Demonstrations

Next we provide hardware examples of the CBF construction methods and theoretical safety guarantees for dual relative degree systems.

We begin by considering the unicycle system with drift:

[12] Implicit in the fact that h satisfies (2.33) with a strict inequality is that $\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}) \neq \mathbf{0}$ for all $\mathbf{x} \in \partial\mathcal{C}$, a regularity condition needed to apply standard CBF results regarding forward invariance [18].

Example 3.20 (Quadruped on a Treadmill Demonstration)**.** *Consider a system with unicycle dynamics and an additional drift term:*

$$\frac{\mathrm{d}}{\Delta_t}\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \underbrace{\begin{bmatrix} d_x \\ d_y \\ 0 \end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{bmatrix}}_{\mathbf{g}_1(\mathbf{x})} v + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{\mathbf{g}_2(\mathbf{x})} \omega, \tag{3.60}$$

where the state $\mathbf{x} = (x, y, \theta) \in \mathbb{R}^2 \times \mathbb{S}^1$ defines the planar position and heading angle. The control input $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2) = (v, \omega) \in \mathbb{R}^2$ represents the linear and angular input velocities. The values $d_x, d_y \in \mathbb{R}$ represent constant drift, motivated by the unicycle operating on a treadmill (e.g., Fig. 3.5). Our control objective is to constrain the position of the unicycle. Thus, we take our outputs as $\mathbf{y}_{\text{out}}(\mathbf{x}) = (x, y) \in \mathbb{R}^2$, which do not have a valid relative degree in the sense of Def. 3.9. However, the unicycle with this choice of outputs has dual relative degree $(r, q) = (1, 1)$ as one may verify that $L_{\mathbf{g}_1}\mathbf{y}_{\text{out}}(\mathbf{x}) = [\cos(\theta)\ \sin(\theta)]^\top$ and $L_{\mathbf{g}_2}L_{\mathbf{g}_1}\mathbf{y}_{\text{out}}(\mathbf{x}) = [-\sin(\theta)\ \cos(\theta)]^\top$, which both have rank 1.

We consider a safety requirement that ensures the unicycle remains within an ellipse centered at $\mathbf{y}^{\text{c}}_{\text{out}} = [x_c, y_c]^\top \in \mathbb{R}^2$:

$$h_\eta(\boldsymbol{\eta}(\mathbf{x})) = 1 - (\mathbf{y}_{\text{out}}(\mathbf{x}) - \mathbf{y}^{\text{c}}_{\text{out}})^\top P(\mathbf{y}_{\text{out}}(\mathbf{x}) - \mathbf{y}^{\text{c}}_{\text{out}}), \tag{3.61}$$

where $P = \operatorname{diag}(p_1, p_2) \in \mathbb{R}^{2\times 2}$ is a diagonal matrix and $p_1, p_2 \in \mathbb{R}_{>0}$ are the weights corresponding to the lengths of the major and minor axes of the ellipse. The output coordinates $\boldsymbol{\eta}(\mathbf{x}) = \mathbf{y}_{\text{out}}(\mathbf{x})$ yield a single-integrator system of the form (3.34)*. We design a differentiable controller $\hat{\mathbf{k}} := [\hat{\mathbf{k}}_x, \hat{\mathbf{k}}_y]^\top : \mathbb{R}^2 \to \mathbb{R}^2$ satisfying* (3.48) *for the single integrator using the methods in [79]. We then leverage the single integrator controller $\hat{\mathbf{k}}$ to generate a safe linear velocity $v = \mathbf{k}_1(\mathbf{x})$ as in* (3.42) *for the unicycle.*

Next we let $\tilde{\mathbf{k}}(\mathbf{x}) := \hat{\mathbf{k}}(\boldsymbol{\eta}) - L_{\mathbf{f}}\mathbf{y}_{\text{out}} = [\hat{\mathbf{k}}_x(\boldsymbol{\eta}) - d_x, \hat{\mathbf{k}}_y(\boldsymbol{\eta}) - d_y]^\top$ and consider the tracking CLF:

$$V(\mathbf{x}) = \frac{\|\tilde{\mathbf{k}}(\mathbf{x})\|^2}{2}\operatorname{tr}\left(\mathbf{I}_{2\times 2} - \mathbf{R}(\theta_{\text{des}}(\mathbf{x}))^\top \mathbf{R}(\theta)\right), \tag{3.62}$$

where for $\|\tilde{\mathbf{k}}(\mathbf{x})\| \neq 0$, the direction of the vector $\tilde{\mathbf{k}}(\mathbf{x})$ provides the desired safe heading angle (i.e., safe yaw) as $\theta_{\text{des}}(\mathbf{x}) = \operatorname{atan2}(\hat{\mathbf{k}}_y(\boldsymbol{\eta}(\mathbf{x})) - d_y, \hat{\mathbf{k}}_x(\boldsymbol{\eta}(\mathbf{x})) - d_x)$, while if $\|\tilde{\mathbf{k}}(\mathbf{x})\| = 0$, then $V(\mathbf{x}) = 0$, making $\theta_{\text{des}}(\mathbf{x})$ a free parameter that may be

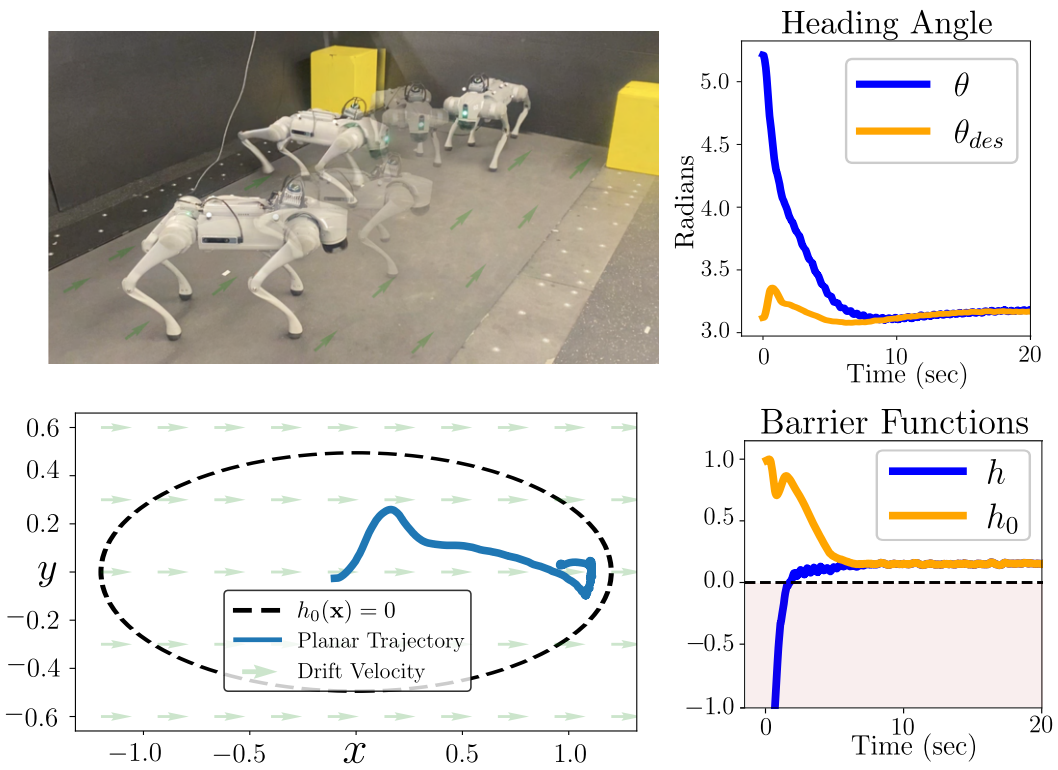


Figure 3.5. Quadruped on a treadmill demonstration. **(top left)** The quadrupedal robot **(top right)** The yaw, θ, of the quadruped in blue and the desired yaw, θ_{des}, from the desired safe controller for the linear system $\hat{\mathbf{k}}(\boldsymbol{\eta})$ in orange. **(bottom left)** (x, y) trajectories of the robot in blue with the drift velocity shown using green arrows and the boundary of $\mathcal{C}_0$ (3.47) shown as a black dotted line. Notably, the trajectories stay in this set and satisfy our safety criterion as desired. **(bottom right)** Our DRD-CBF h in blue and the safety requirement h_0 in orange. Notably, h_0 remains above zero. The robot is initialized with an unsafe yaw θ, causing h to be initially negative (i.e., outside the safe set $\mathcal{C}$ (2.29)). We demonstrate that the geometric tracking CLF (3.62) incorporated in h leads to the convergence of θ to a safe yaw θ_{des} yielding a positive h, enforcing attraction to $\mathcal{C}$. The video of this experiment can be found at [100].

assigned arbitrarily. The term $\mathbf{R} \in SO(2)$ *is a 2D rotation matrix, so* (3.62) *can be rewritten as:*

$$V(\mathbf{x}) = \|\tilde{\mathbf{k}}(\mathbf{x})\|^2(1 - \cos(\theta - \theta_{\text{des}}(\mathbf{x}))), \tag{3.63}$$

which satisfies (3.45) *as shown in the appendix of [50].*

We now consider the DRD-CBF as in (3.49)*, which we show is a CBF for* (3.60)*. We first find that* $L_{\mathbf{g}_2}V(\mathbf{x}) = 0 \implies \theta \in \{\theta_{\text{des}}(\mathbf{x}), \theta_{\text{des}}(\mathbf{x}) + \pi\}$*. Let* $\mu = 0.06$ *and* $\hat{\mathbf{k}}(\boldsymbol{\eta}(\mathbf{x})) = -\rho P^{\frac{1}{2}}\mathbf{y}_{\text{out}}(\mathbf{x})$ *with* $\rho = 0.16$*, then* $L_{\mathbf{g}_1}h(\mathbf{x}) \neq 0$ *when* $\theta - \theta_{\text{des}}(\mathbf{x}) = \pi$ *for all* $\mathbf{x} \in \mathcal{C}_0$ *defined in* (3.47)*. Thus, Collorary 3.19 applies. Note that this does not imply global stability of* θ *on* $\mathbb{S}^1$ *with a continuous controller, but that there exists inputs for each* $\mathbf{x} \in \mathcal{C}_0$ *satisfying the CBF condition* (2.33)*, ensuring safety but not necessarily stability of* $\theta = \theta_{\text{des}}(\mathbf{x})$*.*

Finally, we demonstrate the effectiveness of our proposed CBF (3.49) *in ensuring safety for system* (3.60) *in simulation and hardware. Using the safety specification* (3.61)*, we synthesize a safe controller as in* (2.36) *with the* h *defined in* (3.49) *for* (3.60) *with drift terms* $d_x = 0.35\,m/s$ *and* $d_y = 0$*. For the hardware demonstration, we apply this controller to a Unitree GO2 quadruped for which the unicycle may serve as a ROM*[13]*. On hardware, the drift is applied by placing the quadruped on a treadmill moving at a constant velocity of* $0.35\,m/s$*. The simulated and real-world trajectories can be seen ensuring safety in Fig. 3.5 with a nominal controller of zero linear and angular velocity.*

Next, we deploy our method on a quadrotor drone.

Example 3.21 (Hardware Quadrotor Demonstration)**.** *We use an OptiTrack motion capture system to provide the drone with real-time position measurements and a VectorNav VN-200 IMU for attitude state estimation. All state estimation and control computations are performed onboard at 750 Hz using a Jetson Orin NX. The drone model used is a simplified version of the dynamics in [94] with thrust and desired angle rate inputs as in [59]. The desired angle rates are tracked by a Betaflight flight controller and ESC at 8 kHz.*

To demonstrate the performance of our DRD-CBF (3.49)*, we command the quadrotor to track a sinusoidal reference* $\mathbf{y}^{\mathrm{d}}_{\mathrm{out}}(t) = [-\sin(0.4\pi t),\ 0.0,\ 1.0]^\top$ *in Euclidean space. We then define an* x*-coordinate geofence as the 0-superlevel set of* $h_{\mathrm{geo}}(\boldsymbol{\eta}(\mathbf{x})) = x_{\mathrm{geo}} - x$*, where* $x_{\mathrm{geo}} \in \mathbb{R}$ *is the* x*-position of the geofence [22]. For this particular experiment* $x_{\mathrm{geo}} = 0.2$ m*. Using a high order CBF [75], [76], we extend* h_{geo} *to get* h_η*, a CBF for the quadrotor double-integrator translational dynamics. By enforcing forward invariance of the safe-set defined by* $h_\eta(\boldsymbol{\eta}(\mathbf{x})) \geq 0$*, we ensure the* x*-coordinate of the quadrotor never exceeds the value of* x_{geo}*, irrespective of the commanded reference. Select data are presented in Fig. 3.6 utilizing the DRD-CBF* (3.49) *with the tracking CLF* (3.62) *for 3D orientation in* $SO(3)$*.*

From Fig. 3.6a and Fig. 3.6b, it is clear that the quadrotor drone effectively tracks the sinusoidal reference as long as it stays inside the safe set. However, once the commanded position crosses the geofence, the safety filter intercedes, preventing the drone from violating its safety specification.

[13]The velocity commands generated by our controller are then tracked by Unitree's onboard velocity tracking controller. In general, such an approach will lead to ISSf of the closed-loop system as analyzed in [48].

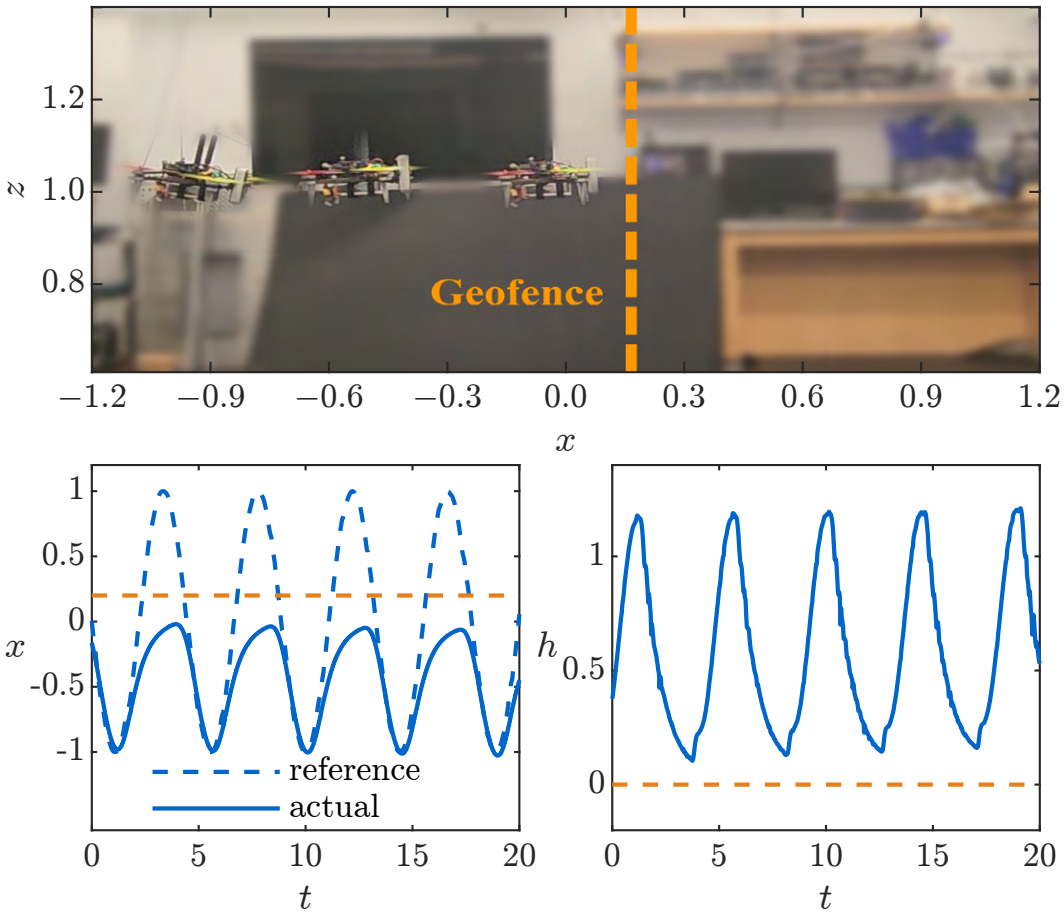


Figure 3.6. 3D quadrotor demonstration. **(top)** A composite image showing the position of the quadrotor drone over the course of the geofencing experiment. **(bottom left)** The x-position reference, which passes beyond the geofence, and the actual x-position, which deviates from the reference to maintain safety. **(bottom right)** The value of the DRD-CBF (3.49), which stays positive throughout the flight, confirming that safety is maintained. The video of this experiment can be found at [100].

Please see [50, Sec. IV and V] for an additional demonstrations on a quadruped without drift and on a simulated planar quadrotor.

Conclusion

This section presents a constructive framework for explicitly synthesizing CBFs and safety-critical controllers for nonlinear systems where output are used to specify safety requirements for dual relative degree systems, a generalization beyond the partially feedback linearizable systems of Section 3.4.

As in all the methods presented throughout this chapter, we see that this involves a safety requirement that defines a set that is potentially not control invariant (e.g., $\mathcal{C}_0$) that is then extended to a valid control invariant set using the structure of the system and a tracking assumption to ensure convergence to safe trajectories of the simplified system.

3.6 Safe-Set Synthesis using Back-up Controllers

The prior methods in this chapter have generally synthesized CBFs by shrinking the safety requirement set $\mathcal{C}_0$ according to some metric of the system's ability to track safe trajectories, thus producing a CBF h and a control invariant set $\mathcal{C} \subset \mathcal{C}_0$. This final section takes a drastically different approach where, instead of shrinking the safety requirement set $\mathcal{C}_0$, we assume the existence of a smaller back-up set $\mathcal{C}_B \subset \mathcal{C}_0$ and expand it to some implicit set $\mathcal{C}_I$ using a known backup controller and model-based prediction over a finite horizon.

This approach expands a known invariant set in a similar way to model predictive control [5], but does so using a fixed controller (instead of optimizing control actions), allowing to to be calculated at very high frequencies [52], [103]. This method also significantly differs from the previous methods in that it allows for the inclusion of input constraints in the synthesis of the safe set. On the other hand, it generally more computationally taxing and only produces an *implicit* understanding of the safe set $\mathcal{C}_I$ and does not directly produce a CBF for this set.

This section does not contain a novel contribution, but instead re-introduces a result from [103] and provides necessary background for Section 4.2.

The text for this section is adapted from:

> R. K. Cosner, A. W. Singletary, A. J. Taylor, T. G. Molnar, K. L. Bouman, and A. D. Ames, "Measurement-robust control barrier functions: certainty in safety with uncertainty in state," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021. DOI: 10.1109/IROS51168.2021.9636584,

Implicit control invariant Sets

Here we consider the desired safety requirement set $\mathcal{C}_0 \subset \mathbb{R}^{n_x}$ defined as the 0-superlevel set of the continuously differentiable function $h_0 : \mathbb{R}^{n_x} \to \mathbb{R}$ as in (2.28) which is not necessarily control invariant. We then assume that there exists a set $\mathcal{C}_B \subset \mathcal{C}_0$, defined as the 0-superlevel set of a continuously differentiable function $h_B : \mathbb{R}^{n_x} \to \mathbb{R}$, which is known *a priori* to be control invariant[14] and can be rendered forward invariant by a known locally Lipschitz continuous *backup*

[14] For $\mathcal{C}_B$, if the backup controllers are simple (such as linear state feedback controllers designed for the linearization of a system) it is possible to find analytical expressions of regions of attraction which can serve as this backup set $\mathcal{C}_B$. Alternatively, numerical tools such as Sums-of-Squares (SOS) may be used to synthesize control invariant sets [104].

controller $\mathbf{k}_B : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$. The goal of this method is to expand the *backup set* $\mathcal{C}_B$ to find a larger control invariant subset of $\mathcal{C}_0$.

We extend the backup set $\mathcal{C}_B$ to the larger implicit control invariant set $\mathcal{C}_I \subset \mathbb{R}^{n_x}$, satisfying $\mathcal{C}_B \subseteq \mathcal{C}_I \subseteq \mathcal{C}$, by considering the *backup trajectory* over a finite and fixed time horizon $T \in \mathbb{R}_{>0}$ as follows. We first assume that, for any $\mathbf{x} \in \mathbb{R}^{n_x}$, there exists a unique solution $\boldsymbol{\varphi} : [0, T] \to \mathbb{R}^{n_x}$ satisfying:

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\boldsymbol{\varphi}(\tau) = \mathbf{f}(\boldsymbol{\varphi}(\tau)) + \mathbf{g}(\boldsymbol{\varphi}(\tau))\mathbf{k}_B(\boldsymbol{\varphi}(\tau)), \qquad \boldsymbol{\varphi}(0) = \mathbf{x}. \tag{3.64}$$

The solution $\boldsymbol{\varphi}$ is the predicted evolution of the system over the interval $[0, T]$ from a state, $\mathbf{x}$, under the backup controller $\mathbf{k}_B$. We denote this prediction as $\boldsymbol{\phi}_\tau^{\mathbf{k}_b}(\mathbf{x}) \triangleq \boldsymbol{\varphi}(\tau)$ which predicts the flow of the trajectory forward τ seconds from the current state, $\mathbf{x}$, under the backup controller $\mathbf{k}_B$.

Using this notation, we define the set $\mathcal{C}_I \subseteq \mathcal{C}_0$ as:

$$\mathcal{C}_I \triangleq \left\{ \mathbf{x} \in \mathcal{C}_0 \;\middle|\; \begin{array}{c} h_0(\boldsymbol{\phi}_\tau^{\mathbf{k}_B}(\mathbf{x})) \geq 0, \forall \tau \in [0, T] \\ \text{and} \\ h_B(\boldsymbol{\phi}_T^{\mathbf{k}_B}(\mathbf{x})) \geq 0 \end{array} \right\}. \tag{3.65}$$

The first inequality implies that the system would remain in $\mathcal{C}_0$ for T seconds if it followed the backup controller from its current position (i.e., $\boldsymbol{\phi}_\tau^{\mathbf{k}_B}(\mathbf{x}) \in \mathcal{C}_0$ for all $\tau \in [0, T]$), and the second inequality implies that the system would reach the control-invariant set $\mathcal{C}_B$ by time T if it followed the backup controller ($\boldsymbol{\phi}_T^{\mathbf{k}_B}(\mathbf{x}) \in \mathcal{C}_B$). Thus, this safe-set synthesis method also relies on a convergence assumption, in this case of, the backup set $\mathcal{C}_B$. Notably, once the system reaches $\mathcal{C}_B$, the backup controller would render that set safe by assumption. The set $\mathcal{C}_I$ is thus control invariant as there exists at least one controller, $\mathbf{k}_B$, which renders it forward invariant. Furthermore, $\mathcal{C}_I$ is control invariant *even in the presence of input bounds* as long as $\mathbf{k}_B$ is constrained to satisfy those input bounds and still renders $\mathcal{C}_B$ forward invariant. A visualization of these set can be found in Fig. 3.7.

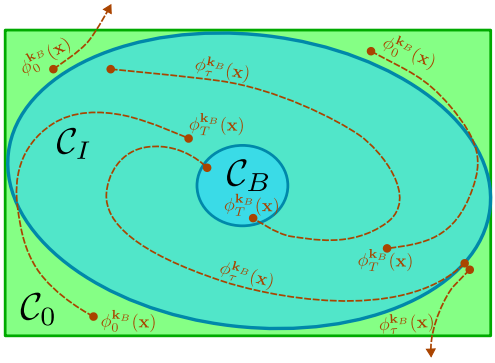


Figure 3.7. A visualization of the sets $\mathcal{C}_B \subseteq \mathcal{C}_I \subseteq \mathcal{C}_0$ in (3.65).

Thus, although we may not be able to render the desired safe set $\mathcal{C}_0$ safe, we can render its subset $\mathcal{C}_I$ safe, and every trajectory that is safe with respect to $\mathcal{C}_I$ also remains in $\mathcal{C}_0$. While $\mathcal{C}_I$ is not necessarily the largest control invariant subset of $\mathcal{C}$ (i.e., it may not be the *viability kernel*, [7]), this method provides a computationally tractable method for finding an implicit definition of a control invariant set.

Using Implicit Safe Sets for Safety-Critical Control

Next, for notational simplicity, we define the continuously differentiable functions $\overline{h}_\tau : \mathbb{R}^{n_x} \to \mathbb{R}$ and $\overline{h}_B : \mathbb{R}^{n_x} \to \mathbb{R}$ as:

$$\overline{h}_\tau(\mathbf{x}) \triangleq h_0(\boldsymbol{\phi}_\tau^{\mathbf{k}_B}(\mathbf{x})), \qquad \overline{h}_B(\mathbf{x}) \triangleq h_B(\boldsymbol{\phi}_T^{\mathbf{k}_B}(\mathbf{x})). \tag{3.66}$$

Given these definitions, the necessary CBF inequalities (2.33) can then be specified for the set $\mathcal{C}_I$ at a point $\mathbf{x} \in \mathcal{C}_I$ as:

$$\begin{aligned} L_{\mathbf{f}}\overline{h}_\tau(\mathbf{x}) + L_{\mathbf{g}}\overline{h}_\tau(\mathbf{x})\mathbf{u} &\geq -\alpha(\overline{h}_\tau(\mathbf{x})), \quad \forall \tau \in [0, T], \\ L_{\mathbf{f}}\overline{h}_B(\mathbf{x}) + L_{\mathbf{g}}\overline{h}_B(\mathbf{x})\mathbf{u} &\geq -\alpha(\overline{h}_B(\mathbf{x})). \end{aligned} \tag{3.67}$$

Any locally Lipschitz continuous controller that takes values satisfying (3.67) for all $\mathbf{x} \in \mathcal{C}_I$ will keep the system (2.2) safe with respect to $\mathcal{C}_I$; see [26, p. 6].

We note that enforcing the first inequality in (3.67) requires that we enforce an infinite number of constraints since it must hold for all $\tau \in [0, T]$. To resolve this, we reduce these infinite constraints to a finite collection of more conservative constraints through constraint tightening (see [103, Thm. 1]). A controller renders $\mathcal{C}_I$ safe using a finite number of tightened constraints is given by the Backup Set Quadratic Program (BS-QP), similar to the CBF-QP safety filter 2.36:

$$\begin{aligned} \mathbf{k}(\mathbf{x}) = \ \underset{\mathbf{u}\in\mathbb{R}^{n_u}}{\operatorname{argmin}} \quad & \frac{1}{2}\|\mathbf{u} - \mathbf{k}_{\text{des}}(\mathbf{x})\|_2^2 \qquad \text{(BS-QP)} \\ \text{s.t.} \quad & L_{\mathbf{f}}\overline{h}_{\tau_j}(\mathbf{x}) + L_{\mathbf{g}}\overline{h}_{\tau_j}(\mathbf{x})\mathbf{u} \geq -\alpha(h_{\tau_j}(\mathbf{x}) - \mu), \ \ \forall \tau_j \in \{0, \Delta_t, \dots, T\} \\ & L_{\mathbf{f}}\overline{h}_B(\mathbf{x}) + L_{\mathbf{g}}\overline{h}_B(\mathbf{x})\mathbf{u} \geq -\alpha(h_B(\mathbf{x})), \end{aligned}$$

where $\Delta_t \in \mathbb{R}_{>0}$ is a time-step such that $T/\Delta_t \in \mathbb{N}$ and $\mu \in \mathbb{R}_{>0}$, which is used to overcome the time-discretization, satisfies:

$$\mu \geq \frac{\Delta_t}{2}\mathfrak{L}_{h_0} \sup_{\mathbf{x}\in\mathcal{C}_0} \|\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{k}_B(\mathbf{x})\|_2, \tag{3.68}$$

where $\mathfrak{L}_{h_0} \in \mathbb{R}_{>0}$ a Lipschitz constant for h_0 on $\mathcal{C}_0$ and this bound captures the worst-case effect of the intersample dynamics on safety under a sampled-data implementation of the backup controller.

Thus, this method provides a way to approximate the largest control invariant subset of a desired safety requirement set $\mathcal{C}_0$ whose safety can then be ensured by the joint enforcement of the CBF condition on $\overline{h}_B$ and $\overline{h}_{\tau_j}$ for a finite number of τ_j. This method was used to enforce safety of a segway robot in [52] where the predictive element was critical to overcome the non-minimum phase nature of that system enabling the simultaneous enforcement of safety criteria that required the system to both remain upright and remain within a certain spatial region. See Section 4.2 for more experimental details and a discussion of how measurement uncertainty affects this system.

Conclusion

Unlike the other methods which shrink the safety requirement set $\mathcal{C}_0$ according to some convergence capability, the method in this section expands an assumed control-invariant subset of $\mathcal{C}_0$ using a backup controller similar to how MPC approaches expand a terminal safe set. This method enables the explicit incorporation of input bounds and produces an implicit control invariant set, but does not directly construct a CBF for $\mathcal{C}_I$, instead using several other CBF-like functions as proxies. In doing so this method can introduce additional computational complexity and discretization error. Despite these drawbacks, this method for ensuring safety has proven to be very highly effective [26], [105] and future work should explore its connections to MPC and its relative benefits and drawbacks.

3.7 Conclusion

This chapter discussed the standard assumption in CBF theory that h_0 satisfies the CBF conditions in Definition 2.19 and that the associated $\alpha \in \mathcal{K}_\infty^e$ is known. This synthesizing functions that are "valid CBFs" is general considered the greatest challenge of CBF-based methods and the fundamental assumption upon which they rely.

To tackle this problem in a general and computationally tractable way, this chapter presented a collection of methods which enable the construction of control invariant sets and CBFs from safety requirements encoded as a function $h_0 : \mathbb{R}^{n_x} \to \mathbb{R}$. These methods leveraged system structure and convergence properties of the entire system in Section 3.2, of the system to safe velocities in Section 3.3, of the system's outputs relevant to safety in Section 3.4, of the system's outputs relevant to safe control actuation in Section 3.5, and finally of an *a priori* known control invariant set for the system under a backup control policy in Section 3.6. Since each of these

methods allows for different forms of safe set and/or CBF synthesis under different structural system assumptions, they all may apply in different circumstances, and serve as a arsenal of tools to tackle safe set synthesis for a large variety of relevant systems.

Ultimately, however, each of these methods is an approximation for finding the viability kernel of the safety criterion set $\mathcal{C}_0$ for the open-loop dynamics (2.1). Future work should consider the effect of bounded inputs on these methods as this is critical to the ability of real-world systems to achieve control invariance and it should explore how computational improvements and new algorithms may improve our ability to find the viability kernel.

Additionally, these methods assume that $\mathcal{C}_0$ is known. I believe that this work establishes the foundation for future work on rapidly synthesizing CBFs for real-world systems when $\mathcal{C}_0$ is not known *a priori* and must instead be determined online from environmental sensors. Where prior work has generally used hand-crafted CBFs that require a strong *a priori* understanding of the environment, recent methods like those in [106] have begun to synthesize safety criterion sets $\mathcal{C}_0$ and functions h_0 directly from environment models built from sensors. By combining those methods with the methods of this chapter, I believe that we will be able to rapidly synthesize control invariant sets and safety constraints in novel environments. I see this as a fruitful open area of research that can explore the human-interpretable meanings of safety and link them with safety-critical control and safety guarantees using safe-set synthesis methods like those presented in this chapter.

Chapter 4

SAFETY UNDER BOUNDED UNCERTAINTY

"*To simplify analysis we begin by assuming that the cow is a perfect sphere of uniform density in a vacuum.*"

"*In theory, there is no difference between theory and practice.*" - Yogi Berra

Theoretical guarantees of safety are powerful tools that can inspire confidence in the deployment of robotic systems, particularly in safety-critical settings. However, that confidence can be dangerously misplaced when built on simplifying assumptions that fail to hold in practice (e.g., perfectly known dynamics, exact measurements, flawless actuation, or spherical cows).

This chapter addresses the mismatch between idealized assumptions and the messy reality of real-world robotics. Specifically, we consider scenarios in which the uncertainties are bounded and we propose methods for retaining meaningful, rigorous safety guarantees despite these deviations from the idealized assumptions of theory.

Abstract

As a tool for achieving robot safety, control barrier functions (CBFs) provide rigorous theoretical guarantees but, to do so, they make several simplifying assumptions. For example, they assume that the dynamics and state are known perfectly, and that the control action is applied exactly as intended. Unfortunately, these strict assumptions are impossible to meet for real-world systems and thus, CBF-based methods can lead to practical safety failures despite the perceived theoretical guarantees.

This chapter addresses this gap by introducing methods for achieving robust safety under bounded uncertainty. I present three core contributions: (1) measurement robust control barrier functions (MRCBFs) that maintain safety despite state measurement error; (2) the tunable robust optimization program (TROP) safety filter that integrates robustness to multiple sources of uncertainty within a single control framework; and (3) the theory of CBF-compliancy that ensures the safety of data-driven controllers with bounded learning-error.

Published content: The MRCBF discussion is adapted from [51] and [52]. The discusion of the TROP safety filter that combines all three forms of robustness was originally presented in [53]. Finally, the CBF compliancy result was originally presented in [54].

4.1 Introduction

The background developed in Chapter 2 introduced robust control concepts such as input-to-state stability (ISS, Def. 2.9 [29], [65]) and input-to-state safety (ISSf, Def. 2.24 [31], [32]), along with associated controllers designed to regulate the effect of disturbances. These and related robust control techniques—such as Hamilton-Jacobi backward reachability [73] and tube-based model predictive control (MPC) [30], [107]—achieve their guarantees by assuming a worst-case disturbance bound and using this bound to predict the impact on safety or stability.

While accounting for disturbances in the dynamics is critical, modern robotic systems also face additional sources of uncertainty that can significantly affect safety. For instance, contemporary robots often rely on complex perception pipelines or data-driven components which introduce uncertainty in both state estimation and control execution. It is therefore essential to develop controllers that preserve safety despite these real-world uncertainties.

Measurement Uncertainty

It is common in safety-critical control to assume that CBF-based controllers have access to perfect state measurements [32], [39], [42], [108]. However, this assumption rarely holds in real-world systems, where measurements are corrupted by sensor noise and errors in the measurement model. Moreover, while systems often exhibit ISS or ISSf robustness to disturbances in the dynamics, similar guarantees do not generally apply for state uncertainty where a bounded error in the state estimate does not necessarily imply a bounded impact on safety [109].

Thus, in the first section of this chapter, we begin by considering the case where the state of our system is not known exactly, $\widehat{\mathbf{x}} \neq \mathbf{x}$. Specifically, we consider the case where the measurement $\widehat{\mathbf{x}}$ is within some error radius $r_e \geq 0$ of the true state $\mathbf{x}$, i.e.,

$$\widehat{\mathbf{x}} \in \mathcal{B}_{r_e}(\mathbf{x}). \tag{4.1}$$

While this chapter considers this form of bounded state uncertainty, safety guarantees in the presence of measurement noise has also been explored from a stochastic perspective [110], [111] and will be addressed in the latter parts of this thesis.

Since the true state is not known, we seek to provide safety guarantees for the system where the measured state $\widehat{\mathbf{x}}$ is used to generate the control actions in the form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{k}(\widehat{\mathbf{x}}). \tag{4.2}$$

Combined Measurement and Dynamics Uncertainty

In Section 4.3, we extend the previous results by developing a unified approach to safety under both measurement uncertainty and dynamics disturbances. Specifically, we combine the measurement robustness techniques introduced earlier in this chapter with the ISSf-CBF formulation from Section 2.3 and the hierarchical safe set synthesis framework from Section 3.3.

The result is the *Tunable Robust Optimization Program (TROP)*—a convex quadratic program that acts as a safety filter, providing formal guarantees under multiple forms of uncertainty. The TROP controller introduces tunable parameters that can be adjusted to reflect varying levels of uncertainty in the system. While this chapter focuses on the safety properties of the TROP controller, Section 5.4 will examine how these parameters can be effectively tuned to achieve a desired trade-off between robustness and performance.

Bounded Imitation Learning Error

Finally, in Section 4.4, we study the case of imitation learning (IL), where a controller is trained to imitate a robust expert policy. IL is a highly effective paradigm in which a policy is learned to mimic expert behavior [112], and it has demonstrated impressive performance across domains including video games [113], humanoid robotics [114], and autonomous driving [115], [116]. However, these methods have largely emphasized performance, with safety typically enforced through fallback mechanisms such as backup controllers [117], [118] or human supervision [116].

While IL-based controllers have achieved safe behavior via training on safe demonstrations [119], we extend this idea by developing theoretical conditions that guarantee the safety of the learned policy. By leveraging a worst-case bound on the imitation learning error, we show that it is possible to transfer safety guarantees from the expert to the learned controller—thereby enabling robust, end-to-end learned control with formal safety assurances.

4.2 Safety Under Measurement Uncertainty

In this section, we address the challenge of achieving safe control when the system state is not directly observed. Motivated by vision-based control systems [115], [120], [121], we consider the common setting in which state information is acquired through a complex sensing pipeline (e.g., a camera), and an inverse mapping (e.g., a convolutional neural network) is used to estimate the underlying state. While perception frameworks of this form are ubiquitous in robotics, the standard approach is to make the measurement-to-state mapping as accurate as possible and then design controllers that ignore any remaining error. For linear systems, the *separation principle* justifies this design strategy by allowing the observer and stabilizer to be constructed independently [90]. However, for nonlinear systems, rigorous analysis of the effects of state estimation error on safety has only recently begun to emerge.

In this section, we introduce measurement-robust control barrier functions (MR-CBFs) as a tool for providing formal safety guarantees under bounded measurement uncertainty. These methods extend classical CBF-based control to explicitly account for uncertainty in the estimated state, enabling robust safety in settings where the true state cannot be directly observed.

The contributions of this section are as follows:

- We define *Measurement-Robust Control Barrier Functions (MRCBFs)*, which extend the standard CBF framework to account for bounded error in measurement models.
- We present a convex optimization-based controller that incorporates MRCBFs and can be solved efficiently in real time.
- We integrate MRCBFs with the backup-controller-based safe set synthesis method introduced in Section 3.6, enabling provable safety guarantees under measurement uncertainty.
- We demonstrate the practical effectiveness of MRCBFs through hardware experiments, achieving safe control of a segway robot using onboard camera-based state estimation. This constitutes the first hardware deployment of MRCBFs.

The text for this section is adapted from:

S. Dean, A. J. Taylor, R. K. Cosner, B. Recht, and A. D. Ames, "Guaranteeing safety of learned perception modules via measurement-robust control barrier functions," *Proceedings of the 2020 Conference on Robot Learning*, vol. 155, pp. 654–670, 2021. [Online]. Available: https://proceedings.mlr.press/v155/dean21a.html,

And

R. K. Cosner, A. W. Singletary, A. J. Taylor, T. G. Molnar, K. L. Bouman, and A. D. Ames, "Measurement-robust control barrier functions: certainty in safety with uncertainty in state," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021. DOI: 10.1109/IROS51168.2021.9636584,

Videos for this section can be found at [122], [123].

Measurement-Robust Control Barrier Functions

In this section we explore the impact of measurement uncertainty on safety guarantees, and propose the notion of MRCBFs as a modified CBF that is robust to such errors.

In many practical applications, the state $\mathbf{x}$ is not directly available to the controller, but instead we have access to a state-dependent sensor measurement:

$$\mathbf{y} = \mathbf{p}_{\mathrm{m}}(\mathbf{x}), \tag{4.3}$$

where $\mathbf{p}_{\mathrm{m}} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ is assumed to be locally Lipschitz continuous. We assume the relationship between the measurement and the true state is deterministic, with stochastic notions considered later in this thesis in Chapters 6 and 7. We further assume that there exists a locally Lipschitz continuous function $\mathbf{q}_{\mathrm{m}} : \mathbb{R}^{n_y} \to \mathbb{R}^{n_x}$ such that for all $\mathbf{x} \in \mathbb{R}^{n_x}$, we can reconstruct the system state as $\mathbf{q}_{\mathrm{m}}(\mathbf{p}_{\mathrm{m}}(\mathbf{x})) = \mathbf{x}$. This assumption implies that the state can be uniquely determined from any given measurement. This bijective relationship would allow the measurements to be redefined as the state of the system if the function $\mathbf{p}_{\mathrm{m}}$ was known, but that is often not the case in many modern control applications (such as when using computer vision).

While the function $\mathbf{p}_{\mathrm{m}}$ is often determined by the physical attributes of a system and its environment, a locally Lipschitz continuous estimate of the function $\mathbf{q}_{\mathrm{m}}$, given by $\widehat{\mathbf{q}}_{\mathrm{m}} : \mathbb{R}^{n_y} \to \mathbb{R}^{n_x}$, is often constructed to destimate the state from a given measurement, $\widehat{\mathbf{x}} = \widehat{\mathbf{q}}_{\mathrm{m}}(\mathbf{p}_{\mathbf{m}}(\mathbf{x}))$. For notational simplicity we define the measurement

estimate function $\widehat{\mathbf{v}} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y} \times \mathbb{R}^{n_x}$ such that $\widehat{\mathbf{v}}(\mathbf{x}) = (\mathbf{p}_{\text{m}}(\mathbf{x}), \widehat{\mathbf{q}}_{\text{m}}(\mathbf{p}_{\text{m}}(\mathbf{x}))$. We also define the set $\mathbf{p}_{\text{m}}(\mathcal{C}) \subset \mathbb{R}^{n_y}$ as the image of the safe set under the measurement function, the set $\widehat{\mathbf{q}}_{\text{m}}(\mathbf{p}_{\text{m}}(\mathcal{C})) \subset \mathbb{R}^{n_x}$ as the image of the safe set for state estimates, and $\widehat{\mathbf{v}}(\mathcal{C})$ as the image of the safe set for the measurement estimates.

The function $\widehat{\mathbf{q}}_{\text{m}}$ is constructed either via system and measurement models, or from data using learning methods, and thus its accuracy in estimating $\mathbf{q}_{\text{m}}$ degrades with imperfections in sensor fabrication and integration, or imperfections in learning models and training data. Thus we assume that our state estimate is related to the true state as follows:

$$\widehat{\mathbf{x}} \triangleq \widehat{\mathbf{q}}_{\text{m}}(\mathbf{y}) = \mathbf{x} + \mathbf{e}(\mathbf{x}), \tag{4.4}$$

for an unknown function $\mathbf{e} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ that is defined implicitly via $\widehat{\mathbf{q}}_{\text{m}}$. In practice, the function $\mathbf{e}$ can often be characterized via upper bounds on model uncertainty or via data-driven arguments for learning models. In particular, we assume that while $\mathbf{e}(\mathbf{x})$ is not known for a particular value of $\mathbf{x}$, it is known that $\mathbf{e}(\mathbf{x}) \in \mathcal{E}(\mathbf{y})$ for a measurement dependent, compact pointwise set $\mathcal{E}(\mathbf{y})$. This leads to the definition of the following two pointwise sets:

$$\widehat{\mathcal{S}}(\mathbf{x}) \triangleq \{\widehat{\mathbf{x}} \in \mathbb{R}^{n_x} \mid \exists\, \mathbf{e} \in \mathcal{E}(\mathbf{p}(\mathbf{x})) \text{ s.t. } \widehat{\mathbf{x}} = \mathbf{x} + \mathbf{e}\}\,, \tag{4.5}$$

$$\mathcal{S}(\mathbf{y}) \triangleq \{\mathbf{x} \in \mathbb{R}^{n_x} \mid \exists\, \mathbf{e} \in \mathcal{E}(\mathbf{y}) \text{ s.t. } \widehat{\mathbf{x}} = \mathbf{x} + \mathbf{e}\}\,. \tag{4.6}$$

The first of these two pointwise sets can be interpreted as all possible state estimates corresponding to a particular state, with uncertainty generated by the possible error dictated by $\mathbf{e} \in \mathcal{E}(\mathbf{p}(\mathbf{x}))$. While $\widehat{\mathcal{S}}(\mathbf{x})$ is not directly computable without knowledge of $\mathbf{p}$, this set will play an important conceptual role in arguing about how data can be used to determine error bounds. The second pointwise, on the other had, may be computed since we have access to $\mathbf{y}$ and this set consists of all potential states that may yield a (measurement, state estimate) pair, $\widehat{\mathbf{v}}$.

Since a controller enforcing the CBF condition (2.33) requires exact knowledge of the state $\mathbf{x}$, we propose an alternative condition which depends on only the set $\mathcal{S}(\mathbf{y})$ and the state estimate $\widehat{\mathbf{x}}$. To ensure safety with a CBF, it is sufficient for the following condition to hold for all $\mathbf{y} \in \mathbf{p}_{\text{m}}(\mathcal{C})$:

$$\sup_{\mathbf{u} \in \mathbb{R}^{n_u}} \inf_{\mathbf{x} \in \mathcal{S}(\mathbf{y})} \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x})(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}) + \alpha(h(\mathbf{x})) \geq 0\,. \tag{4.7}$$

This condition implies that there exists a control input that renders the system safe for all possible states corresponding to a given state estimate. Verifying that this

condition holds can be difficult for an arbitrary CBF, and it is not easily (or possibly) enforced in a convex-optimization based controller. To resolve these problems, we introduce the following definition:

Definition 4.1 (*Measurement-Robust Control Barrier Function (MRCBF)*)**.** *Let $\mathcal{C} \subset \mathbb{R}^{n_x}$ be the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^{n_x} \to \mathbb{R}$ with 0 a regular value. The function h is a Measurement-Robust Control Barrier Function (MRCBF) for the open-loop system* (2.1) *on $\mathcal{C}$ with parameter function $(a, b) : \mathbb{R}^{n_y} \to \mathbb{R}^2_{\geq 0}$ if there exists $\alpha \in \mathcal{K}^e_\infty$ such that for all $(\mathbf{y}, \widehat{\mathbf{x}}) \in \widehat{\mathbf{v}}(\mathcal{C})$:*

$$\sup_{\mathbf{u} \in \mathbb{R}^{n_u}} L_{\mathbf{f}} h(\widehat{\mathbf{x}}) + L_{\mathbf{g}} h(\widehat{\mathbf{x}})\mathbf{u} - (a(\mathbf{y}) + b(\mathbf{y})\|\mathbf{u}\|_2) > -\alpha(h(\widehat{\mathbf{x}})). \tag{4.8}$$

I recognize that directly verifying this condition over $\widehat{\mathbf{v}}(\mathcal{C})$ may not be possible, and point the interested reader to [51, Sec. 4] where we used the set $\widehat{\mathcal{S}}(\mathbf{x})$ to provide sufficient conditions under which (4.8) is met.

The above definition of MRCBFs introduces the non-positive term $-(a(\mathbf{y}) + b(\mathbf{y})\|\mathbf{u}\|_2)$ to the CBF condition, requiring that a stronger degree of safety be enforced compared to the typical CBF condition (2.33). Furthermore, the norm of the input appears in this term, indicating that for large values of b, large inputs can lead to unsafe behavior.

The MRCBF condition (4.8) is equivalently stated as:

$$\|L_{\mathbf{g}} h(\widehat{\mathbf{x}})\|_2 \leq b(\mathbf{y}) \implies L_{\mathbf{f}} h(\widehat{\mathbf{x}}) > -\alpha(h(\widehat{\mathbf{x}})) + a(\mathbf{y}), \quad \forall (\mathbf{y}, \mathbf{x}) \in \widehat{v}(\mathcal{C}). \tag{4.9}$$

In contrast to the implication in (2.35), the size of set for which the antecedent in (4.9) is met may be larger, requiring the natural dynamics to be safe ($L_{\mathbf{f}} h(\widehat{\mathbf{x}}) > -\alpha(h(\widehat{\mathbf{x}})) + a(\mathbf{y})$) in a larger region. Given a MRCBF h for (2.1) on $\mathcal{C}$ with parameter function (a, b) and a corresponding $\alpha \in \mathcal{K}^e_\infty$, we can consider the point-wise set of all control values that satisfy (4.8):

$$\begin{aligned} &\mathscr{K}_{\text{MRCBF}}(\mathbf{y}, \widehat{\mathbf{x}}) \\ &\quad \triangleq \{\mathbf{u} \in \mathbb{R}^{n_u} \mid L_{\mathbf{f}} h(\widehat{\mathbf{x}}) + L_{\mathbf{g}} h(\widehat{\mathbf{x}})\mathbf{u} - (a(\mathbf{y}) + b(\mathbf{y})\|\mathbf{u}\|_2) \geq -\alpha(h(\widehat{\mathbf{x}}))\}, \end{aligned} \tag{4.10}$$

for $(\mathbf{y}, \widehat{\mathbf{x}}) \in \widehat{\mathbf{v}}(\mathcal{C})$. Given this construction, we have the following result relating the existence of a MRCBF to safety under the presence of measurement model uncertainty:

Theorem 4.2 (MRCBF Safety)**.** *Let a set $\mathcal{C} \subset \mathbb{R}^{n_x}$ be defined as the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^{n_x} \to \mathbb{R}$. Assume the functions*

$L_{\mathbf{f}}h : \mathbb{R}^{n_x} \to \mathbb{R}$, $L_{\mathbf{g}}h : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$, and $\alpha \circ h : \mathbb{R}^{n_x} \to \mathbb{R}$ are Lipschitz continuous on $\mathcal{C}$ with Lipschitz coefficients $\mathfrak{L}_{L_{\mathbf{f}}h}$, $\mathfrak{L}_{L_{\mathbf{g}}h}$, and $\mathfrak{L}_{\alpha\circ h}$, respectively. Further assume there exists a locally Lipschitz function $\epsilon_{\mathrm{m}} : \mathbb{R}^{n_y} \to \mathbb{R}_{\geq 0}$, such that $\max_{\mathbf{e}\in\mathcal{E}(\mathbf{y})} \|\mathbf{e}\|_2 \leq \epsilon_{\mathrm{m}}(\mathbf{y})$ for all $\mathbf{y} \in \mathbf{p}_{\mathrm{m}}(\mathcal{C})$. If h is a MRCBF for (2.1) *on $\mathcal{C}$ with parameter function $(\epsilon_{\mathrm{m}}(\mathbf{y})(\mathfrak{L}_{L_{\mathbf{f}}h} + \mathfrak{L}_{\alpha\circ h}), \epsilon_{\mathrm{m}}(\mathbf{y})\mathfrak{L}_{L_{\mathbf{g}}h})$, then any locally Lipschitz continuous controller $\mathbf{k} : \mathbb{R}^{n_y} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$, such that $\mathbf{k}(\mathbf{y}, \widehat{\mathbf{x}}) \in \mathscr{K}_{MRCBF}(\mathbf{y}, \widehat{\mathbf{x}})$ for all $(\mathbf{y}, \widehat{\mathbf{x}}) \in \widehat{\mathbf{v}}(\mathcal{C})$, renders the system* (4.2) *safe with respect to the set $\mathcal{C}$.*

Proof. Define the function $c : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ as

$$c(\mathbf{x}, \mathbf{u}) = \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x})\Big(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}\Big) + \alpha(h(\mathbf{x})) = L_{\mathbf{f}}h(\mathbf{x}) + L_{\mathbf{g}}h(\mathbf{x})\mathbf{u} + \alpha(h(\mathbf{x})).$$

This proof will follow from Theorem 2.20, in that for any $\mathbf{x} \in \mathcal{C}$, with $(\mathbf{y}, \widehat{\mathbf{x}}) = \widehat{\mathbf{v}}(\mathbf{x})$, we will show:

$$c(\mathbf{x}, \mathbf{k}(\mathbf{y}, \widehat{\mathbf{x}})) \geq 0. \tag{4.11}$$

To show that (4.11) is true, consider a measurement-state estimate pair $(\mathbf{y}, \widehat{\mathbf{x}}) \in \widehat{\mathbf{v}}(\mathcal{C})$. A sufficient condition for (4.11) to hold is given by:

$$\inf_{\mathbf{x}\in\mathcal{S}(\mathbf{y})} c(\mathbf{x}, \mathbf{k}(\mathbf{y}, \widehat{\mathbf{x}})) \geq 0\,. \tag{4.12}$$

Recalling that we define $\widehat{\mathbf{x}} = \mathbf{x} + \mathbf{e}(\mathbf{x})$, we have:

$$\begin{aligned}
\inf_{\mathbf{x}\in\mathcal{S}(\mathbf{y})} c(\mathbf{x}, \mathbf{k}(\mathbf{y}, \widehat{\mathbf{x}})) &= \inf_{\mathbf{e}\in\mathcal{E}(\mathbf{y})} c(\widehat{\mathbf{x}} - \mathbf{e}, \mathbf{k}(\mathbf{y}, \widehat{\mathbf{x}})), \\
&= c(\widehat{\mathbf{x}}, \mathbf{k}(\mathbf{y}, \widehat{\mathbf{x}})) + \inf_{\mathbf{e}\in\mathcal{E}(\mathbf{y})} c(\widehat{\mathbf{x}} - \mathbf{e}, \mathbf{k}(\mathbf{y}, \widehat{\mathbf{x}})) - c(\widehat{\mathbf{x}}, \mathbf{k}(\mathbf{y}, \widehat{\mathbf{x}})), \\
&\geq c(\widehat{\mathbf{x}}, \mathbf{k}(\mathbf{y}, \widehat{\mathbf{x}})) - \sup_{\mathbf{e}\in\mathcal{E}(\mathbf{y})} |c(\widehat{\mathbf{x}} - \mathbf{e}, \mathbf{k}(\mathbf{y}, \widehat{\mathbf{x}})) - c(\widehat{\mathbf{x}}, \mathbf{k}(\mathbf{y}, \widehat{\mathbf{x}}))|.
\end{aligned}$$

The assumption on Lipschitz continuity of $L_{\mathbf{f}}h$, $L_{\mathbf{g}}h$, and $\alpha \circ h$ enables the following bound:

$$\begin{aligned}
&|c(\mathbf{x}', \mathbf{u}) - c(\mathbf{x}, \mathbf{u})| && (4.13)\\
&= |L_{\mathbf{f}}h(\mathbf{x}') - L_{\mathbf{f}}h(\mathbf{x}) + L_{\mathbf{g}}h(\mathbf{x}')\mathbf{u} - L_{\mathbf{g}}h(\mathbf{x})\mathbf{u} + \alpha(h(\mathbf{x}')) - \alpha(h(\mathbf{x}))|, && (4.14)\\
&\leq |L_{\mathbf{f}}h(\mathbf{x}') - L_{\mathbf{f}}h(\mathbf{x})| + |L_{\mathbf{g}}h(\mathbf{x}')\mathbf{u} - L_{\mathbf{g}}h(\mathbf{x})\mathbf{u}| + |\alpha(h(\mathbf{x}')) - \alpha(h(\mathbf{x}))|, && (4.15)\\
&\leq \mathfrak{L}_{L_{\mathbf{f}}h}\|\mathbf{x}' - \mathbf{x}\|_2 + \|L_{\mathbf{g}}h(\mathbf{x}') - L_{\mathbf{g}}h(\mathbf{x})\|_2\|\mathbf{u}\|_2 + \mathfrak{L}_{\alpha\circ h}\|\mathbf{x}' - \mathbf{x}\|_2, && (4.16)\\
&\leq (\mathfrak{L}_{L_{\mathbf{f}}h} + \mathfrak{L}_{L_{\mathbf{g}}h}\|\mathbf{u}\|_2 + \mathfrak{L}_{\alpha\circ h})\|\mathbf{x}' - \mathbf{x}\|_2. && (4.17)
\end{aligned}$$

Therefore, using the definition of $\epsilon_{\mathrm{m}}(\mathbf{y})$ we have:

$$\sup_{\mathbf{e}\in\mathcal{E}(\mathbf{y})} |c(\widehat{\mathbf{x}}-\mathbf{e},\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}}))-c(\widehat{\mathbf{x}},\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}}))| \tag{4.18}$$

$$\leq \sup_{\mathbf{e}\in\mathcal{E}(\mathbf{y})} (\mathfrak{L}_{L_{\mathbf{f}}h}+\mathfrak{L}_{L_{\mathbf{g}}h}\|\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}})\|_2+\mathfrak{L}_{\alpha\circ h})\|\mathbf{e}\|_2, \tag{4.19}$$

$$\leq (\mathfrak{L}_{L_{\mathbf{f}}h}+\mathfrak{L}_{L_{\mathbf{g}}h}\|\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}})\|_2+\mathfrak{L}_{\alpha\circ h})\epsilon_{\mathrm{m}}(\mathbf{y}). \tag{4.20}$$

Thus:

$$\inf_{\mathbf{x}\in\mathcal{S}(\mathbf{y})} c(\mathbf{x},\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}}))\geq c(\widehat{\mathbf{x}},\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}}))-(\mathfrak{L}_{L_{\mathbf{f}}h}+\mathfrak{L}_{L_{\mathbf{g}}h}\|\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}})\|_2+\mathfrak{L}_{\alpha\circ h})\epsilon_{\mathrm{m}}(\mathbf{y}).$$

By the MRCBF condition and the design of $\mathbf{k}$ we have that:

$$c(\widehat{\mathbf{x}},\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}}))-\epsilon_{\mathrm{m}}(\mathbf{y})(\mathfrak{L}_{L_{\mathbf{f}}h}+\mathfrak{L}_{\alpha\circ h})-\epsilon_{\mathrm{m}}(\mathbf{y})\mathfrak{L}_{L_{\mathbf{g}}h}\|\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}})\|_2\geq 0, \tag{4.21}$$

implying the condition (4.11). □

To more clearly see how the upper bound on the estimate error, $\epsilon_{\mathrm{m}}(\mathbf{y})$, manifests in the MRCBF condition, we note the particular condition that must be satisfied for this theorem is given by:

$$\sup_{\mathbf{u}\in\mathbb{R}^{n_u}} L_{\mathbf{f}}h(\widehat{\mathbf{x}})+L_{\mathbf{g}}h(\widehat{\mathbf{x}})\mathbf{u}-\epsilon_{\mathrm{m}}(\mathbf{y})(\mathfrak{L}_{L_{\mathbf{f}}h}+\mathfrak{L}_{\alpha\circ h}+\mathfrak{L}_{L_{\mathbf{g}}h}\|\mathbf{u}\|_2)>-\alpha(h(\widehat{\mathbf{x}})). \tag{4.22}$$

Thus as $\epsilon_{\mathrm{m}}(\mathbf{y})$ becomes smaller, the level of robustness required by an MRCBF approaches that of a regular CBF for the same set $\mathcal{C}$, and recovers the original CBF condition with no estimate error. Furthermore, smaller values of $\epsilon_{\mathrm{m}}(\mathbf{y})$ can be interpreted as leading to an enlarging of the region over which the condition (4.9) holds.

While this approach introduces significant conservatism due to the use of several Lipschitz constants and the worst-case upper bound $\epsilon_{\mathrm{m}}(\mathbf{y})$ on the effect of $\mathbf{e}(\mathbf{y})$, one major advantage of this approach is that the constraint in (4.10) remains convex, facilitating real-time robotics applications. This constraint can then be directly integrated into an optimization based controller as:

$$\begin{aligned}\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}})=\ &\underset{\mathbf{u}\in\mathbb{R}^{n_u}}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{u}-\mathbf{k}_d(\widehat{\mathbf{x}})\|_2^2 \qquad \text{(MR-OP)}\\ &\text{s.t.} \quad L_{\mathbf{f}}h(\widehat{\mathbf{x}})-(\mathfrak{L}_{L_{\mathbf{f}}h}+\mathfrak{L}_{\alpha\circ h})\epsilon_{\mathrm{m}}(\mathbf{y})+L_{\mathbf{g}}h(\widehat{\mathbf{x}})\mathbf{u}\\ &\qquad\qquad -\mathfrak{L}_{L_{\mathbf{g}}h}\epsilon_{\mathrm{m}}(\mathbf{y})\|\mathbf{u}\|_2\geq-\alpha(h(\widehat{\mathbf{x}})).\end{aligned}$$

This problem is in fact a second-order cone program (SOCP), with an explicit conversion to standard form provided in [51, Appx. B].

To ensure feasibility, a slack variable, δ, is often added in practice. This relaxation is penalized in the cost with a large coefficient $p \in \mathbb{R}_{>0}$:

$$\mathbf{k}(\mathbf{y}, \widehat{\mathbf{x}}) = \underset{(\mathbf{u},\delta)\in\mathbb{R}^{n_u}\times\mathbb{R}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{u} - \mathbf{k}_d(\widehat{\mathbf{x}})\|_2^2 + p\delta^2 \tag{R-MR-OP}$$
$$\text{s.t.} \quad L_{\mathbf{f}}h(\widehat{\mathbf{x}}) - (\mathfrak{L}_{L_{\mathbf{f}}h} + \mathfrak{L}_{\alpha\circ h})\epsilon_{\mathrm{m}}(\mathbf{y}) + L_{\mathbf{g}}h(\widehat{\mathbf{x}})\mathbf{u} - \mathfrak{L}_{L_{\mathbf{g}}h}\epsilon_{\mathrm{m}}(\mathbf{y})\|\mathbf{u}\|_2 \geq -\alpha(h(\widehat{\mathbf{x}})) - \delta.$$

While this relaxed controller does not necessarily enforce the conservative desired MRCBF safety constraint, if δ remains small the impact on safety can be understood through the notion of projection-to-state safety [108] a variant of the ISSf property discussed in (2.3). Furthermore, this relaxation ensures that the resulting controller is locally Lipschitz continuous as made explicit in [51, Appendix C].

MRCBF Demonstration: Simulation

Next we present simulation results using MRCBFs and data-driven learning models on a simulated robotic segway platform.

Example 4.3. *The segway can be seen in Figure 4.1, and is modeled with system dynamics derived using the unconstrained Euler-Lagrange equations [15]. The system is constrained to planar motion by providing identical input torques about both wheels. The resulting degrees of freedom are the segway's horizontal position r, horizontal velocity $\dot{r}$, pitch angle θ_y, and pitch rate $\dot{\theta}_y$. The nominal controller $\mathbf{k}_d$ is a simple proportional-derivative (PD) controller as in [108].*

The safe set was defined as $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^7 : h_1(\mathbf{x}) \geq 0,\ h_2(\mathbf{x}) \geq 0\}$ with:

$$h_1(\mathbf{x}) = -\dot{\theta}_y + \alpha_e(c - \theta_y + \theta_y^\star), \qquad h_2(\mathbf{x}) = \dot{\theta}_y + \alpha_e(c + \theta_y - \theta_y^\star), \tag{4.23}$$

where $c \in \mathbb{R}_{>0}$, $\alpha_e \in \mathbb{R}_{>0}$, and $\theta_y^\star$ is the pitch angle at equilibrium. The MR-OP Filter constraint in (MR-OP) was applied simultaneously to both safety functions (4.23) and was then implemented using the ECOS SOCP solver [124]. The Lipschitz constants in this constraint were estimated by sampling $\mathfrak{L}_{L_{\mathbf{f}}h}$, $\mathfrak{L}_{L_{\mathbf{g}}h}$ and $\alpha \circ h$ on a set of gridded values around the system's equilibrium point by taking the maximum of the slopes between any two adjacent grid points. As a baseline comparison, a

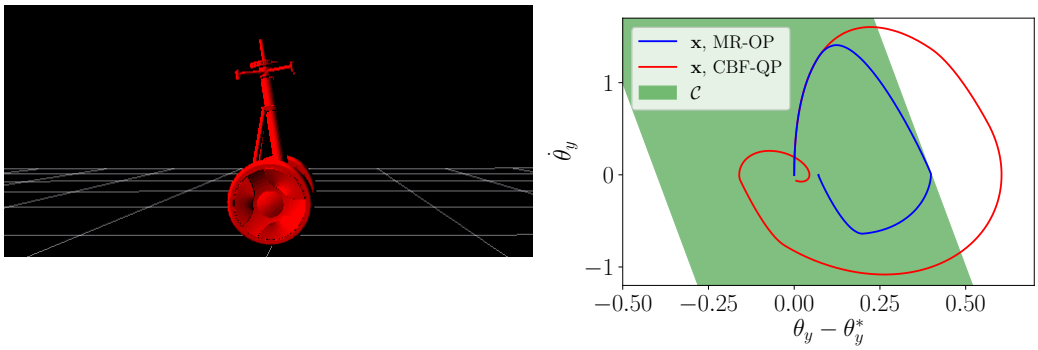


Figure 4.1. Simulated segway demonstration **(Left)** The segway model used in simulation from the perspective of the fixed virtual camera used to estimate its state. **(Right)** Simulation results for worst-case measurement model uncertainty of $\epsilon_{\text{m}} = 0.2$ subtracted from the true pitch angle θ_y when measured. A state trajectory generated using the Standard CBF Filter (red) and the MR-OP Filter (blue) are shown as projections onto their pitch angle and pitch rate components. The safe set is plotted in green. Given the same initial condition, the MR-OP filter ensured safety of the trajectory whereas the Standard CBF Filter did not.

CBF-QP Filter (2.36) was also implemented and applied using both safety functions (4.23). We considered the two following testing scenarios:

1. *Worst-Case Synthetic Measurement Model Uncertainty: In this testing scenario we assumed that direct measurements of the pitch angle θ_y were offset by a constant factor of $\epsilon_{\text{m}} > 0$, such that $\widehat{\theta}_y = \theta_y - \epsilon_{\text{m}}$. Implementing the MR-OP Filter for $\epsilon_{\text{m}} > 0$ ensures safety for this worst-case error of up to ϵ_{m}. The result of this type of worst-case measurement model uncertainty in the segway system with a standard CBF-QP Filter and an MR-OP Filter can be seen in Figure 4.1.*

2. *Data-Driven Sensor Calibration: In this scenario a more realistic form of measurement model uncertainty is introduced through the use of a learned model to estimate the position r and pitch angle θ_y from camera images. In simulation, a virtual camera and lighting source were implemented to provide a 15 Hz video feed with a fixed perspective, an example of which can be seen in Figure 4.1. The labels for this supervised-learning problem were noisy measurements of the position and pitch angle generated by the system's inertial measurement unit, corrupted by Gaussian noise with standard deviation 0.1. We use `sklearn`'s Kernel Ridge Regression with radial basis functions [125] trained using a set of 800 labeled images associated with a gridded range of position and pitch angle values to ensure dense coverage. The hyperparameter values $\alpha = 0, \gamma = 5.4 \times 10^{-8}$ were selected to minimize the average error on an 80% - 20% random train-test split.*

The result of the learning-induced errors in the segway system with a standard CBF

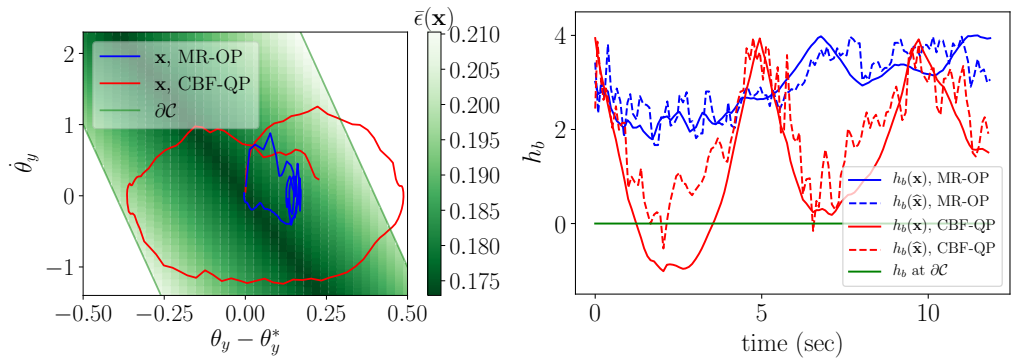


Figure 4.2. Simulation results demonstrating the ability of the MR-OP Filter to mitigate the impact of imperfect learned perception models on safety. **(Left)** The state trajectory generated using the Standard CBF Filter (red) and MR-OP Filter (blue) are shown as projections onto their pitch angle and pitch rate components. Given the same initial condition, the MR-OP Filter generated a safe trajectory whereas the Standard CBF-QP Filter did not. The state estimates for each trajectory had a maximum error of 0.183 and 0.201, respectively. **(Right)** The Boolean composition, $h_b = \min\{h_{e1}, h_{e2}\} = h_{e1} \wedge h_{e2}$ as defined in [20], is plotted for the CBF-QP Filter and MR-OP Filter trajectories for the true and estimated states, $\mathbf{x}$ (solid line) and $\hat{\mathbf{x}}$ (dotted line). The safety violation of the Standard CBF-QP Filter can be seen where $h_b(\mathbf{x})$ crosses 0.

Filter and an MR-OP Filter with $\epsilon_{\mathrm{m}} = 0.2$ can be seen in Figure 4.2. In the left panel, the safe set $\mathcal{C}$ is shaded according to the upper bound on error $\bar{\epsilon}_{\mathrm{m}}(\mathbf{x})$ under which feasibility is guaranteed. As the errors in the learned map do not exceed this value empirically, the set $\mathcal{C}$ is rendered invariant. The expression for $\bar{\epsilon}_{\mathrm{m}}(\mathbf{x})$ in this experimental scenario is presented in [51, Appx. E], along with an empirical validation of learned model errors.

For futher experimental details, such as proofs of MR-OP feasibility for all $\mathbf{x} \in \mathcal{C}$ and additional details regarding the simulation please see the appendix in [51].

MRCBFs as Backup Set CBFs

In order to demonstrate the utlity of MRCBFs on hardware, we unify the robustness of Theorem 4.2 with the back-up CBF safe-set synthesis method outlined in Section 3.6 to create measurement-robust safe sets that realize practically useful safety behavior on the segway system, namely that it both remains upright and within a desired translational region.

To unify the backup-set method for implicit safe-set synthesis of Section 3.6 with the MRCBFs of Theorem 4.2, we alter the conditions of the implicit backup safe set $\mathcal{C}_I$ given in 3.65 using the MRCBF condition (4.8) so that the finite, discretized set

of constraints imposed in the BS-QP become:

$$\begin{aligned} L_{\mathbf{f}}\overline{h}_{\tau_j}(\widehat{\mathbf{x}}) + L_{\mathbf{g}}\overline{h}_{\tau_j}(\widehat{\mathbf{x}})\mathbf{u} - (a_{\tau_j}(\mathbf{y}) + b_{\tau_j}(\mathbf{y})\|\mathbf{u}\|_2) \geq -\alpha(\overline{h}_{\tau_j}(\widehat{\mathbf{x}}) - \mu), \\ L_{\mathbf{f}}\overline{h}_{B}(\widehat{\mathbf{x}}) + L_{\mathbf{g}}\overline{h}_{B}(\widehat{\mathbf{x}})\mathbf{u} - (a_{B}(\mathbf{y}) + b_{B}(\mathbf{y})\|\mathbf{u}\|_2) \geq -\alpha(\overline{h}_{B}(\widehat{\mathbf{x}})), \end{aligned} \tag{4.24}$$

with parameter functions:

$$\begin{aligned} a_{\tau_j}(\mathbf{y}) = (\mathfrak{L}_{L_{\mathbf{f}}\overline{h}_{\tau_j}} + \mathfrak{L}_{\alpha}\mathfrak{L}_{\overline{h}_{\tau_j}})\epsilon_{\mathrm{m}}(\mathbf{y}), \quad b_{\tau_j}(\mathbf{y}) = \mathfrak{L}_{L_{\mathbf{g}}\overline{h}_{\tau_j}}\epsilon_{\mathrm{m}}(\mathbf{y}), \\ a_{B}(\mathbf{y}) = (\mathfrak{L}_{L_{\mathbf{f}}\overline{h}_{B}} + \mathfrak{L}_{\alpha}\mathfrak{L}_{\overline{h}_{B}})\epsilon_{\mathrm{m}}(\mathbf{y}), \quad b_{B}(\mathbf{y}) = \mathfrak{L}_{L_{\mathbf{g}}\overline{h}_{B}}\epsilon_{\mathrm{m}}(\mathbf{y}), \end{aligned} \tag{4.25}$$

for all $\tau_j \in \{0, \Delta_t, \ldots, T\}$ where $T \in \mathbb{R}_{>0}$ is the length of the time horizon and $\Delta_t \in [0, T)$ is the time discretization, and where $\mathfrak{L}$ represents the Lipschitz constant of its subscripted function on $\mathbb{R}^{n_x}$. The unification of these constructions enables the following definition:

Definition 4.4 (*Measurement-Robust Implicit Safe Set*)**.** *The set $\mathcal{C}_I \subseteq \mathcal{C} \subseteq \mathbb{R}^{n_x}$ defined as in* (3.65) *is a Measurement-Robust Implicit Safe Set (MRISS) for the error bound $\epsilon_{\mathrm{m}} : \mathbb{R}^{n_y} \to \mathbb{R}_{\geq 0}$ with parameter functions $a_0, b_0, a_{\Delta_t}, b_{\Delta_t}, \ldots, a_T, b_T, a_B, b_B : \mathbb{R}^{n_y} \to \mathbb{R}_{\geq 0}$ if:*

- *the functions $\{\overline{h}_0, \overline{h}_{\Delta_t}, \ldots, \overline{h}_T, \overline{h}_B\}$, their Lie derivatives, and α are Lipschitz continuous on $\mathcal{C}_I$,*
- *the constant $\mu \in \mathbb{R}_{\geq 0}$ satisfies the worst-case discretization error bound in* (3.68)*,*
- *and for all $\mathbf{x} \in \mathcal{C}_I$ there exists $\mathbf{u} \in \mathbb{R}^{n_u}$ satisfying* (4.24)*.*

Using this definition the safety of such sets can be made robust to measurement model uncertainty as formalized in the following theorem:

Theorem 4.5 (MRISS safety)**.** *Given a MRISS $\mathcal{C}_I$, if $\mathbf{k} : \mathbb{R}^{n_y} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ is a Lipschitz continuous controller that satisfies* (3.68) *with parameter functions* (4.25) *for all $\mathbf{x} \in \mathcal{C}_I$ with $\mathbf{y} = \mathbf{p}_{\mathrm{m}}(\mathbf{x})$ and $\widehat{\mathbf{x}} = \widehat{\mathbf{q}_{\mathrm{m}}}(\mathbf{y})$, then the closed loop system with measurement uncertainty* (4.2) *is safe with respect to $\mathcal{C}_I$.*

Proof. For any function $\overline{h} \in \{\overline{h}_0, \overline{h}_{\Delta_t}, \ldots, \overline{h}_T, \overline{h}_B\}$ let

$$c(\mathbf{x}, \mathbf{k}(\mathbf{y}, \widehat{\mathbf{x}})) = L_{\mathbf{f}}\overline{h}(\mathbf{x}) + L_{\mathbf{g}}\overline{h}(\mathbf{x})\mathbf{k}(\mathbf{y}, \widehat{\mathbf{x}}) + \alpha(\overline{h}(\mathbf{x}) - \nu), \tag{4.26}$$

where we choose $\nu = \mu$ if $\overline{h} = \overline{h}_{\tau_j}$ and $\nu = 0$ if $\overline{h} = \overline{h}_B$. It follows by Lipschitz continuity that:

$$\|L_{\mathbf{f}}\overline{h}(\widehat{\mathbf{x}}) - L_{\mathbf{f}}\overline{h}(\mathbf{x})\|_2 \leq \mathfrak{L}_{L_{\mathbf{f}}\overline{h}}\epsilon_{\mathrm{m}}(\mathbf{y}), \tag{4.27}$$

$$\|\alpha(\overline{h}(\widehat{\mathbf{x}}) - \nu) - \alpha(\overline{h}(\mathbf{x}) - \nu)\|_2 \leq \mathfrak{L}_{\alpha}\mathfrak{L}_{\overline{h}}\epsilon_{\mathrm{m}}(\mathbf{y}), \tag{4.28}$$

$$\|L_{\mathbf{g}}\overline{h}(\widehat{\mathbf{x}}) - L_{\mathbf{g}}\overline{h}(\mathbf{x})\|_2\|\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}})\|_2 \leq \mathfrak{L}_{L_{\mathbf{g}}\overline{h}}\epsilon_{\mathrm{m}}(\mathbf{y})\|\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}})\|_2. \tag{4.29}$$

As $\mathbf{k}$ satisfies (3.68), we have that:

$$c(\mathbf{x},\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}})) = c(\widehat{\mathbf{x}},\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}})) + c(\mathbf{x},\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}})) - c(\widehat{\mathbf{x}},\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}})), \tag{4.30}$$

$$\geq c(\widehat{\mathbf{x}},\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}})) - (a(\mathbf{y}) + b(\mathbf{y})\|\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}})\|_2) \geq 0. \tag{4.31}$$

Since $c(\mathbf{x},\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}})) \geq 0$ and μ satisfies (4.24), we have that the system (4.2) is safe with respect to $\mathcal{C}_I$ by [103, Lemma 2]. □

This result allows us to present an alternative to the BS-QP controller which adds the measurement-robustness of MRCBFs. The constraints (3.68) can be directly integrated into a Measurement-Robust Backup Set Optimization Program controller MR-BS-OP as:

$$\begin{aligned}\mathbf{k}(\mathbf{y},\widehat{\mathbf{x}}) = \underset{\mathbf{u}\in\mathbb{R}^{n_u}}{\operatorname{argmin}} \quad & \frac{1}{2}\|\mathbf{u} - \mathbf{k}_{\text{des}}(\widehat{\mathbf{x}})\|_2^2 \qquad \text{(MR-BS-OP)}\\ \text{s.t.} \quad & L_{\mathbf{f}}\overline{h}_{\tau_j}(\widehat{\mathbf{x}}) + L_{\mathbf{g}}\overline{h}_{\tau_j}(\widehat{\mathbf{x}})\mathbf{u} - (a_{\tau_j}(\mathbf{y}) + b_{\tau_j}(\mathbf{y})\|\mathbf{u}\|_2) \geq -\alpha(\overline{h}_{\tau_j}(\widehat{\mathbf{x}}) - \mu),\\ & L_{\mathbf{f}}\overline{h}_B(\widehat{\mathbf{x}}) + L_{\mathbf{g}}\overline{h}_B(\widehat{\mathbf{x}})\mathbf{u} - (a_B(\mathbf{y}) + b_B(\mathbf{y})\|\mathbf{u}\|_2) \geq -\alpha(\overline{h}_B(\widehat{\mathbf{x}}))\end{aligned}$$

for all $\tau_j \in \{0, \Delta_t, \ldots, T\}$, where again this controller is a second-order cone program (SOCP). Notably, the conservative nature of the method scales with the bound on the measurement-model error $\epsilon_{\mathrm{m}}(\mathbf{y})$ and the MR-BS-OP reduces to the BS-QP when $\epsilon_{\mathrm{m}}(\mathbf{y}) = 0$. We remark that the feasibility of MR-BS-OP for all $\widehat{\mathbf{x}} \in \mathbb{R}^{n_x}$ can be ensured by adding a slack variablez to the optimization problem as in the slackened R-MR-OP controller.

Experimental Results

We demonstrated the efficacy of the proposed MR-BS-OP controller on a modified Ninebot E+ Segway platform in a hardware experiment.

We consider the same segway model and simulation used in Example 4.3. The backup set method of Section 3.6 for generating control invariant sets is particularly relevant for this system due to its non-minimum phase dynamics [90] which make it

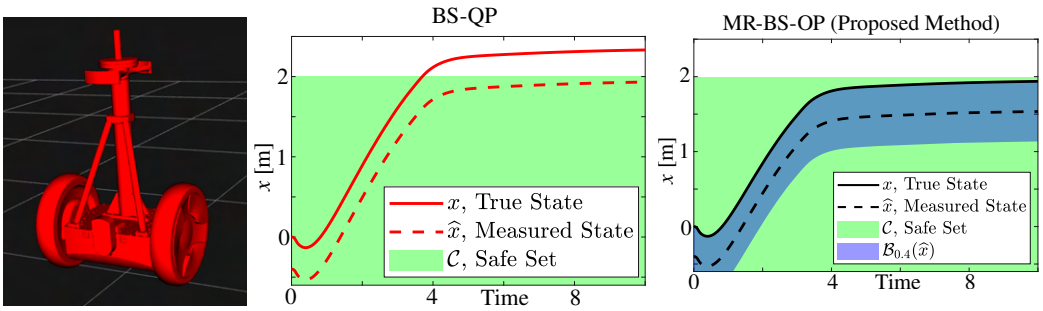


Figure 4.3. Simulation results for a measurement model of $\widehat{x} = x - 0.4$ m and constant desired velocity of 1 m/s. **(Left)** An image of the simulated Segway model. **(Center)** Trajectories generated using the BS-QP. Solid line represents the true state, dashed line shows the estimated state, and green region indicates the safe set $\mathcal{C}$. The true trajectory fails to be safe and exits the safe set at $t = 3$ s. **(Right)** Trajectories generated using the MR-BS-OP. An additional robustness region is plotted in blue to indicate the set of of true states which the control input renders safe. Both the true and measured trajectories are safe demonstrating the robustness of the MR-BS-OP when compared to the BS-QP.

difficult to synthesize control invariant sets that bound both the system position and angle.

The desired safe set was chosen empirically to be the set of states with position less than 2m from the origin, i.e., $\mathcal{C}_0 = \{\mathbf{x} \in \mathbb{R}^{n_x} : x \leq 2\}$ and $h_0(\mathbf{x}) = 2 - x$. The backup controller was an LQR controller on the system dynamics linearized about the upright position and the backup set was an estimate of the region of attraction of the LQR controller to the upright equilibrium state, given by a quadratic Lyapunov function. This set is then translated to match the current position of the Segway, while not allowing it to exceed the set boundary. The functions $\overline{h}_\tau$, $\tau \in [0, T]$ were converted into four CBFs $\overline{h}_{\tau_j}$ via the discretization $\Delta_t = T/3$. Lastly, the Lipschitz constants for $\overline{h}_{\tau_j}$ were found explicitly by inspection of the Segway dynamics and the Lipschitz constants for $\overline{h}_B$ were found by sampling the state space in simulation and taking the largest numerical gradient.

This method was first validated in simulation in a ROS-based environment. Measurement uncertainty was injected by artificially adding a constant error of -0.4m to the true state. The simulated scenario involved using a desired controller $\mathbf{k}_{\text{des}}$ that drove the Segway forward with a constant desired velocity of 1m/s. As seen in Figure 4.3, the MR-BS-OP provided robustness to this error. Importantly, without measurement-robustness, the system would be unsafe due to uncertainty in the state.

The MR-BS-OP was then implemented on hardware. State estimates for the velocity, pitch, and pitch rate were found using wheel incremental encoders and a VectorNav VN-100 IMU. The position estimate for x was obtained from an Intel RealSense

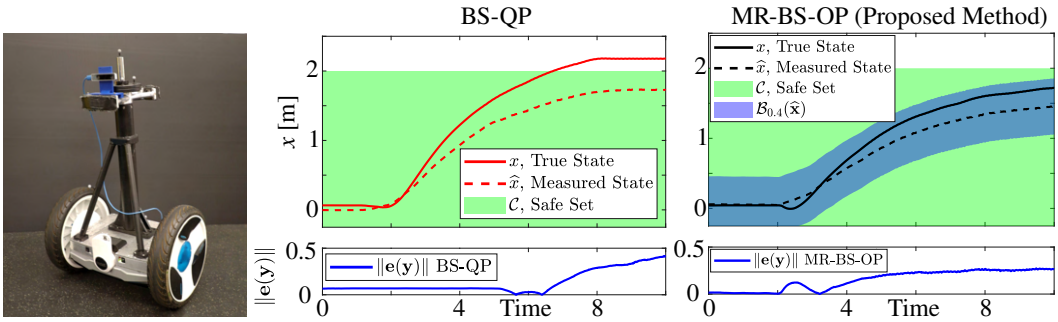


Figure 4.4. Experimental results using SLAM from the onboard Intel RealSense T265 and constant desired velocity of 1 m/s. These experiments can be seen in the supplementary video [126]. The notation and color schemes are the same as in Fig. 4.3. **(Left)** An image of the Segway platform. **(Center)** Trajectories generated using the BS-QP. The true trajectory exits the safe set at $t = 6.7$ s. The measurement error is plotted in blue. **(Right)** Trajectories generated using the MR-BS-OP. Both the true and measured trajectories are safe demonstrating the robustness of the MR-BS-OP when compared to the BS-QP.

T265 onboard camera. Onboard computation was performed by a Jetson TX2 which computes control actions and relays them to the low-level motor controllers. The TX2 concurrently runs Linux with ROS, enabling external communication and logging, and the ERIKA3 real-time operating system, which enables real-time low-level communication and computation of the control action.

As the state estimates provided by the encoders and IMU are highly accurate, we focus on making the system robust to measurement error in its vision-based position estimate $\widehat{x}$. An OptiTrack motion capture system was used in laboratory experiments to provide x estimates which are considered true. These closely matched the encoder position estimates for short trials, so the encoder x estimates were considered true in the outdoor experiments. This data was used to determine the error bound $\epsilon_{\text{m}}(\mathbf{y})$ that appears in the MRCBF constraint when using the onboard camera.

The value $\epsilon_{\text{m}}(\mathbf{y}) = 0.4$ was chosen as an estimated upper bound on the measurement error for all $\mathbf{y} \in \mathbf{p}_{\text{m}}(\mathcal{C})$. The MR-BS-OP was implemented at the embedded level in the ERIKA3 operating system using the ECOS SOCP solver [124]. The desired controller $\mathbf{k}_{\text{des}}$ was a proportional-derivative controller tracking user velocity inputs. The backup trajectory $\phi_\tau^{\mathbf{k}_B}(\widehat{\mathbf{x}})$ and its partial derivatives were approximated via Euler integration using a time step of $\Delta t = 5$ ms and the time used to expand the backup set $\mathcal{C}_B$ to $\mathcal{C}_I$ was $T = 1$ s. The MR-BS-OP ran at 250 Hz with 5 decision variables, 4 linear constraints, and 6 second order cone constraints and with inputs saturated at ± 20 Nm.

To demonstrate the method, a simple scenario is executed on the Segway in which

Figure 4.5. Images from the experiment using the MR-BS-OP controller. The Segway is piloted towards a wall of yellow boxes and the controller ensures that it remains safe, i.e., that it does not crash into the boxes. **(Top)** Time lapse of the Segway trajectory. **(Bottom)** Camera images taken from the perspective of the Segway throughout the experiment. The images are displayed in chronological order from left to right. A video can be found at [123].

it is driven forward at a desired velocity of 1 m/s. This scenario is performed with both the BS-QP and the MR-BS-OP. The results of these experiments can be found in Figure 4.4, images from the experiment can be seen in Figure 4.5, and a video can be found at [123]. With the BS-QP controller the estimated state $\widehat{\mathbf{x}}$ remains safe, but the true state $\mathbf{x}$ becomes unsafe whereas with the MR-BS-OP controller both the estimated and the true state are kept safe. This highlights the importance of providing robustness against measurement uncertainty, as achieved by Theorem 4.5.

Conclusion

In this section, we presented MRCBFs as tools for ensuring safety in the presence of measurement error. The resulting safety condition required by a MRCBF can be directly incorporated into an optimization based controller as a second order cone constraint, preserving the convexity of typical CBF-based controllers and facilitating control implementations on edge computers like the Jetson Tx2 at real-time speeds like 250 Hz. By unifying the robustness of the MRCBFs in Theorem 4.2 with the backup set-based safe set synthesis method of Section 3.6, we deploy our robust safety method on a segway robot with vision-based, inaccurate state estimation, demonstrating the utility and necessity of this robustness paradigm when compared to standard methods.

Later components of this thesis will discuss the case of unbounded measurement uncertainty (Chp. 6), but interesting future work remains in the discovering how measurement model errors can improve from targeted data acquisition and how the real-time utility of these methods can be retained while reducing the conservatism

introduced by the worst-case bounds.

4.3 Safety Under Real-World Uncertainties

So far in this thesis, we have introduced methods for dealing with bounded disturbances through the ISSf framework in Section 2.3, measurement uncertainty through the MRCBF framework in Section 4.2, and safe set synthesis in the context of bounded velocity tracking convergence rates in Section 3.3. In each case, the theoretical approaches successfully achieved safety guarantees in the presence of these real-world complications. Here we show that these methods can be combined to create a general control paradigm that is simultaneously robust to all of these real-world complexities.

The contributions of this section are as follows:

- We combine the robustness properties of MRCBFs [51] with those of ISSf-CBFs [31] and model-free safety-critical control [48] to achieve provable safety guarantees in the presence of measurement and dynamics uncertainty using reduced-order safe set synthesis. This is the first time that these methods have been combined.

The text for this section is adapted from:

> R. K. Cosner, M. Tucker, A. J. Taylor, K. Li, T. G. Molnar, W. Ubellacker, A. Alan, G. Orosz, Y. Yue, and A. D. Ames, "Safety-aware preference-based learning for safety-critical control," *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, Proceedings of Machine Learning Research, vol. 168, pp. 1020–1033, 2022. [Online]. Available: `https://proceedings.mlr.press/v168/cosner22a.html`,

General Robust Safety-Critical Control

Here we present the theoretical unification of the worst-case CBF robustification methods of sections 2.3, 3.3, and 4.2 which will provide a theoretical framework for general real-world safety-critical control that is robust to errors in the state-measurement model, uncertainties in the system dynamics model, and reduced-order model-based safe set synthesis.

Specifically, we start by considering the system with state and matched dynamic uncertainty:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})(\mathbf{v} + \mathbf{d}(t)) \tag{4.32}$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$ represents states of the system model and $\mathbf{v} \in \mathbb{R}^{n_v}$ is the reduced-order control input to the model. Here $\mathbf{v}$ is not necessarily the true input to the system and consider the case when there may be higher-order unmodeled dynamics that effect the ability of the system to track $\mathbf{v}$ as in the model-free discussion in Section 3.3.

Concisely, the following theorem summarizes the safety results achieved with with various CBF methods for a control-invariant safe set $\mathcal{C}$:

Theorem 4.6. *Consider the set $\mathcal{C}$ defined in (2.29).*

1. *Standard CBF: If h is a CBF for* (4.32) *on $\mathcal{C}$, $\mathbf{d}(t) = \mathbf{0}$ for $t \in \mathbb{R}_{\geq 0}$ and $\widehat{\mathbf{x}} = \mathbf{x}$, then there exists a controller $\mathbf{v} = \mathbf{k}(\widehat{\mathbf{x}})$ such that* (4.32) *is safe with respect to $\mathcal{C}$.*
2. *ISSf-CBF: If h is an ISSf-CBF (Def. 2.24) for the system* (4.32) *on $\mathcal{C}$ with parameter[1] $\varphi \in \mathbb{R}_{>0}$ and $\widehat{\mathbf{x}} = \mathbf{x}$, then there exists a controller $\mathbf{v} = \mathbf{k}(\widehat{\mathbf{x}})$ such that* (4.32) *is ISSf with respect to $\mathcal{C}$ with $\gamma(\overline{d}) = -\alpha^{-1}(-\overline{d}^2/(4\varphi))$ where $\alpha^{-1} \in \mathcal{K}_\infty^e$.*
3. *MRCBF: Assume $L_{\mathbf{f}}h$, $L_{\mathbf{g}}h$, and $\alpha \circ h$ are Lipschitz continuous on their domains, and assume that $\|\widehat{\mathbf{x}} - \mathbf{x}\| \leq \epsilon_{\mathrm{m}}$ for some $\epsilon_{\mathrm{m}} \in \mathbb{R}_{\geq 0}$. Then there exists $\underline{a}, \underline{b} \in \mathbb{R}_{\geq 0}$ such that if h is an MRCBF for* (4.32) *on $\mathcal{C}$ with parameters $a, b \in \mathbb{R}_{\geq 0}$ satisfying $a \geq \underline{a}$ and $b \geq \underline{b}$, and $\mathbf{d}(t) = \mathbf{0}$ for $t \in \mathbb{R}_{\geq 0}$, then there exists a controller $\mathbf{v} = \mathbf{k}(\widehat{\mathbf{x}})$ such that* (4.32) *is safe with respect to $\mathcal{C}$.*

Next, we propose a design paradigm that combines these guarantees to achieve general robust safety guarantees in the presence of real-world uncertainties for a multi-layered control system.

Combined Robust CBFs

We now combine the robustness properties of MRCBFs and ISSf-CBFs alongside the model-free safe-critical control method, to account for measurement uncertainty

[1] Unlike the original definition of ISSf-CBFs in Section 2, we use $\varphi = \frac{1}{\epsilon}$ here for notational convenience.

and matched disturbances allowing us to make robust safety guarantees for the full system (4.32). This is formalized in the following theorem:

Theorem 4.7. *Given the set $\mathcal{C}$ defined in* (2.29)*, the dynamics* (4.32)*, and $\alpha \in \mathcal{K}_\infty^e$, suppose the functions $L_\mathbf{f}h$, $L_\mathbf{g}h$, $\|L_\mathbf{g}h\|^2$, and $\alpha \circ h$ are Lip assumed to be Lipschitz continuous on their domains, and assume that $\|\widehat{\mathbf{x}} - \mathbf{x}\| \leq \epsilon_\text{m}$ for some $\epsilon_\text{m} \in \mathbb{R}_{\geq 0}$. Given these assumptions, there exists $\underline{a}, \underline{b} \in \mathbb{R}_{\geq 0}$ such that, if h satisfies:*

$$\sup_{\mathbf{v} \in \mathbb{R}^{n_u}} L_\mathbf{f}h(\widehat{\mathbf{x}}) + L_\mathbf{g}h(\widehat{\mathbf{x}})\mathbf{v} - \varphi\|L_\mathbf{g}h(\widehat{\mathbf{x}})\|^2 - a - b\|\mathbf{v}\| > -\alpha(h(\widehat{\mathbf{x}})), \tag{4.33}$$

for all $\mathbf{x} \in \mathbb{R}^{n_x}$ and some $\varphi \in \mathbb{R}_{>0}$ and $a, b \in \mathbb{R}_{\geq 0}$ satisfying $a \geq \underline{a}$ and $b \geq \underline{b}$, then there exists a controller $\mathbf{k} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ such that (4.32) *is ISSf with respect to $\mathcal{C}$ with $\gamma(\overline{d}) = -\alpha^{-1}\left(-\frac{\overline{d}^2}{4\varphi}\right)$ were $\alpha^{-1} \in \mathcal{K}_\infty^e$.*

Proof. First, we show that satisfying (4.33) for a particular set of $\underline{a}, \underline{b}$ implies satisfaction of the ISSf-CBF constraint (2.41). For this we choose:

$$\underline{a} = \epsilon(\mathfrak{L}_{L_\mathbf{f}h} + \mathfrak{L}_{\alpha\circ h} + \mathfrak{L}_{\varphi\|L_\mathbf{g}h\|^2}), \qquad \underline{b} = \epsilon_\text{m}\mathfrak{L}_{L_\mathbf{g}h}, \tag{4.34}$$

where $\mathfrak{L}$ indicates the Lipschitz coefficient of the subscripted function with respect to argument $\mathbf{x}$. Let us define the function $c : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \to \mathbb{R}$ such that:

$$c(\mathbf{x}, \mathbf{v}) \triangleq L_\mathbf{f}h(\mathbf{x}) + L_\mathbf{g}h(\mathbf{x})\mathbf{v} - \varphi\|L_\mathbf{g}h(\mathbf{x})\|^2 + \alpha(h(\mathbf{x})). \tag{4.35}$$

We then use this definition to construct the following bound on $c(\mathbf{x}, \mathbf{v})$ which holds for any $\mathbf{v} \in \mathbb{R}^{n_v}$:

$$\begin{aligned} c(\mathbf{x}, \mathbf{v}) &= c(\mathbf{x}, \mathbf{v}) + c(\mathbf{x}, \mathbf{v}) - c(\mathbf{x}, \mathbf{v}), && (4.36)\\ &\geq c(\mathbf{x}, \mathbf{v}) - \underbrace{\epsilon_\text{m}(\mathfrak{L}_{L_\mathbf{f}h} + \mathfrak{L}_{\alpha\circ h} + \mathfrak{L}_{\varphi\|L_\mathbf{g}h\|^2})}_{\underline{a}} - \underbrace{\epsilon\mathfrak{L}_{L_\mathbf{g}h}}_{\underline{b}}\|\mathbf{v}\| && (4.37)\\ &\geq c(\mathbf{x}, \mathbf{v}) - a - b\|\mathbf{v}\|. && (4.38) \end{aligned}$$

Above we added zero in (4.36) and used the Lipschitz coefficients and the worst-case uncertainty ϵ to achieve the bound in (4.37). Since $\sup_{\mathbf{v}\in\mathbb{R}^{n_v}} c(\mathbf{x}, \mathbf{v}) - a - b\|\mathbf{v}\| > 0$ holds based on (4.33), inequality (4.38) implies that (2.41) holds for the true parameters, $\boldsymbol{\rho}$. Since (2.41) holds, the conditions of Theorem 4.6 point 2 are satisfied and thus $\mathcal{C}$ is ISSf with $\gamma(\delta) = -\alpha^{-1}(-\delta^2/(4\varphi))$ where $\alpha^{-1} \in \mathcal{K}_\infty^\text{e}$. □

As in [15], (4.33) can be incorporated as a constraint into a safety filter on a locally Lipschitz continuous desired nominal controller $\mathbf{k}_\text{nom} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_v}$. We call this

filter the Tunable Robustified Optimization Program (TR-OP) (4.39) with tunable parameters $\rho, \varphi \in \mathbb{R}_{>0}$ and $, a, b \in \mathbb{R}_{\geq 0}$. We refer to this as *tunable* because these parameters can be tuned to account for the required robustness with respect to $\mathbf{k}(\widehat{\mathbf{x}})$ tracking convergence ($\rho \downarrow$ as tracking gets worse as discussed in Section 3.3), dynamics disturbance ($\varphi \uparrow$ as the disturbance bound grows), and measurement uncertainty ($a, b \uparrow$ as measurement accuracy degrades).

$$\begin{aligned} \mathbf{k}_{\mathrm{T}}(\mathbf{x}) = \underset{\mathbf{v}\in\mathbb{R}^{n_v}}{\operatorname{argmin}} \quad & \|\mathbf{v} - \mathbf{k}_{\text{des}}(\widehat{\mathbf{x}})\|^2 \\ \text{s.t. } & L_{\mathbf{f}}h(\widehat{\mathbf{x}}) + L_{\mathbf{g}}h(\widehat{\mathbf{x}})\mathbf{v} - \varphi\|L_{\mathbf{g}}h(\widehat{\mathbf{x}})\|^2 - a - b\|\mathbf{v}\| \geq -\rho h(\widehat{\mathbf{x}}). \end{aligned} \tag{4.39}$$

Here we use a linear class $\mathcal{K}_{\infty}^{\mathrm{e}}$ function $\alpha(r) = \rho r$ with coefficient $\rho \in \mathbb{R}_{>0}$. As with the MR-OP controller, this safety filter is a convex second-order cone program (SOCP) [64] for which an array of solvers exist [124].

Conclusion

This section provides a theoretical method for overcoming several sources of potential real-world uncertainty, providing a controller that can be tuned to account for hierarchical systems in the presence of measurement and dynamics model errors. It does so by introducing conservatism from several sources through the use of worst-case bounds uncertainty, convergence, and Lipschitz bounds. Due to this increased conservatism, we leave practical application of this controller to Section 5.4 where we will use preference-based learning to tune the parameters of the TR-OP controller (4.39) to desired required levels of conservatism and realize safe desirable real-world behavior.

4.4 End-to-End Safety for Learned Controllers

The previous sections of this chapter presented frameworks for robust safety-critical control under various bounded real-world complexities, including measurement uncertainty and matched disturbances. This constructions of this section most closely match the robustness methods for dynamics disturbances, but do so for the case of controller error, when the desired control action is not the one applied to the system.

In particular, this section generates guarantees in the case imitation learning where the is a bounded discrepancy between (a) an ideal, expert controller known to be robustly safe, and (b) the controller that is actually implemented, which may be a complex, end-to-end, learning-based policy. Using the robustification methods of this chapter, we find that, if the expert controller satisfies certain worst-case

robustness conditions, then its safety guarantees can be formally transferred to the learned controller, based on the worst-case deviation between the two.

The contributions of this section are as follows:

- The definition of *CBF-compiant* controllers, which characterize sufficient conditions for the learned controller to retain safety guarantees.
- Formal guarantees of safety (in an ISSf sense) where properties of the imitation learning problem directly affect the corresponding robust safe set.
- Simulated demonstrations of an inverted pendulum and a vehicle driving around a track using a vision-based end-to-end (i.e., perception-to-control) framework.

The text for this section is adapted from:

> R. K. Cosner, Y. Yue, and A. D. Ames, "End-to-end imitation learning with safety guarantees using control barrier functions," *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 5316–5322, 2022. DOI: 10.1109/CDC51059.2022.9993193,

Imitation Learning

Imitation learning (IL) is a common learning framework in which a mapping between observations and actions is trained using expert demonstrations. Common methodologies in IL include behavioral cloning (a form of supervised learning) and inverse reinforcement learning (IRL) [127] which learns a cost function such that the action or action sequence with minimal cost agrees with the expert demonstrations. We will present our method in the context of behavioral cloning, but note that our method is not specific to this form of IL and can be generalized to provide safety guarantees for IRL since the theory developed in this work depends on the learned controller itself and not the learning framework used to produce it.

For end-to-end IL we model sensor measurements as:

$$\mathbf{y} = \mathbf{p}_{\mathrm{m}}(\mathbf{x}) \tag{4.40}$$

where $\mathbf{y} \in \mathbb{R}^{n_y}$ represents the system observations and $\mathbf{p}_{\mathrm{m}} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ represents the system's sensors which we assume to be locally Lipschitz as in Section 4.2. In

the context of computer vision $\mathbf{y}$ may be a vector representation of image data and $\mathbf{p}_{\mathrm{m}}$ may be the camera sensor which maps from the state to an image.

In order to train an end-to-end controller, we collect a dataset of observation-input pairs using the expert controller $\mathbf{k} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$:

$$\mathfrak{D} = \{D_i\}_{i=1}^{n_x}, \qquad D_i = (\mathbf{p}_{\mathrm{m}}(\mathbf{x}_i), \mathbf{k}(\mathbf{x}_i)) \in \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \tag{4.41}$$

for $N \in \mathbb{N}$ samples[2]. Given a nonlinear function class $\mathcal{H} : \mathbb{R}^{n_y} \to \mathbb{R}^{n_u}$ and a loss function $\mathcal{L} : \mathbb{R}^{n_u} \times \mathbb{R}^{n_u} \to \mathbb{R}$, the learning problem can be expressed as optimizing the parameters θ of the function $\mathbf{k}_\theta \in \mathcal{H}$ via empirical risk minimization:

$$\min_{\mathbf{k}_\theta \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{n_x} \mathcal{L}\left(\mathbf{k}_\theta(\mathbf{p}_{\mathrm{m}}(\mathbf{x}_i)), \mathbf{k}(\mathbf{x}_i)\right). \tag{4.42}$$

This optimization problem attempts to minimize the difference between the learned and expert controllers, producing a hopefully small bound, which we can then use to produce robust control guarantees by treating the difference between the learned and expert controllers as a matched disturbance. The theoretical contribution of this section is can be summarized as ensuring that the generalization of this error bound between controllers does not outpace the expansion of $\mathcal{C}_\delta$ when extrapolating beyond the data set $\mathfrak{D}$.

Robust Safety and Continuity Properties

Behavioral cloning as in (4.42) suffers from compounding errors in the resulting trajectories [113]. However, since our goal is to transfer safety guarantees from the expert controller to the learned controller rather than to exactly mimic the expert behavior, we show that forward-invariance can be achieved despite compounding errors if the expert controller enforces robust forward-invariance.In order to introduce this form of required robustness, we must first introduce a required continuity property of the non-zero level sets of h, defining the potentially expanded sets $\mathcal{C}_\delta$ as in (2.38).

To do this, we define the c-level set of h using the preimage $h^{-1} : \mathbb{R} \rightsquigarrow \mathcal{P}(\mathbb{R}^{n_x})$,

$$h^{-1}(c) = \{\mathbf{x} \in \mathbb{R}^{n_x} \mid h(\mathbf{x}) = c\} \tag{4.43}$$

where $c \in \mathbb{R}$ and $\mathcal{P}(\mathbb{R}^{n_x})$ denotes the power set of $\mathbb{R}^{n_x}$.

One useful continuity property of point-to-set maps is upper semi-continuity (USC):

[2] As in [116], we assume the expert controller has access to the true state.

Definition 4.8 (Upper Semi-Continuity (USC) [69])**.** *A set valued function map $h^{-1} : \mathbb{R} \rightsquigarrow \mathcal{P}(\mathbb{R}^{n_x})$ is upper semi-continuous at $c \in \mathbb{R}$ if and only if for any $\epsilon > 0$, there exists $\eta > 0$ such that $c' \in \mathcal{B}_\eta(c) \implies h^{-1}(c') \subset h^{-1}(c) \oplus \mathcal{B}_\epsilon(\mathbf{0})$.*

where $\mathcal{B}_\eta(c)$ is the Euclidean ball of radius η centered at point c, and $\oplus$ indicates the Minkowski sum.

It was established in [70, Prop. 6] that, under common assumptions for CBFs, the level sets h^{-1} are USC as stated in:

Proposition 4.9 (Upper Semi-Continuity of CBF Level Sets [70, Prop. 6])**.** *Let $h^{-1} : \mathbb{R} \rightsquigarrow \mathcal{P}(\mathbb{R}^{n_x})$ be the preimage* (4.43) *representing the c-level set of some continuously differentiable function $h : \mathbb{R}^{n_x} \to \mathbb{R}$. If 0 is a regular value of h and $\Lambda := \{\mathbf{x} \in \mathbb{R}^{n_x} \mid -\delta \leq h(\mathbf{x}) \leq \delta\}$ is compact for all $\delta \geq 0$, then h^{-1} is upper semi-continuous at 0.*

This proposition relates the regularity of h to the upper semi-continuity of its level sets. In essence, the regularity of h at 0 ensures that small changes in the value defining the level set has a small effect on the level set itself.

Theoretical Result

Next we present the main result of this section that relates the supervised training of end-to-end controllers to intput-to-state safety. For this we consider the following closed loop system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{k}_\theta(\mathbf{p}_\mathrm{m}(\mathbf{x})). \tag{4.44}$$

Next we assume that the expert controller is a TR-OP safety filter (4.39) as in the previous section which provides additional robustness to matched disturbances, measurement uncertainty, and hierchical safe-set synthesis methods. The practical use case for which could involve generating the expert data in $\mathfrak{D}$ by using the TR-OP safety filter to modify human actions, thus adding robustness to the human's control inputs.

With the robust controller $\mathbf{k}_\mathrm{T}$ as the expert controller, we define the properties of *CBF-compliancy* which will allow us to transfer safety guarantees to the system under the end-to-end learned controller $\mathbf{k}_\theta$ as in (4.44).

Definition 4.10 (CBF-Compliancy)**.** *The learned controller* $\mathbf{k}_\theta : \mathbb{R}^{n_y} \to \mathbb{R}^{n_u}$ *is CBF-compliant for* $h : \mathbb{R}^{n_x} \to \mathbb{R}$ *with measurement function* $\mathbf{p}_\mathrm{m} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ *if*

$$\min_{\mathbf{x}\in\mathfrak{D}} \|\mathbf{x}_1 - \mathbf{x}\| \leq r_1, \qquad \forall \mathbf{x}_1 \in \partial\mathcal{C}, \tag{4.45}$$

$$\|\mathbf{k}_T(\mathbf{x}_2) - \mathbf{k}_\theta(\mathbf{p}_\mathrm{m}(\mathbf{x}_2))\| \leq M_e, \qquad \forall \mathbf{x}_2 \in \mathfrak{D}, \tag{4.46}$$

$$\|\mathbf{k}_\theta(\mathbf{p}_\mathrm{m}(\mathbf{x}_3)) - \mathbf{k}_\theta(\mathbf{p}_\mathrm{m}(\mathbf{x}_4))\| \leq \mathfrak{L}_{\mathbf{k}_\theta\circ\mathbf{p}_\mathrm{m}}\|\mathbf{x}_3 - \mathbf{x}_4\|, \quad \forall \mathbf{x}_3, \mathbf{x}_4 \in \partial\mathcal{C} \oplus \overline{\mathcal{B}}_{r_2}(\mathbf{0}), \tag{4.47}$$

where $r_1, r_2 \in \mathbb{R}_{>0}$*,* $\mathfrak{L}_{\mathbf{k}_\theta}, M_e \in \mathbb{R}_{\geq 0}$*, and* $\mathbf{k}_T : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ *is a 4.39 controller for* h *with parameters* $\varphi, a, b \in \mathbb{R}_{\geq 0}$ *and* $\alpha \in \mathcal{K}^e_\infty$*.*

Intuitively, r_1 in (4.45) is a parameter that indicates the sample density of the data set $\mathfrak{D}$ near the boundary of the safe set. M_e in (4.46) represents the maximum amount of matched disturbance error between the expert and learned controllers on the data set. Finally, (4.47) is a requirement on the smoothness of the learned controller which constrains how it can generalize away from data points near the boundary of the safe set.

Next, we use the USC property of h to relate the existence of a CBF-compliant controller to the ISSf property of system (4.44).

Theorem 4.11. *Let* $\mathcal{C} \subset \mathbb{R}^{n_x}$ *be the 0-superlevel set of a function* $h : \mathbb{R}^{n_x} \to \mathbb{R}$ *which satisfies the Proposition 4.9. There exist* $\underline{\varphi}, \underline{a}, \underline{b} \in \mathbb{R}_{\geq 0}$ *such that, if*

- $\mathbf{k}_\theta : \mathbb{R}^{n_y} \to \mathbb{R}^{n_u}$ *is a CBF-compliant controller on* $\mathcal{C}$ *for parameters* $\varphi \geq \underline{\varphi}, a \geq \underline{a}, b \geq \underline{b}$*,* $\alpha \in \mathcal{K}^e_\infty$ *with constants* $r_1, r_2 > 0$
- *and* $L_\mathbf{f}h, L_\mathbf{g}h, \|L_\mathbf{g}h\|^2$*, and* $\alpha \circ h$ *are Lipschitz continuous on* $\partial\mathcal{C} \oplus \overline{B}_{r_2}(\mathbf{0})$*,*

then the closed loop system (4.44) *is ISSf with respect to* $\mathcal{C}$ *and safe with respect to*

$$\mathcal{C}_\delta = \left\{ \mathbf{x} \in \mathbb{R}^{n_x} \middle| h(\mathbf{x}) \geq \alpha^{-1}\left(\frac{-1}{2\varphi}(\mathfrak{L}_{\mathbf{k}_\theta\circ\mathbf{p}_\mathrm{m}} r_3 + M_\mathbf{e})^2\right) \right\} \tag{4.48}$$

where $r_3 \triangleq r_2 + r_1$*.*

The proof of this theorem formalizes the following idea: sufficient sampling of $\partial\mathcal{C}$ and upper semi-continuity of h ensure that the points in $\partial\mathcal{C}_\delta$ remain sufficiently close to $\partial\mathcal{C}$ to limit extrapolation error, thus producing control inputs that are accurate enough to ensure the forward invariance of $\mathcal{C}_\delta$ and prevent the cascading failure mode that is typical of behavioral cloning.

Proof. Consider some state $\mathbf{x}_3 \in \partial\mathcal{C} \oplus \mathcal{B}_{r_2}(\mathbf{0})$. Given $\mathbf{x}_3$, there must exist some $\mathbf{x}_2 \in \partial\mathcal{C}$ such that $\|\mathbf{x}_2 - \mathbf{x}_3\| \leq r_2$. Additionally, since $\mathbf{k}_\theta$ is a CBF-compliant controller with the appropriate parameters, there must be some $\mathbf{x}_1 \in \mathfrak{D}$ such that $\|\mathbf{x}_1 - \mathbf{x}_2\| \leq r_1$ and so $\|\mathbf{x}_1 - \mathbf{x}_3\| \leq r_3$ by the triangle inequality.

The function h satisfies Proposition 4.9 by assumption so by the definition of upper semi-continuity we can bound the expansion of the safe set:

$$\exists \eta > 0 \text{ s.t. } c \in \mathcal{B}_\eta(0) \implies h^{-1}(c) \subset h^{-1}(0) \oplus \mathcal{B}_{r_2}(\mathbf{0}). \tag{4.49}$$

Thus we can choose $\underline{\varphi} > 0$ large enough such that for any $\varphi \geq \underline{\varphi}$ the expansion remains in a r_2 expansion of the original boundary $\partial\mathcal{C}_\delta \subset \partial\mathcal{C} \oplus \mathcal{B}_{r_2}(\mathbf{0})$ for $\mathcal{C}_\delta$ as in (4.48). Next, we choose the remaining parameter bounds to be:

$$a \geq \underline{a} = r_3(\mathfrak{L}_{L_\mathbf{f}h} + \mathfrak{L}_{\alpha\circ h} + \mathfrak{L}_{\varphi\|L_\mathbf{g}h\|^2}), \tag{4.50}$$

$$b \geq \underline{b} = r_3\mathfrak{L}_{L_\mathbf{g}h}. \tag{4.51}$$

where $\mathfrak{L}$ represents the Lipschitz constant of the subscripted function on $\partial\mathcal{C}\oplus\mathcal{B}_{r_2}(\mathbf{0})$.

Using $\mathbf{k}_\theta$ we can bound the time derivative of the CBF at $\mathbf{x}_3 \in \mathcal{C} \oplus \mathcal{B}_{r_2}(\mathbf{0})$ as:

$$\frac{d}{dt}h(\mathbf{x}_3, \mathbf{k}_\theta(\mathbf{p}_\mathrm{m}(\mathbf{x}_3))) \tag{4.52}$$

$$\begin{aligned}
&= L_\mathbf{g}h(\mathbf{x}_1)\mathbf{k}_T(\mathbf{x}_1) + \frac{d}{dt}h(\mathbf{x}_3, \mathbf{k}_\theta(\mathbf{p}_\mathrm{m}(\mathbf{x}_3))) - L_\mathbf{g}h(\mathbf{x}_1)\mathbf{k}_T(\mathbf{x}_1),\\
&\geq L_\mathbf{f}h(\mathbf{x}_3) - L_\mathbf{f}h(\mathbf{x}_1) + r_3\mathfrak{L}_{L_\mathbf{f}h} + \alpha(h(\mathbf{x}_3)) - \alpha(h(\mathbf{x}_1)) + r_3\mathfrak{L}_{\alpha\circ h}, \qquad (4.53)\\
&\quad + \varphi\|L_\mathbf{g}h(\mathbf{x}_1)\|^2 - \varphi\|L_\mathbf{g}h(\mathbf{x}_3)\|^2 + r_3\mathfrak{L}_{\varphi\|L_\mathbf{g}h\|^2},\\
&\quad + L_\mathbf{g}h(\mathbf{x}_3)\mathbf{k}_T(\mathbf{x}_1) - L_\mathbf{g}h(\mathbf{x}_1)\mathbf{k}_T(\mathbf{x}_1) + r_3\mathfrak{L}_{L_\mathbf{g}h}\|\mathbf{k}_T(\mathbf{x}_1)\|,\\
&\quad - \alpha(h(\mathbf{x}_3)) + \varphi\|L_\mathbf{g}h(\mathbf{x}_3)\|^2 + L_\mathbf{g}h(\mathbf{x}_3)(\mathbf{k}_\theta(\mathbf{p}_\mathrm{m}(\mathbf{x}_3)) - \mathbf{k}_T(\mathbf{x}_1)).
\end{aligned}$$

We can now bound the first three lines of the lower bound in (4.53) using the assumed Lipschitz constants. For example,

$$\begin{aligned}
&L_\mathbf{f}h(\mathbf{x}_3) - L_\mathbf{f}h(\mathbf{x}_1) + r_3\mathfrak{L}_{L_\mathbf{f}h} \geq -\|L_\mathbf{f}h(\mathbf{x}_3) - L_\mathbf{f}h(\mathbf{x}_1)\| + r_3\mathfrak{L}_{L_\mathbf{f}h}, \qquad (4.54)\\
&\geq \mathfrak{L}_{L_\mathbf{f}h}(r_3 - \|\mathbf{x}_1 - \mathbf{x}_2 + \mathbf{x}_2 - \mathbf{x}_3\|) \geq \mathfrak{L}_{L_\mathbf{f}h}(r_3 - \|\mathbf{x}_1 - \mathbf{x}_2\| - \|\mathbf{x}_2 - \mathbf{x}_3\|) \geq 0
\end{aligned}$$

since $\|\mathbf{x}_1 - \mathbf{x}_2\| \leq r_1$ and $\|\mathbf{x}_2 - \mathbf{x}_3\| \leq r_2$.

Applying these Lipschitz-based bounds for $L_\mathbf{f}h$, $L_\mathbf{g}h$, $\alpha \circ h$, and $\varphi\|L_\mathbf{g}h\|^2$ in (4.53) yields:

$$\begin{aligned}
\frac{d}{dt}h(\mathbf{x}_3, \mathbf{k}_\theta(\mathbf{p}_\mathrm{m}(\mathbf{x}_3))) \geq &- \alpha(h(\mathbf{x}_3)) + \varphi\|L_\mathbf{g}h(\mathbf{x}_3)\|^2 \qquad (4.55)\\
&+ L_\mathbf{g}h(\mathbf{x}_3)(\mathbf{k}_\theta(\mathbf{p}_\mathrm{m}(\mathbf{x}_3)) - \mathbf{k}_T(\mathbf{x}_1)).
\end{aligned}$$

Additionally we can lower bound the final term using properties (4.46) and (4.47) of CBF-compliant controller as:

$$
\begin{aligned}
&L_{\mathbf{g}}h(\mathbf{x}_3)(\mathbf{k}_\theta(\mathbf{p}_{\mathrm{m}}(\mathbf{x}_3)) - \mathbf{k}_T(\mathbf{x}_1)) \\
&\geq L_{\mathbf{g}}h(\mathbf{x}_3)(\mathbf{k}_\theta(\mathbf{p}_{\mathrm{m}}(\mathbf{x}_3)) - \mathbf{k}_\theta(\mathbf{p}_{\mathrm{m}}(\mathbf{x}_1)) + \mathbf{k}_\theta(\mathbf{p}_{\mathrm{m}}(\mathbf{x}_1)) - \mathbf{k}_T(\mathbf{x}_1)), && (4.56) \\
&\geq -\|L_{\mathbf{g}}h(\mathbf{x}_3)\|\Big(\|\mathbf{k}_\theta(\mathbf{p}_{\mathrm{m}}(\mathbf{x}_3)) - \mathbf{k}_\theta(\mathbf{p}_{\mathrm{m}}(\mathbf{x}_1))\| + \|\mathbf{k}_\theta(\mathbf{p}_{\mathrm{m}}(\mathbf{x}_1)) - \mathbf{k}_T(\mathbf{x}_1)\|\Big), \\
&\geq -\|L_{\mathbf{g}}h(\mathbf{x}_3)\|(\mathfrak{L}_{\mathbf{k}_\theta \circ \mathbf{p}_{\mathrm{m}}} r_3 + M_e). && (4.57)
\end{aligned}
$$

Using (4.57) to lower-bound (4.55) results in:

$$
\begin{aligned}
&\frac{d}{dt}h(\mathbf{x}_3, \mathbf{k}_\theta(\mathbf{p}_{\mathrm{m}}(\mathbf{x}_3))) && (4.58) \\
&\geq -\alpha(h(\mathbf{x}_3)) + \varphi\|L_{\mathbf{g}}h(\mathbf{x}_3)\|^2 - \|L_{\mathbf{g}}h(\mathbf{x}_3)\|(\mathfrak{L}_{\mathbf{k}_\theta \circ \mathbf{p}_{\mathrm{m}}} r_3 + M_e), && (4.59) \\
&\geq -\alpha(h(\mathbf{x}_3)) - \frac{1}{2\varphi}(\mathfrak{L}_{\mathbf{k}_\theta \circ \mathbf{p}_{\mathrm{m}}} r_3 + M_e)^2, && (4.60)
\end{aligned}
$$

where the final bound is achieved by completing the square and removing the positive term.

To achieve forward invariance of $\mathcal{C}_\delta$ we note that

$$
h(\mathbf{x}_3) = \alpha^{-1}\left(-\frac{1}{2\varphi}(\mathfrak{L}_{\mathbf{k}_\theta} r_3 + M_e)\right) \implies \frac{d}{dt}h(\mathbf{x}_3, \mathbf{k}_\theta(\mathbf{x}_3)) \geq 0. \tag{4.61}
$$

Since the bound (4.60) holds for all $\mathbf{x}_3 \in \partial\mathcal{C} \oplus \mathcal{B}_{r_2}(\mathbf{0})$ and $\underline{\varphi}$ was chosen such that $\partial\mathcal{C}_\delta \subset \partial\mathcal{C} \oplus \mathcal{B}_{r_2}(\mathbf{0})$ it is true that $\frac{d}{dt}h(\mathbf{x}_4, \mathbf{k}_\theta(\mathbf{x}_4)) \geq 0$ for all $\mathbf{x}_4 \in \partial\mathcal{C}_\delta$. Thus by Nagumo's theorem [13] the set $\mathcal{C}_\delta$ is forward invariant and $\mathcal{C}$ is ISSf. □

We recognize that finding and using the exact Lipschitz constants may be impractical, but note that due to their conservatism the CBF-compliant controller may be capable of achieving safety with far smaller values as will be later demonstrated in simulation.

The learned controller $\mathbf{k}_\theta$ developed in Theorem 4.11 has mathematical guarantees of safety, but may result in behaviors significantly different than the expert controller in the interior of the safe set $\mathrm{Int}(\mathcal{C})$ when the system is far from the boundary $\partial\mathcal{C}$, since the sampling assumption (4.45) is only required to hold on $\partial\mathcal{C}$. Therefore we present a corollary which generally results in significantly improved behavioral cloning on the interior of $\mathcal{C}$ due to increased sampling. The safety guarantees of this corollary follow immediately from Theorem 4.11.

Corollary 4.12. *Let the dataset $\mathfrak{D}$ satisfy the inequality*

$$\min_{\mathbf{x}\in\mathfrak{D}} \|\mathbf{x}_1 - \mathbf{x}\| \leq r_1, \qquad \forall \mathbf{x}_1 \in \mathcal{C} \tag{4.62}$$

in place of (4.45) *for some $r_1 > 0$. Let the remaining assumptions of Theorem 4.11 hold, then the closed loop system* (4.44) *is ISSf with respect to $\mathcal{C}$ and safe with respect to $\mathcal{C}_\delta$* (4.48)*.*

Proof. $\mathcal{C}$ is a closed set so $\partial\mathcal{C} \subseteq \mathcal{C}$, thus (4.62) $\implies$ (4.45) and the conditions of Theorem 4.11 are met. □

Simulation Results

Next we discuss the simulation results that demonstrate safe vision-based end-to-end control of an inverted pendulum and a simplified car using CBF-compliant controller. In both cases the convolutional neural network used for end-to-end learning was MobileNetV2 [128] with an additional fully-connected layer added to generate control inputs of the proper dimension. The full network has approximately 3.4 million parameters. Training was performed using the ADAM optimizer, an ℓ_2 loss with ℓ_2 weight decay, and batched training. The frequency of the observations was chosen to be 100 Hz and 60 Hz for the pendulum and car, respectively. The simulations were conducted using zero-order-hold control inputs of the same frequency with no latency.

Inverted Pendulum with Image Feedback: We first consider an inverted pendulum system with the states $\mathbf{x} = \begin{bmatrix}\theta & \dot{\theta}\end{bmatrix}^\intercal$ with torque inputs $\tau \in \mathbb{R}$ as shown in Fig. 4.6. The dynamics and observation function of this system are given as:

$$\dot{\mathbf{x}} = \begin{bmatrix}\dot{\theta} \\ \sin\theta\end{bmatrix} + \begin{bmatrix}0 \\ \tau\end{bmatrix}, \qquad \mathbf{y} = \mathbf{p}_{\mathrm{m}}(\mathbf{x}) = \begin{bmatrix}\mathrm{Img}(\mathbf{x}) \\ \dot{\theta}\end{bmatrix} \tag{4.63}$$

where $\mathrm{Img}(\mathbf{x})$ represents the image of the system at state $\mathbf{x}$ as seen from a camera facing the inverted pendulum. A example images can be found in Figure 4.6. The learned controller is a function of the current image and velocity of the system, so an additional fully connected layer was added to incorporate the velocity into the end-to-end controller.

The safe set for the inverted pendulum is chosen to be:

$$h(\mathbf{x}) = c - \mathbf{x}^\intercal \mathbf{p}_{\mathrm{m}} \mathbf{x}, \qquad \mathcal{C} = \{\mathbf{x} \in \mathbb{R}^{n_x} \mid h(\mathbf{x}) \geq 0\} \tag{4.64}$$

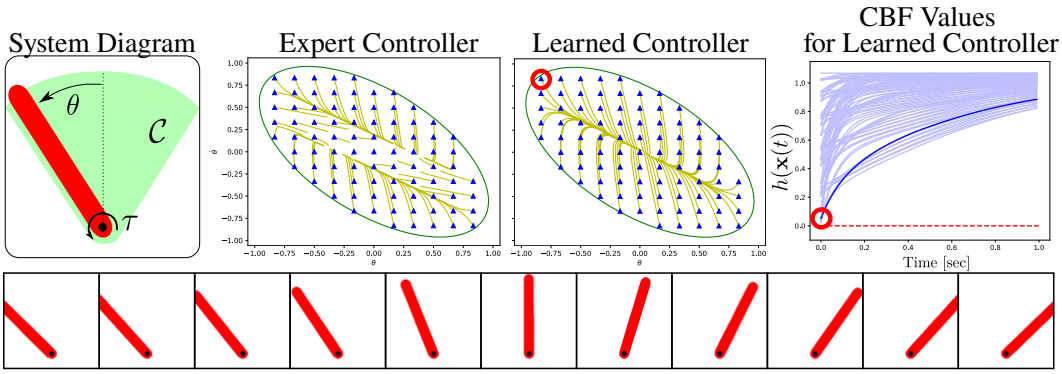


Figure 4.6. Results for the inverted pendulum. **Left:** A diagram of the system where the green represents the safe set $\mathcal{C}$. **Center Left:** One second long trajectories generated by the expert controller $\mathbf{k}_T$ are shown in yellow and plotted for several initial conditions represented by blue triangles. The green ellipse marks the boundary $\partial\mathcal{C}$. **Center Right:** One second long trajectories generated by the learned controller $\mathbf{k}_\theta$ are shown in yellow and plotted for several initial conditions represented by blue triangles. in The green ellipse marks the boundary $\partial\mathcal{C}$. **Right:** CBF values $h(\mathbf{x}(t))$ achieved by the learned controller. Note all are greater than zero indicating safety of the system. The darker blue trajectory begins at the initial condition marked by the red circle in the plot of the learned controller. **Bottom:** Images spanning the safe set $\mathcal{C}$ that are used by $\mathbf{k}_\theta$ for end-to-end control of the system.

where $\mathbf{p}_\text{m} \in \mathbb{R}^{2\times 2}$ is such that $\mathbf{x}^\intercal \mathbf{p}_\text{m} \mathbf{x}$ is a control Lyapunov function derived from the continuous time algebraic Ricatti equation using feedback linearization and c is chosen such that $\max_{\theta\in\mathcal{C}} |\theta| = \pi/4$. This safe set is visualized in Fig. 4.6.

The expert controller is the TR-OP controller (4.39) with parameters $\varphi = 2$, $\alpha(c) = c$, and a and b chosen as the Lipschitz constants of $(L_\mathbf{f} h(\mathbf{x}) + \alpha(h(\mathbf{x})) + \varphi\|L_\mathbf{g} h(\mathbf{x})\|^2))$ and $L_\mathbf{g} h(\mathbf{x})$, respectively, over the compact set $\partial\mathcal{C}$ multiplied by the minimum sampling distance $r_1 = 0.01$. The nominal controller is $\mathbf{k}_\text{nom}(\mathbf{x}) = -0.75\theta$ which provides some torque to counteract gravity, but fails to stabilize the pendulum. The boundary of the safe set, $\partial\mathcal{C}$, is gridded and sampled uniformly with a minimum distance r_1 to create the training dataset $\mathfrak{D}$.

Simplified Race Car with Image Feedback: Next we consider a simplified car given by the unicycle dynamics and observation function:

$$\dot{\mathbf{x}} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \qquad \mathbf{y} = \mathbf{p}_\text{m}(\mathbf{x}) = \text{Img}(\mathbf{x}) \tag{4.65}$$

where the state $\mathbf{x} = \begin{bmatrix} x & y & \theta \end{bmatrix}^\intercal$ is the planar position and heading angle and the input $\mathbf{u} = \begin{bmatrix} v & \omega \end{bmatrix}^\intercal$ is the forward and angular velocities and $\text{Img}(\mathbf{x})$ represents the driver's first-person-view from the car at position $\mathbf{x}$. A series of example first-person-view images can be seen in Figure 4.7.

The safe set for the car is chosen to be the 0-superlevel set of the function $\min\{h_1, h_2\}$ where:

$$h_i(\mathbf{x}) = \delta \widehat{\mathbf{n}}^\intercal \widehat{\mathbf{d}}_i + \psi_i \cdot \begin{cases} \rho_i^2 - \left(\left(x - \frac{\ell}{2}\right)^2 + y^2 \right), & x \geq \frac{\ell}{2} \\ \rho_i^2 - \left(\left(x + \frac{\ell}{2}\right)^2 + y^2 \right), & x \leq \frac{-\ell}{2} \\ \rho_i^2 - y^2, & \text{else} \end{cases}$$

where $\rho_1 = (\ell/\pi + w)$, $\psi_1 = 1$, $\rho_2 = \ell/\pi$, and $\psi_2 = -1$. Additionally, $\delta = 0.1$, $\widehat{\mathbf{n}}$ is the unit vector in the car's heading direction, $\widehat{\mathbf{d}}_1$ is the unit vector pointing perpendicularly inward from the outer boundary of the track through the car's position, $\widehat{\mathbf{d}}_2$ is the unit vector point perpendicularly outward from the inner boundary of the track through the car's position, ℓ is the length of the straight portions of the track, w is the width of the track. An annotated diagram of the track can be found in Figure 4.7. Given these functions, the safe set $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^3 \mid \min\{h_1(\mathbf{x}), h_2(\mathbf{x})\} \geq 0\}$ is a subset of the track with an angle dependence where positions with heading angles pointed towards the center line are considered safer.

The expert controller is the TR-OP controller (4.39) with the constraint simultaneously enforced for both h_1 and h_2 with parameters $\varphi = 0.5$, $a = 10^{-2}$, $b = 10^{-4}$, and $\alpha(c) = 10c$. $\partial\mathcal{C}$ was gridded and sampled uniformly with distance of $r_1 = 0.1$ to generate $\mathfrak{D}$. We use Theorem 4.11 to guide the choice of these constants, but due to the difficulty of estimating the Lipschitz constants and the likely over-conservatism of the resulting controller we choose parameters which are likely much smaller than those required to sufficiently guarantee safety mathematically but we nonetheless succeed in demonstrating safety experimentally.

The nominal controller used in the 4.39 controller is:

$$\mathbf{k}_{\text{nom}} = \begin{bmatrix} K_p |r - r_{\text{mid}}| + F \\ K_r (r - r_{\text{mid}}) + K_{\text{dir}} (\widehat{\mathbf{n}}^\intercal \widehat{\mathbf{e}}_{\text{mid}}) \end{bmatrix} \tag{4.66}$$

where $K_p, F, K_r, K_{\text{dit}} \in \mathbb{R}_{>0}$, $r = \|[x,\ y]\|$, r_{mid} is the distance from the origin to the middle the track along a line passing through the car, $\widehat{\mathbf{e}}_{\text{mid}}$ is the unit vector from the vehicle to middle line of the track. This nominal controller is capable of circumnavigating the track, but is unsafe.

Learning and Results: For both the inverted pendulum and the car, the learned controller $\mathbf{k}_\theta$ is trained until convergence to minimize (4.42) where the (4.39) controller is the expert controller.

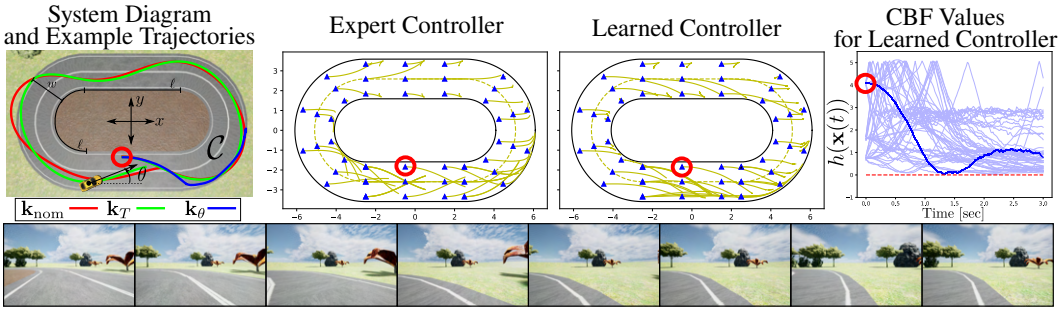


Figure 4.7. Results for the car. **Left:** Three 15 second long trajectories are shown starting from the same initial condition, the nominal controller generates the unsafe red trajectory, the 4.39 controller generates the safe green trajectory, and the learned controller generates the safe blue trajectory. **Center Left:** 3 second long trajectories starting at several initial conditions are shown for the expert controller $\mathbf{k}_T$. The blue triangles represent initial conditions and are all within the safe set. The yellow lines represent the trajectories of the car beginning at a blue triangle. **Center Right:** 3 second long trajectories for starting at several initial conditions shown for the learned controller $\mathbf{k}_\theta$. **Right:** CBF values $\min\{h_1(\mathbf{x}(t)), h_2(\mathbf{x}(t))\}$ achieved by the learned controller. Note all are greater than zero. The darker blue trajectory begins at the initial condition indicated by the red circles in each plot. **Bottom:** Images used by $\mathbf{k}_\theta$ for end-to-end control. From left to right, the first-person view of the trajectory indicated using the red circles starting at time $t = 0$ seconds and increasing by 0.25 seconds.

The safe set for both systems was gridded with initial conditions and simulated forward for 1 second for the inverted pendulum and 3 seconds for the car. For the car, $\theta = 0$ was held constant for each initial condition and the interior of the track was sampled. The trajectories can be seen in Figures 4.6 and 4.7. For each trajectory, the 4.39 controller renders the system safe and this safety is transferred to the learned controller despite having different closed-loop behavior. The minimum of h achieved for the inverted pendulum example was 0.028 and the smallest value of $\min\{h_1(\mathbf{x}), h_2(\mathbf{x})\}$ achieved by the car for all initial conditions was 0.030, indicating safety of both systems.

Even though the system deviates significantly from the expert trajectories, the learned controller successfully keeps the system inside of the safe set. Thus, although additional sampling can be performed to improve the learned behavior, sampling on the boundary of the safe set is sufficient to render it forward invariant.

Conclusion

This section provides a guarantee of robust safety for end-to-end, learning-based controllers by bounding the deviation between an expert and a learned control policy. Rather than requiring precise models or measurements, we assume a worst-case bound on this deviation and establish conditions under which safety can be formally transferred to the learned controller.

The definition of CBF-compliancy and Theorem 4.11 provide a lens through which to view safety-critical imitation learning. Specifically, they emphasize the importance of sampling near the boundary of the safe set, regularizing learned policies to ensure smoothness, minimizing imitation error, and ensuring that the safe set function h is sufficiently regular.

4.5 Conclusion

This chapter presented robustified control methods which can be used to achieve rigorous, mathematical guarantees of safety despite the presence of significant uncertainty arising from realistic sources such as measurement uncertainty, dyanmics uncertainty, and imperfect learning methods.

The unifying tools used by these and many other robust control methods [4], [28], [107] are worst-case bounds of the uncertainty and/or worst-case slope bounds (i.e., Lipschitz constants) of the relevant functions. These two bounds can then be used to translate between the true, unknown value of a function and a the predicted value plus the worst-case effect of the uncertainty. This allows us to analytically bound the worst-case affect of uncertainty which we can then use to generate robust inputs.

Unfortunately, these methods are overly conservative due to their reliance on these worst-case bounds. In general, real-world uncertainties will not take on this sort of adversarial behavior and the true functions do not uniformly follow their slope bounds. Thus, although the guarantees of this section are robust, they are often result in large sacrifices in performance to obtain unnecessary robustness. The following chapters of this thesis will seek to improve performance without sacrificing safety during deployment.

Chapter 5

LEARNING-BASED IMPROVEMENTS

"*The safest way to avoid crashing your car is to never drive it.*"

"*Perfect is the enemy of good*" - Voltaire

"*All models are wrong, but some are useful.*" - George Box

Robust safety guarantees typically rely on worst-case over approximations. As discussed in Chapter 4, these powerful guarantees enable formal assurances of safety even in the face of real-world uncertainty. However, they often achieve this by sacrificing performance. After all, the safest car is one that never leaves the garage.

This chapter addresses the conservatism of robust safety methods by leveraging tools from machine learning to generate preferable *good* behavior in place of *perfect* safety guarantees with poor performance. In particular, we explore how learning can (1) improve the underlying model by reducing the amount of required robustness, and (2) quantify intangible safety parameters like human-tolerable risk or responsibility levels.

Abstract

Robust safety-critical control offers a principled framework for ensuring safety in the presence of complex uncertainty and has enabled the deployment of safety guarantees on real-world robotic systems. Unfortunately, despite this utility, they often rely on worst-case over-approximations of uncertainty, resulting in highly conservative behaviors and significant compromises to system performance.

This chapter presents several learning-based approaches to reduce such conservatism. Since robust formulations typically aim to bound the difference between a model and the real world, a straightforward paradigm to improve performance is simply to improve the model itself. With this goal in mind, Section 5.2 develops an episodic learning method that models uncertainty's effect on safety to enable safe bipedal locomotion over stepping stones and Section 5.3 introduces a self-supervised online learning method for estimating environment-dependent perception uncertainty in a stereo vision system. Alternatively, the second half of

the chapter leverages the mathematical structure of robust safety-critical control in conjunction with data-driven techniques to endow controllers with an understanding of intangible, human-centered concepts like desirable robustness-performance tradeoffs (Section 5.4) and social responsibility (Section 5.5) which are key in characterizing desired closed-loop behavior. The methods presented in this chapter forego the theoretical guarantees of robust safety-critical control and instead use these formulations as a tunable foundation for achieving robot behavior that is both safe and performant.

Published content: The text for this chapter is predominately adapted from four works: the stepping-stone discussion is adapted from [27], the stereo-vision text is adapted from [55], the preference-based learning work is adapted from [53], and the responsibility-learning work is adapted from [56].

5.1 Introduction

Although robust CBF-based methods have seen considerable success in maintaining theoretical safety guarantees under various real-world uncertainties, such as measurement error [51], [52], dynamics disturbances [31], [32], hierarchical control [48], [87], and sampled-data implementations [129], these approaches generally rely on assumed bounds on uncertainty and select control actions that are robust to worst-case, adversarial uncertainties. While some alternative methods have relaxed this boundedness assumption and retained theoretical guarantees by pursuing probabilistic guarantees [21], [130], [131], we defer a more in-depth exploration of robust probabilistic safety guarantees to Chapter 6 and instead, in this chapter, focus on improving the performance of deterministic, bound-based methods using machine learning techniques.

Although safety is paramount in real-world systems, overly conservative controllers that fail to achieve performance goals are often not practically useful. As a result, a significant body of work has moved away from formal robustness guarantees in favor of achieving more desirable real-world behavior. In particular, many recent approaches use learning-based techniques to enable safe and performant closed-loop behavior, even when relying on black-box function approximators. For instance, machine learning has been used to augment model predictive control (MPC) to improve performance while satisfying safety constraints under uncertainty [88], [132], and has also been combined with control barrier functions (CBFs) to model system uncertainty more accurately [39], reducing reliance on worst-case assumptions.

This chapter adopts this paradigm of using machine learning to address real-world uncertainties and extends prior methods in two key directions: (1) by learning improved uncertainty models for high-dimensional dynamics and measurement systems, either in an episodic or online fashion, with applications to new use cases; and (2) by learning mathematical representations of human-centered concepts that are critical for safe and effective behavior, such as risk-performance tradeoffs and responsibility allocation in multi-agent systems.

To explore these directions, this chapter presents four representative examples. First, in Section 5.2, we consider a planar bipedal robot tasked with traversing a set of stepping stones under significant model uncertainty. We apply an episodic learning method to reduce the safety impact of that uncertainty, enabling the robot to complete the task safely and effectively. In Section 5.3, we address measurement uncertainty in stereo vision. We develop an online learning method that estimates the effect of image features on the error distribution of a vision-based estimator which enables safe operation in scenarios where general worst-case bounds would prevent effective operation.

In Sections 5.4 and 5.5, we move beyond learning improvements to existing models and instead focus on learning intangible, human-centered quantities. First, we apply a Preference-Based Learning (PBL) method to tune a conservative safety filter (the TR-OP controller, Eq. 4.39) to achieve improved performance without compromising safe behavior. Then, we consider the case of autonomous vehicles operating in decentralized multi-agent environments with humans, and propose a method for learning context-dependent safety constraints that reflect social responsibility while still enabling effective system behavior.

Together, the methods in this chapter build on the robust safety frameworks developed in earlier chapters to achieve both safety and performance despite significant real-world uncertainty. While they depart from the formal guarantees of the previous chapters, they retain the structure and interpretability of robust safety-critical control theory. This approach allows learned models to remain meaningful and grounded, even as they enable practical, high-performance robotic behavior under real-world uncertainty.

5.2 Learning Dynamics Uncertainty

I first present our work on achieving safe bipedal locomotion over complex terrain. In particular we focus on the “stepping-stone problem” where the robot’s feet must

be placed in precise locations at each step, a difficult task made even more difficult by the significant uncertainties in the robot's dynamics which can lead to safety failures unless properly accounted for.

This stepping-stone task is a historical benchmark for evaluating the safety-critical control of biped platforms [133] that, requires dynamic gaits, due to the underactuated nature of the bipedal robot. During such dynamic motion, satisfaction of dynamic safety constraints is predicated on having perfect model information, a requirement that impossible to meet for real-world systems. However, simply being robust to a worst-case bound on this uncertainty would likely leave the system unable to complete task, thus suggesting the need for model uncertainty reduction.

To attenuate the impact of model uncertainty on safety, we consider a machine learning approach. However, one challenge in applying learning methods to robotics is the need for diverse data capturing system behavior. While it is possible to collect high-coverage data sets in simulated environments that accurately represent how inputs affect the evolution of the system, collecting such data on real physical systems may be prohibitively costly or damaging to the system. The lack of this data can lead to challenges with under-determination in supervised learning problems that seek to preserve the underlying structure of dynamic systems [39]. This data sparsity leads to models with low training loss, but poor closed-loop performance when deployed on the real system.

In this section we present an approach that overcomes these model uncertainty and data-sparsity problems by the learning the effect of model uncertainty on safety episodically over several rollouts. This allows us to improve the understanding of the uncertainty iteratively across each episode, sampling the data most relevant to the closed-loop control problem.

The contributions of this section are as follows:

- An episodic learning framework for iteratively reducing the impact of disturbances on the safety-critical behavior of a system.
- The first experimental demonstration of CBFs for safety-critical control on a bipedal robot.

The text for this section is adapted from:

N. Csomay-Shanklin, R. K. Cosner, M. Dai, A. J. Taylor, and A. D. Ames, "Episodic learning for safe bipedal locomotion with control barrier functions and projection-to-state safety," *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, vol. 144, pp. 1041–1053, 2021. [Online]. Available: `https://proceedings.mlr.press/v144/csomay-shanklin21a.html`,

A video for this section can be found at [134].

Model Uncertainty and Projection-to-State Safety

In practice, the system dynamics (2.1) are not known during control design process due to parametric error and unmodeled dynamics. Instead, a nominal model of the system is utilized:

$$\dot{\widehat{\mathbf{x}}} = \widehat{\mathbf{f}}(\mathbf{x}) + \widehat{\mathbf{g}}(\mathbf{x})\mathbf{u}, \tag{5.1}$$

where $\widehat{\mathbf{f}} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ and $\widehat{\mathbf{g}} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x \times n_u}$ are assumed to be Lipschitz continuous on $\mathbb{R}^{n_x}$. By adding and subtracting the right hand side of (5.1) to (2.1), the dynamics of the system are:

$$\dot{\mathbf{x}} = \widehat{\mathbf{f}}(\mathbf{x}) + \widehat{\mathbf{g}}(\mathbf{x})\mathbf{u} + \overbrace{\underbrace{\mathbf{f}(\mathbf{x}) - \widehat{\mathbf{f}}(\mathbf{x})}_{\mathbf{b}(\mathbf{x})} + \underbrace{(\mathbf{g}(\mathbf{x}) - \widehat{\mathbf{g}}(\mathbf{x}))}_{\mathbf{A}(\mathbf{x})}\mathbf{u}}^{\mathbf{d}(\mathbf{x},\mathbf{u})}, \tag{5.2}$$

where the unknown disturbance $\mathbf{d}(\mathbf{x}, \mathbf{u}) = \mathbf{b}(\mathbf{x}) + \mathbf{A}(\mathbf{x})\mathbf{u}$ is assumed to be time invariant, but depends on the state and input to the system. If the function $h : \mathbb{R}^{n_x} \to \mathbb{R}$ is a CBF for the nominal model (5.1) on $\mathcal{C}$, the uncertainty in the dynamics directly manifests in the time derivative of h:

$$\dot{h}(\mathbf{x}, \mathbf{u}) = \underbrace{\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x})(\widehat{\mathbf{f}}(\mathbf{x}) + \widehat{\mathbf{g}}(\mathbf{x})\mathbf{u})}_{\dot{\widehat{h}}(\mathbf{x},\mathbf{u})} + \underbrace{\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x})\mathbf{b}(\mathbf{x})}_{b(\mathbf{x})} + \underbrace{\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x})\mathbf{A}(\mathbf{x})}_{\mathbf{a}(\mathbf{x})^\top}\mathbf{u}. \tag{5.3}$$

Given that h is a CBF for (5.1) on $\mathcal{C}$, let $\mathbf{k} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ be a Lipschitz continuous state-feedback controller such that $\dot{\widehat{h}}(\mathbf{x}, \mathbf{k}(\mathbf{x})) \geq -\alpha(h(\mathbf{x}))$. Defining the *projected disturbance* as:

$$d_h(\mathbf{x}) \triangleq \dot{h}(\mathbf{x}, \mathbf{k}(\mathbf{x})) - \dot{\widehat{h}}(\mathbf{x}, \mathbf{k}(\mathbf{x})) = b(\mathbf{x}) + \mathbf{a}(\mathbf{x})^\top \mathbf{k}(\mathbf{x}), \tag{5.4}$$

yields:

$$\dot{h}(\mathbf{x}, \mathbf{k}(\mathbf{x})) \geq -\alpha(h(\mathbf{x})) - d_h(\mathbf{x}). \tag{5.5}$$

Assuming that d_h is essentially bounded in time (there exists $M \in \mathbb{R}$, $\overline{d_h} > 0$, such that $\|d_h\|_\infty \triangleq \operatorname{ess\,sup}_{t\geq 0} \|d_h(\mathbf{x}(t))\| < \overline{d_h}$), we may make use of the following definition:

Definition 5.1 (Projection-to-State Safety (PSSf) [108, Def. 6])**.** *Given a feedback controller* $\mathbf{k}$*, the closed-loop system* (2.2)*,* $\dot{\mathbf{x}} = \widehat{\mathbf{f}}(\mathbf{x}) + \widehat{\mathbf{g}}(\mathbf{x})\mathbf{k}(\mathbf{x}) + \mathbf{d}(\mathbf{x})$ *with* $\mathbf{d}(\mathbf{x}) = \mathbf{b}(\mathbf{x}) + \mathbf{A}(\mathbf{x})\mathbf{k}(\mathbf{x})$*, is* projection-to-state safe *(PSSf) on* $\mathcal{C}$ *with respect to the function* h *and projected disturbances* $d_h : \mathbb{R}^{n_x} \to \mathbb{R}$ *if there exists* $\overline{d_h} > 0$ *and* $\gamma \in \mathcal{K}_\infty$ *such that the expanded set* $\mathcal{C}_\delta \supset \mathcal{C}$*,*

$$\mathcal{C}_\delta \triangleq \{\mathbf{x} \in \mathbb{R}^{n_x} : h(\mathbf{x}) \geq -\gamma(\|d_h\|_\infty)\}\,, \tag{5.6}$$

is forward invariant for all d_h *satisfying* $\|d_h\|_\infty \leq \overline{d_h}$*.*

PSSf captures the fact that in the presence of model uncertainty, satisfying the CBF condition (2.33) for the estimated time derivative $\widehat{\dot{h}}$ is not sufficient for safety of $\mathcal{C}$, as the projected disturbance d_h appears in the lower bound on true time derivative of h as in (5.5). This results in a larger forward invariant set, given by $\mathcal{C}_{d_h}$, that grows with the magnitude of the projected disturbance.

Learning Projected Disturbances

Next we explore how an estimate of the projected disturbance d_h can be learned episodically from data and incorporated into control synthesis to improve PSSf behavior.

As in (5.5) the projected disturbance d_h appears in the time derivative of the CBF $\dot{h}$, and potentially leads to unsafe behavior since it compromises the CBF condition (2.33). If an upper bound $\overline{d_h}$ on $\|d_h\|_\infty$ is known (or determined heuristically), it could be directly incorporated into the inequality enforced in the controller:

$$\begin{aligned} \mathbf{k}(\mathbf{x}) = \ \underset{\mathbf{u}\in\mathbb{R}^{n_u}}{\operatorname{argmin}} \quad & \frac{1}{2}\|\mathbf{u} - \mathbf{k}_d(\mathbf{x})\|_2^2 \qquad\qquad (\overline{d_h}\text{-CBF-QP}) \\ \text{s.t.} \quad & \widehat{\dot{h}}(\mathbf{x}, \mathbf{u}) - \overline{d_h} \geq -\alpha(h(\mathbf{x})). \end{aligned}$$

While this will enforce safety of the original set $\mathcal{C}$, it can be exceedingly conservative if $\overline{d_h}$ is larger than the actual projected disturbance. Furthermore, as the projected disturbance is a function of the state, its magnitude (and possibly sign) may change along a trajectory, leading to additional conservatism in this approach.

Instead, we consider a learning approach to resolve the impact of d_h. To motivate such an approach, consider the following setting: in an experiment, the system is

allowed to evolve forward in time from a particular initial condition and under a given state-feedback controller. During this experiment, data is collected which provides a discrete-time history of the CBF, h. This time history is smoothed and numerically differentiated to compute an approximate time history of the true value of the time derivative of the CBF, $\dot{h}$. This yields a collection of input-output pairs:

$$D_i = ((\mathbf{x}_i, \mathbf{k}(\mathbf{x}_i)), \dot{h}_i) \in (\mathbb{R}^{n_x} \times \mathbb{R}^{n_u}) \times \mathbb{R} \tag{5.7}$$

whereby a dataset $\mathfrak{D} = \{D_i\}_{i=1}^{n_{\text{data}}}$ can be constructed. Given a nonlinear function class $\mathcal{H} : \mathbb{R}^{n_x} \to \mathbb{R}$ and a loss function $\mathcal{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, a learning problem can be specified as finding a function $\hat{d}_\theta \in \mathcal{H}$ with parameters $\theta \in \mathbb{R}^{n_\theta}$ to estimate d_h via empirical risk minimization:

$$\inf_{\hat{d}_\theta \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{n_{\text{data}}} \mathcal{L}\left(\widehat{\dot{h}}(\mathbf{x}_i, \mathbf{k}(\mathbf{x}_i)) + \hat{d}_\theta(\mathbf{x}_i), \dot{h}_i\right). \tag{ERM}$$

A controller can then be synthesized which incorporates $\hat{d}_\theta$ as follows:

$$\begin{aligned} \mathbf{k}(\mathbf{x}) = \ & \underset{\mathbf{u} \in \mathbb{R}^{n_u}}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{u} - \mathbf{k}_{\text{des}}(\mathbf{x})\|_2^2 \\ & \text{s.t.} \qquad \widehat{\dot{h}}(\mathbf{x}, \mathbf{u}) + \hat{d}_\theta(\mathbf{x}) \geq -\alpha(h(\mathbf{x})). \end{aligned} \tag{$\hat{d}_\theta$-CBF-QP}$$

Note that compared with the CBF-QP (2.36), the extended safe set generated by $\hat{d}_\theta$-CBF-QP shrinks from (5.6) to

$$\left\{\mathbf{x} \in \mathbb{R}^{n_x} : h(\mathbf{x}) \geq -\gamma(\|d_h - \hat{d}_\theta\|_\infty)\right\}. \tag{5.8}$$

To implement this controller we build upon the episodic learning framework from [39], [135] by seeking to learn d_h. Our approach is outlined in Algorithm 5.2. In each episode, the algorithm runs the current controller to collect data, learns a new $\hat{d}_\theta$ using the newly collected data, and synthesizes a new controller. In this prior work, which was applied to less complex dynamical systems, the collected data was rich enough to determine a control affine structure. In many contexts, such as bipedal robots, such a degree of diversity is infeasible without damaging the system. We instead directly learn d_h as a function of the previous controller $\mathbf{k}$ via a recursive relationship, as updating the estimator leads to the definition of a new projected disturbance $d_h' = b(\mathbf{x}) + \mathbf{a}(\mathbf{x})^\top \mathbf{k}'(\mathbf{x})$. This yields a projected disturbance d_h learned iteratively by modifying $\hat{d}_\theta$ over the course of multiple episodes.

Algorithm 5.2: Projected Disturbance Learning (PDL)

input: CBF h, CBF derivative estimate $\widehat{\dot{h}}$, model class $\mathcal{H}$, loss function $\mathcal{L}$, nominal state-feedback controller $\mathbf{k}_0$, number of episodes T, initial condition $\mathbf{x}_0$

output: Augmented Controller $\mathbf{k}_T$

for $j = 1, \dots, T$ **do**

$\mathfrak{D}_j \leftarrow$ `experiment(`$\mathbf{x}_0, \mathbf{k}_{j-1}$`)` // Execute experiment

$\hat{d}_\theta \leftarrow$ `ERM(`$\mathcal{H}, \mathcal{L}, \mathfrak{D}_j, \widehat{\dot{h}}_0$`)` // Fit estimator

$\widehat{\dot{h}}_j \leftarrow \widehat{\dot{h}}_0 + \hat{d}_\theta$ // Update derivative estimator

$\mathbf{k}_j \leftarrow \hat{d}_\theta$`-CBF-QP (`$\widehat{\dot{h}}_j$`)` // Synthesize new controller

Bipedal Robotics: Dynamics

Next we specify the notion of learning projected disturbances to the setting of bipedal locomotion. We briefly introduce the theory of bipedal locomotion and then describe the barrier function formulations which allow us to achieve safe bipedal locomotion across stepping-stones. A deeper exploration of this material may be found in [136].

The bipedal robotic system we consider is the AMBER-3M robotic platform seen in Figure 5.1, modeled as an underactuated, planar five-link robot with point feet [137] whose physical parameters are reported in [138, Table 1]. The configuration coordinates $\mathbf{q} \in \mathcal{Q} \subset \mathbb{R}^5$ are given by $\mathbf{q} = [q_{sf},\ q_{sk},\ q_{sh},\ q_{nsh},\ q_{nsk}]^\top$, with stance foot angle q_{sf}, stance knee angle q_{sk}, stance hip angle q_{sh}, non-stance hip angle q_{nsh} and non-stance knee angle q_{nsk}. The continuous-time equations of motion, derived from the Euler-Lagrange equations as in (3.5), are given by:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}\mathbf{u}, \tag{5.9}$$

where $\mathbf{D}(\mathbf{q}) \in \mathbb{S}^5_{++}$ is the positive definite mass-inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{5\times 5}$ contains the centrifugal and Coriolis forces, $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^5$ contains the gravitational forces, $\mathbf{B} \in \mathbb{R}^{5\times 4}$ is the actuation matrix, and $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^4$ is the input. For AMBER-3M, the number of inputs is one fewer than the degrees of freedom, meaning the system has one degree of underactuation.

Taking $p^v : \mathcal{Q} \to \mathbb{R}$ to represent the vertical position (height) of the swing foot, the admissible states are given by the *domain* $\mathcal{D} = \{(\mathbf{q}, \dot{\mathbf{q}}) \in T\mathcal{Q} \mid p^v(\mathbf{q}) \geq 0\}$. The switching surface on which the impact events occur, also known as the *guard*, is

defined by:

$$\mathcal{S} = \{(\mathbf{q}, \dot{\mathbf{q}}) \in T\mathcal{Q} \mid p^v(\mathbf{q}) = 0, \dot{p}^v(\mathbf{q}, \dot{\mathbf{q}}) < 0\} \subset \mathcal{D}. \tag{5.10}$$

With the full system state given by $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}}) \in T\mathcal{Q}$, the impact dynamics [139] are defined by a *reset map* $\Delta : \mathcal{S} \to \mathcal{D}$ relating pre-impact states $\mathbf{x}^-(t) \triangleq \lim_{\tau \nearrow t} \mathbf{x}(\tau)$ and post-impact states $\mathbf{x}^+(t) = \lim_{\tau \searrow t} \mathbf{x}(\tau)$ via $\mathbf{x}^+(t) = \Delta(\mathbf{x}^-(t))$. Combining these concepts and rearranging (5.9) into control affine form yields the following *hybrid control system*:

$$\mathcal{HC} = \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} & \mathbf{x}^- \in \mathcal{D} \setminus \mathcal{S}, \\ \mathbf{x}^+ = \Delta(\mathbf{x}^-) & \mathbf{x}^- \in \mathcal{S}. \end{cases} \tag{5.11}$$

Bipedal Robotics: Control

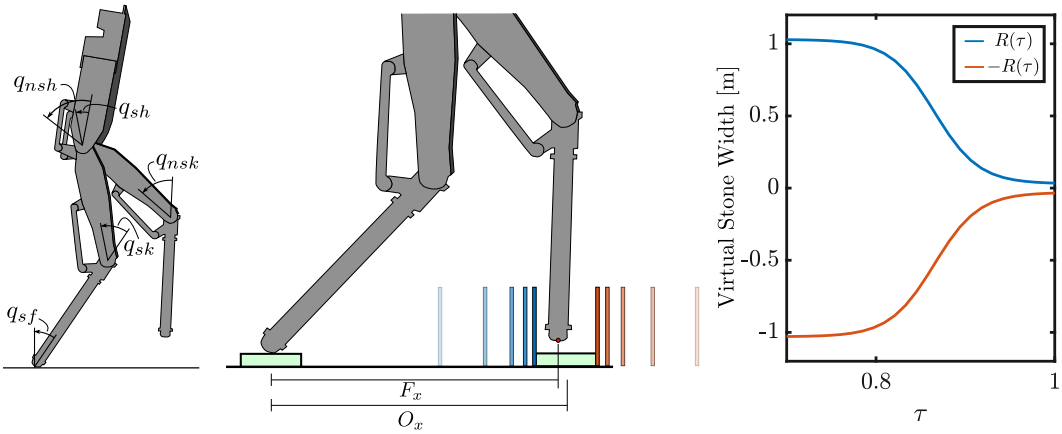
Our control scheme for the planar AMBER-3M bipedal robot centers around a *phasing variable*, $\tau : \mathcal{Q} \to [0, 1]$, given by:

$$\tau(\mathbf{q}) = \frac{\delta_{hip}(\mathbf{q}) - \delta_{hip}^+}{\delta_{hip}^- - \delta_{hip}^+}, \tag{5.12}$$

where $\delta_{hip} : \mathbb{R}^5 \to \mathbb{R}$ defined as $\delta_{hip}(\mathbf{q}) = [-l_t - l_f,\ -l_f,\ 0,\ 0,\ 0]\mathbf{q}$ is the linearized hip position with l_t and l_f the length of the robot's tibia and femur, respectively. The constants δ_{hip}^+ and δ_{hip}^- are the linearized hip positions at the beginning and the end of a step, ensuring that $\tau(\mathbf{q})$ increases monotonically in time within a step. Desired trajectories resulting in walking gaits for the robot can be rapidly synthesized via a hybrid zero dynamics framework [62], [136]. We are now well equipped to define the relative degree 2 (see Def. 3.9) outputs $\mathbf{y} : \mathcal{Q} \to \mathbb{R}^4$ as the difference between the actual output $\mathbf{y}_a$ and the desired output trajectory $\mathbf{y}_d$:

$$\mathbf{y}(\mathbf{q}, \boldsymbol{\alpha}) \triangleq \mathbf{y}_a(\mathbf{q}) - \mathbf{y}_d(\tau(\mathbf{q}), \boldsymbol{\alpha}), \tag{5.13}$$

with $\boldsymbol{\alpha}$ being the coefficients of a Bézier polynomial coming from the trajectory generation step. The actual output is given by the actuated coordinates: $\mathbf{y}_a(\mathbf{q}) = \begin{bmatrix} \mathbf{0}_{4\times1} & \mathbf{I}_{4\times4} \end{bmatrix} \mathbf{q}$. The nominal controller for this system is then given by the proportional-derivative controller $\mathbf{k}_d(\mathbf{x}) = \mathbf{k}_{PD}(\mathbf{x}) \triangleq -\mathbf{K}_P\mathbf{y}(\mathbf{q}) - \mathbf{K}_D\dot{\mathbf{y}}(\mathbf{q})$ with positive definite proportional gain matrices $\mathbf{K}_P \in \mathbb{S}_{++}^4$ and derivative gain $\mathbf{K}_D \in \mathbb{S}_{++}^4$.



Robot schematic and stepping stone definition.

Figure 5.1. (Left): Schematic diagram of the AMBER-3M robot with position coordinates. **(Center):** Schematic of the foot placement in the stepping-stone problem. The boundaries of virtual stepping-stones are captured via the blue and orange vertical lines. **(Right):** Virtual stepping stone width as function of the phase variable $\tau(\mathbf{q})$.

Stepping-Stone CBFs

The stepping-stone problem is encodd through the use of *virtual* stepping-stones, which shrink over the course of a step to confine foot placement to a safe region defined on a targeted stone as in [25]. The safety criteria used to specify these foot position constraints are given by:

$$h_1(\mathbf{q}) = R(\tau(\mathbf{q})) - (O_x - F_x(\mathbf{q})), \tag{5.14}$$

$$h_2(\mathbf{q}) = R(\tau(\mathbf{q})) + (O_x - F_x(\mathbf{q})), \tag{5.15}$$

where $F_x(\mathbf{q})$ is the horizontal position of the swing foot and $O_x > 0$ is the horizontal position of the center of stepping-stone. The virtual stone width is given by the function $R : \mathbb{R} \to \mathbb{R}$:

$$R(\tau(\mathbf{q})) = \frac{ar - 1}{1 + ar(e^{-m(\tau(\mathbf{q})-1)} - 1)} + 1 + r \tag{5.16}$$

where $m > 0$ determines the decay rate of the barrier function, $(1+a)r$ is half of the targeted stone width, and $1+r$ defines the half the width of the virtual stepping-stone when $\tau = 0$. These functions are visualized in Figure 5.1. The safety constraints can be interpreted as keeping the swing foot horizontal position in an interval centered at the middle of the stepping-stone, where the interval shrinks as τ increases. As this formulation of CBFs is position-based and therefore relative degree two, we employ the exponential control barrier function (ECBF) extension technique [75] to both CBFs to attain the relative degree 1 CBFs: $h_{e,i}(\mathbf{x}) \triangleq L_{\mathbf{f}} h_i(\mathbf{x}) + \alpha_e h_i(\mathbf{q})$.

Combining the episodic learning frame work and the stepping stone CBFs with $\hat{\mathbf{f}}$ and $\hat{\mathbf{g}}$ being the nominal model in (5.11), the final Stepping Stone QP (SS-QP)

controller combines the robustifying term of ($\hat{d}_\theta$-CBF-QP) with the stepping-stone ECBF extensions of (5.14) and (5.15):

$$\begin{aligned}\mathbf{k}(\mathbf{x}) = \underset{\mathbf{u}\in\mathbb{R}^{n_u}}{\operatorname{argmin}} \quad & \frac{1}{2}\|\mathbf{u} - \mathbf{k}_{PD}(\mathbf{x})\|_2^2 && \text{(SS-QP)}\\ \text{s.t.} \quad & L_{\hat{\mathbf{f}}}^2 h_1(\mathbf{x}) + L_{\hat{\mathbf{g}}} L_{\hat{\mathbf{f}}} h_1(\mathbf{x})\mathbf{u} + \alpha_e L_{\hat{\mathbf{f}}} h_1(\mathbf{x}) + \hat{d}_{\theta,1}(\mathbf{x}) \geq -\alpha(h_{e,1}(\mathbf{x}))\\ & L_{\hat{\mathbf{f}}}^2 h_2(\mathbf{x}) + L_{\hat{\mathbf{g}}} L_{\hat{\mathbf{f}}} h_2(\mathbf{x})\mathbf{u} + \alpha_e L_{\hat{\mathbf{f}}} h_2(\mathbf{x}) + \hat{d}_{\theta,2}(\mathbf{x}) \geq -\alpha(h_{e,2}(\mathbf{x})).\end{aligned}$$

We assume that this SS-QP is feasible and we encountered no infeasibilities in simulation or experimentation.

Simulation and Experimental Validation

Next, we apply our episodic learning framework (Algorithm 5.2) to the AMBER-3M platform in both simulation with injected model uncertainty and on hardware with the model error inherent to real-world systems. In each instance the estimator $\hat{d}_\theta$ was implemented as a neural network with two hidden layers of 50 hidden units using the ReLU activation function. The network was trained minimizing mean absolute error using mini-batch gradient descent. Mean absolute error was chosen over other loss functions for its robustness to outliers. The same controller (SS-QP) was deployed in the RaiSim [140] simulation environment and on the AMBER-3M hardware platform, as seen in the supplementary video [134].

Simulation: The controllers and learning algorithm were first validated in simulation. Model error was introduced by increasing the inertia of all limbs on the true model by a factor of ten while maintaining constant mass. Due to the underactuated nature of the robot and the relationship between step length and zero dynamics stability, not every set of stepping stones is navigable, even if safety is perfectly enforced with respect to the CBFs. Therefore, a feasible stepping stone configuration was first generated for the robot to traverse with stones of 4 cm in width. Without knowledge of the modified model ($\hat{d}_{\theta,1}(\mathbf{x}) = \hat{d}_{\theta,2}(\mathbf{x}) = 0$), the controller did not satisfy the stepping-stone safety criteria (5.14-5.15), with a maximum violation at foot placement of 2.0 cm, causing the robot to miss the stepping stone and fall over. Three episodes of the PDL algorithm were run, after which the maximum violation was reduced to be 0.3 cm, only 15% of the original violation. Additionally, the $\overline{d_h}$-CBF-QP controller was implemented, which ensured safety but resulted in extremely conservative behavior, resulting in poor qualitative walking, i.e., harsh

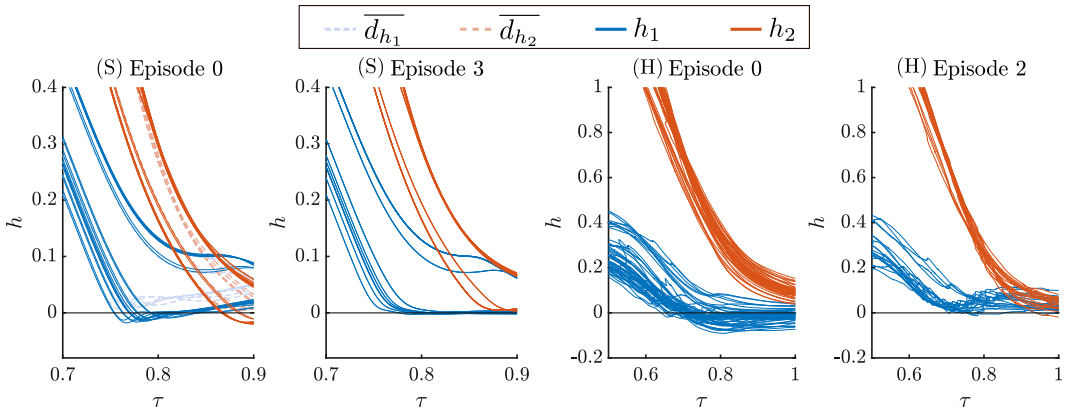


Figure 5.2. Simulation (S) and Hardware (H) data where model mismatch causes violations. **(Far-Left)**: Simulation where the barrier functions h_1 (solid blue) and h_2 (solid orange) are enforced via a 2.36. The $\overline{d_h}$-CBF-QP is also shown for $\overline{d_{h_1}}$ (dashed blue) and $\overline{d_{h_2}}$ (dashed orange), which results in more conservative behavior over many steps. **(Mid-Left)**: After three episodes of learning the SS-QP in simulation, the maximum barrier violation decreases from 2.0 to 0.3 cm. **(Mid-Right)**: Hardware where the barrier functions h_1 (blue) and h_2 (orange) enforced via a 2.36. **(Far-Right)**: After two episodes of learning on hardware, the maximum barrier violation decreases from 9.2 to 1.9 cm via the SS-QP.

foot strikes and an over-bending torso. A comparison of the barrier functions h_1 and h_2 over the steps with these controllers can be seen in Figure 5.2.

Hardware: The same nominal model for the robot was used in the hardware experiments as in simulation, with model uncertainty resulting from significant friction in the joints, imperfect mass and inertia measurements, and several other sources. Algorithm 5.2 was implemented on the AMBER-3M robot across a sequence of two episodes. The controllers ran on an off-board i7-6700HQ CPU @ 2.6GHz with 16 GB RAM, which computed desired torques and communicated them with the ELMO motor drivers on the 137 cm tall, 22 kg robot. The motor driver communication ran at 2kHz, and the SS-QP ran at 1kHz. The stepping stone configuration was specified to the controller with stones of 8 cm in width. As with simulation, the CBF-QP resulted in a maximum violation of the barriers of 9.2 cm due to model error. After running the PDL algorithm for two episodes, the maximum violation of the barriers was 1.9 cm, only 21% of the original violation, as depicted in Figure 5.2. Gait tiles for this improved traversal of the stepping stones are shown in Figure 5.3.

Conclusion

Here, instead of trying to add robustness to all possible disturbances as in Chapter 4, we presented an episodic learning approach for reducing the impact of model uncertainty on safety-critical control using CBFs. Our method is able to learn

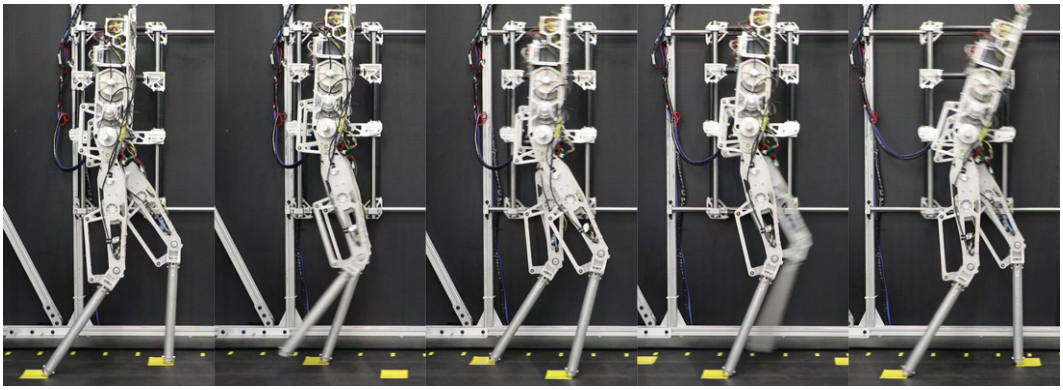


Figure 5.3. Gait tiles for Episode 2 of learning showing the AMBER-3M robot safely traversing a set of stepping stones. Notice the change in step width and added lean of the torso induced by the barrier functions.

a projected disturbance and incorporate the learned model information into an optimization-based controller, as demonstrated in both a high-fidelity simulation and on the AMBER-3M planar bipedal robot hardware platform. Although Algorithm 5.2 improved the safety and performance in our experiments, future work involves analyzing the theoretical convergence properties of this algorithm to ensure that trajectories remain close to previously-seen data points to reduce the effect of generalization error.

5.3 Learning Measurement Uncertainty

Next, I present our work on achieving safe behavior in unstructured environments by estimating measurement uncertainty a computer vision module using online, self-supervised learning.

Computer vision has become an important tool in robotics for sensing environments and identifying obstacles. Despite its utility and ubiquity in robotics, using vision sensors to achieve robust safety is difficult due to the complex environment-dependent error that they generate. For example, error patterns are highly correlated with the textures and appearance of a scene. Supervised methods can identify and model error as it affects safety [27], [39]; however, supervised approaches require ground-truth training data that may be difficult or impossible to obtain for the diversity of potential environments that a robot could experience during deployment.

To overcome these problems of environment-dependent measurement uncertainty, we develop an online, self-supervised learning method [141], inspired by successful demonstrations of online learning in robotics [142]–[144], that is capable of captur-

ing and accounting for the environment-dependent uncertainty during closed-loop deployment. By actively capturing the effects of the current environment, we can significantly reduce the conservatism by only requiring robustness with respect to the current environment, not with respect to all possible environments.

The contributions of this section are as follows:

- An online, self-supervised method for characterizing the uncertainty of disparity errors generated by stereo vision algorithms in novel environments.
- A robustified CBF-based control method which utilizes this error estimate for obstacle avoidance.
- Demonstrations of the proposed methods of error estimation and obstacle avoidance on a quadrupedal robot operating in real time.

The text for this section is adapted from:

> R. K. Cosner, I. D. J. Rodriguez, T. G. Molnar, W. Ubellacker, Y. Yue, A. D. Ames, and K. L. Bouman, "Self-supervised online learning for safety-critical control using stereo vision," *2022 International Conference on Robotics and Automation (ICRA)*, pp. 11 487–11 493, 2022. DOI: 10.1109/ICRA46639.2022.9812183,

A video for this section can be found at [145].

Stereo Vision Uncertainty Quantification

We begin by reviewing stereo vision, a popular tool for determining depth from images. These methods compute a *disparity*: the shift observed in an object's projection onto two camera planes. Using a geometric understanding of the camera setup, pixel-based disparity maps can be converted to depth maps. Errors in the final depth-map result from a combination of pixel-mismatch in disparity estimation and error in the camera parameters used to convert from disparity to depth. The errors in the intrinsic and extrinsic parameters of the camera are usually small and their effect on the resulting depth distribution is easy to compute. On the other hand, pixel-matching errors are much larger and are the result of a much more complicated stereo matching procedure whose effect on the resulting disparity is difficult to quantify and heavily environment-dependent.

For standard stereo vision we adopt the model from [146] for two cameras (left and right) and assume that they are perfectly rectified, vertically aligned and evenly spaced with known distance $b \in \mathbb{R}_{>0}$ between each camera. Pixel coordinates within an image are given by the tuple $p \triangleq (u, v) \in \mathbf{K}$, where $\mathbf{K} \triangleq \{0, \dots, W\} \times \{0, \dots, H\}$ for image width $W \in \mathbb{N}_{>0}$ and image height $H \in \mathbb{N}_{>0}$.

Stereo algorithms such as block matching, semi-global block matching, and efficient large-scale stereo [147] compute disparities by determining the discrete pixel distance between matching regions of two images. Since the disparity represents a shift between pixels of two images, the measured disparity $\widehat{d}$ must be a finite integer value. Assuming that the true disparity d is a finite integer implies that the error $e \triangleq \widehat{d} - d$ must also be a finite integer[1].

To learn the error in disparity, we introduce a three-camera multibaseline stereo system which produces multiple disparity maps that are related through simple functions; deviations from the ideal relationship indicate error in the estimated disparities. By analyzing the correlation of image contents with these errors, a function that estimates disparity error from appearance is learned and used to specify state error-bounds in real-time for use in a robustified CBF.

We introduce a three-element camera system, whose central camera is assumed to be perfectly rectified and vertically aligned with the other two cameras. This third camera is placed between the left and right cameras such that it has a baseline of $b/2$ with both. The three cameras produce a time-synchronized grayscale image triple (I_1, I_2, I_3) where $I_i \in \mathbb{N}^{W \times H}$ for $i \in 1, 2, 3$ and $1, 2, 3$ correspond with left, center, and right, respectively. The disparity between any image pair (I_i, I_j) for $i < j$ is obtained using the stereo-vision algorithm $\text{Disp} : \mathbb{N}^{W \times H} \times \mathbb{N}^{W \times H} \to \Gamma^{W \times H}$, so that $\widehat{d}_{i,j} = \text{Disp}(I_i, I_j)$. Here, $\Gamma \subset \mathbb{N}_{\geq 0}$ is the set of possible disparity values.

Given the measurement $\widehat{d}_{i,j}$, the error appears as $\widehat{d}_{i,j} = d_{i,j} + e_{i,j}$ with error distribution $e_{i,j} \sim \mathcal{P}(I_i, I_j)$ and true disparity $d_{i,j} \in \Gamma^{W \times H}$. We model this error as a discrete random variable with probability $\mathcal{P}(I_i, I_j)$ on $\Gamma^{W \times H}$. This model of disparity errors contrasts sharply with other common error models, such as punctual observation, uniform observation, and Gaussian observation [146], in that it accounts for the discrete nature of stereo-pixel matching algorithms. If ground-truth knowledge of $d_{i,j}$ is obtainable, then supervised learning methods can be implemented to directly estimate this error term. However, it is often the case that

[1]Prior work has been done to interpolate disparities for non-integer subpixel accuracy [148]; however, we restrict our attention to integer disparity values to highlight the error in pixel-matching.

ground-truth knowledge is unavailable; particularly when a domain transfer must occur during operation. Thus we seek a general method to estimate $e_{i,j}$ and $\mathcal{P}(I_i, I_j)$ as functions of the input image for any black-box disparity algorithm without the need for ground-truth data.

We leverage the known geometric relationships between the three cameras to learn a mapping between image appearance and disparity error distribution that can adapt during operation in new environments. Given a multibaseline stereo system, if one ignores occlusions, it is possible to completely reconstruct each disparity map from the other two maps. The relationship to reconstruct $\widehat{d}_{1,3}$ from $\widehat{d}_{1,2}$ and $\widehat{d}_{2,3}$ is shown in Algorithm 5.3; we denote this reconstruction[2] as $\overline{d}_{1,3} \triangleq \widehat{d}_{1,2} \oplus \widehat{d}_{2,3}$.

Algorithm 5.3: Disparity Reconstruction: $\overline{d}_{1,3} = \widehat{d}_{1,2} \oplus \widehat{d}_{2,3}$

$\overline{d}_{1,3} \leftarrow \mathbf{0}_{H \times W}$
for $v \in [1, ..., H]$ **do**
  for $u \in [1, ..., W]$ **do**
    $\widehat{u} \leftarrow n + \widehat{d}_{1,2}(u, v)$
    $\overline{d}_{1,3}(u, v) \leftarrow \widehat{d}_{1,2}(u, v) + \widehat{d}_{2,3}(u, \widehat{v})$

We use the reconstructed disparity $\overline{d}_{1,3}$ to learn the parameters θ of a function $\mathcal{P}^\theta$ that approximates the error distribution $\mathcal{P}$ (refer to Algorithm 5.4). Since this method does not require ground truth information, Algorithm 5.4 can be run online during operation to adapt $\mathcal{P}^\theta$ to new visual environments; however its lack of ground truth information means that this algorithm cannot learn constant bias in the error and assumes that the uncertainty distribution is zero-mean.

Recall that the disparity error, $e_{1,3}$ is discrete in nature. Therefore, the pixel-wise reconstruction error $\text{re}(p) \triangleq \|\widehat{d}^p_{1,3} - \overline{d}^p_{1,3}\|_1$ will also be discrete. For this reason, optimizing the loss L in Alg. 5.4 reduces to a pixel-wise classification problem similar to image segmentation. Thus, as is done in image segmentation, we use pixel-wise cross entropy as the loss function. This method is shown in Algorithm 5.4.

In algorithm 5.4, for each pixel p of the disparity $\widehat{d}_{1,3}$ the corresponding reconstruction error is computed. The loss function in algorithm 5.4 is then equivalent to

[2] I recognize that this is an abuse of notation with the Minkowski sum. In this thesis $\oplus$ indicates the Minkowski sum between sets and Algorithm 5.3 between disparity maps.

Algorithm 5.4: Self-Supervised Stereo Error Estimation Adaptation

$L \leftarrow 0$
while *robot is running* **do**
  $(I_1, I_2, I_3) \leftarrow$ Capture Current Frame
  $(\widehat{d}_{1,2}, \widehat{d}_{2,3}, \widehat{d}_{1,3}) \leftarrow (\text{Disp}(I_1, I_2), \text{Disp}(I_2, I_3), \text{Disp}(I_1, I_3))$
  $\overline{d}_{1,3} \leftarrow \widehat{d}_{1,2} \oplus \widehat{d}_{2,3}$
  $\text{re}(p) \leftarrow \left| \widehat{d}^p_{1,3} - \overline{d}^p_{1,3} \right|$
  $L \leftarrow -\frac{1}{H \times W} \sum_p \mathbb{E}_{\mathbb{1}(\text{re}(p))}[\log \mathcal{P}^\theta(I_i, I_k)]$
  $\theta_{t+1} \leftarrow \theta_t - \eta \frac{\partial L}{\partial \theta}$

the expected negative log likelihood of each pixel under the proposed model $\mathcal{P}^\theta$. An example visualization of lines $3-8$ can be found in Fig. 5.4. Although this algorithm focuses on the reconstructed disparity $\overline{d}_{1,3}$, it can be easily extended to similar reconstructions of $d_{1,2}$ and $d_{2,3}$.

This approach of estimating the uncertainty distribution on the measurement error is similar to that of [149], but does not require ground truth information and instead requires a zero-mean assumption on the uncertainty distribution.

Safe Stereo Vision-Based Control

Next, we propose a CBF-based control strategy that relies directly on the measurements of the stereo system and which incorporates the proposed self-supervised error estimates to enforce robust safety.

First, we construct CBFs for safe vision-based control. Let $\rho_p \in \mathbb{R}^3$ represent the true three-dimensional position of the portion of the scene which generated pixel p. Using this, we can define a CBF $h : \mathbb{R}^n \times \mathbb{R}^3 \to \mathbb{R}$ that relies on both the state $\mathbf{x}$ and three dimensional pixel position ρ_p. The pixel position is a geometric function of the true disparity, $\rho_p = \mathbf{T}(\mathbf{x}, r(p, d^p_{1,3}))$ where $r : \mathbb{N}^2 \times \mathbb{N}$ is the stereo reprojection function and $\mathbf{T} : \mathbb{R}^n \times \mathbb{R}^3 \to \mathbb{R}^3$ is the transformation mapping from the robot's state and relative pixel position to global pixel position.

In order to relate the output of the stereoscopic sensor with safety, we assume that the environment is static and that it is sufficient to enforce safety with respect to the currently measured pixel locations ρ_p for all $p \in \mathbf{K}$.

To combine the pixel-wise constraints, we apply Boolean composition to each CBF

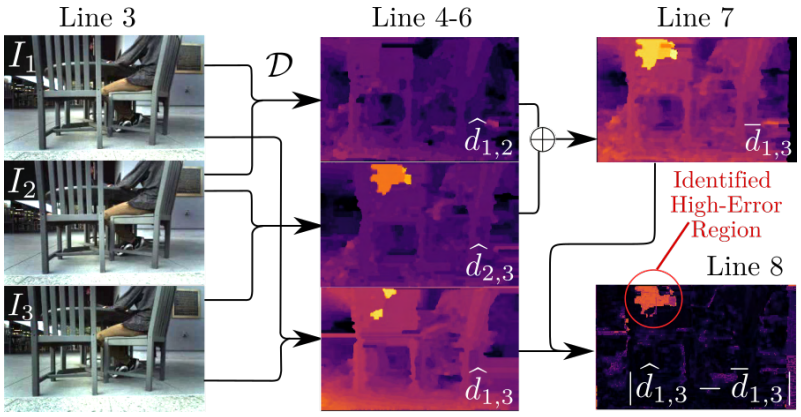


Figure 5.4. Lines 3-8 of algorithm 5.4 illustrated from left to right. Starting from three time-synchronized images three pairwise disparities are computed as shown in the middle column. Two of these disparities are used to build a reconstruction of the third disparity shown in the top right which can then be used to estimate the pixel-wise error of the stereo algorithm shown in the bottom right image. These steps of the algorithm correctly identify that the back of the closest chair is a high-error region without using ground truth information. This information is used to learn a correspondence between visual features and error distributions.

h to produce a single nonsmooth CBF h_{ns},

$$h_{\text{ns}}(\mathbf{x}) \triangleq \min_{p \in K} h(\mathbf{x}, \rho_p), \tag{5.17}$$

and simply enforce the CBF constraint associated with the pixels whose CBFs have the smallest value [92]. In particular, to achieve safety it is sufficient to enforce only the constraints whose indices appear in the locally-encapsulating index set:

$$\Lambda = \{p \in K : h(\mathbf{x}, \rho_p) \leq h_{\text{ns}}(\mathbf{x}) + \delta\}, \tag{5.18}$$

for some $\delta > 0$, as stated formally in [92, Prop. III.6]. For our application, this proposition indicates that enforcing the CBF condition only for the "least safe" pixel is sufficient to ensure safety.

Robustness to Vision-based Uncertainty

Error in the disparity propagates to the controller in the form of the measured 3D pixel position $\widehat{\rho}_p$. The measured value $\widehat{\rho}_p$ lies in a neighborhood $\mathcal{E}_p$ of the true value ρ_p, which is characterized by the error distribution $\mathcal{P}(I_i, I_j)$. We assume that the distribution $\mathcal{P}(I_i, I_j)$ is zero-mean and symmetric about the measured value and define the pixel-wise uncertainty set:

$$\mathcal{E}_p \triangleq \left\{ \rho \in \mathbb{R}^3 \;\middle|\; \begin{array}{c} \rho = \mathbf{T}(\mathbf{x}, r(p, \xi)), \quad \xi \in \Gamma \\ \mathcal{P}^\theta\{e_{1,3}(p) < |\xi - \widehat{d}(p)|; I_1, I_3\} \geq \sigma \end{array} \right\} \tag{5.19}$$

where $\sigma > 0$ is a parameter defining the desired uncertainty robustness.

To achieve safety, one must determine which pixels are safety-critical given $\mathcal{E}_p$ and then enforce robust safety with respect to those pixels. The safety-critical pixels can be determined by expanding the index set Λ using the uncertainty:

$$\Lambda \subseteq \left\{ p \in \mathbf{K} \;\middle|\; h(\mathbf{x}, \rho_p) \leq \max_{\rho_p \in \mathcal{E}_p} \min_{p \in \mathbf{K}} h(\mathbf{x}, \rho_p) + \delta \right\}. \tag{5.20}$$

This can further be expanded to an easily calculable index set $\widehat{\Lambda} \supseteq \Lambda$ by minimizing the left-hand-side of the inequality condition and using the max-min inequality [64]:

$$\widehat{\Lambda} = \left\{ p \in K \;\middle|\; \min_{\rho_p \in \mathcal{E}_p} h(\mathbf{x}, \rho_p) \leq \min_{p \in \mathbf{K}} \max_{\rho_p \in \mathcal{E}_p} h(\mathbf{x}, \rho_p) + \delta \right\}. \tag{5.21}$$

This expanded index set $\widehat{\Lambda}$ accounts for uncertainty and indicates which pixels are safety-critical and which constraints must be enforced to achieve safety given the pixel-wise uncertainty sets $\mathcal{E}_p$.

Measurement-Robust Control Barrier Functions (MRCBFs) as outlined in [51] and Section 4.2 of this thesis are a general method for accounting for state uncertainty in CBFs. We can use this method for each pixel $p \in \widehat{\Lambda}$ to ensure that the safety constraint is satisfied despite the uncertainty. The resulting constraint is:

$$L_{\mathbf{f}} h(\mathbf{x}, \widehat{\rho}_p) + L_{\mathbf{g}} h(\mathbf{x}, \widehat{\rho}_p)\mathbf{u} - \left(\mathfrak{L}_{L_f h} + \mathfrak{L}_{\gamma \circ h_{\text{ns}}} + \mathfrak{L}_{L_g h} \|\mathbf{u}\|_2 \right) \epsilon_p \geq -\gamma(h_{\text{ns}}(\mathbf{x})), \tag{5.22}$$

for all $p \in \widehat{\Lambda}$ where $\mathfrak{L}$ is the Lipschitz constant of the subscripted function and

$$\epsilon_p \geq \max_{\rho_p \in \mathcal{E}_p} \|\rho_p - \widehat{\rho}_p\|_2 \tag{5.23}$$

is a quantile bound on the pixel position error. Since $\Lambda \subseteq \widehat{\Lambda}$ and the MRCBF condition implies the CBF condition (2.33), satisfying (5.22) also satisfies the CBF condition for each pixel providing safety of the system if $\sigma = 1$ and $\mathcal{P}^\theta = \mathcal{P}$.

Application to Quadruped Obstacle Avoidance

We evaluate our approach on a Unitree A1 quadruped. With these experiments we aim to demonstrate: 1) our method is capable of keeping the system safe in a simple do-not-collide task, and 2) our method can adapt online to measurement uncertainty in different environments without ground-truth data.

For the hardware experiments we designed a custom camera array with three equally spaced inexpensive CMOS, global shutter, time-synchronized Arducam cameras.

An Nvidia Jetson Nano is used to capture, downsize, and greyscale the stereo images. The images are then sent to an external computer that receives the images and outputs the filtered control input at a frequency of at least 10 Hz. The robot receives virtual inputs of velocity and angle rate, $\mathbf{u} = \begin{bmatrix} v & \omega \end{bmatrix}^\intercal$ and uses a 1 kHz Inverse Dynamics Quadratic Program (ID-QP) walking controller designed using the concepts in [95] to track these virtual inputs. Stereo pixel-matching calculations were performed using Efficient LArge-scale Stereo (ELAS) [147].

The architecture of the model used to estimate $\mathcal{P}^\theta$ is a modified version of the Hierarchical Multi-Scale Attention for Semantic Segmentation introduced in [150]; this model is relatively lightweight, consisting of only 196 thousand parameters. The robustness threshold used was $\sigma = 0.99$ and the online learning rate was 0.001. We pretrain the model until convergence on a dataset of 6000 stereo image triples collected by manually moving the camera array through a variety of environments.

In order to control the system we consider a reduced-order model of the system dynamics given by the standard unicycle model (3.16). A formal analysis of CBFs which utilize reduced-order velocity input models is described in Section 3.3 and [48].

For this system we consider the pixel-wise CBFs,

$$h(\mathbf{x}, \rho_p) = \frac{1}{2}\left(\left\| \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} \rho_{p,x} \\ \rho_{p,y} \end{bmatrix} \right\|_2^2 - c^2 \right) \tag{5.24}$$

where $\rho_{p,x}$ and $\rho_{p,y}$ indicate the global real-world x and y positions of pixel p. This function characterizes safety as remaining a planar distance $c > 0$ from ρ_p. This can be thought of as buffering surfaces in the environment by a radius c.

To illustrate the efficacy of our method we use two controllers in our experiments. A standard, unrobustified controller:

$$\begin{aligned} \mathbf{k}_{\text{cbf}}(\mathbf{x}) = \underset{\mathbf{u}\in\mathbb{R}^2}{\operatorname{argmin}} \quad & \frac{1}{2}\|\mathbf{k}_{\text{des}}(\mathbf{x}) - \mathbf{u}\|_2^2 \\ \text{s.t.} \quad & \underbrace{-\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\intercal r(p, \widehat{d_p}) v}_{\dot{h}} \geq -\gamma(\min_{p\in K} h(\mathbf{x}, \widehat{\rho_p})), \qquad \forall p \in \Lambda \end{aligned} \tag{5.25}$$

and a robustified controller:

$$\mathbf{k}^*_{\text{cbf}}(\mathbf{x}) = \underset{\mathbf{u}\in\mathbb{R}^2}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{k}_{\text{des}}(\mathbf{x}) - \mathbf{u}\|_2^2 \tag{5.26}$$
$$\text{s.t.} \ -v \geq \frac{-\gamma(\min_{p\in K} h(\mathbf{x}, \rho^*_p))}{\begin{bmatrix}1 & 0 & 0\end{bmatrix}^\intercal r(p, d^*_p)}, \quad \forall p \in \widehat{\Lambda},$$

where $\mathbf{k}_{\text{des}} : \mathbb{R}^m \to \mathbb{R}^n$ is a desired controller, d^* is the maximum disparity for any $\rho_p \in \mathcal{E}_p$, and ρ^*_p is pixel location associated with d^*_p.

Controller (5.26) is obtained by first replacing the index set Λ with the $\widehat{\Lambda}$. Next we note that $\begin{bmatrix}1 & 0 & 0\end{bmatrix}^\intercal r(p, \widehat{d}_p)$ is strictly positive. After dividing by this quantity, the constraint in (5.25) is robustified to account for the worst-case error as is done with MRCBFs. Experimentally, this controller was implemented with $\delta = 0$ and a maximum of 4000 constraints.

The system was run in 4 different environments (see Fig. 5.5). The CBF (5.24) was used with a safe radius of $c = 0.33$ m. The intended obstacle in the 4 different environments were (**A**) a tree, (**B**) a backpack, (**C**) a chair, (**D**) and a glass window. A desired constant forward velocity $v = 0.2$ m/s was used in each experiment and the robot was started approximately 1.3 m away from the obstacle. Since ground-truth measurements were unavailable, we use a yellow line on the ground to indicate the true location of the barrier.

For each environment three different tests were performed. First, controller (5.25) was used. Since this did not consider measurement uncertainty it failed to achieve safety in every environment; in all experiments the stereo vision overestimated the distance to objects at some point during the run and the quadruped ran directly into the obstacles. Second, the controller (5.26) was used with an error estimate computed through a fixed, pretrained error distribution estimate $\mathcal{P}^\theta$; this succeeded in providing safety, but was found to be overly conservative and did not allow the quadruped to approach the obstacle as desired. Third, the controller (5.26) was used with a $\mathcal{P}^\theta$ that adapted to the environment according to Algorithm 5.4. In this case, safety of the system was generally maintained and over time the system was able to approach the boundary of the safe set. Even when small safety violations occurred, the system eventually corrected and came to rest at a safe steady-state. These results can be seen in Figure 5.5. A video of the experiments can be found at [145].

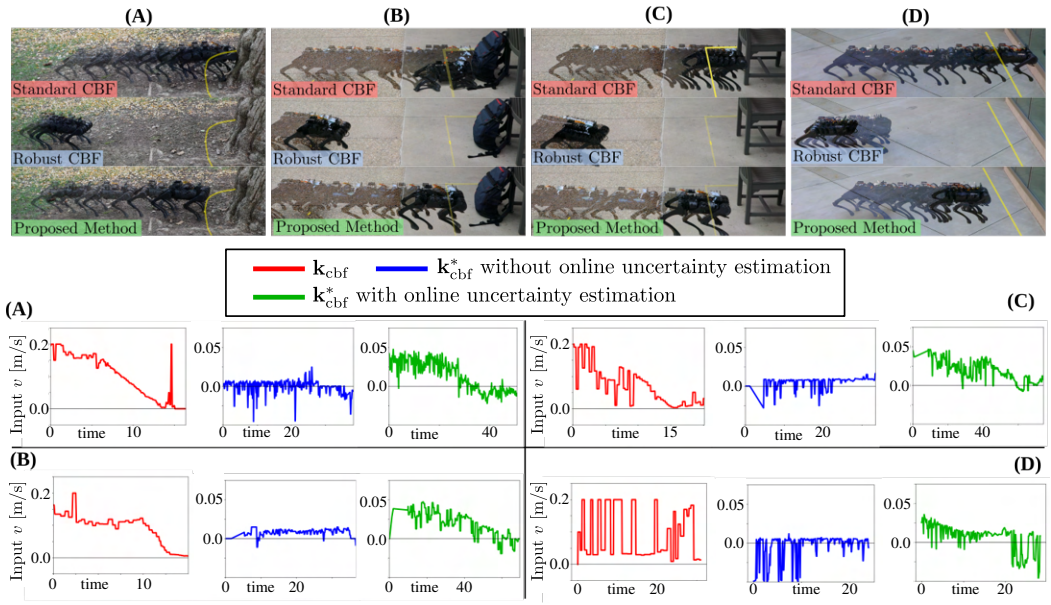


Figure 5.5. Demonstration of our method in a variety of environments. From left to right the goal is to maintain a safe distance from (**A**) a tree, (**B**) a backpack, (**C**) a chair, (**D**) and a glass window. The distance to the barrier is measured and marked on the floor with a yellow tape for visualization purposes – we emphasize this tape is not used for depth estimation. Notice that the barrier is assumed to be a sphere around an obstacle but in the case on the glass, this sphere degenerates into a plane. The quadrupedal robot is given a desired control input of 0.2 m/s. In all cases, a naive barrier implementation that simply takes the noisy measurements from a stereo vision system fails to keep the system safe. The robustified controller (5.26) with a pretrained model consistently shows overly conservative behavior. Finally, with online learning, the robot converges to the barrier without exhibiting conservative behavior, except for the glass environment where the robot is overly conservative and walks away from the barrier due to the perceived uncertainty. The (**A-D**) corresponding plots below show the control input filtered by the barrier in each of the three robustification cases.

Conclusion

In this section, we presented a framework for achieving safety of a stereo vision-based system using self-supervised online uncertainty estimation and robustified CBFs. Refining the uncertainty estimate model online was shown to achieve significantly better performance than when a worst-case bound was used that was required to hold for all environments. We validated our approach across several environments and successfully achieved robust safety with minimal violations and conservatism. While the method in this chapter had limited applications to stereo vision systems, I believe that this method of determining environment-dependent error bounds is a useful way forward to achieve behavior that achieves performances goals while being sufficiently robust when the environment demands it.

5.4 Learning Preferred Robustness

Despite the utility of the uncertainty-learning methods of the previous two sections, these approaches assume the existence of reasonably accurate models. However, in many robotics applications, "preferred" behavior and "preferred" levels of robustness are difficult to define mathematically and are intangible concepts for which we do not have *a priori* models. In such cases, rather than reducing model inaccuracies, it may be more appropriate to learn behavior improvements directly from human preferences.

With the goal of identifying user preferences, preference-based learning (PBL) has shown to be a powerful tool for converting subjective user preferences into quantitative adjustments to design parameters. When used in an online, episodic fashion, PBL is capable of interactively inferring a user's latent utility function using only subjective feedback such as pairwise preferences and ordinal labels [151]–[153]. For applications with actions that may be classified as safe or unsafe, *safety-critical* PBL algorithms have been demonstrated to prevent unsafe actions from being sampled [154], [155]. However, these safety-critical algorithms require worst-case approximations which may cause performant and safe actions to be characterized as catastrophically unsafe. Thus, we seek to formulate a *safety-aware* approach to PBL that generally avoids unsafe actions without being overly conservative.

In this section we, we use PBL to determine the desirable parameter values for robust safety-critical control using the TR-OP controller (4.39), thus creating an algorithm that is able to identify the user-preferred behavior of the closed-loop system by iteratively adjusting its robustness levels.

The contributions of this section are as follows:

- The Safety-Aware LineCoSpar (SA-LineCoSpar), a modified version of LineCoSpar [156] capable of high-dimensional preference-based Bayesian optimization while also accounting for safety in the learning process.
- Demonstrations of Safety-Aware LineCoSpar in conjunction with the TR-OP controller (4.39) (the contribution of Section 4.3) to achieve safety-critical control of a quadrupedal robot in simulation and hardware in laboratory and outdoor settings.
- The first use of PBL to tune a CBF controller.

The text for this section is adapted from:

R. K. Cosner, M. Tucker, A. J. Taylor, K. Li, T. G. Molnar, W. Ubellacker, A. Alan, G. Orosz, Y. Yue, and A. D. Ames, "Safety-aware preference-based learning for safety-critical control," *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, Proceedings of Machine Learning Research, vol. 168, pp. 1020–1033, 2022. [Online]. Available: `https://proceedings.mlr.press/v168/cosner22a.html`,

A video for this section can be found at [126].

Safety-Aware LineCoSpar

Preference-Based Learning (PBL) provides an approach for searching complex parameter spaces via subjective feedback, without an explicitly defined reward function. This is particularly relevant for safety-critical systems and tuning the robustness parameters of the TR-OP controller (4.39), as quantifying the user-preferred trade-off between robustness and performance is difficult. Moreover, poorly defined reward functions often result in "reward hacking" [157], in which undesirable actions achieve high rewards. Here, we propose Safety-Aware LineCoSpar (SA-LineCoSpar), outlined in Algorithm 5.5. This is a modification of the LineCoSpar algorithm [156], which iteratively selects actions to query the user for subjective feedback and updates its belief of the user's underlying utility function via Bayesian inference.

Problem Setup: Let $\mathbf{a}$ denote an action, such as a collection of l parameters used in a feedback controller, that takes values in a finite search space $A \subset \mathbb{R}^l$. We assume that each action $\mathbf{a} \in A$ has an unknown utility to the user, defined by a function $r : A \to \mathbb{R}$. These utilities are given by $\mathbf{r}_A = [r(\mathbf{a}_1), \ldots, r(\mathbf{a}_{|A|})]^\top \in \mathbb{R}^{|A|}$. In each iteration, $s \in \mathbb{N}$ actions are sampled from A and executed. Then, the user is queried for two forms of feedback: pairwise preferences and ordinal labels, describing *performance* and *safety*, respectively. This feedback is collected into dataset $\mathfrak{D}$.

Modeling the Utility Function: Since collecting an exhaustive dataset to estimate the unknown utility $\mathbf{r}_A$ is expensive for non-trivial action spaces, we use Bayesian optimization (BO), a sampling efficient paradigm for identifying the optimizer. In BO, $\mathbf{r}_A$ is modeled as a Gaussian process with prior $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}^{\text{pr}})$, where each element of the covariance matrix $\mathbf{\Sigma}^{\text{pr}} \in \mathbb{S}_{\succ 0}^{|A|\times|A|}$ is computed as $\mathbf{\Sigma}^{\text{pr}}_{ij} = k(\mathbf{a}_i, \mathbf{a}_j)$ with a kernel function $k : A \times A \to \mathbb{R}$ and $\mathbf{a}_i \in A$ denoting the i^{th} action in A. We select k to be

the squared exponential kernel, yielding a prior given by the multivariate Gaussian:

$$\mathbb{P}(\mathbf{r}_A) = \frac{1}{(2\pi)^{|A|/2}\left|\boldsymbol{\Sigma}^{\text{pr}}\right|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{r}_A^\top (\boldsymbol{\Sigma}^{\text{pr}})^{-1}\mathbf{r}_A\right). \tag{5.27}$$

Given a dataset $\mathfrak{D}$, the posterior is proportional to the likelihood and the prior by Bayes' theorem, i.e., $\mathbb{P}(\mathbf{r}_A \mid \mathfrak{D}) \propto \mathbb{P}(\mathfrak{D} \mid \mathbf{r}_A)\mathbb{P}(\mathbf{r}_A)$. We denote the maximum a posteriori (MAP) estimate of the posterior by $\hat{\mathbf{r}}_A \in \mathbb{R}^{|A|}$, which is defined as $\hat{\mathbf{r}}_A \triangleq \text{argmax}_{\mathbf{r}_A \in \mathbb{R}^{|A|}} \mathbb{P}(\mathbf{r}_A \mid \mathfrak{D})$, noting that $\hat{\mathbf{r}}_A$ is equivalent to the minimizer of $\mathcal{S}(\mathbf{r}_A) = -\ln(\mathbb{P}(\mathfrak{D} \mid \mathbf{r}_A)) + \frac{1}{2}\mathbf{r}_A^T \left(\Sigma^{\text{pr}}\right)^{-1}\mathbf{r}_A$. As is common in BO, we model the posterior as a multivariate Gaussian centered at $\hat{\mathbf{r}}_A$ with the covariance $\boldsymbol{\Sigma}_A \in \mathbb{S}_{\succ 0}^{|A|\times|A|}$ defined as $\boldsymbol{\Sigma}_A = (\frac{\partial^2 \mathcal{S}}{\partial \mathbf{r}_A^2}(\hat{\mathbf{r}}_A))^{-1}$ [158][3]. Additionally, we can improve tractability of calculating $\hat{\mathbf{r}}_A$ by reducing the action space A to a subset $S \subset A$, forming a partial characterization of the utilities denoted by $\mathbb{P}(\mathbf{r}_S \mid \mathfrak{D}) \approx \mathcal{N}(\hat{\mathbf{r}}_S, \boldsymbol{\Sigma}_S)$, with $\mathbf{r}_S, \hat{\mathbf{r}}_S \in \mathbb{R}^{|S|}$.

Preference Likelihood Function: A pairwise preference is defined as a relation between two actions $\mathbf{a}_1, \mathbf{a}_2 \in A$, where $\mathbf{a}_1 \succ \mathbf{a}_2$ if action $\mathbf{a}_1$ is preferred to $\mathbf{a}_2$. Since user preferences are expected to be corrupted by noise, we model individual pairwise preferences via a likelihood function:

$$\mathbb{P}(\mathbf{a}_1 \succ \mathbf{a}_2 | r(\mathbf{a}_1), r(\mathbf{a}_2)) = g_p\left(\frac{r(\mathbf{a}_1) - r(\mathbf{a}_2)}{c_p}\right), \tag{5.28}$$

where $g_p : \mathbb{R} \to [0, 1]$ is any monotonically-increasing link function, and $c_p \in \mathbb{R}_{>0}$ accounts for preference noise. We select g_p to be the sigmoid function, i.e., $g_p(x) = 1/(1 + e^{-x})$. Assuming conditional independence, the likelihood function for a collection of $K \in \mathbb{N}$ preferences, $\mathfrak{D}_p$, can be modeled as the product of each individual preference likelihood:

$$\mathbb{P}(\mathfrak{D}_p | r(\mathbf{a}_{11}), r(\mathbf{a}_{12}), \cdots, r(\mathbf{a}_{K2})) = \prod_{k=1}^{K} \mathbb{P}(\mathbf{a}_{k1} \succ \mathbf{a}_{k2} | r(\mathbf{a}_{k1}), r(\mathbf{a}_{k2})), \tag{5.29}$$

where $\mathbf{a}_{k1}, \mathbf{a}_{k2} \in A$ are the preferred and non-preferred actions, respectively, in the k^{th} preference.

Ordinal Likelihood Function: We partition the action space into "unsafe" and "safe" actions by leveraging the ordinal nature of these definitions (i.e., unsafe actions are always considered worse than safe actions). A user provides this feedback as

[3] This is known as the Laplace approximation of the distribution $\mathbb{P}(\mathbf{r}_A \mid \mathfrak{D})$, i.e., $\mathbb{P}(\mathbf{r}_A \mid \mathfrak{D}) \approx \mathcal{N}(\hat{\mathbf{r}}_A, \boldsymbol{\Sigma}_A)$.

an ordinal label, which assigns an action to a discrete ordered category such as "bad" and "good" [159]. While ordinal labels can be generalized to any number of ordinal categories (c.f. [160]), we utilize just two categories to represent "unsafe" and "safe." In this case, the action space is decomposed into two disjoint sets, $A = O_1 \cup O_2$, with $\mathbf{a} \in O_1$ if $r(\mathbf{a}) < \beta$ and $\mathbf{a} \in O_2$ if $r(\mathbf{a}) \geq \beta$, with the ordinal threshold $\beta \in \mathbb{R}$. As with preferences, we assume that ordinal label feedback is corrupted by noise and is modeled as:

$$\mathbb{P}(\mathbf{a} \in O_1 \mid r(\mathbf{a})) = g_o\left(\frac{\beta - r(\mathbf{a})}{c_o}\right), \quad \mathbb{P}(\mathbf{a} \in O_2 \mid r(\mathbf{a})) = 1 - g_o\left(\frac{\beta - r(\mathbf{a})}{c_o}\right), \tag{5.30}$$

where $g_o : \mathbb{R} \to [0, 1]$ is any monotonically-increasing link function and c_o quantifies the noise in the ordinal label feedback. Again, we select g_o to be the sigmoid function $g_o(x) = 1/(1 + e^{-x})$. Assuming conditional independence of ordinal label queries, the likelihood function for a collection of $M \in \mathbb{N}$ ordinal labels, $\mathfrak{D}_o$, can be modeled as the product of each individual ordinal likelihood:

$$\mathbb{P}(\mathfrak{D}_o \mid r(\mathbf{a}_1), \cdots, r(\mathbf{a}_k)) = \prod_{k=1}^{M} \mathbb{P}\left(\mathbf{a}_k \in O_{o(k)} \mid r(\mathbf{a}_k)\right), \tag{5.31}$$

where $\mathbf{a}_k \in A$ refers to the action corresponding to the k^{th} ordinal label, $o(k) \in \{1, 2\}$. For our simulation and experiments, the hyperparameters c_p, c_o, β are determined in advance. Lastly, assuming conditional independence of the feedback mechanisms, the combined likelihood function is calculated as the product of the individual likelihoods, $\mathbb{P}(\mathfrak{D} \mid r) = \mathbb{P}(\mathfrak{D}_p \mid r)\mathbb{P}(\mathfrak{D}_o \mid r)$.

Sampling New Actions: In the first iteration ($i = 1$), $s \in \mathbb{N}$ actions are sampled randomly from A, recorded as the set of visited actions $V_1 = \{\mathbf{a}_1^{(1)}, \ldots, \mathbf{a}_1^{(s)}\}$, executed on the system, and the preferences and ordinal labels are collected into a dataset $\mathfrak{D}_1$. In each subsequent iteration ($i > 1$), s new actions are sampled using Thompson sampling, which is shown to have desirable regret minimization properties [161]. Ideally, Thompson sampling draws s samples from the posterior $\mathbb{P}(\mathbf{r}_A \mid \mathfrak{D}_{i-1})$, i.e., $\mathbf{r}^{(j)} \sim \mathcal{P}(\mathbf{r}_A \mid \mathfrak{D}_{i-1})$ for $j \in \{1, \ldots, s\}$, and the action $\mathbf{a}_i^{(j)} \in A$ maximizing each $\mathbf{r}^{(j)}$ is selected to execute on the system. These sampled actions $\{\mathbf{a}_i^{(1)}, \ldots, \mathbf{a}_i^{(s)}\}$ are concatenated with V_{i-1} to produce V_i, executed on the system, and the resulting preferences and ordinal labels are concatenated with $\mathfrak{D}_{i-1}$ to produce $\mathfrak{D}_i$. However, since it is intractable to approximate $\mathcal{P}(\mathbf{r}_A \mid \mathfrak{D})$ for high-dimensional action spaces, we utilize a dimensionality-reduction technique introduced in [156] that instead

updates the posterior over a subset $S_i \subset A$. Motivated by [162], we construct the subset as $S_i = L_i \cup V_{i-1}$, where $L_i \subset A$ is the collection of $n_e \in \mathbb{N}$ actions in A closest to a randomly drawn line $\ell_i \subset \mathbb{R}^l$. This line is drawn to intersect with the believed best action, computed as $\hat{\mathbf{a}}^*_{i-1} = \text{argmax}_{\mathbf{a} \in V_{i-1}} \hat{\mathbf{r}}_{V_{i-1}}(\mathbf{a})$ where $\hat{\mathbf{r}}_{V_{i-1}}$ is the MAP estimate of the posterior $\mathcal{P}(\mathbf{r}_{V_{i-1}} \mid \mathfrak{D}_i)$. See [156] for more details.

Safety-Aware Sampling: It is important to avoid unsafe actions during sequential decision making in certain applications, such as learning robotic controllers on hardware, where low-reward actions might lead to physical damage of the platform. Prior safe exploration algorithms [155], [163] considered the setting where actions below a prespecified safety threshold are catastrophic and must be avoided at all cost. In our work, we rely on the natural conservatism of robust control methods like the TR-OP controller (4.39) and adopt a more optimistic learning approach called *safety-aware*. In this case, actions labeled by a human as "unsafe" are not catastrophic but undesirable. Thus, the algorithm *avoids* these actions; whereas the safe exploration algorithms guarantee that no such actions are sampled which can be sometimes exceedingly conservative in settings like ours.

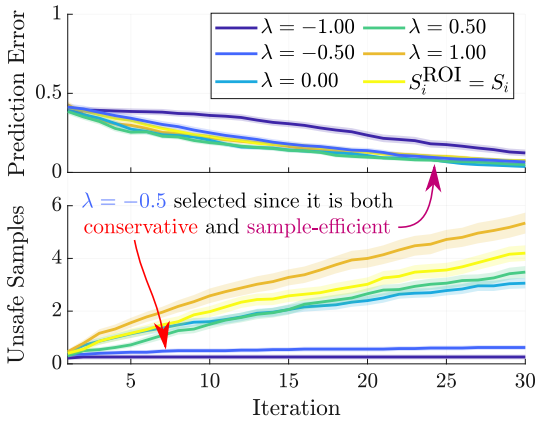


Figure 5.6. A comparison of SA-LineCoSpar and standard LineCoSpar on a synthetic utility function (drawn from the Gaussian prior) averaged over 50 runs with standard error shown by the shaded region. The safety-aware criteria reduces the number of sampled unsafe actions with a minimal effect on the prediction error, defined as $|\hat{\mathbf{a}}^*_i - \mathbf{a}^*|$ with $\hat{\mathbf{a}}^*_i \triangleq \text{argmax}_{\mathbf{a}} \hat{\mathbf{r}}_{S_i}$ and $\mathbf{a}^* \triangleq \text{argmax}_{\mathbf{a}} r(\mathbf{a})$.

To achieve this safety-awareness, we leverage the approach introduced in [160], which uses ordinal labels to identify a *region of interest* (ROI) in A. In this work, the ROI is defined to be the actions labeled as "safe." In each iteration i we estimate an ROI within the set S_i as:

$$S_i^{\text{ROI}} = \{\mathbf{a} \in S_i \mid \hat{\mathbf{r}}_{S_i}(\mathbf{a}) + \lambda \boldsymbol{\sigma}_{S_i}(\mathbf{a}) > \beta\}, \tag{5.32}$$

where $\hat{\mathbf{r}}_{S_i}(\mathbf{a})$ and $\boldsymbol{\sigma}_{S_i}(\mathbf{a})$ are the posterior mean and standard deviation, respectively, evaluated at the action $\mathbf{a} \in S_i$. The variable $\lambda \in \mathbb{R}$ determines how conservative the algorithm would be in estimating the safety region, as illustrated in Figure 5.6. We

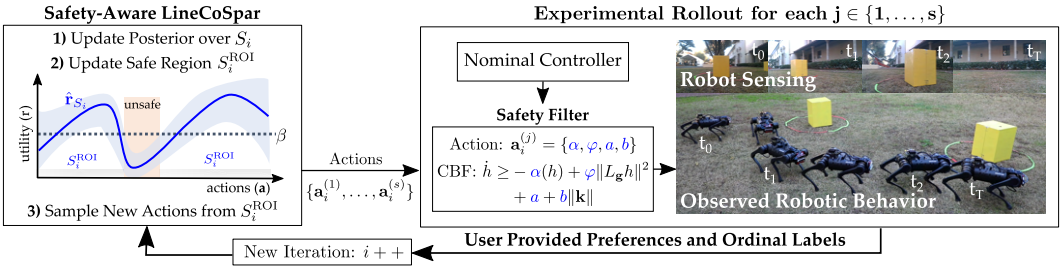


Figure 5.7. An overview of the Safety-Aware Preference-Based Learning design paradigm. Safety-Aware LineCoSpar is used to generate actions which are rolled out in experiments as parameters of the CBF-based safety filter to obtain user preferences and safety ordinal labels which are then used to update the user's estimated utility and generate new actions.

see that lower (negative) values of λ result in fewer unsafe actions being sampled, with only a slight effect on sample-efficiency. The restriction to S_i^{ROI} is added to LineCoSpar by only considering actions in S_i^{ROI} during Thompson sampling. We refer to this as Safety-Aware LineCoSpar (SA-LineCoSpar), with the full algorithm outlined in Algorithm 5.5.

Algorithm 5.5: Safety-Aware LineCoSpar

input: s uniform random actions ($V_1 \subset A$), corresponding feedback ($\mathfrak{D}_1$),
for $i = 2, \ldots, N$ **do**
  Update posterior over V_{i-1}
  $\hat{\mathbf{a}}_{i-1}^* \leftarrow \text{argmax}_{\mathbf{a} \in V_{i-1}} \hat{\mathbf{r}}_{V_{i-1}}(\mathbf{a})$
  $L_i \leftarrow$ New linear subspace intersecting $\hat{\mathbf{a}}_{i-1}^*$
  Construct subspace $S_i = L_i \cup V_{i-1}$
  Update the model posterior over S_i
  Determine region of interest S_i^{ROI}
  for $j = 1, \ldots, s$ **do**
    $r^{(j)} \sim \mathcal{N}(\hat{\mathbf{r}}_{S_i}, \mathbf{\Sigma}_{S_i})$
    $\mathbf{a}_i^{(j)} \leftarrow \text{argmax}_{\mathbf{a} \in S_i^{\text{ROI}}} r^{(j)}$
  Deploy $\{\mathbf{a}_i^{(1)}, \ldots \mathbf{a}_i^{(s)}\}$ on system
  $V_i \leftarrow V_{i-1} \cup \{\mathbf{a}_i^{(1)}, \ldots \mathbf{a}_i^{(s)}\}$
  $\mathfrak{D}_i \leftarrow \mathfrak{D}_{i-1} \cup$ new preferences $\cup$ new ordinal labels

Integrating Learning to Tune the Control Barrier Function: The parameter selection process of the TR-OP controller (4.39) is particularly important, since the parameters $\underline{a}$ and $\underline{b}$ guaranteed to exist by Theorem 4.7 are worst-case approximations of the uncertainty's effect on safety. Such approximations often lead to undesired conservatism and may render the system incapable of performing its goal (as seen in Figure 5.8). Thus, as illustrated in Figure 5.7, we propose utilizing SA-

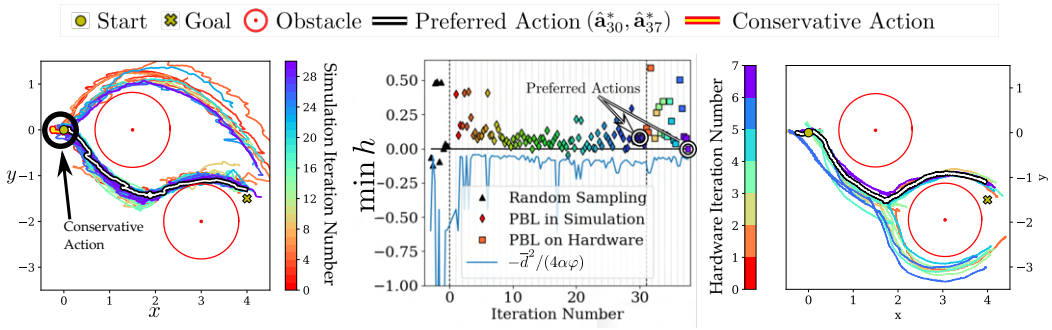


Figure 5.8. Preference-based learning Experiment plots. **(Left)** Actions sampled during simulation in 30 iterations with 3 new actions in each iteration. The preferred action, $\hat{\mathbf{a}}_{30} = (3, 0.6, 0.5, 0.015)$, is shown in black and white. A conservative action, $\mathbf{a} = (2, 0.5, 0.0651, 0.485)$, is indicated by the black circle, where a and b were determined by estimating the Lipschitz coefficients present in the proof of Theorem 4.7. The conservative action fails to progress whereas SA-LineCoSpar provides an action which successfully navigates between obstacles. **(Center)** The minimum value of h that occurred in each iteration. Triangles, diamonds, and squares represent actions that are sampled randomly, by PBL in simulation and on hardware in an indoor setting, respectively. Colors correlate to iteration number. The lower bound $-\frac{\overline{d}^2}{4\alpha\varphi}$ for the expanded set $\mathcal{C}_\delta = \{\mathbf{x} \in \mathbb{R}^{n_x} \mid h(\mathbf{x}) \geq -\frac{\overline{d}^2}{4\alpha\varphi}\}$ with $\overline{d} = 1$ is plotted. The preferred actions for simulation and hardware experiments are circled. **(Right)** Seven additional iterations of 3 actions executed indoors. The preferred action, $\hat{\mathbf{a}}^*_{37} = (4, 0.6, 0.4, 0)$, successfully traverses between the obstacles.

LineCoSpar to identify user-preferred parameters of the TR-OP controller. This relaxes the worst-case over-approximation to experimentally realize performant and safe behavior. This design paradigm relies on the *tunable* construction of the TR-OP controller (4.39), allowing us to define the actions for SA-LineCoSpar to the parameters $\mathbf{a} = (\alpha, \varphi, a, b)$ of the TR-OP controller (4.39). We note that the construction of the tunable parameters in Theorem 4.7 assures that unsafe actions are not necessarily catastrophic, as any $\alpha, \varphi, a, b > 0$ endows the system with a non-zero degree of robustness to disturbances and measurement error. This assurance allows us to utilize a safety-aware approach where unsafe actions are considered undesirable as opposed to more conservative safety-critical approach to learning where unsafe actions are considered catastrophic.

If we wish to enforce multiple safety constraints, such as in obstacle avoidance with several obstacles, $\widehat{\boldsymbol{\rho}}_i$ can be used to indicate the measured parameters of the i^{th} obstacle, with $N_o \in \mathbb{N}$ being the total number of obstacles. Enforcing this constraint for $N_o > 1$ can be viewed as Boolean composition of safe sets [20].

Experimental Results

We applied the proposed design paradigm to a perception-based obstacle avoidance task with a Unitree A1 quadrupedal robot (Figure 5.7) in simulation and on hardware

hyperparameter	value
λ	-0.5
β	0

name	min.	max.	Δ
α	0.5	5	0.5
φ	0	1	0.1
a	0	1	0.1
b	0	0.05	0.005

Table 5.1. The safety-aware hyperparameters, and action space bounds (min. and max.) with discretizations Δ. (Left) Safety-aware region of interest parameters. (Right) TR-OP tunable parameters and discretizations that define the action space.

for both indoor and outdoor environments (see video: [126]). The action space A and hyperparameters of PBL are defined in Table 5.1. We used the unicycle model with disturbance as our simplified model with the desired, nominal controller $\mathbf{k}_{\text{des}}$:

$$\underbrace{\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\psi}\end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix}0\\ 0\\ 0\end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix}\cos\psi & 0\\ \sin\psi & 0\\ 0 & 1\end{bmatrix}}_{\mathbf{g}(\mathbf{x})} \left(\underbrace{\begin{bmatrix}v\\ \omega\end{bmatrix}}_{\mathbf{v}} + \mathbf{d}(t) \right), \tag{5.33}$$

$$\mathbf{k}_{\text{des}}(\mathbf{x}) = \begin{bmatrix} K_v d_{\text{g}} + C \\ -K_\omega(\sin\psi - (y_{\text{g}} - y)/d_g) \end{bmatrix}, \tag{5.34}$$

where (x, y) is the planar position of the robot, ψ is the yaw angle, $(x_{\text{g}}, y_{\text{g}})$ is the goal position of the robot, $d_{\text{g}} = \|(x_{\text{g}} - x, y_{\text{g}} - y)\|$ is the distance to the goal, and K_v, K_ω, and C are positive constants. Obstacle avoidance is encoded via the 0-superlevel set of the function:

$$h(\mathbf{x}) = d_{\text{obs},i} - r_{\text{obs}} - \zeta\cos(\psi - \theta_i), \tag{5.35}$$

where the state $\mathbf{x}$ is extended by the i^{th} obstacle location $\boldsymbol{\rho}_i = [x_{\text{obs},i}, y_{\text{obs},i}]$ which is assumed to contain all of the measurement uncertainty, $d_{\text{obs},i} = \|(x_{\text{obs},i} - x, y_{\text{obs},i} - y)\|$ and $\theta_i = \arctan((y_{\text{obs},i} - y)/(x_{\text{obs},i} - x))$ are the distance and angle from the i^{th} obstacle, r_{obs} is the sum of the radii of the obstacle and robot, and $\zeta > 0$ determines the effect of the heading angle on safety. The controller used to drive the system is the 4.39 controller with the nominal controller $\mathbf{k}_{\text{nom}}$ from (5.33) and (5.34). In practice, infeasibilities of this safety filter were considered unsafe and the inputs were saturated such that $v \in [-0.2, 0.3]$ m/s and $\omega \in [-0.4, 0.4]$ rad/s. The velocity command $\mathbf{v}$ is computed at 20 Hz and error introduced by this sampling scheme is captured by the tracking error $\mathbf{d}(t)$. Tracking of $\mathbf{v}$ is performed by an inverse dynamics quadratic program (ID-QP) walking controller designed using the concepts in [95], which realizes a stable walking gait for (4.2) at 1 kHz.

Simulation results: We simulated the quadruped executing the proposed controller with parameters provided by SA-LineCoSpar. The resulting trajectories and the position of the obstacles are shown in Figure 5.8. We ran 30 iterations, with 3 new

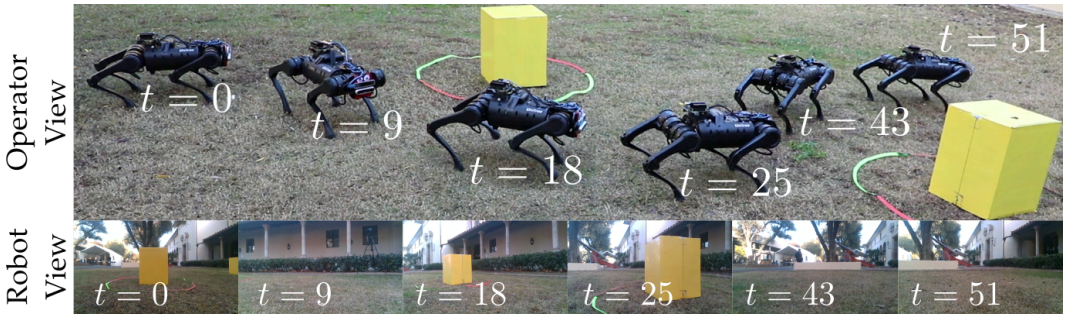


Figure 5.9. The preferred action, $\hat{\mathbf{a}}^*_{40} = (5, 0.1, 0.4, 0.02)$, after simulation, indoor experiments, and 3 additional iterations of 3 actions in an outdoor environment is shown alongside views from the onboard camera.

actions sampled in each iteration ($s = 3$), and obtained user preferences and ordinal labels in between each set of actions. To simulate perception error, the measurements of the obstacles were shifted by -0.1 m in the y-direction. The parameters found with SA-LineCoSpar allow the robot to navigate between obstacles. For comparison, a conservative action is also shown, which is safe but fails to progress towards the goal. SA-LineCoSpar eliminates this conservatism with only minor safety violations and determines a parameter set which is both safe and performant, foregoing the rigorous theoretical guarantees to instead find desireable closed-loop behavior.

Hardware results: After the simulation experiments, we continued the learning process on hardware in a laboratory setting for 7 additional iterations until the user was satisfied with the experimental behavior. The robot and obstacle positions were estimated using Intel RealSense T265 and D415 cameras to perform SLAM and segmentation. Centroids of segmented clusters in the occupancy map were used as the measured obstacle positions $\widehat{\boldsymbol{\rho}}_i$. The true robot and obstacle positions were obtained for comparison using an OptiTrack motion capture system. The results of these experiments can be seen in Figure 5.8. Afterwards, three additional iterations were conducted outdoors on grass until again the user was satisfied with the experimental behavior. The resulting best trajectory which safely satisfied the safety and performance goals can be seen in Figure 5.9. The preferred action was also tested on a variety of other obstacle arrangements to confirm its generalizability. The performance of the final preferred action for these obstacle configurations can be seen in the supplementary video [53].

Conclusion

In this section we proposed a design paradigm for control systems in which the robust safety requirements of a provably safe, but conservative controller are relaxed,

and controller parameters are instead chosen using a safety-aware preference-based learning algorithm called SA-LineCoSpar. Using our algorithm, we were able to learn a set of parameters that leads to user-preferred balance between safety and robustness on a quadrupedal robot platform.

While robust safety-critical control methods can be highly conservative, they show incredible utility in identifying tunable parameters that can be adjusted to produce robust behaviors. Their underlying theoretical proofs, establishes an interpretable understanding to the effect of tuning any particular parameter as discussed of the TR-OP controller's paraters 4.39 in Section 4.3.

5.5 Learning Responsibility

In this section we consider the context of decentralized multi-agent systems, like humans driving on a crowded street, where it is impractical if not impossible for a single agent to assume responsibility for the whole system's safety. Being robust to every possible action of every other agent would make driving impossible. Instead, drivers have a duty to exercise reasonable care when interacting with other road users [164]. The assumption that other road users will exercise *responsible* behaviors enables everyone to maintain safety without explicit coordination.

To safely and fluently interact with other agents, it is critical to understand how much responsibility any particular agent must take for achieving safety, and to provide robots with mathematical or algorithmic representations of this responsibility. Unfortunately, most existing techniques make strong assumptions about how other agents will act which often results in defensive or erratic behavior [165]–[167]; while these assumptions allow for strong theoretical guarantees, they are often impractical as they result in infeasible planning problems or induce overly conservative behaviors. Therefore, a key challenge is developing safety controllers that are simultaneously robust to decentralized multi-agent uncertainty while also being capable of accounting for context-dependent social norms that effect how agents should implicitly coordinate.

Unfortunately, social responsibility is a complex, intangible concept that is often shared asymmetrically and whose allocation is highly dependent on context and social norms. Given the strong situational dependence of responsibility, we will focus on learning it in the case of autonomous driving, for which there is a large amount of high quality data [168] and for which the robot (in the form of an autonomous vehicle (AV)) is generally expected to behave similarly to a human

agent.

To tackle this challenge, many works focus on modeling and estimating drivers' social preferences to synthesize socially-aware driving behaviors. For instance, [169] estimates the "social value orientation" of other drivers and formulates a game-theoretic planner and [170] crafts a planning reward function that incentivizes an AV to be more cooperative. While these approaches demonstrate that accounting for social preferences can lead to more human-like AV behaviors, they do not provide any assurances or quantification of AV safety. This limitation inspired several works [171]–[173] that investigate collision-avoidance responsibility using safety-critical control techniques. However, these methods consider either centralized control or centrally-defined social preferences, which does not apply to the autonomous driving setting where the social preferences of other agents are not known precisely and cannot be assigned.

To design systems that behave in accordance with intangible, context-dependent notions of responsibility, we propose combining safety-critical control with data-driven learning. While previous learning-based safety methods [41], [174]–[176] focus on learning safety constraints from data, we introduce a framework that starts from a centralized, multi-agent safety specification and then derives decentralized constraints that encode an individual agent's social responsibility from human driving data.

We formalize this framework as *Responsibility-Aware Control Barrier Functions (RACBFs)*. RACBFs enable AVs to reason about and enforce safety in a manner that is both decentralized and grounded in context-sensitive social responsibility, allowing for safe, interpretable, and human-aligned behavior in multi-agent environments.

The contributions of this section are as follows:

- The concept of Responsibility-Aware Control Barrier Functions (RACBFs) which extends the standard CBF to account for asymmetric sharing of responsibility between multiple agents.
- A data-driven constraint-learning algorithm to infer the responsibility allocations modeled in the RACBF formulation. A simulated demonstration of the utility of RACBFs and their learned responsibility allocations in safe closed-loop AV control.

The text for this section is adapted from:

> R. K. Cosner, Y. Chen, K. Leung, and M. Pavone, "Learning responsibility allocations for safe human-robot interaction with applications to autonomous driving," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9757–9763, 2023. DOI: 10.1109/ICRA48891.2023.10161112,

Responsibility-aware Decentralized Multi-agent Safety

In this section, we extend the CBF framework to a decentralized multi-agent setting and introduce additional terms to account for asymmetrically shared responsibility.

We extend the open-loop nonlinear control-affine system dynamics (2.1) to multiple agents:

$$\dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}_i) + \mathbf{g}_i(\mathbf{x}_i)\mathbf{u}_i \tag{5.36}$$

where $\mathbf{x}_i \in \mathbb{R}^{n_{x,i}}$, $\mathbf{u}_i \in \mathcal{U}_i \subset \mathbb{R}^{m_{u,i}}$, $\mathbf{f}_i : \mathbb{R}^{n_{x,i}} \to \mathbb{R}^{n_{x,i}}$, $\mathbf{g}_i : \mathbb{R}^{n_{x,i}} \to \mathbb{R}^{m_{u,i}}$ represent the state, input, drift, and actuation matrix of agent i. For the entire system of $N \in \mathbb{N}$ agents, let $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1^\top & \cdots & \mathbf{x}_N^\top \end{bmatrix}^\top \in \mathbb{R}^{n_x}$ denote the concatenated state and the dynamics for $\mathbf{x}$ be denoted as in (2.1):

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{f}_1(\mathbf{x}_1) \\ \vdots \\ \mathbf{f}_N(\mathbf{x}_N) \end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix} \mathbf{g}_1(\mathbf{x}_1) \\ \vdots \\ \mathbf{g}_N(\mathbf{x}_N) \end{bmatrix}}_{\mathbf{g}(\mathbf{x})} \underbrace{\begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_N \end{bmatrix}}_{\mathbf{u}}. \tag{5.37}$$

If the multi-agent system is governed by a centrallized controller, the CBF inequality (2.33) can be checked directly and used as a constraint in an optimization-based controller to obtain safe inputs [10]. However, centralized control is often unrealizable for AVs due to communication and scalability issues as well as the presence of human actors. Thus, we focus on a decentralized variant of the CBF inequality and assume that each agent can measure the states of the other agents, but independently generates its own input according to some controller unknown to the other agents.

One common method for retaining safety guarantees in the context of decentralized control, is to ensure robustness with respect to all possible actions of the other agents (including the worst-case inputs) as in [166]. In this case, the CBF constraint (2.33)

from the perspective of agent i becomes:

$$\sup_{\mathbf{u}_i \in \mathcal{U}_i} \inf_{\substack{\mathbf{u}_j \in \mathcal{U}_j, \\ j \neq i}} L_{\mathbf{f}} h(\mathbf{x}) + L_{\mathbf{g}} h(\mathbf{x}) \mathbf{u} \geq -\alpha(h(\mathbf{x})). \tag{5.38}$$

This is a conservative constraint which ensures the safety of the system even when other agents act adversarially.

Despite their safety-guarantees, worst-case constraints like (5.38) are highly conservative and prevent the system from achieving performant closed-loop behaviors [53]. It is therefore desirable to find a less conservative safety constraint that is more cognizant of the social interactions between agents even when the controllers of the other agents are unknown. For this purpose, we consider a novel CBF framework that models *social responsibility*.

Responsible-Aware Control Barrier Functions

In multi-agent systems of human actors, the responsibility for maintaining safety is typically shared among several people. For example, humans exhibit social behavior in crowd navigation and driving where the burden of maintaining safety is distributed between everyone [169], [177]. Equipped with the notion that agents share the responsibility for maintaining safety, we move away from worst-case behavioral assumptions, and instead, *learn* the responsibility allocation from data. First, we define *responsibility allocation functions*:

Definition 5.6 (Responsibility Allocation Function)**.** *A function* $\gamma_{\mathrm{r}} : \mathbb{N} \times \mathbb{R}^{n_x} \to \mathbb{R}$ *is a responsibility allocation function for* $N \in \mathbb{N}$ *if for all* $\mathbf{x} \in \mathbb{R}^{n_x}$*:*

$$\sum_{i=1}^{N} \gamma_{\mathrm{r}}(i, \mathbf{x}) \geq 0. \tag{5.39}$$

For agent i in a multi-agent system at state $\mathbf{x}$, $\gamma_{\mathrm{r}}(i, \mathbf{x}) > 0$ indicates increased responsibility, $\gamma_{\mathrm{r}}(i, \mathbf{x}) = 0$ indicates evenly shared responsibility, and $\gamma_{\mathrm{r}}(i, \mathbf{x}) < 0$ indicates decreased responsibility. The sum of $\gamma_{\mathrm{r}}(i, \mathbf{x})$ is lower bounded by zero to ensure that the total allocated responsibility must be greater than or equal to that of even sharing.

Using these responsibility allocation functions we can present our definition of Responsibility-Aware Control Barrier Functions (RACBFs) which consider responsibility allocation in their decentralized multi-agent safety constraint:

Definition 5.7 (Responsibility-Aware Control Barrier Function[4])**.** *Let $\mathcal{C} \subset \mathbb{R}^{n_x}$ be the 0-superlevel set of some continuously differentiable function $h : \mathbb{R}^{n_x} \to \mathbb{R}$ with 0 a regular value. Additionally, let $\gamma_{\mathrm{r}} : \mathbb{N} \times \mathbb{R}^{n_x} \to \mathbb{R}$ be a responsibility allocation function for $N \in \mathbb{N}$. The function h is a Responsibility-Aware CBF for the system* (5.37) *and responsibility allocation function γ_{r} if there exists an extended class $\mathcal{K}_\infty$ function α such that for all $\mathbf{x} \in \mathcal{C}$ and all $i \in \{1, \dots, N\}$:*

$$\sup_{\mathbf{u}_i \in \mathcal{U}_i} \overbrace{\underbrace{L_{\mathbf{g}_i} h(\mathbf{x})\mathbf{u}_i + \frac{1}{N}\Big(\alpha(h(\mathbf{x})) + L_{\mathbf{f}} h(\mathbf{x})\Big)}_{\text{RACBF Constraint}(i,\mathbf{x},\mathbf{u},\gamma_{\mathrm{r}}) \triangleq} - \gamma_{\mathrm{r}}(i, \mathbf{x})}^{\mathbf{c}_i(\mathbf{x},\mathbf{u}) \triangleq} \geq 0. \tag{5.40}$$

With this definition we can now make safety guarantees for the multi-agent system assuming that all of the agents agree on a single responsibility allocation function γ_{r}.

Theorem 5.8 (Responsibility-Aware Safety)**.** *Given a set $\mathcal{C} \subset \mathbb{R}^{n_x}$ defined as the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^{n_x} \to \mathbb{R}$ with 0 a regular value, if h is an RACBF for* (5.36) *and the responsibility allocation function $\gamma_{\mathrm{r}} : \mathbb{N} \times \mathbb{R}^{n_x} \to \mathbb{R}$ for $N \in \mathbb{N}$, then any locally Lipschitz controller $\mathbf{k} : \mathbb{R}^{n_x} \to \mathbb{R}^m$ that satisfies* (5.40) *for all $\mathbf{x} \in \mathcal{C}$ and $i \in \{1, \dots, N\}$, can be used with the dynamics* (5.36) *to make a safe closed-loop system with respect to $\mathcal{C}$.*

Proof. First let $\mathbf{c}_i(\mathbf{x}, \mathbf{k}(\mathbf{x})) \triangleq L_{\mathbf{g}_i} h(\mathbf{x})\mathbf{k}_i(\mathbf{x}) + \frac{1}{N}\Big(\alpha(h(\mathbf{x})) + L_{\mathbf{f}} h(\mathbf{x})\Big)$ for all $i \in \{1, \dots, N\}$. Since the $\mathbf{k}_i$ satisfies (5.40),

$$0 \geq -\mathbf{c}_i(\mathbf{x}, \mathbf{k}(\mathbf{x})) + \gamma_{\mathrm{r}}(i, \mathbf{x}) \geq -\sum_{i \in \{1,\dots,N\}} \mathbf{c}_i(\mathbf{x}, \mathbf{k}(\mathbf{x})) + \sum_{i \in \{1,\dots,N\}} \gamma_{\mathrm{r}}(i, \mathbf{x}), \tag{5.41}$$

$$\geq -\sum_{i \in \{1,\dots,N\}} \mathbf{c}_i(\mathbf{x}, \mathbf{k}(\mathbf{x})). \tag{5.42}$$

The second in inequality (5.41) follows from the decentralized constraint (5.40) for all i and (5.42) holds since γ_{r} is a responsibility allocation function for N. Since the final inequality (5.42) is equivalent to the centralized CBF constraint (2.33), Theorem 2.20 implies the safety of system (2.2) with respect to $\mathcal{C}$. □

[4]For generality, RACBFs are presented for N agents but in practice it is common to enforce CBF constraints between each pair of agents where the number of constraints enforced on agent i's input grows linearly with the number of agents [92]. In this case there would be several pairwise RACBF constraints with $N = 2$. We take this approach in our application of RACBFs.

In summary, instead of considering the worst-case inputs from other agents, our RACBF approach uses $\gamma_{\mathrm{r}}(i, \mathbf{x})$ to allow agent i's required contribution to decentralized multi-agent safety to vary depending on the state $\mathbf{x}$ of all of the agents in the scene. Also, instead of explicitly considering the uncertainty in the other agents' actions, one perspective of the responsibility allocation function is that it models a bound on the *projection* [108] of the other agents' inputs onto the CBF time derivative. Thus we can learn the effect of the other agents' actions as a scalar adjustment to $\frac{dh}{dt}$ as opposed to predicting their exact trajectories.

Our responsibility model is similar to that of [171] which instead uses a multiplicative term and is limited to driftless systems (i.e., systems where $\mathbf{f}(\mathbf{x}) \equiv 0$). By using an additive term, our model is generally applicable to control affine systems and is capable of accounting for the effect of responsibility even when the unforced dynamic are unsafe, i.e., $\alpha(h) + L_{\mathbf{f}} h(\mathbf{x}) \leq 0$.

Learning Responsibility Allocations

In this section we formalize the problem of learning responsibility allocation functions $\gamma_{\mathrm{r}}(i, \mathbf{x})$ from data and describe our method for learning γ_{r} from expert demonstrations, given a known safe set $\mathcal{C}$ and associated CBF h.

We assume that agent i strives to minimize some unknown function $Q_i : \mathbb{R}^{n_{x,i} \times m_{u,i}} \to \mathbb{R}$ and does so according to a constrained optimal control policy:

$$\begin{aligned} \mathbf{k}_i(\mathbf{x}) = \underset{u_i \in \mathcal{U}_i}{\operatorname{argmin}} \quad & Q_i(\mathbf{x}_i, \mathbf{u}_i) \\ \text{s.t.} \quad & \text{RACBF Constraint}(i, \mathbf{x}, \mathbf{u}, \gamma_{\mathrm{r}}) \geq 0 \end{aligned} \tag{5.43}$$

where $Q_i(\mathbf{x}_i, \mathbf{u}_i)$ represents agent i's cost for input $\mathbf{u}_i$ at state $\mathbf{x}_i$. Although the cost function is unknown, we assume that all agents obey the RACBF constraint for some γ_{r} that we seek to learn, thus framing the problem of learning responsibility allocations as constraint learning.

Next let $\mathfrak{D} = \{\mathbf{u}^k, \mathbf{x}^k\}_{k=1}^{N_d}$ be a dataset of state-input pairs gathered from expert (human) demonstrations where $N_d \in \mathbb{N}$ represents the total number of data points collected. Since the cost function Q_i can vary during data collection, it is possible for a state to have several associated expert inputs.

Our goal is find some responsibility allocation function γ_{r} such that the RACBF constraint is satisfied for all state-input pairs in the expert demonstrations $\mathfrak{D}$. This

can be written as the constrained optimization problem:

$$\begin{aligned} \gamma_{\mathrm{r}}^{*} = \underset{\gamma_{\mathrm{r}}}{\operatorname{argmin}} \quad & \|\gamma_{\mathrm{r}}\| \\ \text{s.t.} \quad & \text{RACBF Constraint}(i, \mathbf{x}, \mathbf{u}, \gamma_{\mathrm{r}}) \geq 0, \ \forall i \in \{1, ..., N\}, \\ & \sum_{i \in \{1,...,N\}} \gamma_{\mathrm{r}}(i, \mathbf{x}) \geq 0, \qquad \text{for all } (\mathbf{x}, \mathbf{u}) \in \mathfrak{D}, \end{aligned} \tag{5.44}$$

where the constraints enforce satisfaction of the RACBF and ensure that γ_{r} is a responsibility allocation function.

To find an approximate solution to this problem we take inspiration from prior CBF and Lyapunov learning methods [41], [175] and relax (5.44) to the following unconstrained loss function:

$$\begin{aligned} \mathcal{L}(\mathfrak{D}, \gamma_{\mathrm{r}}) = & \|\gamma_{\mathrm{r}}\| + \lambda_1 \sum_{(\mathbf{x},\mathbf{u}) \in \mathfrak{D}} \sum_{i=1}^{n_x} \Big[- \mathbf{c}_i(\mathbf{x}, \mathbf{u}) + \gamma_{\mathrm{r}}(i, \mathbf{x}) \Big]_+ \\ & + \lambda_2 \sum_{(\mathbf{x},\mathbf{u}) \in \mathfrak{D}} \left[\sum_{i=1}^{n_x} -\gamma_{\mathrm{r},i}(i, \mathbf{x}) \right]_+ \end{aligned} \tag{5.45}$$

where $\lambda_1, \lambda_2, \in \mathbb{R}_{\geq 0}$ are hyperparameters which adjust the constraint relaxations and $[\,\cdot\,]_+ \triangleq \max\{\cdot, 0\}$. This loss function can then be used find approximate solutions to (5.44):

$$\gamma_{\mathrm{r}}^{*} \approx \underset{}{\operatorname{argmin}} \quad \mathcal{L}(\mathfrak{D}, \gamma_{\mathrm{r}}). \tag{5.46}$$

Responsibility Regularization

However, the loss function used in the unconstrained optimization (5.46) is insufficient since, as in Inverse Reinforcement Learning, the problem of learning the constraint in (5.43) is poorly defined since the optimal input generated by (5.43) is a function of both the unknown cost function Q_i and unknown responsibility allocation function γ_{r}. Intuitively, this is because we cannot answer the question "did the agent act that way because it wanted to (i.e., cost minimization) or because it had to (i.e., safety constraint satisfaction)?" To better define the constraint learning problem we take an approach similar to [178] and regularize γ_{r} by maximizing the likelihood that it was used in (5.43) to generate the expert demonstrations $\mathfrak{D}$.

Following the maximum entropy model presented in [127] with the variant for continuous-time nonlinear systems presented in [179] we wish to solve the opti-

mization problem:

$$\gamma_{\text{r,reg}}^{*} = \text{argmax} \sum_{(\mathbf{x},\mathbf{u})\in\mathfrak{D}} \mathcal{P}(\mathbf{u} \mid \mathbf{x}, \gamma_{\text{r}}). \tag{5.47}$$

We approximate the probability of a given $\mathbf{u}$, by choosing $\text{disc}(\mathcal{U})$ to be a finite discretization of the bounded input set $\mathcal{U}$ such as $\text{disc}(\mathcal{U}) = \{\mathbf{u} \in \mathcal{U} \mid \delta\lfloor \mathbf{u}/\delta \rceil\}$ for some $\delta > 0$ where $\lfloor\cdot\rceil$ rounds each component to the nearest integer. Mimicking the forms presented in [178], [179], the approximate probability of an input $\mathbf{u} \in \mathcal{U}$ given the system state $\mathbf{x}$ and responsibility allocation γ_{r} is:

$$\mathcal{P}(\mathbf{u} \mid \mathbf{x}, \gamma_{\text{r}}) = \frac{e^{R(\mathbf{x},\mathbf{u})}}{Z_{\gamma_{\text{r}}}} \mathbb{1}^{\gamma_{\text{r}}}(\mathbf{x}, \mathbf{u}), \qquad Z_{\gamma_{\text{r}}} = \sum_{\boldsymbol{\upsilon}\in\text{disc}(\mathcal{U})} e^{R(\mathbf{x},\boldsymbol{\upsilon})} \mathbb{1}^{\gamma_{\text{r}}}(\mathbf{x}, \boldsymbol{\upsilon}) \tag{5.48}$$

where $Z_{\gamma_{\text{r}}}$ is the partition function, $R : \mathbb{R}^{n_x} \times \mathbb{R}^{m} \to \mathbb{R}$ is the reward function, and $\mathbb{1}_{\text{r}}^{\gamma}(\mathbf{x}, \mathbf{u}) \mapsto \{0, 1\}$ indicates satisfaction of the RACBF constraints given $\mathbf{x},\ \mathbf{u}, \gamma_{\text{r}}$.

To maximize the likelihood of the demonstration we minimize the number of feasible inputs while retaining the feasibility of the expert demonstrations. We note the total number of feasible inputs in $\text{disc}(\mathcal{U})$ decreases as $\gamma_{\text{r}}(i, \mathbf{x})$ increases, regardless of R, so we can maximize $\mathcal{P}(\mathbf{u}|\mathbf{x}, \gamma_{\text{r}})$ without knowledge of the agents' reward functions by maximizing γ_{r} while maintaining feasibility of the expert demonstrations. This can be expressed as the optimization problem:

$$\begin{aligned} \gamma_{\text{r, reg}}^{*} \approx \text{argmax}_{\gamma_{\text{r}}} \quad & \sum_{(\mathbf{x},\mathbf{u})\in\mathfrak{D}} \sum_{i=1}^{n_x} \gamma_{\text{r}}(i, \mathbf{x}) \\ \text{s.t.} \quad & \text{RACBF Constraint}(i, \mathbf{x}, \mathbf{u}, \gamma_{\text{r}}) \geq 0, \\ & \text{for all } i \in \{1, \ldots, N\} \text{ and } (\mathbf{x}, \mathbf{u}) \in \mathfrak{D}. \end{aligned} \tag{5.49}$$

Since constraint feasibility on $\mathfrak{D}$ is already accounted for in (5.45), this regularization can be added to the loss as:

$$\mathcal{L}_{\text{reg}}(\mathfrak{D}, \gamma_{\text{r}}) \triangleq \mathcal{L}(\mathfrak{D}, \gamma_{\text{r}}) + \lambda_3 \sum_{(\mathbf{x},\mathbf{u})\in\mathfrak{D}} \sum_{i=1}^{n_x} -\gamma_{\text{r}}(i, \mathbf{x}) \tag{5.50}$$

with hyperparamter $\lambda_3 \in \mathbb{R}_{\geq 0}$ which can be used in the final regularized optimization problem to estimate γ_{r}:

$$\gamma_{\text{r}}^{*} = \text{argmin}\ \ \mathcal{L}_{\text{reg}}(\mathfrak{D}, \gamma_{\text{r}}). \tag{5.51}$$

Application to Autonomous Driving

In this section we apply our RACBF and responsibility allocation learning method to urban driving using the Boston Seaport data in the nuScenes dataset [168].

We assume that all agents in the scene are vehicles (i.e., there are no pedestrians) and we model each agent as:

$$\underbrace{\begin{bmatrix}\dot{x}_i\\ \dot{y}_i\\ \dot{v}_i\\ \dot{\theta}\end{bmatrix}}_{\dot{\mathbf{x}}_i} = \underbrace{\begin{bmatrix}v_i\cos(\theta_i)\\ v_i\sin(\theta_i)\\ 0\\ 0\end{bmatrix}}_{\mathbf{f}_i(\mathbf{x}_i)} + \underbrace{\begin{bmatrix}0 & 0\\ 0 & 0\\ 1 & 0\\ 0 & 1\end{bmatrix}}_{\mathbf{g}_i(\mathbf{x}_i)} \underbrace{\begin{bmatrix}a_i\\ \omega_i\end{bmatrix}}_{\mathbf{u}_i}, \tag{5.52}$$

where $(x_i, y_i) \in \mathbb{R}^2$, $v_i, \theta_i, a_i, \omega_i \in \mathbb{R}$ represent the position, velocity, yaw, acceleration, and yaw rate[5] of vehicle i.

Safe Set Synthesis: To define the safety criterion function $h_0 : \mathbb{R}^{n_x} \to \mathbb{R}$, we begin by assuming all vehicles must maintain a minimum inter-vehicle distance $\underline{d} > 0$. With this in mind, let $d_{\text{min}} : \mathbb{R}^4 \times \mathbb{R}^4 \to \mathbb{R}$ be the minimum distance between two agents. We can then define the pairwise safe set between agents i and j to be:

$$\mathcal{C}_0^{ij} = \big\{\mathbf{x} \in \mathbb{R}^{n_x} \mid \underbrace{d_{\text{min}}(\mathbf{x}_i, \mathbf{x}_j) - \underline{d}}_{h_{ij}(\mathbf{x}) \triangleq} \geq 0\big\}. \tag{5.53}$$

However this function is of relative degree 2 w.r.t. a_i (i.e., $\frac{dh_0}{dt}$ is not directly affected by the inputs a_i or ω_i) and describes safety by only considering the instantaneous current position.

In order to incorporate a temporal aspect and ensure that the time derivative of h_0^{ij} is affected by both vehicles' acceleration and angle rate we take inspiration from the backup set safe set synthesis method of [103] and Section 3.6 and forward project the current state using the backup controller $\mathbf{k}_B : \mathbb{R}^4 \to \mathbb{R}^2$ over a time interval $[0, T]$ for $T \in \mathbb{R}_{>0}$. By assumption, for any $\mathbf{x}_i(t) \in \mathbb{R}^4$ there exists a unique solution $\boldsymbol{\phi} : [0, T] \to \mathbb{R}^4$ satisfying:

$$\frac{d}{dt}\boldsymbol{\phi}(\tau) = \mathbf{f}_i(\boldsymbol{\phi}(\tau)) + \mathbf{g}_i(\boldsymbol{\phi}(\tau))\mathbf{k}_B(\boldsymbol{\phi}(\tau)), \tag{5.54}$$

$$\boldsymbol{\phi}(0) = \mathbf{x}_i(t). \tag{5.55}$$

[5]Bezier curves are fit to position and yaw data and then differentiated to obtain velocity, acceleration, and yaw rate. The code repository for learning the responsibility allocation function γ_{r} can be found https://github.com/rkcosner/learning_responsibility_allocation.

The solution $\boldsymbol{\phi}$ starting at $\mathbf{x}_i(t)$ is the flow under $\mathbf{k}_B$, and is denoted as $\boldsymbol{\varphi}_\tau(\mathbf{x}_i) \triangleq \boldsymbol{\phi}(\tau)$. Similar ideas of forward projection are also seen in Velocity Obstacles [180], Safety Force Fields [165], and Responsibility-Sensitive Safety [167].

Using the flow $\boldsymbol{\varphi}$ and the distance function d_{ij} we can find the minimum distance that would be achieved during the interval $[t, T+t]$ if $\mathbf{k}_B$ were the controller for both vehicles:

$$h_{ij}^{\boldsymbol{\varphi}}(\mathbf{x}) = \min_{\tau \in [0,T]} d_{\text{min}}(\boldsymbol{\varphi}_\tau(\mathbf{x}_i), \boldsymbol{\varphi}_\tau(\mathbf{x}_j)) - \underline{d}, \tag{5.56}$$

which has the associated safe set $\mathcal{C}_{ij}^{\boldsymbol{\varphi}} \subseteq \mathcal{C}_{ij} \subset \mathbb{R}^{n_x}$

$$\mathcal{C}_{ij}^{\boldsymbol{\varphi}} = \left\{ \mathbf{x} \in \mathbb{R}^{n_x} \mid h_{ij}^{\boldsymbol{\varphi}}(\mathbf{x}) \geq 0 \right\}. \tag{5.57}$$

To compute $h_{ij}^{\boldsymbol{\varphi}}$ the interval $[0, T]$ was discretized at 100 Hz as in [52], [103] and soft minimum functions were used to ensure differentiability. It is shown in [181] that the CBF $h_{ij}^{\boldsymbol{\varphi}}$ constructed from the backup controller is guaranteed to be of relative degree 1 under mild assumptions.

For the backup controller we choose $\mathbf{k}_B(\mathbf{x}_i) = \mathbf{0}$ which approximates idling. Unlike other methods [167], [182] which assume maximum braking for their predictors, we choose an idling controller since it better approximates nominal driving behavior and does not introduce worst-case assumptions.

Given these pairwise safe sets $\mathcal{C}_{ij}^{\boldsymbol{\varphi}}$, we can define the a global safe set $\mathcal{C}^{\boldsymbol{\varphi}} \subseteq \mathcal{C}_{ij}^{\boldsymbol{\varphi}} \subseteq \mathcal{C}_{ij}$ for all $i \neq j$ as:

$$\mathcal{C}^{\boldsymbol{\varphi}} = \bigcap_{i \neq j} \mathcal{C}_{ij}^{\boldsymbol{\varphi}} \quad \text{with} \quad h(\mathbf{x}) = \min_{i \neq j} h_{ij}^{\boldsymbol{\varphi}}(\mathbf{x}). \tag{5.58}$$

The intersection of safe sets has been studied in [183] and safety of such sets can be achieved by enforcing the safety constraint for all $i \neq j$.

Learning Setup: The inputs of the responsibility allocation function are an image with semantic labels as see in Figure 5.10 and the relative vehicle states of agents i and j. The image is processed by ResNet-18 [184] and the 256 dimensional output is concatenated to the vehicle states and processed by a multi-layer perceptron (MLP) with 2 hidden layers of size 128 and a single dimensional output.

The hyperparameters chosen were $\lambda_1 = 1$, $\lambda_2 = 10$, $\lambda_3 = 0.01$, $\alpha = 0.5$, $T = 1$, $\underline{d} = 0.4$, $\ell_1 = 0.1$, $\ell_2 = 0.01$ and $\theta_{\text{max}} = 100^\circ$ where ℓ_1 and ℓ_2 were the negative

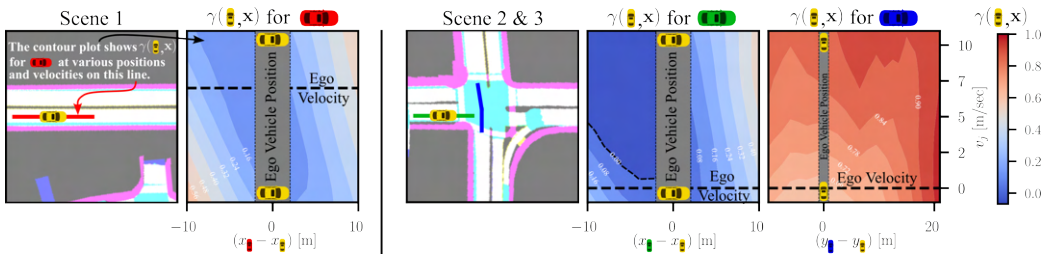


Figure 5.10. The learned responsibility allocation surface is visualized for a range of velocities and relative positions. **Scene 1:** the ego vehicle (yellow) is driving on a two lane road. In all cases, $\gamma_{\text{r}}(\text{ego}, \mathbf{x}) > 0$ indicating a degree of conservative driving. Generally, $\gamma_{\text{r}}(\text{ego}, \mathbf{x})$ is larger when the other agent (red) is in front of the ego vehivlethan when behind, indicating increased responsibility when driving behind another vehicle. **Scene 2:** The ego vehicle is stopped at a four-way intersection with the other vehicle (green) ahead or behind it (and no blue agent). Again the ego vehicle (yellow) is more responsible when the green vehicle is in front of it than when it is behind it. **Scene 3:** The ego vehicle (yellow) is stopped at a four-way intersection with the other vehicle (blue) crossing from top to bottom (and no green agent). $\gamma_{\text{r}}(\text{ego}, \mathbf{x})$ is large for all positions and velocities of the blue vehicle showing that the ego agent takes is more responsible in this situation.

slopes of the MLP's leaky ReLU activation functions and θ_{max} is used to filter the dataset such that only interactions between vehicles whose headings are within $\pm\theta_{\text{max}}$ are considered. The parameter θ_{max} is necessary since our data does not include lane direction annotation. We note that this does limit the applicability of this network and plan to include lane direction information in future work.

The network was trained on the NuScenes Boston Seaport dataset. Example responsibilities generated by our learned model can be found in Fig. 5.10. These figures show that our model conforms to the general intuition that the vehicle behind is more responsible than the vehicle in front for avoiding collisions between them, and the vehicle stopped at an intersection is responsible for not interfering with a vehicle already crossing the intersection.

Closed-Loop Testing: We use our RACBF framework with a learned responsibility allocation function as a safety-filter in closed-loop control and simulate human-like driving using the Bi-Level Imitation for Traffic Simulation (BITS) model [185]. The ego agent follows (5.43) where $Q_i(\mathbf{x}_i, \mathbf{u}_i) = \|\mathbf{k}^+_{\text{bits}}(\mathbf{x}_i) - \mathbf{u}_i\|^2$ and $\mathbf{k}^+_{\text{bits}}$ is the BITS controller with an additional 1 $\frac{\text{m}}{\text{sec}^2}$ acceleration added to generate irresponsible desired behavior that must be filtered to ensure safety. The RACBF constraint is applied for each pairwise vehicle j and slack variables are used to ensure feasibility. We compare our method to the same controller with two other baseline constraints: (i) *"Worst-Case"* constraint (5.38), and (ii) *"Even-Sharing"* constraint which is the RACBF constraint with $\gamma_{\text{r}}(i, \mathbf{x}) \equiv 0$.

The closed-loop system was run in 120 scenarios sampled from NuScenes for 10

Closed-Loop Simulation Results

	Worst-Case	Even-Split	Our Method
Validation Constraint Violation	43.99%	8.13%	9.51%
Closed-Loop Safety Violation	0.833%	2.50%	0.833%
Time Spent Off Road	1.48%	0.59%	0.54%
Distance Covered Metric	290.21	309.21	307.84

Table 5.2. Results for the closed-loop experiments.

seconds at 10 Hz. Table 5.2 contains metrics comparing the controllers. The Worst-Case controller has the fewest safety violations (as expected), but worse compatibility with the expert demonstrations as indicated by the large constraint violation, smallest distance covered, and significant amount of time off of drivable surfaces. The Even-Sharing controller has fewer constraint violations on the validation data and the most distance covered, but allows for more closed-loop collisions. Our method has a slightly higher number of constraint violations on the validation data set and slightly less distance covered, but achieves a better trade-off between performanc (distance covered) and closed-loop safety.

Conclusion

This section presented Responsibility-Aware Control Barrier Functions (RACBFs) as a framework to synthesize safe actions with an understanding of multi-agent social responsibility. RACBFs are designed to capture the asymmetric sharing of responsibility between multiple (human) agents and we present a method to learn context-dependent responsibility allocations from data. We then demonstrated the efficacy and utility of our method by training on human data and deploying in a closed-loop driving simulation. This work enables various exciting future directions which include incorporating explicit traffic rules into our responsibility-learning paradigm, comparing how responsibility allocations vary across geographical regions, and exploring other application domains such as crowd navigation.

5.6 Conclusion

The contributions in this chapter extend robust safety-critical control techniques by incorporating learning-based methods. Specifically, I presented strategies that (1) reduce system uncertainty through learned models of dynamics and perception, and (2) enable controllers to reason about abstract, human-centered concepts that are essential for safe and effective behavior, such as preferred risk-performance tradeoffs and social responsibility in multi-agent environments.

These methods build on the formal safety tools developed in earlier chapters by introducing learning-tuned adaptations that reduce conservatism to levels that are

practically, even if not *formally*, sufficient. By reducing uncertainty and operating below the worst-case theoretical thresholds, these systems gain the flexibility needed to achieve high-performance behavior while still practically achieving safety. In doing so, this chapter merges theoretical and end-to-end approaches in a way that preserves a level of interpretability and safety-awareness while also enabling more capable behavior and adaptive behavior.

The methods and philosophies introduced here point to several promising directions for future research, where robust safety-critical control frameworks are embedded within, combined with, or applied to learning systems to create high-performance solutions that are deployable in safety-critical settings and which are capable of learning across their lifetime. Exciting work in safe learning [155], safe reinforcement learning [42], [43], learning with safety guarantees [40], and out-of-distribution detection in human-robot interaction [186] demonstrates the field's momentum in this direction [187].

Chapter 6

STOCHASTIC ROBUSTNESS

> *"What do you mean you have safety guarantees? Do they still hold if the sky starts falling and aliens attack?"*
>
> *"It is far better to foresee even without certainty than not to foresee at all."* - Henri Poincaré
>
> *"In Pokémon, no one every really catches them all. That's the slogan, but it's not really the point of the game."* - Andrew J. Taylor

Robust theoretical guarantees of safety can be incredibly motivating and useful: if we can guarantee that our system is safe, then we can deploy it with confidence. However, as discussed in the previous chapters, traditional robust guarantees that rely on worst-case bounds of uncertainty can be difficult to translate to real world systems. After all, how can we bound all uncertainty? What's the bounded effect of the sky falling or aliens invading?

Some events are so unlikely that it probably is not worth being robust to them. This realization motivates an alternative, more flexible perspective on safety: one based on probability. In this chapter, we move beyond worst-case guarantees to introduce a stochastic approach to safety. Instead of guarantees that hold with 100% certainty, we explore the more nuanced understanding of *probabilistic safety*, acknowledging that maybe robustifying against *all* possible uncertainties, like catching all the Pokémon, is not really the point.

Abstract

Our world is inherently chaotic and unpredictable. While worst-case guarantees from robust safety-critical control provide confidence in the face of uncertainty, they inherently fail to capture the randomness of the real world. If we wish to retain meaningful guarantees under *unbounded* disturbances, we must relinquish the notion of absolute certainty and adopt a more nuanced understanding of safety. By reasoning about the distribution of uncertainty, we can construct *probabilistic safety guarantees* that remain valid even without conservative worst-case bounds.

This chapter presents a stochastic approach to safety, risk-based robust controllers, and rigorous theoretical guarantees that hold in this context. Since the mathematical tools for reasoning about stochastic processes differ substantially from the background developed in Chapter 2, we begin by introducing the necessary theoretical preliminaries. Section 6.3 then establishes a connection between control barrier functions (CBFs) and martingales (a well-studied class of stochastic processes) and uses this connection to construct rigorous guarantees and practical algorithms for CBF-based controllers under unbounded uncertainty. In Section 6.4, we explore an alternative approach based on a different martingale concentration inequality that yields tighter probability bounds on safety in certain cases. The methods present a markedly different approach than those introduced in the prior chapters of this thesis. Instead of conservatively enforcing worst-case guarantees (Chp. 4) or foregoing those guarantees in search of data-driven performance (Chp. 5), this chapter provides a middle ground, where we achieve performance improvements while retaining theoretical guarantees by tolerating some closed-loop risk.

Published content: The text for this chapter is adapted from [57] and [58].

6.1 Introduction

This chapter diverges methodologically from the previous contributions of this thesis and adopts a stochastic, discrete-time perspective on robot safety. Here I present several methods that provide theoretical guarantees of safety despite stochastic and potential unbounded disturbances. In lieu of the absolute certainty provided by the worst-case bounds of Chapter 4, this chapter generates risk-based safety guarantees which allow a controls engineer to modulate the acceptable risk in robot deployment, providing additional nuance and allowing for increased flexibility and performance based on risk-tolerance.

The probabilistic bounds of this chapter are particularly useful for real-world robotic systems where disturbances are often modeled as continuous random variables with unbounded support (e.g., zero-mean, additive Gaussian noise). For such systems, it is impossible to give an absolute bound on the disturbance magnitude, so the results of Chapter 4 do not directly apply. Alternatively, a wide variety of stochastic safety methods exist that do tackle this problem including: stochastic reachability-based methods [188], [189], constrained coherent risk measures [190], sampling-based general risk measures [191], and martingale-based methods [131], [192] among many others. In this work, we will focus on martingale-based methods due to their

ability to generate trajectory-long guarantees, their relative simplicity as a method which relies primarily on only a distribution's first-moment, and their inherent connections to CBFs. In particular we will focus on the discrete-time stochastic safety filter, a controller type which has recently begun to gain popularity [57], [131] and which differs significantly from many of the previous methods which assumed the presence of a nominally safe, stabilizing controller [192].

In order to best represent the uncertainty that might appear from sources such as discrete-time perception errors or sampled-data modeling errors, this chapter focuses on generating probabilistic bounds of safety and stability for discrete-time (DT) stochastic systems. While, continuous-time stochastic safety methods have successfully achieved strong probabilistic safety guarantees [35], [130], [193], [194], they generally require controllers with functionally infinite bandwidth, a strong assumption for real-world systems with discrete-time sensing and actuation. Alternatively, discrete-time methods have achieved success while also capturing the sampled-data complexities of most real-world systems [57], [192], [195], [196]. Although CBFs are normally applied in continuous time, they admit a discrete-time counterpart (discrete time CBFs (DCBFs)) that were first introduced in [197] and have gained popularity due to their compatibility with planners based on model predictive control (MPC) [46], [198], [199], reinforcement learning (RL) [43], and Markov decision processes [200].

Previous work has studied martingale-based techniques to establish safety guarantees [192], [195], yet these works have limited utility when analyzing the safety of discrete-time CBF-based controllers. In particular, the "c-martingale" condition used in [192] does not admit a multiplicative scaling of the barrier function, and therefore, at best, provides a weak worst-case safety bound for CBF-based controllers that grows linearly in time. The work of [195], which builds upon [35], is largely focused on offline control synthesis to achieve a desired safety bound (as opposed to the online, optimization-based control studied in this work). Also, this method can only generate controllers for affine barriers, which severely limits its applicability to general safety requirements. Both [192] and [195] depend on sum-of-squares (SoS) programming [89] for control synthesis/system verification, thereby requiring an offline step that scales poorly with the state dimension. The goal of this chapter is to extend the results of [35] in a different direction, and thereby enable the synthesis of online controllers that can be realized on robotic systems to achieve probabilistic safety guarantees in practice.

To develop these probabilistic guarantees and stochastic controllers, we begin by introducing relevant background material in Section 6.2, including the notion of discrete-time control barrier functions (DCBFs) and the definition of K-step exit probability, the notion of safety that we will adopt in the case of stochastic uncertainty. Next, in Section 6.3, we introduce a probabilistic safety guarantee that builds on the stochastic Lyapunov results in [35] and which holds in the presence of unbounded uncertainty. We additionally provide practical, computationally tractable methods for enforcing this guarantee through DCBF-based safety filters. While the theoretical results of Section 6.3 are compelling, they are also quite loose, relying on weak martingale-based concentration inequality (i.e., Ville's inequality [201]). To improve upon this, Section 6.4 proposes the use of an alternative, stronger martingale concentration inequality (i.e., Freedman's inequality [202]) to generate probabilistic safety guarantees. While the assumptions required for this alternative safety guarantee to hold are more restrictive, it results in generally tighter, better-calibrated risk bounds.

6.2 Background on Discrete-time Safety and Stochastic Safety

First, we provide the mathematical preliminaries for the remainder of this thesis. In particular, we shift from a continuous-time framework to a discrete-time one and introduce discrete-time control barrier functions (DCBFs). Next, we discuss probability spaces, disturbance distributions, martingales, and martingale concentration inequalities from which we will construct stochastic safety guarantees.

Robotics Motivation for Discrete-time Stochastic Systems

When discussing stochastic control processes, I choose to work in a discrete-time framework because it better reflects the reality of practical robotics implementations. For example, several authors have achieved arbitrarily good safety probabilities when working with continuous-time stochastic differential equations (SDEs) [110], [130], [193]; however, this takes advantage of the fact that SDEs allow for infinite controller bandwidth and have covariances over infinitesimal time intervals that are instantaneously zero. While the sampled-data approximation for deterministic systems generally leads to small errors [129], [203], that approximation can be much worse for sampled-data stochastic systems. For example, while authors can guarantee safety with 100% probability for SDEs, similar guarantees cannot be generally made for discrete-time systems:

Example 6.1 ([204, Sec. IV]). *Consider the system:* $\mathbf{x}_{k+1} = \mathbf{u}_k + \mathbf{d}_k$*, where* $\mathbf{x} \in \mathbb{R}, \mathbf{u} \in \mathbb{R}, \mathbf{d} \sim \mathcal{N}(0,1)$*, and* $\mathcal{C} = \{\mathbf{x} \in \mathbb{R} \mid |\mathbf{x}| < 1\}$*. At every step* $\mathbf{P}\{\mathbf{x}_{k+1} \in \mathcal{C} \mid \mathbf{x}_k\}$ *is maximized with* $\mathbf{u}_k = 0$*, but then even over a single discrete step, there is at least a 30% chance of failure. As time continues, this constant risk of failure at every step makes infinite horizon guarantees impossible to achieve for this system.*

While a full exploration of safety for sampled-data stochastic systems is an interesting direction for future work, this example motivates our decision to study discrete-time stochastic system which better capture the realities of robotic systems with stochastic uncertainties and limited controller bandwidth.

Discrete-time Control Barrier Functions

We begin by introducing the traditional notion of discrete-time control barrier functions (DCBFs) in the deterministic setting as first introduced in [197].

For this, we consider discrete-time systems of the form:

$$\mathbf{x}_{k+1} = \mathbf{F}_0(\mathbf{x}_k, \mathbf{u}_k), \tag{6.1}$$

with the state $\mathbf{x}_k \in \mathbb{R}^{n_x}$, input $\mathbf{u}_k \in \mathbb{R}^{n_u}$, time index $k \in \mathbb{N}_0$, and dynamics $\mathbf{F}_0 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ which capture discrete updates in the system. This discrete-time model can be generated for the continuous-time system (2.1) with a zero-order-hold, sampled-data controller implementation using piecewise solutions to the flow (2.3) over the sampling interval.

As with continuous time systems (2.1,2.2), we can add a state-feedback controller, $\mathbf{k} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ to yield a discrete-time closed-loop system of the form,

$$\mathbf{x}_{k+1} = \mathbf{F}_0(\mathbf{x}_k, \mathbf{k}(\mathbf{x}_k)), \tag{6.2}$$

and we can define deterministic safety as the forward invariance of this system:

Definition 6.2 (Discrete-time Forward Invariance and Safety[1]). *] A set* $\mathcal{C} \subset \mathbb{R}^{n_x}$ *is forward invariant for the system* (6.2)*, if* $\mathbf{x}_0 \in \mathcal{C}$ *implies that* $\mathbf{x}_k \in \mathcal{C}$ *for all* $\mathbf{k} \in \mathbb{N}$*. In this case, we call the system* (6.2) *safe with respect to* $\mathcal{C}$*.*

[1] With this discrete-time definition of safety (Def. 6.2) it is possible that a sampled-data robotic system would experience inter-sample safety failures. Since samples times are usually very fast and inter-sample failures usually quite small, the remainder of this thesis we will focus on the safety exclusively at samples times as in [197] and [203]. We refer to [129] for an analysis of CBF-based intersample safety.

The CBF definition (Def. 2.19) can then be modified to produce its discrete-time variant:

Definition 6.3 (Discrete-time Control Barrier Functions (DCBFs) [197])**.** *Let $\mathcal{C} \subset \mathbb{R}^{n_x}$ be the 0-superlevel set of a function $h : \mathbb{R}^{n_x} \to \mathbb{R}$. The function h is a discrete-time control barrier function (DTCBF) for* (6.2) *on $\mathcal{C}$ if there exists an $\alpha \in [0, 1]$ such that for each $\mathbf{x} \in \mathbb{R}^{n_x}$, there exists a $\mathbf{u} \in \mathbb{R}^{n_u}$ such that:*

$$h(\mathbf{F}_0(\mathbf{x}, \mathbf{u})) \geq \alpha h(\mathbf{x}). \tag{6.3}$$

For DCBFs, the CBF assumption that $\alpha \in \mathcal{K}_\infty^e$ is replaced with the assumption that $\alpha \in [0, 1]$ for simplicity.

DTCBFs differ from their continuous-time counterparts in that they satisfy an inequality constraint on their *finite difference* instead of their derivative[2]. On the other hand, they are similar in their ability to create *safety filters* for desired nominal controllers $\mathbf{k}_{\text{des}} : \mathbb{R}^{n_x} \times \mathbb{Z} \to \mathbb{R}^{n_u}$ of the form:

$$\begin{aligned} \mathbf{k}(\mathbf{x}) = \underset{\mathbf{u} \in \mathbb{R}^{n_u}}{\operatorname{argmin}} \quad & \|\mathbf{u} - \mathbf{k}_{\text{des}}(\mathbf{x}, k)\|^2 \\ \text{s.t.} \quad & h(\mathbf{F}_0(\mathbf{x}, \mathbf{u})) \geq \alpha h(\mathbf{x}). \end{aligned} \tag{6.4}$$

Assuming feasibility, the controller in (6.4) guarantees safety for the closed-loop system (6.2) by selecting inputs that satisfy (6.3) as formalized in:

Theorem 6.4 (DCBF safety, [197, Prop. 4])**.** . *Let $\mathcal{C} \subset \mathbb{R}^{n_x}$ be the 0-superlevel set of a function $h : \mathbb{R}^{n_x} \to \mathbb{R}$. If h is a DTCBF for* (6.2) *on $\mathcal{C}$, then the set*

$$\mathscr{K}_{\text{CBF}}(\mathbf{x}) = \{\mathbf{u} \in \mathbb{R}^{n_x} \mid h(\mathbf{F}_0(\mathbf{x}, \mathbf{u})) \geq \alpha h(\mathbf{x})\} \tag{6.5}$$

is non-empty for all $\mathbf{x} \in \mathbb{R}^{n_x}$ and, for any state-feedback controller $\mathbf{k}$ with $\mathbf{k}(\mathbf{x}) \in \mathscr{K}_{\text{CBF}}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^{n_x}$, the closed-loop system (6.2) *is safe with respect to $\mathcal{C}$.*

Proof. $h(\mathbf{x}_k) \geq \alpha^k h(\mathbf{x}_0)$ is lower-bounded by 0, thus ensuring the safety of $\mathcal{C}$. □

Remark 6.5. *If* (6.4) *is infeasible, a slack variable can be added to recover feasibility and its effect on safety can be analyzed using a discrete-time variant the ISSf framework [31]. Additionally, unlike the affine inequality constraint that arises*

[2]The standard continuous-time CBF condition $\dot{h}(\mathbf{x}) \geq -\overline{\gamma} h(\mathbf{x})$ for $\overline{\gamma} > 0$ becomes $h(\mathbf{x}_{k+1}) - h(\mathbf{x}_k) \geq -\gamma h(\mathbf{x}_k)$ for $\gamma \in [0, 1]$ in discrete-time; defining $\alpha = 1 - \gamma$ recovers the condition $h(\mathbf{x}_{k+1}) \geq \alpha h(\mathbf{x}_k)$.

with continuous-time CBFs (2.33)*, the optimization problem* (6.4) *is not necessarily convex. To ameliorate this issue, it is often assumed that* $h \circ \mathbf{F}_0$ *is concave with respect to* $\mathbf{u}$ *[46], [197], [205].*

Remark 6.6. *As a safety filter on* $\mathbf{k}_{\text{des}}$*, the closed-loop performance of system* (6.2) *using the controller* (6.4) *is indirectly achieved through the nominal controller. If the safety constraint and the performance objective of the nominal controller do not conflict, then the desired nominal controller allows the system to achieve its performance goal. However, if they do conflict, then myopic pointwise modifications are made to the nominal controller that enforce safety but may destroy the performance capabilities of the system [16].*

Stochastic Preliminaries

Next we introduce mathematical background for our stochastic understanding of safety. For this, let $(\Omega, \mathscr{F}, \mathbb{P})$ be a probability space and let $\mathscr{F}_0 \subset \mathscr{F}_1 \subset \cdots \subset \mathscr{F}$ be a filtration of $\mathscr{F}$ and consider discrete-time dynamical systems of the form:

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k), \quad \forall k \in \mathbb{N}_0 \tag{6.6}$$

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is the state, $\mathbf{u}_k \in \mathbb{R}^{n_u}$ is the input, $\mathbf{d}_k$ is an $\mathscr{F}_{k+1}$ measurable random disturbance which takes values in $\mathbb{R}^{n_d}$, and $\mathbf{F} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_x}$ is the discrete update dynamics. Throughout this work we assume that all random variables and functions of random variables are integrable.

To create a closed-loop system, we add a state-feedback controller $\mathbf{k} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$:

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{k}(\mathbf{x}_k), \mathbf{d}_k), \quad \forall k \in \mathbb{N}_0 \tag{6.7}$$

The goal of the remainder of this thesis will be to provide probabilisic safety guarantees for this closed-loop system.

For deterministic systems, infinite-horizon safety guarantees like those of Theorem 6.4 are common. However, for discrete-time stochastic systems, when the disturbance is bounded, infinite-horizon guarantees fail to capture the nuances of variable risk levels and, when the disturbance is unbounded, infinite-horizon guarantees can be impossible to achieve as outlined in Example 6.1. In order to provide an achievable risk-based guarantee we choose to analyze finite-time safety probabilities as in [35], [131], [192], [194] instead of infinite-time safety guarantees. In particular we consider the K-step exit probabilities of the safe set $\mathcal{C}$:

Definition 6.7 (K-step Exit Probability)**.** *For any $K \in \mathbb{N}_1$ and initial condition $\mathbf{x}_0 \in \mathbb{R}^{n_x}$, the K-step exit probability of the set $\mathcal{C}$ for the closed-loop system* (6.2) *is:*

$$P_u(K, \mathbf{x}_0) = \mathbb{P}\left\{\mathbf{x}_k \notin \mathcal{C} \text{ for some } k \leq K\right\}. \tag{6.8}$$

This describes the probability that the system will leave the safe set $\mathcal{C}$ within K time-steps given that it started at $\mathbf{x}_0$.

In the remainder of this thesis, we will generate bounds on K-step exit probabilities using martingale-based concetration inequalities. Martingales are a class of stochastic processes that satisfy an expectation-based relationship between their mean and previous value.

Definition 6.8 (Martingale [206], [192])**.** *Let $(\Omega, \mathscr{F}, \mathbb{P})$ be a probability space with a filtration $\{\mathscr{F}_0, \mathscr{F}_1, \ldots, \mathscr{F}\}$. A stochastic process W_k that is adapted to the filtration and is integrable at each k is a martingale if*

$$\mathbb{E}[\, W_{k+1} \mid \mathscr{F}_k \,] = W_k, \quad \forall k \in \mathbb{N}_0, \quad \text{(a.s.)}. \tag{6.9}$$

Furthermore, if W_k satisfies:

$$\mathbb{E}[\, W_{k+1} \mid \mathscr{F}_k \,] \leq W_k + c, \quad \forall k \in \mathbb{N}_0, \quad \text{(a.s.)}, \tag{6.10}$$

with $c = 0$ then it is a supermartingale and if it satisfies (6.10) *with $c \geq 0$ then it is a $c-$martingale.*

Many concentration inequalities can be used to bound the spread of a martingale over time. One particularly useful bound that will be used to generate stochastic safety guarantees is Ville's inequality [201] which bounds the probability that a supermartingale W_k rises above a threshold $\lambda > 0$.

Lemma 6.9 (Ville's Inequality [201])**.** *If W_k is a nonnegative supermartingale, then for all $\lambda > 0$,*

$$\mathbb{P}\left\{\sup_{k \in \mathbb{Z}} W_k > \lambda\right\} \leq \frac{\mathbb{E}[W_0]}{\lambda}. \tag{6.11}$$

Ville's inequality is a direct application of Markov's inequality [206, Chp. 7.2, Lem. 7] in conjunction with martingale stopping-times [206, Chp. 12.4, Def 1.]. A proof of Ville's inequality is provided in [207, Appx. A].

Lastly, as we will see that when synthesizing safety-critical controllers in the presence of stochastic disturbances, we will need to enforce conditions on the expectation of a DCBF. In doing so, we will need to relate the expectation of the next DCBF value, $h(\mathbf{x}_{k+1})$, to the expectation of the next state, $\mathbf{x}_{k+1}$. This will be achieved using Jensen's inequality:

Theorem 6.10 (Jensen's Inequality [208])**.** *Consider a continuous function* $h : \mathbb{R}^{n_x} \to \mathbb{R}$ *and a random variable* $\mathbf{x}$ *that takes values in* $\mathbb{R}^{n_x}$ *with* $\mathbb{E}[\|\mathbf{x}\|] < \infty$*. We have that:*

$$\begin{cases} \textit{if } h \textit{ is convex,} & \textit{then } \mathbb{E}[h(\mathbf{x})] \geq h(\mathbb{E}[\mathbf{x}]), \\ \textit{if } h \textit{ is concave,} & \textit{then } \mathbb{E}[h(\mathbf{x})] \leq h(\mathbb{E}[\mathbf{x}]). \end{cases} \tag{6.12}$$

6.3 Stochastic Safety Guarantees using DCBFs

In order to generate bounds on the safety probability, represented by $K-$step exit probability (Def. 6.7), we provide a theoretical connection between DCBFs and supermartingales and show how they can be used to achieve safety for stochastic and potentially unbounded disturbances. This understanding will allow us to provide nuanced theoretical guarantees that incorporate a tunable understanding of risk (i.e., tolerable failure probability).

The contributions of this section are as follows:

- A translation of the stochastic Lyapunov stability result in [35] to a safety setting.
- A new, more intuitive and complete proof of the result in [35] with connections to existing ISSf results [31] for bounded-uncertainty.
- An algorithmic method based on Jensen's inequality to account for the effects of process noise on a DCBF-based controller.
- Applications of this method to a variety of systems in simulation to analyze the tightness of our bound and demonstrate its utility.

The text for this section is adapted from:

> R. K. Cosner, P. Culbertson, A. J. Taylor, and A. D. Ames, "Robust safety under stochastic uncertainty with discrete-time control barrier functions," *Proceedings of Robotics: Science and Systems*, 2023. DOI: 10.15607/RSS.2023.XIX.084,

Ville's-based DCBF Safety Guarantees

Our first result is a reframing of the stochastic invariance theorems in [35], [195] using the standard formulation of DCBFs. Additionally, we produce a probability bound for both $\mathcal{C}$ (defined as the 0-superlevel set of h) and all non-positive superlevel sets of h (i.e., $\mathcal{C}_\delta$ (2.38)) resulting in a discrete-time stochastic variant of the ISSf property [31].

Theorem 6.11 (DCBF Safety using Ville's-inequality)**.** *Let $h : \mathbb{R}^{n_x} \to \mathbb{R}$ be a continuous, upper-bounded function with upper bound $B \in \mathbb{R}_{>0}$. Suppose there exists an $\alpha \in (0, 1)$ and $c \leq B(1-\alpha)$ such that the closed-loop system* (6.7) *satisfies:*

$$\mathbb{E}[\, h(\mathbf{F}(\mathbf{x}, \mathbf{k}(\mathbf{x}), \mathbf{d})) \mid \mathbf{x} \,] \geq \alpha h(\mathbf{x}) + c, \tag{6.13}$$

for all $\mathbf{x} \in \mathbb{R}^{n_x}$, with $\mathbf{d} \sim \mathcal{D}$. For any $K \in \mathbb{N}$ and $\gamma \in \mathbb{R}_{\geq 0}$, if $c < -\gamma(1-\alpha)$, we have that:

$$P_u \leq \left(\frac{N - h(\mathbf{x}_0)}{B + \gamma}\right) \alpha^K + \frac{M(1-\alpha) - c}{B + \gamma} \sum_{i=1}^{K} \alpha^{i-1}. \tag{6.14}$$

Alternatively if $c \geq -\gamma(1-\alpha)$, then:

$$P_u \leq 1 - \frac{h(\mathbf{x}_0) + \gamma}{B + \gamma} \left(\frac{B\alpha + \gamma + c}{B + \gamma}\right)^K. \tag{6.15}$$

Remark 6.12. The upper bound $c \leq B(1-\alpha)$ is relatively non-restrictive, as not only is c typically negative, but it must hold such that, in expectation, $h(\mathbf{x}_{k+1})$ cannot rise above the upper bound B on h. The switching condition between (6.14) and (6.15) of $c = \gamma(1-\alpha)$ corresponds to whether, in expectation, the one-step evolution of the system remains in the set $\mathcal{C}_\gamma = \{\mathbf{x} \in \mathbb{R}^{n_x} \mid h(\mathbf{x}) \geq -\gamma\}$ when it begins on the boundary of $\mathcal{C}_\gamma$.

The full proof of Theorem 6.11 is provided in my publication [57, Thm. 5]. For simplicity, this thesis presents a simplified version of this theorem and proof which have a slightly more limited application, but use a similar structure and which I believe are more useful for developing intuition regarding the proof method.

For this simplification of Theorem 6.11, consider the case where $h(\mathbf{x}_k)$ is upper bounded by $B \in \mathbb{R}_{>0}$ and satisfies one of the following expectation conditions:

$$\mathbb{E}[\, h(\mathbf{F}(\mathbf{x}_k, \mathbf{k}(\mathbf{x}_k), \mathbf{d}_k)) \mid \mathscr{F}_k \,] \geq \alpha h(\mathbf{x}_k), \tag{6.16}$$

$$\mathbb{E}[\, h(\mathbf{F}(\mathbf{x}_k, \mathbf{k}(\mathbf{x}_k), \mathbf{d}_k)) \mid \mathscr{F}_k \,] \geq h(\mathbf{x}_k) - c, \tag{6.17}$$

for some $\alpha \in (0,1)$ or $c \geq 0$, where an expectation-based version of the DCBF condition (6.3) for stochastic dynamics. This results in the following probability bound[3]:

Corollary 6.13 (Simplified DCBF Safety using Ville's-inequality[4], [35], [57], [131], [192])**.** *If, for some $B > 0$ and $K \in \mathbb{N}_1$, the function $h : \mathbb{R}^{n_x} \to \mathbb{R}$ satisfies:*

$$h(\mathbf{x}) \leq B, \quad \text{for all } \mathbf{x} \in \mathbb{R}^{n_x}, \tag{6.18}$$

$$\text{then:} \qquad P_u(K, \mathbf{x}_0) \leq 1 - \frac{\lambda}{B}, \tag{6.19}$$

$$\text{where } \lambda = \begin{cases} \alpha^K h(\mathbf{x}_0), & \text{if (6.7) satisfies (6.16) } \forall k \leq K \\ h(\mathbf{x}_0) - cK, & \text{if (6.7) satisfies (6.17) } \forall k \leq K. \end{cases}$$

Proof. We prove the two cases separately:

Case 1:

We first prove the case when the constraint (6.16) is satisfied. Let $W_k \triangleq B\alpha^{-K} - \alpha^{-k} h(\mathbf{x}_k)$. This is a nonnegative supermartingale for $k \leq K$:

$$W_k = \alpha^{-K} B - \alpha^{-k} h(\mathbf{x}_k) \geq \alpha^{-k}(B - h(\mathbf{x}_k)) \geq 0 \tag{6.20}$$

$$\mathbb{E}[W_{k+1} | \mathscr{F}_k] = \alpha^{-K} B - \alpha^{-(k+1)} \mathbb{E}[h(\mathbf{x}_{k+1}) | \mathscr{F}_k] \tag{6.21}$$

$$\leq \alpha^{-K} B - \alpha^{-k} h(\mathbf{x}_k) = W_k. \tag{6.22}$$

Apply Ville's inequality (Lem. (6.9)) to W_k to find:

$$\mathbb{P}\left\{ \max_{k \leq K} W_k > \lambda \right\} \leq \frac{\mathbb{E}[W_0]}{\lambda}. \tag{6.23}$$

Next note that the implication:

$$\exists k \leq K \text{ s.t. } h(\mathbf{x}_k) < 0 \implies \exists k \leq K \text{ s.t. } W_k > \alpha^{-K} B \tag{6.24}$$

ensures that $P_u(K, \mathbf{x}_0) \leq \mathbb{P}\left\{ \max_{k \leq K} W_k > \alpha^{-K} \right\}$. Choose $\lambda = \alpha^{-K} B$ to achieve:

$$P_u(K, \mathbf{x}_0) \leq \frac{\alpha^{-K} B - h(\mathbf{x}_0)}{\alpha^{-K} B} = 1 - \frac{h(\mathbf{x}_0)}{B} \alpha^K. \tag{6.25}$$

[3] We note that the result in Corollary 6.13 is a special case of Theorem 6.11.

[4] See [58, Appx. C] for a discussion of notational differences between this presentation of Cor. 6.13 and that in [195] and [192].

Case 2: Next we prove the case when (6.17) is satisfied. Let $W_k^c \triangleq B - h(\mathbf{x}_k) + (K-k)c$. This is a non-negative supermartingale for $k \leq K$:

$$W_k^c = B - h(\mathbf{x}_k) + (K-k)c \geq 0, \tag{6.26}$$

$$\mathbb{E}[W_{k+1}^c \mid \mathscr{F}_k] = B - \mathbb{E}[h(\mathbf{x}_{k+1}) \mid \mathscr{F}_k] + (K-k-1)c, \tag{6.27}$$

$$\leq B - h(\mathbf{x}_k) + c + (K-k-1)c, \tag{6.28}$$

$$= B - h(\mathbf{x}_k) + (K-k)c = W_k^c. \tag{6.29}$$

Apply Ville's inequality (6.9) to W_k^c to find:

$$\mathbb{P}\left\{\max_{k\leq K} W_k^c > \lambda\right\} \leq \frac{\mathbb{E}[W_0^c]}{\lambda}. \tag{6.30}$$

Next note that the implication:

$$\exists k \leq K \text{ s.t. } h(\mathbf{x}_k) < 0 \implies \exists k \leq K \text{ s.t. } W_k^c > B \tag{6.31}$$

ensures that $P_u(K, \mathbf{x}_0) \leq \mathbb{P}\{\max_{k\leq K} W_k^c > \lambda\}$. Choose $\lambda = M$ to achieve:

$$P_u(K, \mathbf{x}_0) \leq \frac{B - h(\mathbf{x}_0) + Kc}{B} = 1 - \frac{h(\mathbf{x}_0) - Kc}{B}. \tag{6.32}$$

□

This theorem and corollary summarize the results of several works, namely [35], [57], [131], [192], in the context of DCBFs. They guarantee that the risk of the becoming unsafe is upper bounded by a function which decays to 1 with time and which depends on the system's initial safety "fraction," $h(\mathbf{x}_0)/B$ where safety increases with α and decreases with c.

Practical Considerations for Enforcing Stochastic DCBFs

Theorem 6.11 allows us to reason about the finite-time safety probabilities of systems governed by DCBFs. To utilize the results of this theorem in a control setting, we aim to use DCBFs to synthesize which enforce the expectation condition:

$$\mathbb{E}[h(\mathbf{F}_d(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k)) \mid \mathbf{x}_k] \geq \alpha h(\mathbf{x}_k). \tag{6.33}$$

If such a condition can be enforced, then the result of Theorem 6.11 can be directly applied to provide probabilistic bounds on the system's safety. However, since the composition of system dynamics with the disturbance in (6.2) may make computing

this expectation difficult, we instead focus on systems with additive disturbances of the form:

$$\mathbf{x}_k = \mathbf{F}_a(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{d}_k, \tag{6.34}$$

where $\mathbf{d}_k$ takes values in $\mathbb{R}^{n_x}$ and the expectation-based DCBF condition (6.16) for Theorem 6.11 becomes,

$$\mathbb{E}[h(\mathbf{F}_a(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{d}_k) \mid \mathbf{x}_k] =\geq \alpha h(\mathbf{x}_k). \tag{6.35}$$

Like the deterministic DCBF controller in (6.4), we apply this constraint in an optimization-based controller that enforces safety while achieving pointwise minimal deviation from a desired nominal controller $\mathbf{k}_{\text{des}} : \mathbb{R}^{n_x} \times \mathbb{N}_0 \to \mathbb{R}^{n_u}$ in the form of an Expectation-based DCBF (6.36) controller:

$$\begin{aligned} \mathbf{k}_{\text{ED}}(\mathbf{x}_k) = \underset{\mathbf{u}\in\mathbb{R}^{n_u}}{\operatorname{argmin}} \quad & \|\mathbf{u} - \mathbf{k}_{\text{des}}(\mathbf{x}_k, k)\|^2 \\ \text{s.t.} \quad & \mathbb{E}[h(\mathbf{F}_a(\mathbf{x}_k, \mathbf{u}) + \mathbf{d}_k) \mid \mathbf{x}_k] \geq \alpha h(\mathbf{x}_k). \end{aligned} \tag{6.36}$$

The expectation in (6.36) adds complexity that is not generally considered in the application of deterministic DCBFs. More commonly, CBF-based controllers solve "certainty-equivalent" optimization programs, like this Certainty-Equivalent DCBF CED controller (6.37) , that replaces the expected barrier value $\mathbb{E}[h(\mathbf{x}_{k+1}) \mid \mathbf{x}_k]$ with the barrier evaluated at the expected next state, $h(\mathbb{E}[\, \mathbf{x}_{k+1} \mid \mathbf{x}_k \,])$:

$$\begin{aligned} \mathbf{k}_{\text{CED}}(\mathbf{x}_k) = \underset{\mathbf{u}\in\mathbb{R}^{n_u}}{\operatorname{argmin}} \quad & \|\mathbf{u} - \mathbf{k}_{\text{nom}}(\mathbf{x}_k, k)\|^2 \\ \text{s.t.} \quad & h(\mathbf{F}_a(\mathbf{x}_k, \mathbf{u}) + \mathbb{E}[\mathbf{d}_k]) \geq \alpha h(\mathbf{x}_k), \end{aligned} \tag{6.37}$$

where $\mathbb{E}[\mathbf{F}_a(\mathbf{x}_k, \mathbf{u}_k)|\mathbf{x}_k] = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k)$ and $\mathbb{E}[\mathbf{d}_k|\mathbf{x}_k] = \mathbb{E}[\mathbf{d}_k]$. This constraint is often easier to evaluate than (6.33) since it allows control actions to be selected with respect to the expected disturbance $\mathbb{E}[\mathbf{d}_k]$ without needing to model the full disturbance distribution $\mathcal{D}$. If the disturbance is zero-mean, then this form of the constraint is implicitly enforced by DCBF controllers such as those presented in [46], [197]. However, when replacing the 6.36 controller with (6.37) it is important to consider the effect of Jensen's inequality in Theorem 6.10.

If the "certainty-equivalent" constraint in (6.37) is strictly concave[5], then we can apply the results of Theorem 6.11 directly since Jensen's inequality tightens the constraint and ensures satisfaction of the expectation condition (6.13). Unfortunately,

[5]The constraint $h(\mathbf{x}_k + \mathbf{u}) \geq \alpha h(\mathbf{x}_k)$ is concave in $\mathbf{u}$ when h is convex and it is convex in $\mathbf{u}$ when h is concave.

using such a controller is a non-convex optimization program which can be impractical to solve at real-time speeds. If, instead, the constraint is convex, then (6.37) is a convex program, but does not necessarily enforce the expectation condition (6.13) in Theorem (6.11) due to the gap introduced by Jensen's inequality. While the impact of this gap on safety can be analyzed using the additive c term in Thm. 6.11, we instead seek to directly compensate for Jensen's inequality.

In order to apply the results of Theorem 6.11 to controllers of the form (6.37) with convex constraints, we must first provide a bound on the gap introduced by Jensen's inequality. In particular, for any concave function $h : \mathbb{R}^{n_x} \to \mathbb{R}$ and random variable $\mathbf{d} \sim \mathcal{D}$, we seek to determine a value $\psi_{\text{J}} \in \mathbb{R}_{\geq 0}$ such that, for all $\mathbf{x} \in \mathbb{R}^{n_x}$ and $\mathbf{u} \in \mathbb{R}^{n_u}$:

$$\mathbb{E}[h(\mathbf{F}_a(\mathbf{x}, \mathbf{u}) + \mathbf{d}) \mid \mathbf{x}] \geq h(\mathbf{F}_a(\mathbf{x}, \mathbf{u}) + \mathbb{E}[\mathbf{d}]) - \psi_{\text{J}}, \tag{6.38}$$

thus quantifying the gap introduced by Jensen's inequality (Thm. 6.10).

A large body of work has studied methods for finding the smallest possible ψ_{J} that satisfies (6.38). Here we adapt a result in [209] to achieve a relatively loose, but straightforward bound:

Lemma 6.14. *Consider a twice-continuously differentiable, concave function* $h : \mathbb{R}^{n_x} \to \mathbb{R}$ *with* $\sup_{\mathbf{x} \in \mathbb{R}^{n_x}} \|\nabla^2 h(\mathbf{x})\|_2 \leq \lambda_{\max}$ *for some* $\lambda_{\max} \in \mathbb{R}_{\geq 0}$*, and a random variable* $\mathbf{x}$ *that takes values in* $\mathbb{R}^{n_x}$ *with* $\mathbb{E}[\|\mathbf{x}\|] < \infty$ *and* $\|\text{cov}(\mathbf{x})\| < \infty$*. Then we have that:*

$$\mathbb{E}[h(\mathbf{x})] \geq h(\mathbb{E}[\mathbf{x}]) - \frac{\lambda_{\max}}{2}\text{tr}(\text{cov}(\mathbf{x})). \tag{6.39}$$

The proof can be found in [57, Appx. B]. We note that although this value of $\psi_{\text{J}} = \frac{\lambda_{\max}}{2}\text{tr}(\text{cov}(\mathbf{x}))$ is easy to interpret, tighter bounds exist which have less restrictive assumptions than a globally bounded Hessian [208]. We also note that one could also use sampling-based methods to approximately satisfy the constraint (6.38) by estimating ψ_{J} empirically.

Next, we present a controller which combines the mean-based control of the "certainty equivalent" (6.37) while also accounting for Jensen's inequality. This Jensen-Enhanced DCBF Controller (JED) includes an additional control parameter $c_{\text{J}} \geq 0$ to account for Jensen's inequality:

$$\begin{aligned} \mathbf{k}_{\text{JED}}(\mathbf{x}_k) = \underset{\mathbf{u} \in \mathbb{R}^{n_u}}{\text{argmin}} \quad & \|\mathbf{u} - \mathbf{k}_{\text{nom}}(\mathbf{x}_k, k)\|^2 \\ \text{s.t.} \quad & h(\mathbf{F}_a(\mathbf{x}_k, \mathbf{u}_k) + \mathbb{E}[\mathbf{d}_k]) - c_{\text{J}} \geq \alpha h(\mathbf{x}_k). \end{aligned} \tag{6.40}$$

Given this controller and a method for bounding ψ_{J}, we can now apply Theorem 6.11 while accounting for (or analyzing) the effects of Jensen's inequality on the 6.40 controller:

Theorem 6.15. *Consider the system* (6.34) *and let* $h : \mathbb{R}^{n_x} \to \mathbb{R}$ *be a twice-continuously differentiable, concave function such that* $\sup_{\mathbf{x}\in\mathbb{R}^{n_x}} h(\mathbf{x}) \leq B$ *for* $B \in \mathbb{R}_{>0}$ *and* $\sup_{\mathbf{x}\in\mathbb{R}^{n_x}} \|\nabla^2 h(\mathbf{x})\|_2 \leq \lambda_{\max}$ *for* $\lambda_{\max} \in \mathbb{R}_{\geq 0}$. *Suppose there exists an* $\alpha \in (0,1)$ *and a* $c_{\text{J}} \in [0, \frac{\lambda_{\max}}{2}\text{tr}(\text{cov}(\mathbf{d})) + B(1-\alpha)]$ *such that:*

$$h(\mathbf{F}_a(\mathbf{x}, \mathbf{k}(\mathbf{x})) + \mathbb{E}[\mathbf{d}]) - c_{\text{J}} \geq \alpha h(\mathbf{x}), \tag{6.41}$$

for all $\mathbf{x} \in \mathbb{R}^{n_x}$ *with* $\mathbf{d} \sim \mathcal{D}$. *Then we have that:*

$$\mathbb{E}[\, h(\mathbf{F}_a(\mathbf{x}, \mathbf{k}(\mathbf{x})) + \mathbf{d}) \mid \mathbf{x}\,] \geq \alpha h(\mathbf{x}) + c, \tag{6.42}$$

for all $\mathbf{x} \in \mathbb{R}^{n_x}$ *with* $\mathbf{d} \sim \mathcal{D}$ *and* $c = c_{\text{J}} - \frac{\lambda_{\max}}{2}\text{tr}(\text{cov}(\mathbf{d}_k))$.

Proof. Given $\mathbf{x} \in \mathbb{R}^{n_x}$, Lemma 6.14 ensures that:

$$0 \leq h(\mathbf{F}_a(\mathbf{x}, \mathbf{k}(\mathbf{x})) + \mathbb{E}[\mathbf{d}]) - c_{\text{J}} - \alpha h(\mathbf{x}) \tag{6.43}$$

$$\leq \mathbb{E}[h(\mathbf{F}_a(\mathbf{x}, \mathbf{k}(\mathbf{x})) + \mathbf{d}) \mid \mathbf{x}] + \psi_{\text{J}} - c_{\text{J}} - \alpha h(\mathbf{x}) \tag{6.44}$$

where $\psi_{\text{J}} = \frac{\lambda_{\max}}{2}\text{tr}(\text{cov}(\mathbf{d}))$. Letting $\delta = c_{\text{J}} - \frac{\lambda_{\max}}{2}\text{tr}(\text{cov}(\mathbf{d}))$ yields the desired result. □

Thus, the JED controller compensates for Jensen's inequality using the c_{J} and ensures that the results of Theorem 6.11 apply, resulting in rigorous probabilistic guarantees of safety for stochastic systems (6.34) controlled by the JED controller.

Example Simulations

Next, to demonstrate the utility of this approach, we consider a variety of simulation examples that highlight the key features of our approach.

We begin by with a simple example using a linear 1D system:

Example 6.16 (Linear 1D System)**.** *Here we analyze our bounds by considering the case of unbounded i.i.d. disturbances* $d_k \sim \mathcal{N}(0,1)$ *for the one dimensional system* $(x, u, \in \mathbb{R})$ *and safe set:*

$$x_{k+1} = x_k + 2 + u_k + \sigma d_k, \;\; \mathcal{C} = \{x \mid 1 - x^2 \geq 0\}. \tag{6.45}$$

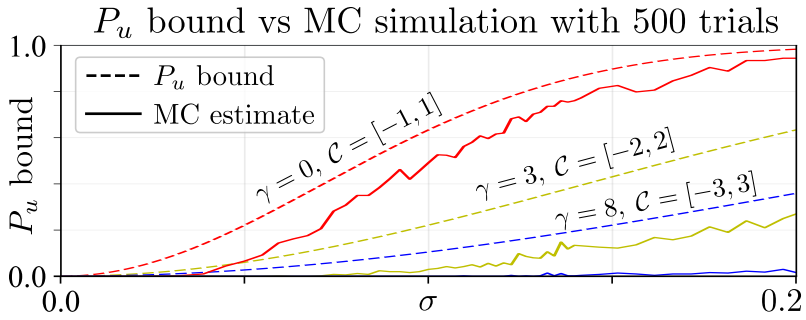


Figure 6.1. The dashed lines represent the theoretical probability bounds for the system as in Theorem 6.11. The solid lines represent the Monte Carlo (MC) estimated P_u across 500 experiments.

The Jensen gap for this system and DCBF is bounded by $\psi_{\text{J}} = \sigma^2$. For simulation, we employ the 6.40 controller with $c_{\text{J}} = \sigma^2$, $\alpha = 1 - \sigma^2$, and nominal controller $\mathbf{k}_{nom}(\mathbf{x}_k, k) = 0$. Figure 6.1 shows the results of 500 one second long trials run with a variety of $\sigma \in [0, 0.2]$ and also displays how the bound on P_u decreases as γ increases.

Next, we consider an inverted pendulum about its upright equilibrium point with the discrete dynamics:

Example 6.17 (Simple Pendulum)**.** *Consider the dynamics:*

$$\begin{bmatrix} \theta_{k+1} \\ \dot{\theta}_{k+1} \end{bmatrix} = \begin{bmatrix} \theta_k + \Delta t \dot{\theta}_k \\ \dot{\theta}_k + \Delta t \sin(\theta_k) \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta t \mathbf{u} \end{bmatrix} + \mathbf{d}_k, \tag{6.46}$$

with time step $\Delta_t = 0.01$ sec, i.i.d disturbances $\mathbf{d}_k \sim \mathcal{N}(\mathbf{0}_2, \text{Diag}([0.005^2, 0.025^2]])$, and[6] safe set:

$$\mathcal{C} = \left\{ \mathbf{x} \in \mathbb{R}^{n_x} \;\middle|\; \underbrace{1 - \frac{6^2}{\pi^2} \mathbf{x}^\top \begin{bmatrix} 1 & 3^{-\frac{1}{2}} \\ 3^{-\frac{1}{2}} & 1 \end{bmatrix} \mathbf{x}}_{h_{pend}(\mathbf{x})} \geq 0 \right\} \tag{6.47}$$

which is constructed using the continuous-time Lyapunov equation as in [203] and for which $|\theta| \leq \pi/6$ for all $\mathbf{x} \in \mathcal{C}$. Figure 6.2 shows the results of 500 one second long trials for each $\mathbf{x}_0 \in \mathcal{C}$ using the 6.40 controller with parameters $\alpha = 1 - \psi_{\text{J}}, c_{\text{J}} = \psi_{\text{J}}$, where $\psi_{\text{J}} = \frac{\lambda_{max}}{2} tr(cov(\mathbf{d}_k))$. This figure highlights the influence of $\mathbf{x}_0$ and shows how the bound on P_u increases as $h(\mathbf{x}_0)$ decreases.

We also consider the problem of controlling a planar system with unit-mass double-integrator dynamics to remain inside a convex polytope (in particular, a unit square centered at the origin).

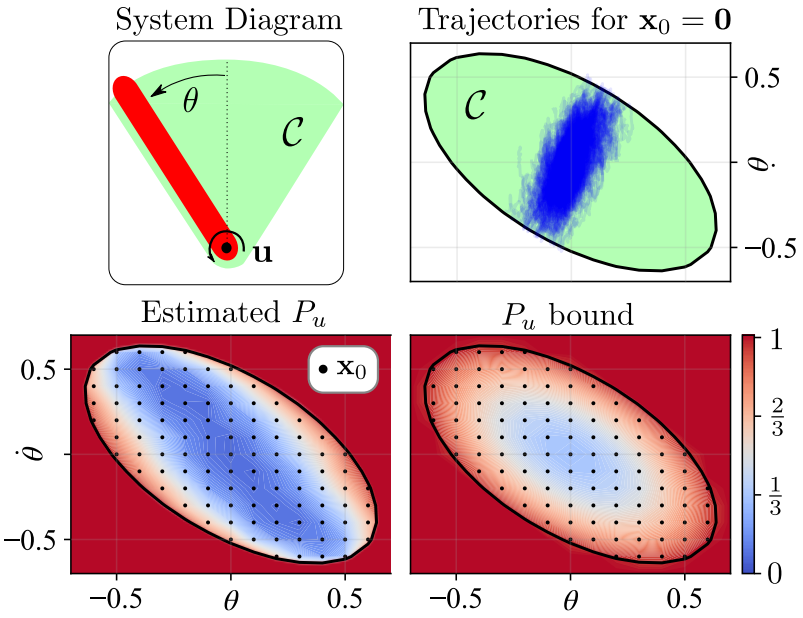


Figure 6.2. Stochastic pendulum example. **(Top Left)** System diagram of the inverted pendulum. **(Top Right)** 500 one second long example trajectories starting at $\mathbf{x}_0 = 0$. **(Bottom Left)** Monte Carlo estimates of P_u for $\gamma = 0$ using 500 one second long trials for each initial conditions represented by a black dot. **(Bottom Right)** Our (conservative) theoretical bounds on P_u from Theorem 6.11

Example 6.18 (Planar Double Integrator)**.** *Consider the discrete-time dynamics:*

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{I}_2 & \Delta t\, \mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \frac{\Delta t^2}{2}\mathbf{I}_2 \\ \Delta t \mathbf{I}_2 \end{bmatrix} \mathbf{u}_k + \mathbf{d}_k, \tag{6.48}$$

$$\triangleq \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{d}_k, \tag{6.49}$$

where Δt is the integration time step and $\mathbf{d}_k \sim \mathcal{N}(\mathbf{0}_4, \mathbf{Q})$ is a zero-mean Gaussian process noise added to the dynamics. Here we use $\Delta t = 0.05$ sec, and $\mathbf{Q} = \mathbf{B}\mathbf{B}^T$, which corresponds to applying a disturbance force $\mathbf{f}_k \sim \mathcal{N}(0, \mathbf{I}_2)$ to the system at each timestep.

To keep the system inside a convex polytope, we seek to enforce the affine inequalities $\mathbf{C}\mathbf{x} \leq \mathbf{w}$ for $\mathbf{C} \in \mathbb{R}^{n_c \times n_x}, \mathbf{w} \in \mathbb{R}^{n_c}$. Thus, we define our barrier $h(\mathbf{x}) = -\max(\mathbf{C}\mathbf{x} - \mathbf{w})$, where $\max(\cdot)$ defines the element-wise maximum, and $h(\mathbf{x}) \geq 0$ if and only if the constraint $\mathbf{C}\mathbf{x} \leq \mathbf{w}$ holds. Implementing the 6.36 controller for this system is non-trivial, since the expectation of $h(\mathbf{x})$ for a Gaussian-distributed $\mathbf{x}$ does not have a closed form. Similarly, implementing the 6.40 controller to account for Jensen's inequality is non-trivial since h is not twice continuously differentiable. We instead choose to enforce a conservative approximation of the barrier condition (6.33) *using the log-sum-exp function. [57, Appx.*

[6]Diag: $\mathbb{R}^{n_x} \to \mathbb{R}^{n \times n}$ generates a square diagonal matrix with its argument along the main diagonal.

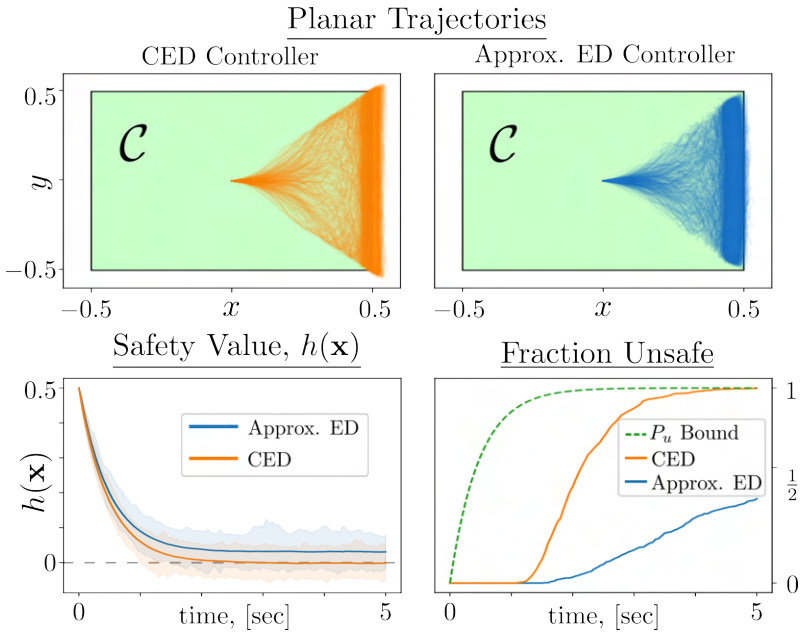


Figure 6.3. Simulation results for double integrator over 500 trials. **(Top left):** Planar (x, y) trajectories for the approximated 6.36 controller, with the safe set (a unit square) plotted in green. **(Top right):** Planar (x, y) trajectories for a CED controller (6.37) . **(Bottom left):** The $h(\mathbf{x}_k)$ for both controllers, with the max and min values shaded. **(Bottom right):** Percent of trajectories that have remained safe over time. We also plot our (conservative) bound (6.15) on the unsafe probability P_u.

C] shows how this approximation yields an analytic lower bound (derived using the moment-generating function of Gaussian r.v.s) on $\mathbb{E}[h(\mathbf{x}_{k+1})]$ *which can be imposed via a convex constraint.*

Figure 6.3 plots the results of 500 simulated trajectories for the double integrator system using the proposed 6.36 controller, and the certainty equivalent CED controller (6.37) *that neglects the presence of process noise. Both controllers have a nominal controller* $\mathbf{k}_{\text{des}}(\mathbf{x}) = [50, 0]$ *which seeks to drive the system into the right wall. All trajectories start from the origin. We note the proposed controller is indeed more conservative than the CED controller* (6.37) *, yielding both fewer and smaller violations of the safe set. In the bottom right, we also plot our bound as a function of the time horizon, which we note is quite conservative compared to our Monte Carlo estimate of the safety probability, motivating future work.*

Finally, we consider the problem of controlling a simulated quadrupedal robot locomoting along a narrow path.

Example 6.19 (Simulated Quadruped)**.** *The simulation is based on a Unitree A1 robot as shown in Figure 6.4 which has 18 degrees of freedom and 12 actuators. with*

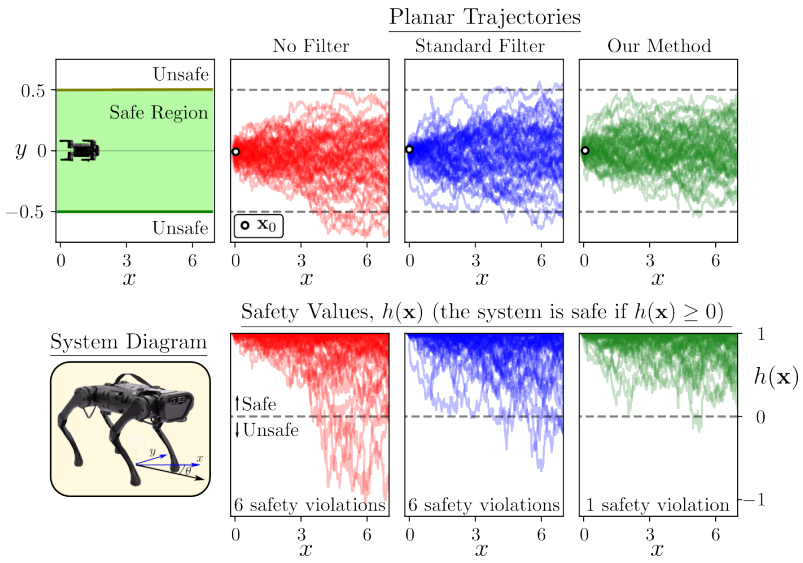


Figure 6.4. Safety of a simulated quadrupedal robot locomoting on a narrow path for a variety of controllers. **(Top Left)** The safe region that the quadruped is allowed to traverse. **(Bottom Left)** A system diagram depicting the states of the quadruped $\begin{bmatrix} x, y, \theta \end{bmatrix}^\top$. **(Top Right)** 50 trajectories for 3 controllers: one without any knowledge of safety ($\mathbf{k}_{\text{nom}}$), one with a standard DTCBF safety filter (6.4), and finally our method which accounts for stochasticity (6.40). **(Bottom Right)** Plots of $h(\mathbf{x})$, a scalar value representing safety. The system is safe (i.e., in the green safe region) if $h(\mathbf{x}) \geq 0$.

configuration space coordinates $\mathbf{q} \in \mathbb{R}^{18}$*, the full state is given by* $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{36}$*. For simulated walking, a no-slip condition,* $\mathbf{c}(\mathbf{q}) = \mathbf{0} \in \mathbb{R}^{n_c}$*, is enforced on the feet where* n_c *depends on the number of feet in contact with the ground. As discussed in [96], when* $\mathbf{c}(\mathbf{q})$ *is differentiated twice, D'Alembert's principle applied to the constrained Euler-Lagrange equations yields the robotic system dynamics:*

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}\mathbf{u} + \mathbf{J}(\mathbf{q})^\top \boldsymbol{\lambda}, \tag{6.50}$$

$$\mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = 0, \tag{6.51}$$

where $\mathbf{D}(\mathbf{q}) \in \mathbb{R}^{18\times 18}$ *is the mass-inertia matrix,* $\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{18}$ *contains the Coriolis and gravity terms,* $\mathbf{B} \in \mathbb{R}^{18\times 12}$ *is the actuation matrix,* $\mathbf{J}(\mathbf{q}) = \partial \mathbf{c}(\mathbf{q})/\partial \mathbf{q} \in \mathbb{R}^{c\times 18}$ *is the Jacobian of the holonomic constraints, and* $\boldsymbol{\lambda} \in \mathbb{R}^c$ *is the constraint wrench. These full-system dynamics including ground contacts were used in our simulations.*

In order to represent the error caused by uncertain terrain, zero mean Gaussian disturbances are added to the quadruped's (x, y) *body position and velocity at 1kHz with variances of* 2.25×10^{-6} *and* 0.01*, respectively. This noise was chosen to qualitatively match the rough-terrain walking that we have observed in experiments;*

a video comparing our simulated walking to rough-terrain walking can be found at [210].

For joint-level torque control, an ID-QP controller designed using concepts in [95] and implemented at 1kHz is used with dynamics (6.50, 6.51) to track center-of-mass velocities and angle rates with swing legs following a Reibert-style trajectory in a diagonal walking gait using the motion primitive framework in [96]. We simulate the entire quadruped's dynamics (6.50, 6.51), but follow a similar reduced-order-modeling methodology to [48] and consider the following simplified discrete-time single-integrator system for DCBF-based control:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \begin{bmatrix} \cos\theta_k & -\sin\theta_k & 0 \\ \sin\theta_k & \cos\theta_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_k^x \\ v_k^y \\ \theta_k \end{bmatrix} + \mathbf{d}_k. \tag{6.52}$$

where $\mathbf{x}_k = \begin{bmatrix} x_k, & y_k, & \theta_k \end{bmatrix}^\top$ *and* $\Delta t = 0.05$ *seconds. Here the i.i.d. random process* $\mathbf{d}_k$ *models the random disturbances introduced to the planar* (x, y) *position and velocity as well as the dynamics-mismatch between the full-order quadrupedal dynamics (6.50, 6.51) and the simplified model* (6.52)*.*

Using the motion primitive framework presented in [96], the quadruped is commanded to stand and then traverse a 7 meter path that is 1 meter wide, with the safe set $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^{n_x} \mid 0.5^2 - y^2 \geq 0\}$*. For this simulation, three controllers are compared: a simple nominal controller* $\mathbf{k}_{nom}(\mathbf{x}) = \begin{bmatrix} 0.2, & 0, & -\theta \end{bmatrix}^\top$ *with no understanding of safety, the nominal deterministic DCBF controller* (6.4) *with* $\alpha = 0.99$*, and our proposed JED controller* (6.40) *with* $\alpha = 0.99$ *and* $c_{\text{J}} = \psi_{\text{J}}$ *using the mean and covariance estimates,* $\mathbb{E}[\mathbf{d}_k] \approx \begin{bmatrix} -0.0132, & -0.0034, & -0.0002 \end{bmatrix}^\top$ *and tr(cov*$(\mathbf{d}_k)) \approx \psi_{\text{J}} = 0.000548$*, which were estimated using 15 minutes of 20 Hz walking data controlled by* $\mathbf{k}_{\text{des}}$ *and which characterize the effect of both the planar disturbances and the model-mismatch between (6.50, 6.51) and* (6.52)*.*

The results of 50 trials for each controller can be seen in Figure 6.4. As expected, $\mathbf{k}_{\text{des}}$ *generated the largest safety violations and JED* (6.40) *the smallest and fewest safety violations.*

Conclusion

In this section, we developed a bound for the finite-time safety of stochastic discrete-time systems by connecting DCBFs to existing work in martingale and stochastic process theory. Additionally, we presented a method for practically implementing

convex optimization-based controllers that satisfy this bound by accounting for or analyzing the effect of Jensen's inequality. We presented several examples which demonstrate the efficacy of our bound and our proposed 6.36 and JED 6.40 controllers. This theoretical lens, and the CED controller in particular, also presents a useful perspective through which to view the inherent probabilistic robustness properties of all DCBF methods, even when stochastic uncertainty is not explicitly accounted for.

6.4 Tightened Stochastic Safety Guarantees

The Ville's based bound of the previous section is notably weak as can be seen in Figures 6.1 and 6.3. This is in part due to the generality of the guarantee which used a worst-case bound on the Hessian of h and very little information regarding the disturbance distribution. This section extends the work of the previous section on martingale-based safety guarantees by replacing Ville's inequality (Lem. 6.9) with an alternative concentration inequality (Freedman's inequality, Lem. 6.23) to obtain tighter safety probability bounds. By additionally assuming that the martingale differences and predictable quadratic variation are bounded, this inequality relaxes the nonnegativity assumption required by Ville's inequality while also providing generally tighter bounds. Since a worst-case bound is assumed, the results of this section also have a direct relationship to the worst-case bound results of ISSf (Sec. 2.3) and we find that the results of this section provide additional risk-based nuance in the context of these worst-case bounds.

The contributions of this section are as follows:

- Novel Freedman-based safety probabilities for DCBFs and c-martingales.
- A characterization of the range of parameter values for which the Freedman-based safety bound is tighter than existing discrete-time martingale-based safety results.
- A comparison of the Freedman-based bound with traditional ISSf safety.
- An application of our theoretical result to a simulated bipedal obstacle avoidance scenario (Fig. 6.7), using a reduced-order model of the step-to-step dynamics.

The text for this section is adapted from:

R. K. Cosner, P. Culbertson, and A. D. Ames, "Bounding stochastic safety: Leveraging freedman's inequality with discrete-time control barrier functions," *IEEE Control Systems Letters*, pp. 193 7 –1942, 2024. DOI: 10.1109/LCSYS.2024.3409105,

Safety Guarantees using Freedman's Inequality

Here we present K-step exit probability bounds for DCBFs and c-martingales generated using Freedman's inequality[7], a particularly strong martingale concentration inequality. After presenting this result, this section explores comparisons with the previously introduced Ville's-based safety and ISSf methods.

Before presenting Freedman's inequality, we must define the predictable quadratic variation (PQV) of a process which is a generalization of variance for stochastic processes.

Definition 6.20 (Predictable Quadratic Variation (PQV) [206])**.** *The PQV of a martingale W_k at $K \in \mathbb{N}_1$ is:*

$$\langle W \rangle_K \triangleq \sum_{i=1}^{K} \mathbb{E}[(W_i - W_{i-1})^2 \mid \mathscr{F}_{i-1}]. \tag{6.53}$$

Unlike Ville's inequality, Freedman's inequality does not require nonnegativity of the martingale W_k, thus removing the upper-bound requirement (6.18) on h. In place of nonnegativity, we require two alternative assumptions:

Assumption 6.21 (Upper-Bounded Differences)**.** *We assume that the martingale differences are upper-bounded by 1 (i.e., $W_{k+1} - W_k \leq 1$, similar to Azuma-Hoeffding methods [206]).*

Assumption 6.22 (Bounded PQV)**.** *We assume that the PQV is upper-bounded by $\xi^2 \in \mathbb{R}_{>0}$.*

Given the PQV of the process, Freedman's inequality provides the following bound:

Lemma 6.23 (Freedman's Inequality [202, Thm. 4.1])**.** *If, for some $K \in \mathbb{N}_1$ and $\xi > 0$, W_k is a supermartingale with $W_0 = 0$ such that:*

$$(W_k - W_{k-1}) \leq 1 \quad \text{for all } k \leq K, \tag{Assumption 6.21}$$

$$\langle W \rangle_K \leq \xi^2, \tag{Assumption 6.22}$$

[7] For this we use the simpler historical version as presented by Freedman [202]; see [211] for historical context and a new, tighter alternative which could also be used.

then, for any $\lambda \geq 0$*,*

$$\mathbb{P}\left\{\max_{k\leq K} W_k \geq \lambda\right\} \leq H(\lambda,\xi) \triangleq \left(\frac{\xi^2}{\lambda+\xi^2}\right)^{\lambda+\xi^2} e^{\lambda}. \tag{6.54}$$

See [207, Appx. D] for a restatement of the proof of this lemma.

We can now use Freedman's inequality to create probabilistic safety guarantees that reflects the structure of Corollary 6.13 for systems that satisfy the expectation-based DCBF (6.16) or c-martingale (6.17) conditions.

Theorem 6.24 (DCBF Safety using Freedman's Inequality)**.** *If, for some* $K \in \mathbb{N}_1, \sigma > 0$*, and* $\delta > 0$*, the following bounds*[8] *on the difference*[9] *between the true and predictable update* (6.55) *and the conditional variance* (6.56) *hold for all* $k \leq K$*:*

$$\mathbb{E}[\, h(\mathbf{x}_k) \mid \mathscr{F}_{k-1} \,] - h(\mathbf{x}_k) \leq \delta, \tag{6.55}$$

$$\text{Var}(\, h(\mathbf{x}_{k+1}) \mid \mathscr{F}_k \,) \leq \sigma^2, \tag{6.56}$$

then the K*-step exit probability is bounded as:*

$$P_u(K, \mathbf{x}_0) \leq H\left(\tfrac{\lambda}{\delta}, \tfrac{\sigma\sqrt{K}}{\delta}\right), \tag{6.57}$$

$$\text{where } \lambda = \begin{cases} \alpha^K h(\mathbf{x}_0), & \text{if (6.7) satisfies (6.16) } \forall k \leq K, \\ h(\mathbf{x}_0) - cK, & \text{if (6.7) satisfies (6.17) } \forall k \leq K. \end{cases}$$

To apply Freedman's inequality (Lem. 6.23) to achieve Theorem 6.24 we follow this proof structure: **(Step 1)** normalize h and use it to construct a candidate supermartingale W_k, **(Step 2)** verify that W_k is indeed a supermartingale with $W_0 = 0$, **(Step 3)** use Doob's decomposition [206, Thm 12.1.10] to produce a martingale M_k from W_k in order to remove the negative effect of safe, predictable jumps from the PQV, **(Step 4)** verify that M_k satisfies Assumptions 6.21 and 6.22, **(Step 5)** choose $\lambda \geq 0$ such that a safety failure implies $\{\max_{k\leq K} W_k \geq \lambda\}$ as in (6.54), and **(Step 6)** specialize to specific values of α and c for each case.

Proof. **(Step 1)** Consider the case, for $\tilde{\alpha} \in (0, 1]$ and $\tilde{c} \geq 0$, where

$$\mathbb{E}[h(\mathbf{x}_{k+1})|\mathscr{F}_k] \geq \tilde{\alpha} h(\mathbf{x}_k) - \tilde{c}, \quad \text{for all } k \leq K. \tag{6.58}$$

[8] Only upper-bounds on δ and σ^2 are required for (6.57) to hold and this guarantee is robust to changes in distribution that still satisfy (6.55) and (6.56). For real-world systems, distribution-learning will be explored in the next chapter.

[9] See [207, Appx. G] for a constructive method for determining δ and σ.

First, define the normalized safety function $\eta(\mathbf{x}) \triangleq \frac{h(\mathbf{x})}{\delta}$ to ensure that the martingale differences will be bounded by 1. Next, use η to define the candidate supermartingale[10]

$$W_k \triangleq -\tilde{\alpha}^{K-k}\eta(\mathbf{x}_k) + \tilde{\alpha}^K\eta(\mathbf{x}_0) - \textstyle\sum_{i=1}^{k} \tilde{\alpha}^{K-i}\frac{\tilde{c}}{\delta}. \tag{6.59}$$

(Step 2) This satisfies[11] $W_0 = 0$ and is a supermartingale:

$$\begin{aligned}\mathbb{E}[W_{k+1}|\mathscr{F}_k] &= -\tilde{\alpha}^{K-(k+1)}\mathbb{E}[\eta(\mathbf{x}_{k+1})|\mathscr{F}_k] + \tilde{\alpha}^K\eta(\mathbf{x}_0) - \textstyle\sum_{i=1}^{k+1} \tilde{\alpha}^{K-i}\frac{\tilde{c}}{\delta}, &(6.60)\\ &\leq -\tilde{\alpha}^{K-k}\eta(\mathbf{x}_k) + \tilde{\alpha}^K\eta(\mathbf{x}_0) - \textstyle\sum_{i=1}^{k} \tilde{\alpha}^{K-i}\frac{\tilde{c}}{\delta} = W_k, &(6.61)\end{aligned}$$

which can be seen by applying the bound from (6.58).

(Step 3) The martingale from Doob's decomposition is:

$$\begin{aligned}M_k &\triangleq W_k + \textstyle\sum_{i=1}^{k}(W_{i-1} - \mathbb{E}[W_i|\mathscr{F}_{i-1}]), &(6.62)\\ &= W_k + \textstyle\sum_{i=1}^{k} \underbrace{\frac{\tilde{\alpha}^{K-i}}{\delta}(\mathbb{E}[h(\mathbf{x}_i)|\mathscr{F}_{i-1}] - \tilde{\alpha}h(\mathbf{x}_{i-1}) + \tilde{c})}_{\geq 0} \geq W_k &(6.63)\end{aligned}$$

where the bound comes from (6.58) and positivity of $\tilde{\alpha}$ and δ.

(Step 4) Furthermore, we can show that M_k satisfies Assp. 6.21:

$$\begin{aligned}M_k - M_{k-1} &= W_k - \mathbb{E}[W_k|\mathscr{F}_{k-1}], &(6.64)\\ &= \tilde{\alpha}^{K-k}(\mathbb{E}[\eta(\mathbf{x}_k)|\mathscr{F}_{k-1}] - \eta(\mathbf{x}_k)) \leq \tilde{\alpha}^{K-k}\tfrac{\delta}{\delta} \leq 1, &(6.65)\end{aligned}$$

since we assume in (6.55) that $\mathbb{E}[h(\mathbf{x}_k) \mid \mathscr{F}_{k-1}] - h(\mathbf{x}_k) \leq \delta$.

Next, $\tilde{\alpha} \in (0, 1]$ and (6.56) ensure that M_k satsifes Assp. 6.22:

$$\begin{aligned}\langle M\rangle_K &= \textstyle\sum_{i=1}^{K} \mathbb{E}[\tilde{\alpha}^{2(K-i)}(\eta(\mathbf{x}_i) - \mathbb{E}[\eta(\mathbf{x}_i)|\mathscr{F}_{i-1}])^2|\mathscr{F}_{i-1}],\\ &= \textstyle\sum_{i=1}^{K} \frac{\tilde{\alpha}^{2(K-i)}}{\delta^2}\mathrm{Var}(h(\mathbf{x}_i)|\mathscr{F}_{i-1}) \leq \sum_{i=1}^{K} \tilde{\alpha}^{2(K-i)}\frac{\sigma^2}{\delta^2} \leq \frac{\sigma^2 K}{\delta^2}. &(6.66)\end{aligned}$$

(Step 5) Now, to relate the unsafe event $\{\min_{k\leq K} h(\mathbf{x}_k) < 0\}$ to our martingale M_k we consider the implications:

$$\begin{aligned}&\min\nolimits_{k\leq K} h(\mathbf{x}_k) < 0 \implies \min\nolimits_{k\leq K} h(\mathbf{x}_k) \leq 0 &(6.67)\\ \iff &\max_{k\leq K} -\tilde{\alpha}^{K-k}\eta(\mathbf{x}_k) \geq 0, \quad \text{since } \tilde{\alpha} > 0, \delta > 0 &(6.68)\\ \iff &\max_{k\leq K} W_k \geq \tilde{\alpha}^K\eta(\mathbf{x}_0) - \textstyle\sum_{i=1}^{k} \tilde{\alpha}^{K-i}\frac{\tilde{c}}{\delta} &(6.69)\\ \implies &\max_{k\leq K} M_k \geq \tilde{\alpha}^K\eta(\mathbf{x}_0) - \textstyle\sum_{i=1}^{k} \tilde{\alpha}^{K-i}\frac{\tilde{c}}{\delta} &(6.70)\\ \implies &\max_{k\leq K} M_k \geq \tilde{\alpha}^K\eta(\mathbf{x}_0) - \textstyle\sum_{i=1}^{K} \tilde{\alpha}^{K-i}\frac{\tilde{c}}{\delta}, &(6.71)\end{aligned}$$

[10] We use the "empty sum" convention that $\sum_{i=1}^{0} \rho = 0$ for any $\rho \in \mathbb{R}$.

[11] $W_0 = 0$ since $\mathbf{x}_0$ is known and randomness first enters through $\mathbf{d}_0$.

where (6.68) is due to multiplication by a value strictly less than zero, (6.69) is due to adding zero, (6.70) is due to $M_k \geq W_k$ as in (6.62), and (6.71) is due to $k \leq K$ and the nonnegativity of α, δ, and $\tilde{c}$. Thus, the unsafe event satisfies the containment:

$$\left\{ \min_{k \leq K} h(\mathbf{x}_k) < 0 \right\} \subseteq \left\{ \max_{k \leq K} M_k \geq \tilde{\alpha}^K \eta(\mathbf{x}_0) - \sum_{i=1}^{K} \tilde{\alpha}^{K-i} \frac{\tilde{c}}{\delta} \right\}.$$

Since M_k satisfies $M_0 = 0$, $M_k - M_{k-1} \leq 1 \; \forall k \leq K$, and $\langle M \rangle_K \leq \frac{\sigma^2 K}{\delta^2}$, we can apply Thm. 6.23 (Freedman's Ineq.) with $\lambda = \frac{\tilde{\alpha}^K h(\mathbf{x}_0) - \sum_{i=1}^{K} \tilde{\alpha}^{K-i} \tilde{c}}{\delta}$ to achieve the probability bound[12]: $P_u(K, \mathbf{x}_0) \leq H\left(\frac{\tilde{\alpha}^K h(\mathbf{x}_0) - \sum_{i=1}^{K} \tilde{\alpha}^{K-i} \tilde{c}}{\delta}, \frac{\sigma\sqrt{K}}{\delta} \right)$.

(Step 6) If the system (6.7) satisfies the expectation-based DCBF condition (6.16), then (6.58) holds with $(\tilde{\alpha} = \alpha, \tilde{c} = 0)$ so the desired bound is achieved with $\lambda = \alpha^K h(\mathbf{x}_0)/\delta$ and if the system (6.7) satisfies the $c-$martingale condition (6.17), then (6.58) holds with $(\tilde{\alpha} = 1, \tilde{c} = c)$ so the desired bound is achieved with $\lambda = h(\mathbf{x}_0)/\delta - Kc$. □

Bound Tightness Comparisons

We now relate the Freedman-based safety of Theorem 6.24 to the Ville's-based safety of Theorem 6.11 and Corollary 6.13. For systems where the results of both theorems apply, i.e., those with an upper-bounded h (6.18), a lower-bounded error (6.55), and a bounded conditional variance (6.56), we provide a range of values for σ, δ, K, B, and λ for which Theorem 6.24 provides a tighter bound. The following proposition provides a direct theoretical comparison (after changing notation) to the Ville's-based bounds in [35], [57], [131], [192].

Proposition 6.25. *For some $\sigma, \delta, B > 0$, $\lambda \geq 0$ and $K \in \mathbb{N}_1$, consider the conditions*

$$\lambda\delta \geq \sigma^2 K, \qquad \lambda \leq B - \tfrac{\delta}{\varphi}, \tag{6.72}$$

where $\varphi = 2\ln(2) - 1$. If these conditions hold, then

$$H\left(\tfrac{\lambda}{\delta}, \tfrac{\sigma\sqrt{K}}{\delta} \right) \leq 1 - \tfrac{\lambda}{B}. \tag{6.73}$$

Proof of this proposition is is provided in [207, Appx. E].

Intuitively, conditions (6.72) stipulate that the conditional variance σ^2 and number of steps K must be limited by $\lambda\delta$, which is a function of the initial condition times

[12] The proof can end after Step 5 and can be applied to any system satisfying (6.58). We specialize to DCBFs and c-martingales for clarity.

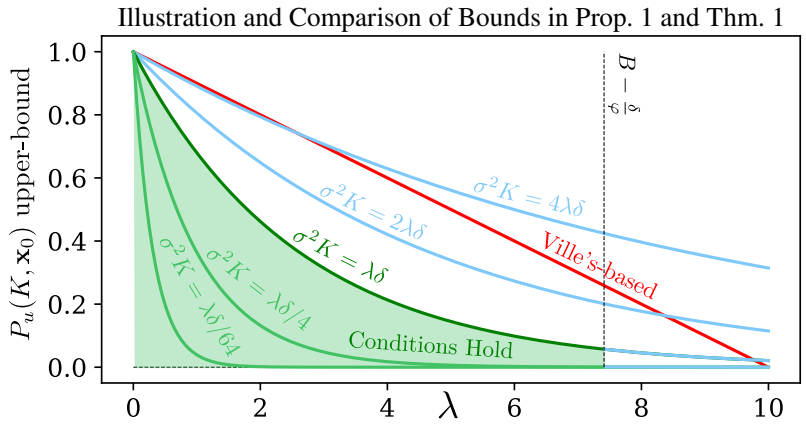


Figure 6.5. Comparison for Prop. 6.25 with $B = 10, K = 100, \delta = 1$, and varying σ and λ. The Freedman-based bounds are shown in green when the conditions of Prop. 6.25 hold and blue when they do not. The Ville's-based bound is shown in red. Code to reproduce this plot can be found at [212].

the maximum single-step disturbance to $h(\mathbf{x}_k)$. Additionally, the initial condition must be less than the maximum safety bound B by an amount proportional to δ. The exact value of φ is a result of the first assumption ($\lambda\delta \geq \sigma^2 K$) and alternative values can be found by changing this assumption; for clarity of presentation, we leave exploration of these alternative assumptions to future work. The safety bounds for various λ and σ are shown in Fig. 6.5 where it is clear that these conditions provide a *conservative* set of parameters over which this proposition holds.

Additionally, since Thm. 6.24 assumes that h has lower-bounded errors (6.55), we can directly compare our method with Input-to-State Safety (ISSf) [31], which provides almost-sure safety guarantees.

In the context of our stochastic, discrete-time problem setting, the ISSf property over finite-time can be reformulated as:

Proposition 6.26 (Finite-time Input-to-State Safety)**.** *If the closed-loop system* (6.2) *satisfies the expectation-based DCBF condition* (6.16) *and the bounded-jump condition* (6.55) *(a.s) for some* $\alpha \in [0, 1)$ *and* $\delta > 0$*, then* $h(\mathbf{x}_k) \geq \alpha^k h(\mathbf{x}_0) - \sum_{i=0}^{k-1} \alpha^i \delta$ *for all* $k \geq 0$ *and*

$$\mathcal{C}_\delta = \left\{ \mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}) \geq \frac{-\delta}{1-\alpha} \right\} \tag{6.74}$$

is safe (a.s.).

Proof. By combining the bounds (6.16) and (6.55):

$$h(\mathbf{x}_{k+1}) \geq \mathbb{E}[h(\mathbf{x}_{k+1}) \mid \mathscr{F}_k] - \delta \geq \alpha h(\mathbf{x}_k) - \delta \quad \text{(a.s.)}. \tag{6.75}$$

Thus, for all $k \in \mathbb{N}_1$, we have the lower-bound $h(\mathbf{x}_k) \geq \alpha^k h(\mathbf{x}_0) - \sum_{i=0}^{k-1} \alpha^i \delta$ (a.s). Furthermore, for all time, $h(\mathbf{x}_k) \geq \frac{-\delta}{1-\alpha} \implies h(\mathbf{x}_{k+1}) \geq \frac{-\delta}{1-\alpha}$, so $\mathcal{C}_\delta$ is safe (a.s.). □

To compare with ISSf's worst-case safe set $\mathcal{C}_\delta$, we wish to use Theorem 6.24 to bound the probability that our system leaves some expanded safe set $\mathcal{C}_\epsilon = \{\mathbf{x} \in \mathbb{R}^n | h(\mathbf{x}) \geq -\epsilon\}$ with $\epsilon \geq 0$ in finite time.

Corollary 6.27. *If the hypotheses of Theorem 6.24 are satisfied and the closed-loop dynamics* (6.7) *satisfy the expectation-based DCBF condition* (6.16) *for some* $\alpha \in (0, 1)$*, then for any value* $\epsilon \geq 0$ *and any* $K \in \mathbb{N}_1$*,*

$$P\left\{ \min_{k \leq K} h(\mathbf{x}_k) < -\epsilon \right\} \leq H\left(\lambda, \tfrac{\sigma}{\delta}\left(\tfrac{1-\alpha^{2K}}{1-\alpha^2}\right)^{\frac{1}{2}}\right) \mathbb{1}_{\left\{-\epsilon \geq \alpha^K h(\mathbf{x}_0) - \sum_{i=0}^{K-1} \alpha^i \delta\right\}} \quad (6.76)$$

where $\lambda = \frac{\alpha^K}{\delta}(h(\mathbf{x}_0) + \epsilon)$.

Proof. The expectation-based DCBF condition (6.16) ensures that, for any $\epsilon \geq 0$:

$$\mathbb{E}[\, h(\mathbf{x}_{k+1}) + \epsilon \mid \mathscr{F}_k \,] \geq \alpha(h(\mathbf{x}_k) + \epsilon) + \epsilon(1 - \alpha) \geq \alpha(h(\mathbf{x}_k) + \epsilon). \quad (6.77)$$

We apply the same proof as for Theorem 6.24 starting at (6.59) with ($\eta(\mathbf{x}_k) = \frac{h(\mathbf{x}_k)+\epsilon}{\delta}, \tilde{\alpha} = \alpha, \tilde{c} = 0$). Choosing $\lambda = \alpha^K \eta(\mathbf{x}_0)$ and bounding[13] $\langle M \rangle_K \leq \sum_{i=1}^{K} \alpha^{2(K-i)} \frac{\sigma^2}{\delta^2} = \frac{\sigma^2(1-\alpha^{2K})}{\delta^2(1-\alpha^2)}$ as in (6.66) yields the desired bound without the indicator function by applying Theorem. 6.23. The indicator function is a result of applying the lower bound on the safety value from Proposition 6.27, i.e., $h(\mathbf{x}_k) \geq \alpha^k h(\mathbf{x}_0) - \sum_{i=0}^{k-1} \alpha^i \delta$ (a.s.) for $k \in \mathbb{N}_0$. □

Next, we perform a simulated comparison of Corollary 6.27 and Proposition 6.26 for various ϵ and distributions (truncated normal, categorical): [14]

Example 6.28 (Prop. 6.26 and Cor. 6.27 Comparison Simulations)**.** *Consider the simple system:*

$$\mathbf{x}_{k+1} = \alpha \mathbf{x}_k + \mathbf{d}_k \quad (6.78)$$

[13] This bound on $\langle M \rangle_K$ uses the finite geometric series identity and can also be applied for a tighter Thm. 6.24 and Prop. 6.25.

[14] Code for these simulations can be found at https://github.com/rkcosner/freedman.git

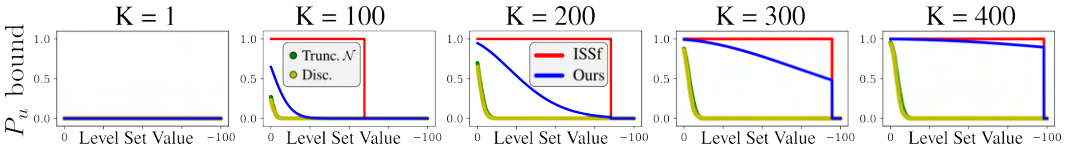


Figure 6.6. Probability that the system is unsafe: our bound from Cor. 6.27 (blue), ISSf bound (red). The x-axis is the level set expansion $-\epsilon$ and the y-axis is the failure probability (lower is better). The plots from left to right indicate safety for $K = 1, 100, 200, 300$, and 400 steps. Simulations where $\mathbb{E}[h(\mathbf{x}_k)|\mathscr{F}_{k-1}] = \alpha h(\mathbf{x}_k)$ and approximate probabilities from 1000 samples are shown for simulations where $h(\mathbf{x}_k)$ is sampled from 3 different conditional distributions: uniform (pink), truncated Gaussian (green), and a categorical (yellow) all which satisfy Cor. 6.27. Code for these plots is can be found at [212].

for $\mathbf{x} \in \mathbb{R}^1$, $\alpha = 0.99$, and zero-mean disturbances $\mathbf{d}_k$ sampled from a variety of distributions for up to $K = 400$ steps. This system naturally satisfies the expectation-based DCBF constraint (6.16)*:*

$$\mathbb{E}[h(\mathbf{x}_{k+1})|\mathscr{F}_k] \geq \alpha h(\mathbf{x}_k) \text{ with } h(\mathbf{x}) = \mathbf{x}, \tag{6.79}$$

so we seek to provide guarantees of its inherent safety probabilities. In particular, in three different experiments we consider $\mathbf{d}_k$ sampled from one of three zero-mean distributions that all satisfy $|\mathbf{d}| \leq 1$ and $\sigma \leq \frac{1}{3}$: a uniform distribution $\mathcal{U}_{[-1,1]}$, a standard normal distribution truncated at -1 and 1, and a categorical distribution where $\mathbb{P}\{\mathbf{d} = -1\} = \frac{1}{6}$ and $\mathbb{P}\{\mathbf{d} = \frac{1}{5}\} = \frac{5}{6}$ to ensure 0 mean.

The results of these simulations are shown in Figure 6.6 where we can see that our method successfully upper-bounds the sampled safety probabilities with risk-sensitive guarantees that are much less conservative than the worst-case bounds provided by ISSf.

These simulations also show that, although our method is conservative compared to the Monte Carlo approximations, it provides useful risk-based safety probabilities for a variety of $\mathcal{C}_\epsilon$ level sets whereas ISSf only provides a worst-case almost-surely bound.

Finally, we apply our method to a simplified model of a bipedal walking robot.

Example 6.29 (Simulated Bipedal Robot)**.** *In this example we use the Hybrid Linear Inverted Pendulum (HLIP) model [213] which approximates a bipedal robot as an inverted pendulum with a fixed center of mass (COM) height $z_0 \in \mathbb{R}_{>0}$. Its states are the planar position, relative COM-to-stance foot position, and COM velocity $\mathbf{p}, \mathbf{c}, \mathbf{v} \in \mathbb{R}^2$. The step-to-step dynamics are linear and the input is the relative foot placement, $\mathbf{u}_k \in \mathbb{R}^2$. The matrices $\mathbf{A} \in \mathbb{R}^{6\times 6}$ and $\mathbf{B} \in \mathbb{R}^{6\times 2}$ are determined*

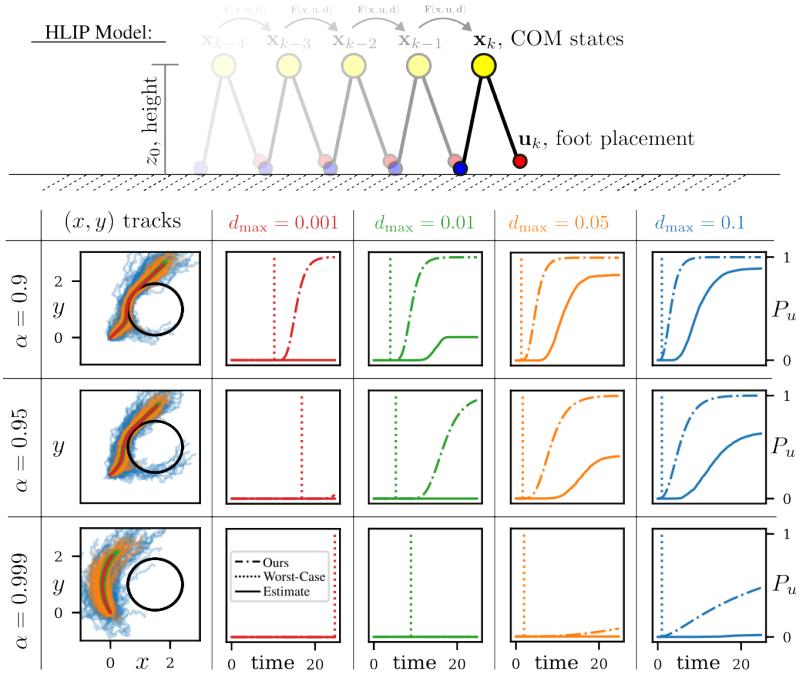


Figure 6.7. Safety results for a bipedal robot navigating around an obstacle using our method. **(Top)** Visualization of the Hybrid Linear Inverted Pendulum (HLIP) model. Yellow indicates the center-of-mass (COM), blue is the stance foot, and red is the swing foot. The states $\mathbf{x}_k$ are the global COM position, the relative COM position, and COM velocity, and the input is the relative position of the feet at impact. **(Bottom)** A table with variable maximum disturbance value ($d_{\max}$) and controller parameter (α) shows our (dashed lines) theoretical bound on safety failure from Thm. 6.24, (dotted lines) the shortest first-violation time based on the worst-case disturbance approximation, and (solid lines) approximated probabilities from 5000 trials (lower is safer). On the left, the trajectories of the COM are shown walking from bottom left towards the top right while avoiding the obstacle with each color corresponding to a different $d_{\max}$. The robot attempts to avoid the obstacle (black).

by z_0 and gait parameters including the stance and swing phase periods. The HLIP model with an added disturbance matrix $\mathbf{D} \in \mathbb{R}^{6\times 4}$ and disturbance $\mathbf{d} \in \mathbb{R}^4$ affecting position and velocity is:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{D}\mathbf{d}_k, \quad \mathbf{d}_k \sim \mathcal{D},$$

where $\mathbf{x}_k = \begin{bmatrix}\mathbf{p}_k^\top & \mathbf{c}_k^\top & \mathbf{v}_k^\top\end{bmatrix}^\top$. We augment the standard HLIP model and assume that $\mathbf{d}$ enters linearly and $\mathcal{D}$ is a 4-dimensional, 0-mean uniform distribution [15] with $\|\mathbf{d}\| \leq d_{\max}$.

We define safety for this system as avoiding a circular obstacle of radius $r > 0$ located at $(x, y) = \boldsymbol{\rho} \in \mathbb{R}^2$, so safety can be defined using the signed-distance function $h(\mathbf{x}) = \|\mathbf{p} - \boldsymbol{\rho}\|_2 - r$. Notably, this function has no upper bound and therefore the Ville's-based Cor. 6.13 does not apply.

[15] See [207, Appx. H] for bounds for δ and σ given this problem structure.

Since $h(\mathbf{x})$ is not convex, we use a conservative halfspace convexification instead:

$$h(\mathbf{x}_{k+1}) \geq \widehat{\mathbf{e}}(\mathbf{p}_k)^\top (\mathbf{p}_{k+1} - \boldsymbol{\rho}) - r \triangleq \bar{h}(\mathbf{x}_{k+1}), \tag{6.80}$$

where $\widehat{\mathbf{e}}(\mathbf{p}) = \frac{(\mathbf{p}-\boldsymbol{\rho})}{\|\mathbf{p}-\boldsymbol{\rho}\|}$ and we apply the controller:

$$\begin{aligned} \mathbf{u}^* = \min_{\mathbf{u}\in\mathbb{R}^2} \quad & \|\mathbf{u} - \mathbf{k}_{\text{nom}}(\mathbf{x}_k)\| \\ \text{s.t.} \quad & \mathbb{E}\left[\, \bar{h}(\mathbf{x}_{k+1}) \mid \mathscr{F}_k \,\right] \geq \alpha \bar{h}(\mathbf{x}_k) \end{aligned} \tag{6.81}$$

with $\alpha \in (0, 1]$ and where $\mathbf{k}_{\text{nom}}$ tracks a desired velocity.

We ran 5000 trials with 3 steps per second and compared against the theoretical bound from Thm. 6.24. Those values and planar pose trajectories can be seen in Fig. 6.7.

Despite the relative tightness guarantee of Proposition 6.25, the probability guarantees of our method are still not necessarily tight, as can be seen in Fig. 6.6. Optimization of h without changing $\mathcal{C}$ as in [192] through methods such as sum-of-squares is a promising direction further tightening.

6.5 Conclusion

This section presented methods for generating robust guarantees of safety and stability in the presence of stochastic and potentially unbounded uncertainty. This work extends the worst-case robustness methods of Chapter 4 to provide guarantees when those do not hold (when the uncertainty is unbounded) and to provide additional nuance when both results hold.

The stochastic results of this section represent a middle middle ground between the robust theoretical results of Chapter 4 and the data-driven performant robotic demonstrations of Chapter 5. Here we retain theoretical guarantees and the level of tolerable risk can be explicitly tuned[16], retaining guarantees, but allowing for performance improvements.

The theoretical results in this chapter rely on a structural connection between Lyapunov methods, like CBFs and CLFs, and martingales. This connection is predicated on the self-referential property of these Lyapunov methods where the next value is bounded by a function of the previous value. This constraint allows for the construction of a supermartingale and the subsequent application of martingale-based

[16] As opposed to the indirect tuning of the PBL method in Section 5.4. In this case, we consider risk to be exactly the K-step exit probability.

concentration inequalities to generate probabilistic safety guarantees, with additional applications to robust stability explored in [35], [204].

This method does have its limitations. Although it produces trajectory-long safety guarantees, these results remain very conservative and future work should investigate calibration methods for generating more-accurate probabilistic bounds that generate controllers that can still be deployed in real-time applications. Interesting recent work has explored set-erosion with sub-Gaussian disturbances as an interesting future direction for potential tightened guarantees [189].

Additionally, one significant limitation of the work presented in this chapter is that, in order to deploy these controllers, one must have some understanding the underlying disturbance distribution since this is used to calculate the disturbances's mean and covariance used in several of the controllers in this chapter. To remove this limitation, the next chapter will discuss methods for modeling these disturbance distributions using generative modeling techniques. Together with the methods in this chapter, this will allow us to deploy these controllers on real-world hardware systems.

Chapter 7

DEPLOYING RISK-AWARE DYNAMIC SAFETY

"*Real artists ship.*" - Steve Jobs

"*Build it, break it, fix it.*" - Marc Raibert

"*Theory is cheap, show me the experiments.*" - Magnus Egerstedt

In addition to theoretical generality and mathematical guarantees, a roboticist must also focus on practical deployment. The true test of a control algorithm is in its ability to "ship," i.e., to be deployed reliably and effectively on dynamic, real-world systems.

This chapter focuses on the real-world deployment of the stochastic control algorithms introduced in the previous chapter. Moving beyond theory and simulation, we present hardware demonstrations showcasing how these methods can achieve performant, safety-critical behavior under significant uncertainty.

Abstract

Safety-critical control frameworks such as Control Barrier Functions (CBFs) provide a powerful foundation for achieving robust safety guarantees in robotics. Recent work has extended CBFs to handle a wide range of uncertainties, including worst-case bounded disturbances (Chapter 4) and potentially unbounded stochastic disturbances (Chapter 6). While stochastic CBF methods can more accurately reflect real-world uncertainty, they have been generally constrained to theory and simulation with limited practical demonstrations. Two key challenges contribute to this gap: (1) incorporating stochastic robustness often increases computational complexity, hindering real-time implementations; and (2) these methods require *a priori* knowledge about the disturbance distribution, which is rarely known in practice.

This chapter addresses these limitations by demonstrating real-world deployment of the stochastic discrete-time CBF (DCBF) framework from Chapter 6. In Sections 7.2

and 7.3, we deploy these methods on quadrotor and bipedal robot platforms, respectively, operating under substantial uncertainty. We leverage generative modeling to learn disturbance distributions and develop computationally tractable DCBF controllers capable of real-time closed-loop deployment. Section 7.4 then discusses the problem of achieving desirable closed-loop behavior, beyond just practical safety guarantees. This section shows that a horizon-based, model predictive control (MPC) approach that incorporates a stochastic DCBF constraint can lead to both probabilistic safety guarantees *and* performance improvements. This result is then validated through hardware experiments on both quadrupedal and quadrotor robots performing vision-based dynamic obstacle avoidance. In the quadrotor experiments, we focus on the real-world deployment of this method with the computer vision, obstacle estimation, and control algorithms running exclusively on an onboard edge computer.

Published content: The text for this chapter is adapted from [17], [59], [60].

7.1 Introduction

The theoretical developments of the previous chapter present a promising direction for achieving guarantees of safety in the face of real-world, stochastic uncertainty. However, hardware demonstrations of these methods are rare, in part because of the *a priori* uncertainty knowledge required to implement controllers like (6.40).

Learning Disturbance Distributions

The first two sections of this chapter will engage directly with this challenge of unknown disturbance distributions when implementing the guarantees of Chapter 6 on hardware systems.

In general, it is not uncommon for theoretical stochastic control methods to assume significant prior knowledge such as the value at risk (VaR) or conditional value at risk (CVaR) of $h(\mathbf{x}_{k+1})$ [205] or some statistics of the disturbance distribution [57], [214]. While this assumption is markedly different than the global upper bound assumption common in deterministic robust control [4], [32], [52], [65], it is still unrealistic to assume perfect, *a priori* knowledge relating to the disturbance before operating the system, and impractical / unprincipled to estimate these values by hand. To address these issues, we propose the use of generative modeling techniques to learn a state-dependent conditional distribution of the dynamics residuals from data.

In particular, Sections 7.2 and 7.3 of this chapter deploy the probabilistic guaran-

tees and control methods of the previous chapter by using deep generative models (DGMs) [215]–[217] which approximate the disturbance distributions. DGMs are a broad class of methods that use neural networks to approximate the probability distribution underlying a given dataset. These models can be used for density estimation, which provides a likelihood model for the data, or to sample (i.e., "generate") new data points from the approximated distribution. Beyond their traditional applications, like generating image [218] and text [219] data, these models have been applied to a broad range of robotics tasks including SLAM [220], imitation learning [221], motion planning [222], anomaly detection [186], and dynamics learning [223].

For the work in this chapter, we chose to employ conditional variational autoencoders (CVAEs) [224] which are a class of DGMs and a generalization of variational autoencoders (VAEs) that allow one to condition the generating process on a context variable (e.g., the current state $\mathbf{x}$). CVAEs have been used to recreate hand-written images of numbers given the desired digit [224] or to predict trajectories given state and environment understanding [225]. Since the generative process for a CVAE only requires two neural network forward passes and normal distribution samples, they are computationally efficient and particularly well suited for real-time robotics applications [226].

The first section of this chapter, Section 7.2, unifies DCBF and CVAE methods in the form of the Online Risk-Informed Optimization (ORIO) controller, a risk-based safety framework that learns to ensure safety in the presence of stochastic dynamics uncertainty and we provide demonstrations of ORIO on a quadrotor drone carry a slung mass attached with a flexible cable. Section 7.3 then extends this idea to the context of hierarchical control, in an architecture similar to [48] and Section 3.3, where we generate safe reference signals that are then tracked by a complex full-order system, such as a humanoid robot with a reinforcement learning (RL)-based, velocity-tracking locomotion controller. The main difference here is that we treat the difference between the reduced-order model (ROM) and the full-order system as a history-dependent stochastic disturbance instead of assuming tracking convergence. This idea is formally presented as the SHIELD (Safety on Humanoids via CBFs In Expectation on Learned Dynamics) safety filter paradigm which leverages the stochastic safety-critical control methods of the previous chapter to improve tracking and achieve safe collision avoidance of a humanoid robot.

Improving Performance Alongside Safety

Additionally, in order to achieve *dynamic* safety, it critical that, whenever possible, robots retain the ability to accomplish their task; they should not generally forego all performance metrics in favor of safety[1].

In general, the common methodologies for safety-critical control take very different approaches to achieve desirable performance. DCBF-based methods [31] generally take a myopic, pointwise approach; state-constrained model predictive control (MPC) [5] jointly optimizes for safety and performance over a finite horizon; and backwards Hamilton-Jacobi (HJ) methods like [73] optimize a plan and ensure safety of a tracking controller separately. While HJ methods provide strong guarantees of safety, they often have limited applications to high-dimensional and/or nonlinear systems due to their computational complexity [4], even when used in conjunction with simplified planners. Alternatively, despite more restrictive theoretical assumptions required for their implementation, widespread experimental success has been achieved for both MPC [227], [228] and CBF-based [19], [22] methods by enforcing computationally simple safety constraints while optimizing for tractable proxies of performance.

By optimizing performance over a receding horizon, MPC methods generally achieve significantly better performance than CBF methods which often encounter undesirable conflict between safety and performance goals [16]. On the other hand DCBF-based safety constraints display desirable robustness guarantees, as illustrated in the previous chapter, that do not generally hold for MPC methods. To unify the horizon-based performance benefits of MPC with the robustness of CBFs, many combinations of these two approaches have been proposed that involve hierarchical multi-rate paradigms [25], [228] and extended discrete-time predictions [229]. Our discussion in Section 7.4 aligns most closely with the combinations presented in [46], [47], [198], [230] which apply the DCBF condition as a constraint in the MPC's finite-time optimal control problem (FTOCP). Section 7.4 significantly extends these works by considering the closed-loop feasibility improvements and robustness properties (both worst-case and probabilistic) of a unified MPC+DCBF controller with demonstrations of dynamic obstacle avoidance on quadrupedal and quadrotor robots.

[1] For example, see how Asimov's second law supersedes the third in Definition 1.0.

7.2 Risk-Aware Control of a Quadrotor with a Slung Mass

In this section, we explore how deep generative models [215]–[217] can be used to approximate dynamics uncertainty distributions and thus enable the deployment of the stochastic control methods of Chapter 6.

In particular, this section studies the problem of safe flight for a quadrotor robot experience significant dynamics uncertainty including ground effects and unmodeled payloads attached via flexible cables. We find that in the case of large, difficult to predict disturbances, the unification of generative modeling and stochastic DCBF-based controllers enables safe and performant flight. The work in this section on modeling dynamics residuals is similar to [40] and can be thought of as a probabilistic generalization.

The contributions of this section are as follows:

- The Online Risk-Informed Optimization (ORIO) controller: a unified framework for dynamics distribution learning and deployment of DCBF-based stochastic safety.
- Simulation and hardware demonstrations of the real-time application of ORIO on a quadrotor robot with a slung load for safe flight.

The text for this section is adapted from:

> R. K. Cosner, I. Sadalski, J. K. Woo, P. Culbertson, and A. D. Ames, "Generative modeling of residuals for real-time risk-sensitive safety with discrete-time control barrier functions," *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024. DOI: 10.1109/ICRA57147.2024.10611355,

A video for this section can be found at [231].

In this work, we consider applications of safe control in the presence of unmodeled disturbances. Specifically, we consider systems with discrete-time dynamics of the form:

$$\mathbf{x}_{k+1} = \mathbf{F}_a(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{d}_k, \quad \forall k \in \mathbb{N} \tag{7.1}$$

with state $\mathbf{x}_k \in \mathbb{R}^{n_x}$, input $\mathbf{u}_k \in \mathbb{R}^{n_u}$, unmodeled residual dynamics $\mathbf{d}_k$ that take values in $\mathbb{R}^{n_d}$ and are sampled from some unknown, state-dependent distribution

$p(\mathbf{d}|\mathbf{x})$, and modeled dynamics $\mathbf{F}_a : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$. A state-feedback controller $\mathbf{k} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ yields the discrete-time closed-loop system:

$$\mathbf{x}_{k+1} = \mathbf{F}_a(\mathbf{x}_k, \mathbf{k}(\mathbf{x}_k)) + \mathbf{d}_k, \quad \forall k \in \mathbb{N}. \tag{7.2}$$

This is a simplification of the general stochastic dynamics uncertainty model of (6.2) where here we assume that the dynamics residuals are input-independent and additive. This is a common structure often assumed in robust control theory for discrete-time systems [232]–[234].

Generative Modeling using Conditional Variational Autoencoders (CVAEs)
To account for the unmodeled disturbances, we first seek a generative model that can approximate the conditional distribution $p(\mathbf{d}|\mathbf{x})$ given a dataset $\mathfrak{D} = \{(\mathbf{x}_i, \mathbf{d}_i)\}_{i=1}^{n_x}$. We do this by fitting a parametric distribution to $\mathfrak{D}$ which attempts to maximize the likelihood of the observed data with respect to the learned distribution.

While there exist many (learning- and learning-free) methods for generative modeling, in this section, we look to Conditional Variational Autoencoders (CVAEs) [224], a variant of Variational Autoencoders (VAEs) [235] that allows the learned models to be conditioned on observations, $\mathbf{x}$. CVAEs assume there exists a latent variable $\mathbf{z}$ which captures the "unobserved" information explaining any non-random variation in the data distribution. For example, in the setting of robot safety, the latent codes $\mathbf{z}$ could represent state-dependent modeling errors, or other hidden variables (e.g., higher-order dynamics, time delays) that could influence the difference between the observed next state $\mathbf{x}_{k+1}$, and the modeled dynamics $\mathbf{F}_a(\mathbf{x}_k, \mathbf{u}_k)$.

Specifically, CVAEs represent the conditional distributions $p_\theta(\mathbf{d}|\mathbf{x}, \mathbf{z})$ and $q_\varphi(\mathbf{z}|\mathbf{x}, \mathbf{d})$, and the latent prior $p_\phi(\mathbf{z}|\mathbf{x})$ as multilayer perceptions (MLPs) with corresponding parameters θ, φ, ϕ, and seek to optimize these parameters such that the learned data likelihood $p_{\theta,\phi}(\mathbf{d}|\mathbf{x})$ is maximized. Traditionally q_φ is called an "encoder," since it maps states $\mathbf{x}$ and disturbances $\mathbf{d}$ to distributions over the latent codes $\mathbf{z}$, and similarly p_θ is called a "decoder," since it decodes latent codes $\mathbf{z}$ and states $\mathbf{x}$ into disturbance distributions $\mathbf{d}$. While maximizing $p_{\theta,\phi}(\mathbf{d}|\mathbf{x})$ directly is intractable, we can instead optimize it by maximizing the evidence lower bound (ELBO) as a proxy:

$$\log p_{\theta,\phi}(\mathbf{d}|\mathbf{x}) \geq \mathbb{E}_{q_\varphi}[\log p_\theta(\mathbf{d}|\mathbf{x}, \mathbf{z})] - KL\big(q_\varphi(\mathbf{z}|\mathbf{x}, \mathbf{d}) \| p_\phi(\mathbf{z}|\mathbf{x})\big) \tag{7.3}$$

where KL is the Kullback-Liebler divergence [206]. In practice, each network represents its corresponding distribution as a conditional Gaussian, e.g., $p_\theta(\mathbf{d}|\mathbf{x}, \mathbf{z}) =$

$\mathcal{N}(\mathbf{d}; \boldsymbol{\mu}_\theta(\mathbf{x}, \mathbf{z}), \boldsymbol{\Sigma}_\theta(\mathbf{x}, \mathbf{z}))$, where $\mathcal{N}(\cdot\,; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the probability density function of a multivariate Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, and $\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta$ are neural networks parameterized by θ that represent the parameters of the distribution.

Once trained, a CVAE can be used to estimate the disturbance's conditional likelihood, generate new samples, or approximate the mean and covariance of the true distribution.

In particular, the risk-sensitive DCBF-based controller (6.40) requires an approximation of the mean and covariance of the disturbance distribution $\mathbf{d}$. To do this, we use the following estimatorfor $p_{\theta,\phi}(\mathbf{d}|\mathbf{x})$:

$$p_{\theta,\phi}(\mathbf{d}|\mathbf{x}) \approx \frac{1}{S}\sum_{s=1}^{S} p_\theta(\mathbf{d}|\mathbf{x}, \mathbf{z}^{(s)}) = \frac{1}{S}\sum_{s=1}^{S} \mathcal{N}\left(\mathbf{d}\,;\ \boldsymbol{\mu}_\theta(\mathbf{x}, \mathbf{z}^{(s)}), \boldsymbol{\Sigma}_\theta(\mathbf{x}, \mathbf{z}^{(s)})\right) \quad (7.4)$$

where $\mathbf{z}^{(s)} \sim p_\phi(\mathbf{z}|\mathbf{x})$ is one of S samples drawn from the prior distribution. Since this approximation is a Gaussian mixture model (GMM) we can obtain its mean and expectation in closed form as:

$$\mathbb{E}_{p_{\theta,\phi}}[\mathbf{d}|\mathbf{x}] \approx \frac{1}{S}\sum_{s=1}^{S} \mu_\theta(\mathbf{x}, \mathbf{z}^{(s)}) \triangleq \overline{\boldsymbol{\mu}}_{(\theta,\phi)}(\mathbf{x}), \quad (7.5)$$

$$\begin{aligned}\mathrm{cov}_{p_{\theta,\phi}}(\mathbf{d}|\mathbf{x}) \approx &\frac{1}{S}\left(\textstyle\sum_{s=1}^{S} \Sigma_\theta(\mathbf{x}, \mathbf{z}^{(s)}) + \mu_\theta(\mathbf{x}, \mathbf{z}^{(s)})\mu_\theta(\mathbf{x}, \mathbf{z}^{(s)})^T\right), \\ &- \overline{\boldsymbol{\mu}}_{(\theta,\phi)}(\mathbf{x})\overline{\boldsymbol{\mu}}_{(\theta,\phi)}(\mathbf{x})^\top \triangleq \overline{\boldsymbol{\Sigma}}_{(\theta,\phi)}(\mathbf{x}). \end{aligned} \quad (7.6)$$

To demonstrate the capabilities of CVAEs to learn complex dynamics residuals, we consider the following example using a simple double integrator system:

Example 7.1 (Residual Dynamics Modeling of a Double Integrator Example)**.** *Consider the dynamics:*

$$\mathbf{x}_{k+1} = \begin{bmatrix} x \\ v \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \Delta_t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix}_k + \Delta_t \mathbf{d}_k \quad (7.7)$$

with a state-dependent Gaussian residual distribution, $\mathbf{d}_k \sim p(\mathbf{d}_k|\mathbf{x}_k)$ *with mean and covariance:*

$$\boldsymbol{\mu}(\mathbf{x}) = \begin{bmatrix} 0, \sin(x) \end{bmatrix}^\top,\ \boldsymbol{\Sigma}(\mathbf{x}) = \frac{1}{2}\begin{bmatrix} 2+\cos(x) & \exp(-|x|) \\ \exp(-|x|) & 2+\sin(x) \end{bmatrix}.$$

The system was initialized at $\mathbf{x}_0 = \mathbf{0}$ *with* $\Delta_t = 0.01$ *and simulated for 35 five-second trials to collect data. Then a CVAE was trained to approximate the distribution. The*

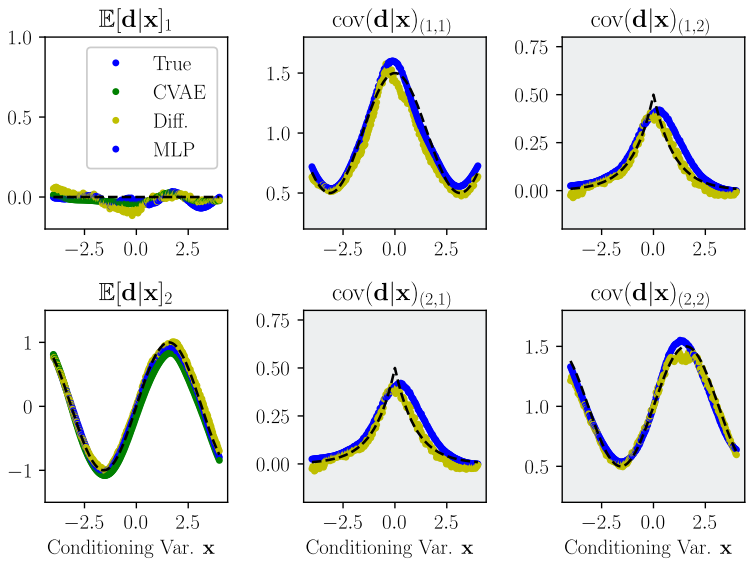


Figure 7.1. Learning heteroschedastic disturbance of double integrator system using 3 minutes of data at 100 Hz (36 five second long trajectories). The approximated mean and covariance values are scaled by the time step and plotted against the true values in black using the CVAE in blue, diffusion model in yellow, and MLP (mean only) in green.

CVAE accurately learns the nonlinear heteroschedastic disturbance with a relatively small amount (3 minutes of simulation time) of data, as can be seen in Fig. 7.1.

We compare the CVAE to two baselines: a conditional diffusion model [236], which is a another popular generative model that has recently seen interest as a policy representation for robotics [221], and a simple MLP trained to map the state $\mathbf{x}$ *to a fixed, deterministic disturbance* $\mathbf{d}(\mathbf{x})$*. The results of this are shown in Fig. 7.2 and Table 7.1. There we can see that the MLP, despite being significantly faster, tends to overfit to the noise causing higher mean error. Alternatively, the diffusion model accurately learns the distribution, but is nearly two orders of magnitude slower than the CVAE. Additionally, two approximation methods are used for the CVAE: the GMM-based estimator in (7.5, 7.6) and a simple two-step sampling estimator using the population mean and covariance calculations from samples of* $p_{\theta,\phi}(\mathbf{d}|\mathbf{x})$*. The GMM-based method is shown to be slightly faster and results in less average error and variance.*

The Online Risk-Informed Optimization (ORIO) Controller

The Ville's inequality-based theoretical guarantees of the previous chapter, specifically Corollary 6.13 and Theorem 6.15, can be combined and restated as:

Theorem 7.2 (Probabilistic Safety Guarantees with Tractable DCBF Constraints)**.**

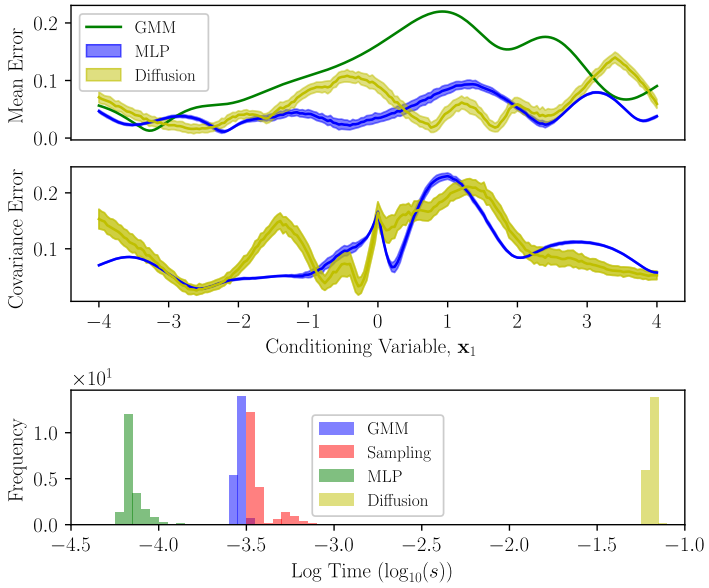


Figure 7.2. Comparing generative models. **(Top)** Mean error $\|\mathbb{E}_{p_{\theta,\phi}}[\mathbf{d}|\mathbf{x}] - \mathbb{E}_{\text{true}}[\mathbf{d}|\mathbf{x}]\|$ vs. state for GMM-based CVAE sampling method (blue), MLP (green), diffusion model sampling mean estimate (yellow). One standard deviation estimated from 100 samples per state $\mathbf{x}$ is plotted around the CVAE and diffusion model curves. **(Middle)** Covariance error $\|\text{cov}_{p_{\theta,\phi}}(\mathbf{d}|\mathbf{x}) - \text{cov}_{\text{true}}(\mathbf{d}|\mathbf{x})\|_2$ vs. state for GMM-based CVAE method (blue) and diffusion model sampling-mean estimate (yellow) both using 10,000 samples is shown with one standard deviation. **(Bottom)** Evaluation time for each model (time in log scale). Calculations were performed on a desktop computer with a Nvidia 3090Ti GPU with $S = 10,000$ for the stochastic methods.

Consider the system (7.2) *and let* $h : \mathbb{R}^{n_x} \to \mathbb{R}$ *be a twice-continuously differentiable, concave function such that* $\sup_{\mathbf{x}\in\mathbb{R}^{n_x}} h(\mathbf{x}) \leq B$ *for* $B \in \mathbb{R}_{>0}$ *and* $\sup_{\mathbf{x}\in\mathbb{R}^{n_x}} \|\nabla^2 h(\mathbf{x})\|_2 \leq \lambda_{\max}$ *for* $\lambda_{\max} \in \mathbb{R}_{\geq 0}$. *If there exists an* $\alpha \in (0, 1]$ *such that:*

$$h(\mathbf{F}_a(\mathbf{x}, \mathbf{k}(\mathbf{x})) + \mathbb{E}[\mathbf{d}|\mathbf{x}]) - \frac{\lambda_{\max}}{2}\text{tr}(\text{cov}(\mathbf{d}|\mathbf{x})) \geq \alpha h(\mathbf{x}) \tag{7.8}$$

for all $\mathbf{x} \in \mathcal{C}$ *with* $\mathbf{d} \sim p(\mathbf{d}|\mathbf{x})$, *then for any* $K \in \mathbb{N}_1$ *the following probability bound holds on the* $K-$*step exit probability (Def. 6.7):*

$$P_u(K, \mathbf{x}_0) \triangleq \mathbb{P}\{\mathbf{x}_k \notin \mathcal{C} \text{ for some } k \leq K\} \leq 1 - \frac{h(\mathbf{x}_0)}{B}\alpha^K. \tag{7.9}$$

This theorem provides a practical method for enforcing the probabilistic guarantees of Section 6.3 when the conditional mean and covariance of the disturbance distribution are known and we can then use a CVAE to approximate these values. This unification, which we call the ORIO (Online Risk-Informed Optimization) controller, provides a tractable way for enforcing the probabilistic safety of Theorem

	μ Err. Avg. $\pm 2\sigma$	Σ Err. Avg. $\pm 2\sigma$
GMM	0.04512 ± 0.00433	0.09518 ± 0.00296
Sampling	0.04604 ± 0.00989	0.09710 ± 0.01419
Diffusion	0.05866 ± 0.00942	0.1025 ± 0.01363

Table 7.1. The statistics for the mean and covariance estimates of each estimation method obtained from 100 estimates at 201 states. The average error is similar for each model, but the GMM-based method has smaller variance which is important when using its outputs in closed-loop control. Estimates for each method are calculated using $S = 10,000$.

7.2[2]:

$$\mathbf{k}(\mathbf{x}) = \underset{\mathbf{u}\in\mathbb{R}^{n_u}}{\operatorname{argmin}} \|\mathbf{u} - \mathbf{k}_{\text{nom}}(\mathbf{x})\|^2 \tag{7.10}$$
$$\text{s.t. } h\left(\mathbf{F}_a(\mathbf{x},\mathbf{u}) + \overline{\boldsymbol{\mu}}_{(\theta,\phi)}(\mathbf{x})\right) - \frac{\lambda_{\max}}{2}\text{tr}\left(\overline{\boldsymbol{\Sigma}}_{(\theta,\phi)}(\mathbf{x})\right) \geq \alpha h(\mathbf{x}).$$

Here the true values of $\mathbb{E}[\mathbf{d}|\mathbf{x}]$ and $\text{cov}(\mathbf{d}|\mathbf{x})$ are approximated using the outputs, $\overline{\boldsymbol{\mu}}_{(\phi,\theta)}(\mathbf{x})$ and $\overline{\boldsymbol{\Sigma}}_{(\phi,\theta)}(\mathbf{x})$, of the CVAE and then are used to enforce the DBCF constraint (7.8). In the remainder of this section we will see how this controller can be used to achieve risk-aware safety on robots with limited compute resources and large, chaotic disturbances.

Safety for Quadrotor Drone: Theory and Experiments

Next we show the utility of the ORIO controller (7.10) in achieving safe flight on a quadrotor drone, and evaluated in both simulation and hardware.

Quadrotor Dynamics Model: We consider a quadrotor drone and model its continuous-time dynamics as:

$$\underbrace{\frac{d}{dt}\begin{bmatrix}\mathbf{p}\\ q\\ \mathbf{v}\end{bmatrix}}_{\dot{\mathbf{x}}} = \begin{bmatrix}\mathbf{v}\\ \mathbf{0}\\ -\mathbf{e}_z g\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \boldsymbol{\omega}\\ \frac{1}{m}\mathbf{R}(q)\mathbf{e}_z\tau\end{bmatrix} \tag{7.11}$$

where the state $\mathbf{x} = (\mathbf{p} \in \mathbb{R}^3, q \in \mathbb{S}^3, \mathbf{v} \in \mathbb{R}^3)$ represents the position, orientation, and velocity, g is gravity, m is the drone mass, and the system has inputs of angular rate $\boldsymbol{\omega} \in \mathbb{R}^3$ and thrust force $\tau \in \mathbb{R}$. Here $\mathbf{e}_z$ is a unit vector in the z direction and $\mathbf{R} : \mathbb{S}^3 \to \text{SO}(3)$ maps the quaternion representation of orientation to the respective rotation matrix. For simulation, these dynamics are approximated in

[2]We note that in order for the ORIO controller (7.10) to provide the theoretical guarantees of Theorem 7.2 we additionally require that the CVAE perfectly model the disturbance distribution. In practice this may not be true and an interesting direction for future work involves analyzing the accuracy and confidence in the predictions of these generative models.

discrete-time using Euler integration on manifolds and for the DCBF standard Euler integration (denoted $\mathbf{F}_{\text{Eul}}(\mathbf{x}, \mathbf{u})$) is used for ease of computation, [203] shows that this approximation is theoretically well justified for DCBFs with short time steps.

Quadrotor DCBF Synthesis: The safety criteria for our quadrotor is to avoid collisions with the ground or roof. We can encode this safety criterion using the Lyapunov-based CBF synthesis method similar to that of Section 3.2:

$$h_{\text{des}}(\mathbf{x}) = C - \boldsymbol{\zeta}^\top \mathbf{P} \boldsymbol{\zeta} \tag{7.12}$$

for some $C > 0$ where $\boldsymbol{\zeta} = \begin{bmatrix} z - z_0, & v_z \end{bmatrix}^\top$ and $V(\boldsymbol{\zeta}) = \boldsymbol{\zeta}^\top \mathbf{P} \boldsymbol{\zeta}$ is a Lyapunov function generated by the Discrete-time Algebraic Ricatti Equation (DARE) for discrete-time double integrator dynamics. However, this is not necessarily a DCBF since the quadrotor's orientation may render it unable to track double integrator trajectories.

To avoid this issue, we add an penalty term to ensure correct orientation at the boundary of the desired safe set:

$$h(\mathbf{x}) = h_{\text{des}}(\mathbf{x}) - \lambda(1 - \mathbf{e}_z R(q) \mathbf{e}_z), \text{ with } \lambda > 0. \tag{7.13}$$

This DCBF is motivated by differential flatness of the quadrotor dynamics [237] and the system's ability to track double integrator trajectories. Additionally, this DCBF synthesis method inspired the DRD-CBF synthesis method presented in Section 3.5.

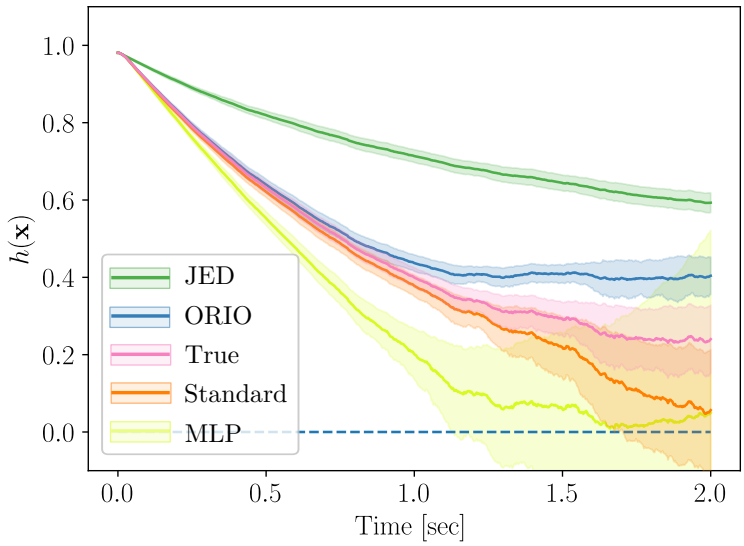
Importantly, (7.13) is a valid DCBF for the Euler-approximated dynamics and there are bounds for $\overline{\boldsymbol{\mu}}_{(\theta,\phi)}$ and $\frac{\lambda_{\max}}{2}\text{tr}(\overline{\boldsymbol{\Sigma}}_{(\theta,\phi)}(\mathbf{x}))$ such that the ORIO controller (7.10) is feasible for all $\mathbf{x} \in \mathcal{C} = \{\mathbf{x} \in \mathbb{R}^{n_x} \mid h(\mathbf{x}) \geq 0\}$. This is formally stated and proved in [238, Thm. 4].

Controllers under Consideration: For comparison, we implement several controllers in addition to ORIO (7.10). Each controller has the structure:

$$\begin{aligned} \mathbf{k}(\mathbf{x}) = \underset{\mathbf{u} \in \mathbb{R}^{n_u}}{\text{argmin}} \quad & \|\mathbf{u} - \mathbf{k}_{\text{Nom}}(\mathbf{x}, k)\|^2 \\ \text{s.t.} \quad & h(\mathbf{F}_{\text{Eul}}(\mathbf{x}, \mathbf{u}) + \mathbf{m}(\mathbf{x})) - c(\mathbf{x}) \geq \alpha h(\mathbf{x}) \end{aligned} \tag{7.14}$$

with the following ablations:

- *Standard:* where $\mathbf{m}(\mathbf{x}) = \mathbf{0}$ and $c(\mathbf{x}) = 0$. This is the standard DCBF controller [197] where the modeled dynamics are assumed to be correct.



$\mathbb{P}_{\text{bound}}$	MLP	Standard	JED	True	ORIO
0.82	0.69	0.56	0.00	0.32	0.11

Figure 7.3. Quadrotor simulation results. (**Figure**) The mean of 100 trajectories for each controller is plotted with 1/2 standard deviation around it. (**Table**) The $K-$step probability bound for the 2 second long trial from Thm. 6.15 and the approximated $K-$step probability experienced on in simulation over 100 trials.

- *JED* (6.40)*:* where $\mathbf{m}(\mathbf{x})$ is the constant sample mean of the training dataset $\mathfrak{D}$ and $c(\mathbf{x})$ is the trace of the sample covariance times $\sup_{\mathbf{x}\in\mathbb{R}^{n_x}} \|\nabla^2 h(\mathbf{x})\|_2$. This is the JED controller from section (6.40) from Section 6.3.

- *MLP:* where $\mathbf{m}(\mathbf{x})$ is an MLP that is trained on the dataset $\mathfrak{D}$ to approximate the dynamics residuals and $c(\mathbf{x}) = 0$.

- *True:* where $\mathbf{m}(\mathbf{x})$ is the true dynamics residual mean and $c(\mathbf{x})$ is the trace of the true covariance times $\sup_{\mathbf{x}\in\mathbb{R}^{n_x}} \|\nabla^2 h(\mathbf{x})\|_2$.

Simulation Results: For simulation, we use the dynamics residual model:

$$p(\mathbf{d}|\mathbf{x}) = \mathcal{N}(\mathbf{d}; \underbrace{\mathbf{0}_9}_{\boldsymbol{\mu}(\mathbf{x})}, \underbrace{I_9 \times (1 + 50e^{-30z^2}) \times 10^{-5}}_{\text{cov}(\mathbf{x})}) \tag{7.15}$$

where the disturbance grows as the drone approaches the ground to approximate complicated ground effects.

To collect training data (13,320 data points), we flew the drone in simulation using an SE(3) stabilization controller [94] from 1 meter in the air to the ground 20 times for 2 seconds each with a control and data collection frequency of 333Hz.

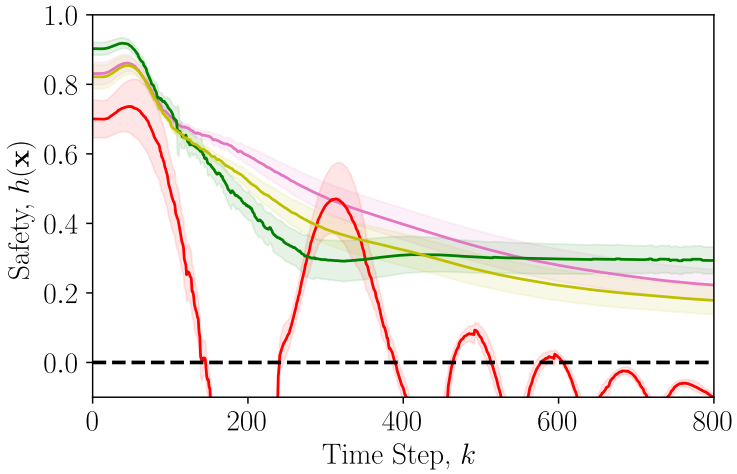


Figure 7.4. Mean and one standard deviation of $h(\mathbf{x})$ for the "drop" test case in hardware, which drops the drone with $\mathbf{k}_{\text{nom}} = 0$ from a height of roughly two meters. We compare our proposed controller (7.10) with three ablations: a deterministic MLP, a simple mean and covariance estimation across all trajectory data (*JED*), and a standard DCBF controller. All controllers except the standard DCBF satisfy the safety constraint $h(\mathbf{x}) \geq 0$, where the standard DCBF fails due to the inaccurately modelled dynamics. While the residual dynamics include complex aerodynamic effects, in this case they are low-variance, so the MLP and 7.10 to perform similarly as expected.

Each controller was simulated for 100 two-second long trajectories at 333Hz with $\alpha = 0.9975$. Results for these simulations are shown in Fig. 7.3. The looseness of the probability bound is in part due to the fact that the covariance is small for a large portion of the trajectory, which is not leveraged by the martingale-based bound for additional bound-tightness. Despite the loose risk probability bound, (7.10) produces behavior which is similar to (*True*) and which is less conservative than (*JED*) while still being more robust than (*Standard*) and (*MLP*).

Hardware Platform: Next, we deploy our ORIO controller (7.10) on a quadrotor drone flying aggressively near the ground. For all tests, we use a motion capture system to provide the drone with real-time position measurements. For onboard computation, the drone is equipped with an Nvidia Jetson Tx2 that is used to perform all neural network forward passes and evaluate the optimization-based controllers. The mean and covariance of the dynamics residuals are approximated using the CVAE with S =200 samples at 100 Hz and the ORIO optimization problem in (7.10) is an SOCP that is solved using an embedded conic solver [124] at 300 Hz. Approximately 2 minutes of training data was collected via human-operated flight for both experiments.

Hardware Experiment 1: Ground Effects Our first experiment is a "drop test" where

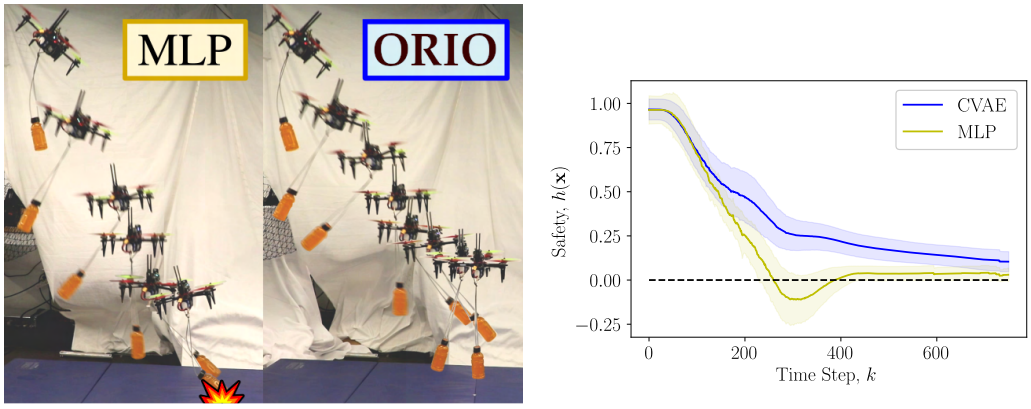


Figure 7.5. Slung mass quadrotor experiments. **(Top)** The drone is dropped from the top left and moves to the right as it falls while carrying an orange payload. The left shows a failure case when controlled by the MLP controller and the right shows a success from the same initial condition when controlled by the ORIO controller. **(Bottom)** The average of 14 trajectories is plotted with one standard deviation shading. The ORIO controller successfully keeps the system safe while the MLP-based controller results in safety failures. The video of these experiments can be found at [231].

we drop the drone from a hover at approximately 2m and enforce the barrier constraint (7.13) with $\alpha = 0.9975$ for positions above the ground; this case has low noise but requires accurate estimation of the quadrotor's thrust / ground effects for the barrier to be effective in preventing ground collision. Figure 7.4 plots the mean barrier value $h(\mathbf{x})$ over 50 trials for each ablation of our method, with one standard deviation shaded around the mean. All controllers except the standard CBF (which nearly immediately becomes unsafe due to the inaccurate modeling) exhibit safe behavior. Of particular interest is the extremely similar behavior of the simple MLP and CVAE methods; this result is intuitive since the low-variance disturbance allows the MLP to accurately capture the unmodeled dynamics. This provides an interesting insight: learning residual dynamics via simple regression, as in [37], [40], [239], is well-posed for systems subject to deterministic, low-variance disturbances, and can yield safe, performant behavior without reasoning about stochasticity.

Hardware Experiment 2: Slung Mass In our second test, the quadrotor is carrying a slung, unmodeled load of 0.55kg attached via a flexible cable, which induces large, chaotic disturbances that are not uniquely determined by the current state of the drone $\mathbf{x}$.Here we again define safety using (7.13) which is adjusted to prevent the slung mass from contacting the ground[3] and we implement the controllers with a heuristically chosen $\alpha = 0.995$.

[3]To define safety using (7.13) for the hardware experiments with the slung payload, we assume the length of the cable and the payload are known *a priori*, but we do not assume that the payload mass is known.

For this test, since the state of the slung load is unknown and since its motion is chaotic, the disturbances appear to be random and high variance when conditioned on only the drone's current state. Here we compare only our proposed method (7.10) and the MLP; as expected, in this noisy case the our CVAE-based method performs significantly better as seen in Fig. 7.5, and has no safety violations, whereas the MLP controller leads to safety failures. Supplementary videos of the experiments can be found at [231]. This experiment demonstrates the CVAE's ability to learn a sufficiently accurate stochastic model of very noisy dynamics (including trajectories where the slung load reached nearly 90 degree angles) and also the importance of accounting for stochasticity to ensure safety in scenarios where uncertainties are large.

Conclusions

In this section, we presented a unified framework for risk-sensitive control that combines CVAEs, which learn stochastic dynamics residual models from trajectory data, with DCBFs, which provide probabilistic safety guarantees for stochastic systems. We demonstrate the real-time utility of this framework by running the full pipeline (after training) at 100Hz onboard a quadrotor drone performing aggressive flight, including a free fall and flight with a slung payload.

7.3 Obstacle Avoidance Using a Humanoid with RL-based Locomotion

Next, we extend the ideas of the pervious section to the problem of collision avoidance for a humanoid robot.

As learning-based controllers achieve remarkable success in complex robotic tasks such as legged locomotion [36], [38], [240] they bring with them a fundamental tension: the black-box, data-driven nature of many learning-based controllers, which enables their robust performance, simultaneously obscures our ability to provide formal safety guarantees or modify their constraints without expensive retraining. As more roboticists begin to deploy controllers trained using strategies like reinforcement learning (RL), developing ways to flexibly and adaptively constrain their behavior online to ensure safety remains an open problem [187]. Solving this problem is especially critical for humanoid robots, which have many human-interactive use cases.

With the successful deployment of the probabilistic safety methods of Chapter 6 in the previous section, we look to use a similar approach to achieve safe behavior on a humanoid robot operating in complex environments with an RL-based locomotion

controller.

To tackle this problem, this section presents SHIELD (Safety on Humanoids via CBFs In Expectation on Learned Dynamics), a layered safety framework that enables safety for hierarchical systems with signifcant uncertainty by: (1) training a generative, stochastic dynamics residual model using real-world data from hardware rollouts of a nominal, RL-based locomotion controller, capturing system behavior and uncertainties; and (2) adding a safety layer on top of the nominal controller that leverages this model via a stochastic discrete-time CBF formulation enforcing safety constraints in probability. We then deploy this method on a Unitree G1 humanoid robot to enable safe navigation (obstacle) avoidance through varied indoor and outdoor environments using an RL-based locomotion controller and onboard perception.

The contributions of this section are as follows:

- The SHIELD (Safety on Humanoids via CBFs In Expectation on Learned Dynamics) paradigm for improved tracking and guaranteed safety of stochastic, hierarchical robotic systems.
- Experimental demonstrations of this framework on a Unitree G1 Humanoid robot conducting comprehensive obstacle avoidance experiments, including in unstructured environments, that show distinct improvements over traditional DCBF safety methods.

The text for this section is adapted from:

> L. Yang, B. Werner, R. K. Cosner, D. Fridovich-Keil, P. Culbertson, and A. Ames, "Shield: Safety on humanoids via cbfs in expectation on learned dynamics," *submitted to the 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025. [Online]. Available: `https://arxiv.org/pdf/2505.11494`,

A video for this section can be found at [241].

This section introduces SHIELD, a novel paradigm for guaranteeing safety in robotic systems that bridges the gap between data-driven and model-based safety methods. SHIELD is specifically designed for systems with complex, robust, but ultimately stochastic, low-level controllers, such as RL policies used by humanoid robots for

locomotion. Unlike traditional safety filters [26], SHIELD functions as a safety layer that sits "above" the nominal learning-based controller in the autonomy stack, modulating the reference signal rather than directly filtering control outputs, an approach similar to the model-free safety of Section 3.3.

SHIELD is constructed through a three-step process:

Step 1: Constraint specification. The user specifies a safety requirement on a subset of the robot states (e.g., the pose of the robot torso). The low-level policy does not need to be trained to satisfy this constraint but can instead be designed to track general reference commands provided to the reduced-order model (as is typical for RL [36], [242]).

Step 2: Dynamics residual learning. The user collects real-world data of the low-level policy being executed and trains a conditional variational autoencoder (CVAE) to model the difference between the desired motion of the reduced-order model, and the closed-loop system's real-world tracking of these commands.

Step 3: Safety-aware reference generation. The learned residual distribution from the CVAE is used to compute reference commands that reduces the difference between the desired and executed motion while also satisfying a stochastic discrete-time control barrier function (DCBF) [59], [131] constraint. The result is improved tracking (Fig. 7.6) and risk-aware safe behavior (Fig. 7.7).

Reduced-Order Model and Safety Goal

In this section we consider robots whose full-order model is given by a discrete-time dynamical systems of the form:

$$\mathbf{s}_{k+1} = \mathbf{\Phi}(\mathbf{s}_k, \mathbf{a}_k) \tag{7.16}$$

where $\mathbf{s}_k \in \mathbb{R}^{n_s}$ is the state of the system and $\mathbf{a} \in \mathbb{R}^{n_a}$ is the system input. This may be the high-dimensional representation of the system where $\mathbf{s}$ includes global pose, joint angles, joint angular velocities, etc., and $\mathbf{a}$ may be joint torques, voltages, desired joint positions, etc. For this complex system, we assume that we have some controller $\boldsymbol{\pi} : \mathbb{R}^{n_s} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_a}$ that takes the current system state $\mathbf{s}$ and user command $\mathbf{u} \in \mathbb{R}^{n_u}$ to produce full-order system inputs $\mathbf{a}$. Using this controller yields:

$$\mathbf{s}_{k+1} = \mathbf{\Phi}(\mathbf{s}_k, \boldsymbol{\pi}(\mathbf{s}_k, \mathbf{u}_k)). \tag{7.17}$$

For navigation purposes, we consider a reduced-order representation of the system $\mathbf{x} \in \mathbb{R}^{n_x}$ where $n_x < n_s$ and $\mathbf{x} = \mathbf{p}(\mathbf{s})$ for some projection $\mathbf{p} : \mathbb{R}^{n_s} \to \mathbb{R}^{n_x}$ that projects the full-order state $\mathbf{s}$ onto the reduced-order state $\mathbf{x}$. Here $\mathbf{x}$ may be the outputs of the system that are considered in safety and navigation, such as its center of mass position, similar to the model-free safe-set synthesis approach of Section 3.3 or the output-based back-stepping aproach of Section 3.4.

We can then represent the discrete-time dynamics of this reduced-order state of the system as:

$$\mathbf{x}_{k+1} = \mathbf{p}(\mathbf{\Phi}(\mathbf{s}_k, \boldsymbol{\pi}(\mathbf{s}_k, \mathbf{u}_k))), \tag{7.18}$$

$$\approx \mathbf{F}_R(\mathbf{x}_k) + \mathbf{G}_R(\mathbf{x}_k)\mathbf{u}_k + \mathbf{d}_k \tag{7.19}$$

where $\mathbf{F}_R(\mathbf{x}_k) + \mathbf{G}_R(\mathbf{x}_k)\mathbf{u}_k$ represents a simplified model of the system and $\mathbf{d}$ is the difference between the full-order model and this reduced-order model, also called the dynamics residual. To capture the complexities of the full-order dynamics $\mathbf{\Phi}$ and the controller $\boldsymbol{\pi}$, we consider $\mathbf{d}_k$ to be a random disturbance sampled from a distribution $\mathcal{D}(\mathbf{s}_{k:0}, \mathbf{a}_{k:0})$ that is dependent on the history of full states and from time 0 through k, denoted as $\mathbf{s}_{k:0}$ and $\mathbf{a}_{k:0}$, respectively, that define the filtration $\mathscr{F}_k \triangleq \{\mathbf{s_k}, \mathbf{s_{k-1}}, ..., \mathbf{s_0}, \mathbf{a_k}, \mathbf{a_{k-1}}, ..., \mathbf{a_0}\} \subset \mathscr{F}_{k+1} \subset \cdots \subset \mathscr{F}$.

This represents a specific instantiation of the discrete-time stochastic system (6.7) considered throughout Chapters 6 and 7 where the model of the discrete-time model is control affine and the disturbances are a function of the mismatch between the full-order system with state $\mathbf{s}$ and the reduced-order representation with state $\mathbf{x}$.

For this system, we will again consider $K-$step exit probability (Def. 6.7) as our safety metric over a finite horizon of length $K \in \mathbb{N}_1$.

In particular, this work will seek to realize the probabilistic DCBF safety bound based on Freedman's inequality of Section 6.4, Theorem 6.24 on a bipedal system of the form (7.19).

To apply this theorem, we require two assumptions: first, a bound on the safety variance as in (6.56); second, a bound on the difference between the true safety value $h(\mathbf{x}_k)$ and the expected value as in (6.55). The first assumption is not very restrictive and allows for a large class of potential functions h, dynamics, and disturbance distributions. The second assumption is more restrictive, but generally applies in our setting, as the worst-case falling behavior would lead to a bounded difference between the commanded and true reduced-order-model behavior.

Disturbance Learning

While theoretical safety guarantees such as those in Chapter 6 provide powerful methods for analyzing and synthesizing risk-aware controllers, their guarantees fundamentally depend on accurate characterization of the disturbance distribution $\mathcal{D}$. Rather than assuming that this distribution is known *a priori* or constrained to a simplified parametric form (e.g., additive Gaussian noise), we use a data-driven approach, similar to Section 7.2, that leverages generative modeling to learn these distributions directly from empirical trajectories of the system. This approach enables us to capture complex, non-Gaussian, and state-dependent uncertainty distributions that more faithfully represent the actual disturbances encountered during hardware operation.

To account for these dynamics residuals, we seek to first train a generative model to approximate their distribution. To do this, we collect a dataset of state, command, and disturbance tuples $\mathfrak{D} = \{(\mathbf{x}_i, \mathbf{u}_i, \mathbf{d}_i)\}_{i=1}^{N_{\text{samples}}}$. We then train a Conditional Variation Autoencoder (CVAE) [224] on this dataset, which yields a generative disturbance model $p_\theta(\mathbf{d}_k | \mathbf{x}_{k:k-N}, \mathbf{u}_{k:k-N})$. In contrast to the previous section, the model is conditioned on a *context window* of length $N \in \mathbb{N}$, this was found to be crucial to allow the model to capture temporal effects such as higher-order state derivatives or time delays. We find that providing this context greatly boosts modeling accuracy for a complex, hierarchical systems of the form (7.18,7.19).

We note that any class of generative disturbance model (e.g., diffusion [236], flow matching [243], etc.) can be used with our proposed safety framework. For SHIELD we choose to use CVAEs due to their expressivity and fast inference time, as shown empirically in the previous section (and [59]).

Stochastic Tracking and Safety with Learned Disturbances

The proposed method of this section, SHIELD, distinguishes itself from conventional safety layers through how it modulates control signals. While traditional approaches [6], [10], [59] operate by modifying low-level signals (such as joint torques or raw actuation commands) to maintain safety, SHIELD instead modulates higher-level signals, i.e., the reference commands provided to the reduced-order model. This architecture is similar to that of a *reference governor* [244], which modulates reference or command signals into the controller/plant; the key difference is we modulate these signals with a CBF and without knowledge of the actual controller and plant dynamics. This modification enables the definition of safety

constraints on simpler, more semantically meaningful states, making the system both more interpretable and manageable.

SHIELD recognizes that the ultimate objective is to achieve the intended system behavior, meaning the system should accurately track the reduced-order model's intended trajectory. To derive this "best-tracking" control, we define the optimal reference command $\mathbf{u}_k^*$ as the one that minimizes the expected difference between the next state of the reduced-order model under the desired command and the next state of the actual system:

$$\mathbf{u}_k^* = \underset{\mathbf{u}_k \in \mathcal{U}}{\operatorname{argmin}} \ \mathbb{E}[||\overline{\mathbf{x}}_{k+1} - (\mathbf{F}_R(\mathbf{x}_k) + \mathbf{G}_R(\mathbf{x}_k)\mathbf{u}_k + \mathbf{d}_k)||^2 | \mathscr{F}_k] \tag{7.20}$$

where $\overline{\mathbf{x}}_{k+1}$ is the desired next position. Assuming pseudo-invertibility of $\mathbf{G}_R(\mathbf{x}_k)$, the optimal $\mathbf{u}$ is[4]:

$$\mathbf{u}_k^* = \mathbf{G}_R^\dagger(\mathbf{x}_k)(-\mathbf{F}_R(\mathbf{x}_k) + \overline{\mathbf{x}}_{k+1} - \mathbb{E}[\mathbf{d}_k | \mathscr{F}_k]). \tag{7.21}$$

However, since we do not have access to the true expectation $\mathbb{E}[\mathbf{d}_k|\mathscr{F}_k]$, we approximate this with the learned expectation computed from samples generated by the CVAE:

$$\mathbf{u}_k^* = \mathbf{G}_R^\dagger(\mathbf{x}_k)(-\mathbf{F}_R(\mathbf{x}_k) + \overline{\mathbf{x}}_{k+1} - \mathbb{E}_{p_\theta}[\mathbf{d}_k | \mathbf{x}_{k:k-N}, \mathbf{u}_{k:k-N}]). \tag{7.22}$$

This $\mathbf{u}_k^*$ uses the learned disturbance distribution to select the command which reduces the mean squared error to the desired next state $\overline{\mathbf{x}}_k$.

In addition to using the learned dynamics residual to improve tracking, we can also use it to enforce safety. To do this, we select a maximum allowable risk level $P \in (0, 1)$. Given the horizon length $K \in \mathbb{N}_1$, the initial safety value $h(\mathbf{x}_0)$, the step-wise bound δ from assumption (6.55), and the variance bound σ from assumption (6.56) we can solve for the α that will result in the desired risk level bound $P_r \in (0, 1)$:

$$\alpha = L(P_r, K, h(x_0), \delta, \sigma). \tag{7.23}$$

In practice, we approximate $L : (0, 1) \times \mathbb{N} \times \mathbb{R}_{>0} \times \mathbb{R}_{>0} \times \mathbb{R}_{>0} \to (0, 1)$ numerically due to the complexity of the analytic solution.

[4]The derivation of this follows from the equality $\mathbb{E}[||\overline{\mathbf{x}}_{k+1} - (\mathbf{F}_R(\mathbf{x}_k) + \mathbf{G}_R(\mathbf{x}_k)\mathbf{u}_k + \mathbf{d}_k)||^2 | \mathscr{F}_k] = ||\overline{\mathbf{x}}_{k+1} - (\mathbf{F}_R(\mathbf{x}_k) + \mathbf{G}_R(\mathbf{x}_k)\mathbf{u}_k + \mathbb{E}[\mathbf{d}_k|\mathscr{F}_k])||^2 + \mathbb{E}[\|\mathbf{d}\|^2|\mathscr{F}_k] - \|\mathbb{E}[\mathbf{d}|\mathscr{F}_k]\|^2$. Since this is true, it suffices to find the optimal $\mathbf{u}$ for $||\overline{\mathbf{x}}_{k+1} - (\mathbf{F}_R(\mathbf{x}_k) + \mathbf{G}_R(\mathbf{x}_k)\mathbf{u}_k + \mathbb{E}[\mathbf{d}_k|\mathscr{F}_k])||^2$ which is (7.20).

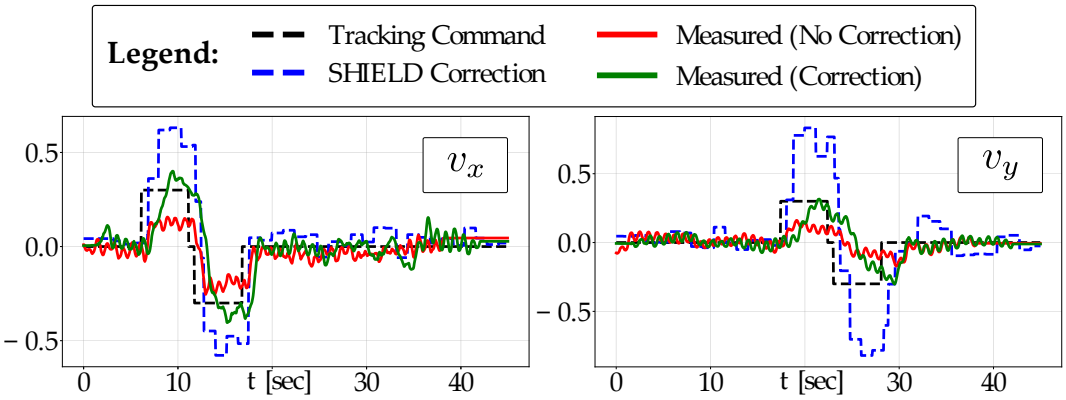


Figure 7.6. SHIELD improves tracking performance by correcting learned disturbances. After applying the SHIELD correction as shown by the blue dashed lines, the robot's tracking of the user's intended velocities (shown as a black dashed lines) improves.

Next to apply Theorem 6.24 to our application, we address each assumption, (6.56) and (6.55). Firstly, for assumption (6.56), the variance bound σ^2 is approximated from the sampled dataset $\mathfrak{D}$. Secondly, for assumption (6.55), we derive a bound from our application to bipedal robots and bound the difference between the true and predicted update for $h(\mathbf{x}_k)$ based upon the maximum step distance which can be measured in practice:

$$\delta \triangleq 2(h(\mathbf{x}_{\text{footstep } k}) - h(\mathbf{x}_{\text{footstep } k+1})). \tag{7.24}$$

In addition to meeting assumptions (6.56) and (6.55), we must also enforce the expectation-based DCBF inequality (6.16) inequality, which we incorporate, for concave and continuously-differentiable h with bounded second derivative (i.e., $\sup_{\mathbf{x}\in\mathbb{R}^{n_x}} \|\nabla^2 h(\mathbf{x})\|_2 \leq \lambda_{\max}$ for some $\lambda_{\max} \in \mathbb{R}_{\geq 0}$), as a constraint in the safety filter using the alternative, lower-bounding condition from Theorem 6.15:

$$\begin{aligned} \mathbf{u}^*_{\text{safe}} = \underset{\mathbf{u}\in\mathcal{U}}{\text{argmin}} \quad & \|\mathbf{u} - \mathbf{u}^*_k\| \\ \text{s.t. } & h(\mathbf{F}_R(\mathbf{x}_k) + \mathbf{G}_R(\mathbf{x}_k)\mathbf{u}_k + \mathbb{E}_{p_\theta}[\mathbf{d}_k|\mathbf{x}_{k:0}, \mathbf{u}_{k:0}]) \\ & - \frac{\lambda_{\max}}{2}\text{tr}(\text{cov}_{p_\theta}(\mathbf{d}_k|\mathbf{x}_{k:0}, \mathbf{u}_{k:0}) \geq \alpha h(\mathbf{x}_k), \end{aligned} \tag{7.25}$$

where we can approximate $\mathbb{E}[\mathbf{d}_k|\mathscr{F}_k]$ and $\text{cov}(\mathbf{d}_k|\mathscr{F}_k)$ using the learned dynamics residual distribution $p_\theta(\mathbf{d}_k|\mathbf{x}_{k:0}, \mathbf{u}_{k:0})$.

In summary, SHIELD uses the learned dynamics residual distribution from the CVAE to compute two distinct quantities: (1) the optimal reference command (7.22) that minimizes the expected tracking error between the true system and desired next

state, and (2) a minimally adjusted safe reference command (7.25) that enforces probabilistic safety constraints. We emphasize that these components are fully modular. The tracking-optimized input can be used independently to reduce the sim-to-real gap, while the safety adjustment can be applied separately to enhance real-world safety guarantees. Alternatively, both components can be combined sequentially to simultaneously improve tracking performance and safety assurance, providing flexibility for different application requirements.

Dynamic Obstacle Avoidance on Stochastic Reduced-Order Models

Next, we detail our approach to improve tracking and safety under random uncertainty with a stochastic reinforcement learning-based controller $\boldsymbol{\pi}$. In particular, we use a PPO Actor-Critic learned controller $\boldsymbol{\pi}_{\text{PPO}}$. This takes into account histories of proprioceptive and extereoceptive states $\mathbf{s}$ and a commanded velocity vector $\mathbf{u} = (v_x, v_y, \omega)$ using an LSTM and uses those to generate joint positions, $\mathbf{a}$.

To characterize the stochasticity of this controller, we use a CVAE to learn the distribution of the dynamics residual $\mathbf{d}$ conditioned on the last four[5] system states and commands, i.e., $(\mathbf{x}_{k:k-3}, \mathbf{u}_{k:k-3})$. Specifically, we use a single integrator system with an additive disturbance as our simplified model:

$$\underbrace{\begin{bmatrix} p_x \\ p_y \\ \theta \end{bmatrix}_{k+1}}_{\mathbf{x}_{k+1}} = \underbrace{\begin{bmatrix} p_x \\ p_y \\ \theta \end{bmatrix}_{k}}_{\mathbf{F}_R(\mathbf{x}_k)} + \underbrace{\Delta_t \mathbf{I}_3}_{\mathbf{G}_R(\mathbf{x}_k)} \underbrace{\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}_{k}}_{\mathbf{u}_k} + \underbrace{\Delta_t \begin{bmatrix} d_x \\ d_y \\ d_\theta \end{bmatrix}}_{\mathbf{d}_k} \tag{7.26}$$

where $p_x, p_y \in \mathbb{R}$, $\theta \in [0, 2\pi)$, and $\Delta_t > 0$ represent the x and y position, the yaw angle, and the state-update period and where $\mathbf{d}_k$ is a random disturbance that models the difference between the simplified model and the true dynamics.

The tracking improvement method (7.22) leads us to the optimal tracking command:

$$\mathbf{u}_{\text{adjusted}} = \frac{\overline{\mathbf{x}}_{k+1} - \mathbf{x}_k}{\Delta_t} - \mathbb{E}_{p_\theta}[\mathbf{d} | \mathbf{x}_{k:k-3}, \mathbf{u}_{k:k-3}] \tag{7.27}$$

where $\mathbb{E}_{p_\theta}[\mathbf{d} | \mathbf{x}_{k:k-3}, \mathbf{u}_{k:k-3}]$ is the mean disturbance learned by the CVAE. After modifying the command velocity with the predicted dynamics residual to improve tracking, we apply our safety filter which minimally modifies that command to enforce our safety constraint. For application, we consider obstacle avoidance with

[5] In practice, we condition on the last $N = \min(k, 4)$ states and commands for the algorithm to run at start time.

respect to $N_{\text{obs}} \in \mathbb{N}$ obstacles as characterized by the signed distance function (sdf):

$$\text{sdf}(\mathbf{x}) = \min_{i \in \{1,\dots N\}} \left\| \begin{bmatrix} p_x \\ p_y \end{bmatrix} - \boldsymbol{\rho}_i \right\| - R_i \tag{7.28}$$

where $\boldsymbol{\rho}_i \in \mathbb{R}^2$ is the planar position of obstacle i and $R_i > 0$ is the robot radius plus the obstacle radius. The SDF is then used to produce the function defining the safety requirement:

$$h_{\text{smooth}}(\mathbf{x}_k) = \lambda(1 - e^{-\gamma \text{sdf}(\mathbf{x}_k)}) \tag{7.29}$$

where $\lambda > 0$, $\gamma > 0$ are positive constants controlling the maximum magnitude and smoothness of safety.

Since we are only considering the closest obstacle, we make the following concave approximation:

$$\widehat{h}(\mathbf{x}) = \lambda \left(1 - e^{-\gamma((\mathbf{p}-\boldsymbol{\rho}_i)^T \mathbf{e}_i - R_i)}\right) \tag{7.30}$$

$$\tilde{h}(\mathbf{x}) = \begin{cases} \widehat{h}(\mathbf{x}), & \text{if } (\mathbf{p} - \boldsymbol{\rho}_i)^\top \mathbf{e}_i \geq 0 \\ \nabla_{\mathbf{x}} \widehat{h}(\mathbf{x}) + \lambda(1 - e^{\gamma R_i}), & \text{else} \end{cases} \tag{7.31}$$

where $\mathbf{p} \triangleq [p_x, p_y]^T$ and $\mathbf{e}_i \in \mathbb{R}^2$ with $||\mathbf{e}_i||_2 = 1$ is the unit direction towards the closest obstacle from the previous timestep, i.e., $(\mathbf{p}_k - \boldsymbol{\rho}_i)/\|\mathbf{p}_k - \boldsymbol{\rho}_i\|$.

In the case of a single obstacle, we provide the following inequality which will allow us to build conditions that enforce a bound on the $K-$step failure probability in practice:

Proposition 7.3 (Single-Obstacle Avoidance with Concave Barrier Functions)**.** *Consider the function $\tilde{h}$ as in* (7.31) *with $N = 1$ and a random variable $\mathbf{x}$ that takes values in $\mathbb{R}^{n_x}$ with $\mathbb{E}[\|\mathbf{x}\|_2] < \infty$ and $\|\text{cov}(\mathbf{x})\| < \infty$. This function $\tilde{h}$ and random variable $\mathbf{x}$ satisfy:*

$$\mathbb{E}\left[\tilde{h}(\mathbf{x})\right] \geq \tilde{h}(\mathbb{E}[\mathbf{x}]) - \frac{\lambda_{\max}}{2} \mathbf{e}_1^T \text{cov}(\mathbf{x}) \mathbf{e}_1. \tag{7.32}$$

Please see the appendix of the extended version of this paper for the proof [17].

This allows us to enforce the expectation-based DCBF constraint (6.16) for concave, continuously differentiable h indirectly by instead enforcing the tightened constraint:

$$\begin{aligned} h(\mathbf{F}_R(\mathbf{x}_k) + \mathbf{G}_R(\mathbf{x}_k)\mathbf{u}_k + \mathbb{E}_{p_\theta}[\mathbf{d}_k | \mathbf{x}_{k:k-3}, \mathbf{u}_{k:k-3}]) \\ - \frac{\lambda_{\max}}{2} \mathbf{e}_i^T \text{cov}_{p_\theta}(\mathbf{d}_k | \mathbf{x}_{k:k-3}, \mathbf{u}_{k:k-3}) \mathbf{e}_i \geq \alpha h(\mathbf{x}_k) \end{aligned} \tag{7.33}$$

where we can approximate $\mathbb{E}[\mathbf{d}_k|\mathscr{F}_k]$ and $\text{cov}(\mathbf{d}_k|\mathscr{F}_k)$ using the learned dynamics residual distribution $p_\theta(\mathbf{d}_k|\mathbf{x}_{k:k-3}, \mathbf{u}_{k:k-3})$. In practice, we find that the utility of SHIELD generalizes to multiple obstacles; however, we leave a rigorous theoretical analysis of the nonconcave $\tilde{h}$ with multiple obstacles for future work.

Algorithm 7.4: SHIELD: Deployment Phase

Initialize $k \leftarrow 0, \mathbf{x} \leftarrow \mathbf{x}_0$
Initialize P, δ, α
while *true* **do**
  obstacles $\leftarrow \{\boldsymbol{\rho}_1, ..., \boldsymbol{\rho}_M\}$
  $h_k \leftarrow \min_i \tilde{h}(\mathbf{x}, \boldsymbol{\rho}_i)$, $i^* \leftarrow \arg\min_i \tilde{h}(\mathbf{x}, \boldsymbol{\rho}_i)$
  if k modulo $K = 0$ **then**
    $\Sigma \leftarrow \text{cov}_{p_\theta}(\mathbf{d}|\mathbf{x}_{k:k-N}, \mathbf{u}_{k:k-N})$
    $\alpha \leftarrow L(K, h_k, P, \delta, \Sigma)$
  Get $\mathbf{u}_{\text{cmd}}$ as input
  $\mathbf{u}_{\text{adjusted}} \leftarrow \mathbf{u}_{\text{cmd}} - \mathbb{E}_{p_\theta}[\mathbf{d}|\mathbf{x}_{k:k-3}, \mathbf{u}_{k:k-3}]$
  $\mathbf{e} \leftarrow \frac{\mathbf{p}_k - \boldsymbol{\rho}_{\text{obs},i^*}}{||\mathbf{p}_k - \boldsymbol{\rho}_{\text{obs},i^*}||}$, $\lambda \leftarrow \lambda_{\max}(\boldsymbol{\rho}, \mathbf{e})$
  $\mathbf{u}^*_{\text{safe}} \leftarrow \min_{\mathbf{u}} \|\mathbf{u} - \mathbf{u}_{\text{adjusted}}\|^2$
    s.t. $\tilde{h}(\mathbf{F}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}) - \frac{\lambda}{2}\mathbf{e}^T\Sigma\mathbf{e} \geq \alpha h_k$
  Apply command $\mathbf{u}^*_{\text{safe}}$, $\mathbf{x_k} \leftarrow \mathbf{x_{k+1}}$, $k \leftarrow k+1$

To determine the appropriate α for a particular risk level, we use the L function in (7.23). To calculate α, a desired risk level P_r is chosen, the current safety value is noted as $h(\mathbf{x}_k)$, the worst-case δ is approximated as in (7.24), and the covariance σ is set to the maximum value experienced in the experimental data. Furthermore, to extend the guarantee beyond K steps, we recalculate α every K steps. Thus, each successive K steps satisfies the bound in Theorem 6.15 and they can be connected using the union bound:

$$\mathbb{P}\left\{\min_{k\in[0,K\times F]} h(\mathbf{x}_k) < 0\right\} \leq \sum_{i=0}^{F} \mathbb{P}\left\{\min_{k\in[Ki,K(i+1)]} h(\mathbf{x}_k) < 0\right\}$$

where $F \in \mathbb{N}_1$ is the number of K-step intervals in the experiment. We show the SHIELD deployment stage with combined tracking and safety improvements in Algorithm 7.4.

Experimental Deployment: Bipedal Obstacle Avoidance

Finally, we demonstrate the validity of SHIELD on a Unitree G1 humanoid robot, to show the method's adaptable conservativeness, performance, and robustness.

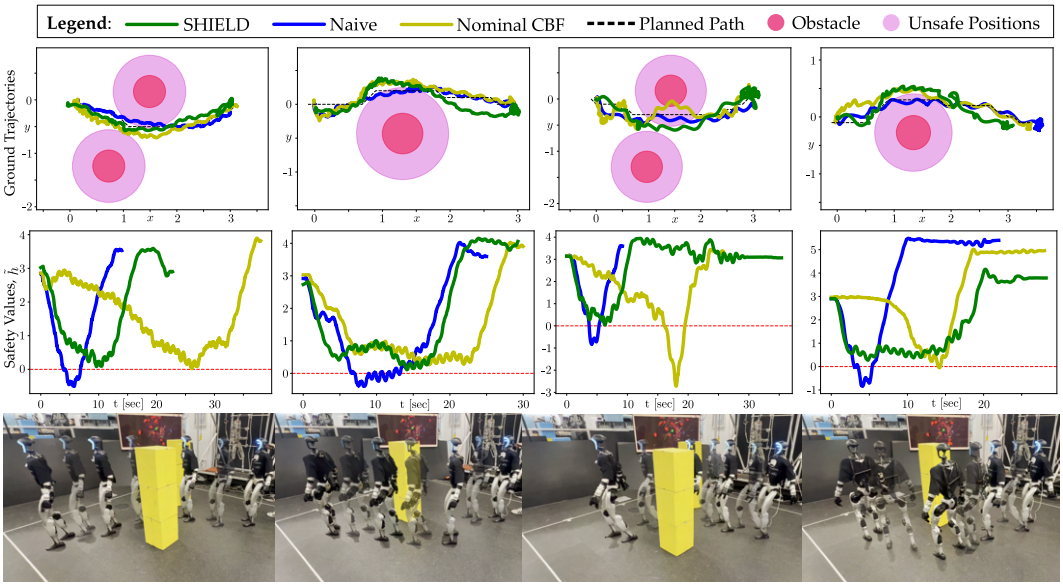


Figure 7.7. SHIELD enforces safety in collision avoidance with adaptive conservatism. The A* planner path is not necessarily safe even it does not cross the obstacle, thus naively following the path would result in collisions or scrapes. Nominal CBF, due to not accounting for the inaccurate reduced-order model, would also result in collisions or be extremely conservative.

Hardware Setup The Unitree G1 humanoid robot has a height of 1.32 meters and weighs approximately 40kg, with 23 actuated degrees of freedom. We employ an onboard Jetson Orin NX for computation, a Livox Mid-360 LiDAR for sensing the environment, and an Intel T265 to localize the robot. Euclidean clustering [245] is applied to the LiDAR pointcloud to locate obstacles of interest in the scene.

To test the generalization of SHIELD in deployment, we conduct experiments with two different walking controllers: (1) *built-in*: the Unitree built-in controller [246] and (2) *custom*: a custom RL locomotion controller trained in IsaacLab [247] using standard rewards from [248].

Approximately 6 minutes of training data are collected for each controller to train the CVAE for both the *built-in* and *custom* controllers. We query the CVAE to update the mean and covariance of the disturbance distribution at 0.83Hz, and we filter the command velocity at 100Hz.

Hardware: Learned Tracking Improvements We first test the velocity tracking capabilities of the SHIELD framework. In these experiments, we send a pre-set sequence of velocity commands through the framework to the controller and compare our resulting velocities to the command sequence. We achieve noticeable improvements in tracking as shown in Fig. 7.6.

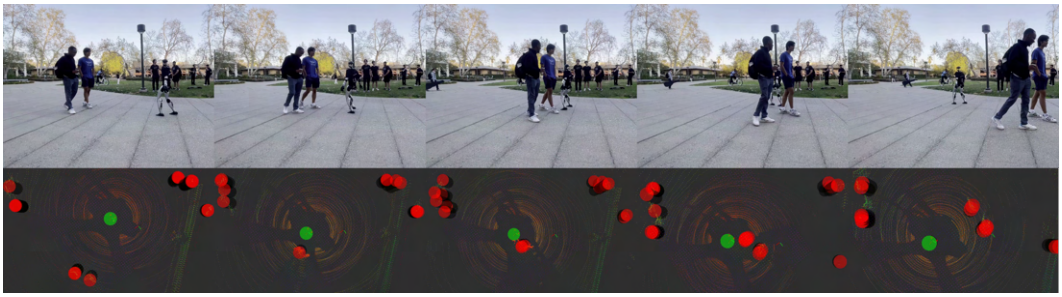


Figure 7.8. SHIELD enables real-world pedestrian avoidance with a humanoid robot, using a "general-purpose" RL policy. *Top:* Our robot safely walks among pedestrians using SHIELD's stochastic safety framework. *Bottom:* The robot relies solely on onboard perception to detect and avoid obstacles. Experimental video of this experiment can be found at [241].

Hardware: Laboratory Obstacle Avoidance First, we conduct controlled experiments with fixed obstacles. We define success as the robot walking past obstacles without making contact. We model the detected obstacles as cylinders of radius 0.3m and the robot to have a safety margin of 0.38m from the center of mass. To navigate, we first use A* [249] to first plan a path through free space, we then generate nominal velocities by directing the robot from its current position to the next node on the path and filter the commanded velocities with SHIELD. We present both single-obstacle and multi-obstacle cases. In single-obstacle experiments, naively following the A* path alone does not completely avoid obstacles due to state tracking errors. The nominal DCBF filter, being unaware of the dynamics residual, either collides into the obstacle or exhibits extremely conservative behavior with $\alpha = 0.99$. However, SHIELD enables the robot to completely bypass the obstacle. We observe similar behavior in multi-obstacle scenarios, where SHIELD is able to adjust conservativeness online to only enforce maximum safety conditions when needed, resulting in more dynamic behavior. The results of these experiments can be seen in Fig. 7.7. We do not however, that the probability bound of Theorem 6.24 is not well-calibrated for this scenario and instead functions best as a "tuning knob" to encourage riskier behavior at the cost of a higher chance of collision (though still far below the target percentage).

Hardware: Outdoor Obstacle Avoidance We also perform experiments in unstructured outdoor environments for further validation. In these tests, a user provides joystick inputs to the robot for safety reasons and would either control the robot to walk directly towards people or provide no input and let the robot stay in place unless people encroach on its safety boundary. These experiments can be seen in Fig. 7.8 and in the experimental video [241].

Conclusion

This section presented SHIELD: a safety layer that leverages stochastic DCBFs to achieve risk-aware safety. Importantly, SHIELD can be added to an existing autonomy stack, wherein the dynamics of the full-order system including the nominal reference-tracking controller can be learned as the residual on a simplified model. SHIELD then filters the nominal reference commands to produce safe inputs using the expectation-based DCBF condition (6.16). This framework is instantiated on a humanoid robot in the context of collision avoidance, where it is shown to outperform a nominal safety filter in hardware experiments on the Unitree G1 humanoid.

7.4 Dynamic Obstacle Avoidance

In this section, to study the problem of realizing performant risk-aware safety, we consider the problem of achieving collision avoidance with highly dynamic obstacles. While this problem appears to be a natural extension of the safety problems studied throughout the previous two sections of this work, this section will demonstrate that naive implementations of DCBF safety-filtering methods will quickly lead to practical safety failures due to the myopia of the DCBF-OP safety filters (6.4).

To overcome this shortfall, we combine the robust safety guarantees of DCBF methods with the horizon-based planning and optimization of model predictive control (MPC) to find that the unification of these methods results in mutual benefits and inherent robustness properties beyond the capabilities of either method alone. Critically, switching from the strict state-constraints typical of MPC to the decay-based constraint of DCBFs provides inherent robustness to stochastic uncertainty which will be proved in this section. On the other hand, switching from the myopic optimization of the DCBF-OP safety filter (6.4), improves the closed-loop behavior by allowing the robot to generate plans that avoid getting stuck states that are only marginally safe or from which it will eventually become unsafe.

We demonstrate benefits of the unification of these two techniques on the practical example of quadrupedal and quadrotor robots performing dynamic obstacle avoidance.

The contributions of this section are as follows:

- Improved guarantees of closed-loop feasibility (despite reduced pointwise feasibility) of the MPC-DCBF controller when compared to traditional MPC

methods.

- Demonstrations of the improved convergence properties of the MPC+DCBF and its ability to overcome the stable undesired equilibria that arise with naive DCBF-based safety filters (6.4).
- Risk-based guarantees naturally arising from the MPC-DCBF framework that extend beyond the prior work presented in Chapter 6 to consider state uncertainty.
- Examples of the MPC+DBCF controller achieving dynamic obstacle avoidance on quadrupedal and quadrotor robots, with the quadrotor example being performed with all computer vision, obstacle state estimation, and controller algorithms being performed entirely onboard.

The text for this section is adapted from:

> R. K. Cosner, R. M. Bena, and A. D. Ames, "Unified mpc+cbf control for performant safety: Mutual benefits and inherent robustness properties," *submitted to IEEE Transactions on Robotics*, 2025. [Online]. Available: `http://www.rkcosner.com/assets/files/dodgeball_paper.pdf`,

A video for this section can be found at [250].

Related Work

The stochastic safety guarantees of this section diverge significantly from the standard quantile-based methods of the MPC literature [214], [251]–[253] and leverage the safety-decay property of the DCBF constraint to create simpler-to-enforce martingale-based guarantees as in [35], [57], [58], [192], but which extends their utility to scenarios with state uncertainty and improves their performance through the addition of a planning horizon. Notably, the methods in this chapter assume the existence of a nominal or *a priori* learned models of the obstacle dynamics. Alternatively, see [214], [254] for a data-driven method that develops a model of the obstacle dynamics during deployment given noisy obstacle measurements.

The experimental demonstrations of quadrotor-based dynamic obstacle avoidance provided in this paper are similar to previous learning + event cameras [255], artificial potential fields + event cameras [256], and MPC + motion capture [257]

works on collision avoidance, but with onboard RGBd camera-based obstacle detection and improved theoretical safety guarantees and horizon-based optimization. In comparison with [254], we leverage onboard, vision-based sensing to estimate the obstacle state, employ a simpler Kalman filter-based state estimator that assumes a noisy ballistic trajectory, and use a higher-order model (3rd vs. 1st order) in the planning problem, which together allow for significantly more dynamic hardware demonstrations, despite less generality in the possible obstacle dynamics.

Model Predictive Control and Safety

We begin by introducing the MPC and MPC+DCBF control frameworks and, before generalizing to uncertain systems, we consider them in the context of deterministic discrete-time as discussed in Section 6.2 in equations (6.1, 6.2).

MPC is a a control methodology which leverages a model-based prediction of the system dynamics along a finite-horizon to compute control actions. In MPC, at each time-step k the controller plans a sequence of open-loop control actions to minimize a cost function and then the first action is applied to the system. The plan of actions is then recalculated using the updated state, and the new first control action is applied, creating a state-feedback controller.

In MPC, to calculate the plan of control actions, the following discrete, finite-time optimal control problem (FTOCP) is solved at each time-step k:

$$\begin{aligned} \min_{\substack{\boldsymbol{\xi}_{0:N}\in\mathbb{R}^{n_x}\\ \boldsymbol{\nu}_{0:N-1}\in\mathbb{R}^{n_u}}} \quad & \sum_{i=0}^{N-1} c(\boldsymbol{\xi}_i, \boldsymbol{\nu}_i) + V(\boldsymbol{\xi}_N) && \text{(FTOCP)}\\ \text{s.t.} \quad & \boldsymbol{\xi}_{i+1} = \mathbf{F}(\boldsymbol{\xi}_i, \boldsymbol{\nu}_i), \quad \forall i \in \{0,\dots,N-1\} \\ & \boldsymbol{\xi}_i \in \mathcal{C}, \quad \boldsymbol{\nu}_i \in \mathcal{U}, \quad \forall i \in \{0,\dots,N-1\} \\ & \boldsymbol{\xi}_0 = \mathbf{x}_k, \quad \boldsymbol{\xi}_N \in \mathcal{C}_N \end{aligned}$$

where $\mathcal{U} \subset \mathbb{R}^{n_u}$ represents the input constraint set (e.g., torque limits), $c : \mathbb{R}^{n_x} \times \mathcal{U} \to \mathbb{R}_{\geq 0}$ is the stage cost, and $V : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ is the terminal cost used to approximate the infinite-horizon optimal control problem. Here we use the variables $\boldsymbol{\xi}_i \in \mathbb{R}^{n_x}$ and $\boldsymbol{\nu}_i \in \mathbb{R}^{n_u}$ to represent the planned sequence of states and inputs given the current state $\mathbf{x}_k$, i.e., if the dynamics and the state are known exactly then using $\mathbf{u}_k = \boldsymbol{\nu}_0$ results in the plan being precisely executed so that $\mathbf{x}_{k+1} = \boldsymbol{\xi}_1$ for the $\boldsymbol{\xi}_1$ generated at $\mathbf{x}_k$.

In the FTOCP, the first constraint incorporates the discrete time model of the system (6.1) along the horizon of length $N \in \mathbb{N}_1$, the state constraint $\boldsymbol{\xi}_i \in \mathcal{C}$ (equivalently

$h(\mathbf{x}) \geq 0$) ensures that each state in the plan is safe, the input constraint $\boldsymbol{\nu}_i \in \mathcal{U}$ ensures that the inputs are realizable on the system, the initial condition constraint $\boldsymbol{\xi}_0 = \mathbf{x}_k$ aligns the plan with the current state, and the terminal state constraint $\boldsymbol{\xi}_N \in \mathcal{C}_N \subset \mathcal{C}$ is used to achieve recursive feasibility of the feedback controller. In general it is assumed that $\mathcal{C}_N$ is a safe, control invariant set for the inputs $\mathbf{u} \in \mathcal{U}$, in which case the MPC controller can be thought of as a domain-of-attraction expander for $\mathcal{C}_N$, similar to the backup set safe-set synthesis method of Section 3.6. For additional discussion this FTOCP, please see [5].

To generate the MPC input, the optimal plan of inputs for the FTOCP is computed as $[\boldsymbol{\nu}_0^*(\mathbf{x}_k), \ldots, \boldsymbol{\nu}_{N-1}^*(\mathbf{x}_k)]$ and then the first action is applied to the system, defining the MPC controller:

$$\mathbf{k}^{\text{MPC}}(\mathbf{x}_k) = \boldsymbol{\nu}_0^*(\mathbf{x}_k). \tag{MPC}$$

By enforcing safety in the form of a state constraint, $\boldsymbol{\xi}_i \in \mathcal{C}$, in the FTOCP, $\mathbf{k}^{\text{MPC}}(\mathbf{x})$ selects control actions which ensures the safety of the system at each discrete update of the closed-loop dynamics (6.2).

As an alternative to the standard safety constraint, a DCBF can instead be enforced along the MPC horizon. The main theoretical focus of this section is to consider the safety, performance, and robustness properties of this unified controller generated from the FTOCP+DCBF problem:

$$\begin{aligned}
\min_{\substack{\boldsymbol{\xi}_{0:N} \in \mathbb{R}^{n_x} \\ \boldsymbol{\nu}_{0:N-1} \in \mathbb{R}^{n_u}}} \quad & \sum_{i=0}^{N-1} c(\boldsymbol{\xi}_i, \boldsymbol{\nu}_i) + V(\boldsymbol{\xi}_N) && \text{(FTOCP+DCBF)} \\
\text{s.t.} \quad & \boldsymbol{\xi}_{i+1} = \mathbf{F}(\boldsymbol{\xi}_i, \boldsymbol{\nu}_i), \quad \forall i \in \{0, \ldots, N-1\} \\
& h(\boldsymbol{\xi}_{i+1}) \geq \alpha h(\boldsymbol{\xi}_i), \quad \forall i \in \{0, \ldots, N-1\} \\
& \boldsymbol{\nu}_i \in \mathcal{U}, \quad \forall i \in \{0, \ldots, N-1\} \\
& \boldsymbol{\xi}_0 = \mathbf{x}_k
\end{aligned}$$

where we replace the state constraint $\boldsymbol{\xi}_i \in \mathcal{C}$ (i.e., $h(\boldsymbol{\xi}_i) \geq 0$) with the DCBF constraint (6.3) for some $\alpha \in [0, 1]$ and we remove the terminal constraint $\boldsymbol{\xi}_N \in \mathcal{C}_N$.

As with MPC, we can then derive a controller from the FTOCP+DCBF by using the first input of the open-loop plan:

$$\mathbf{k}^{\text{MPC+DCBF}}(\mathbf{x}_k) = \boldsymbol{\nu}_0^*(\mathbf{x}_k). \tag{MPC+DCBF}$$

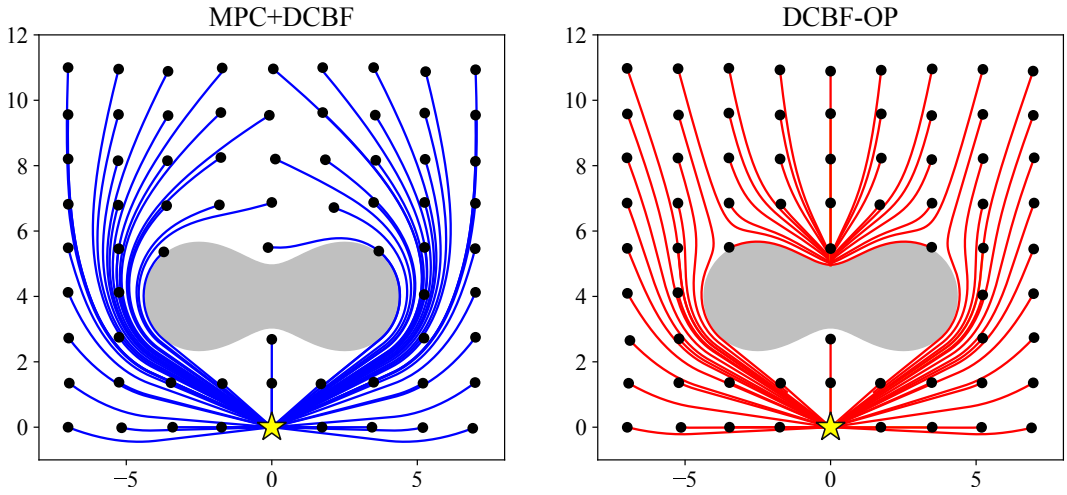


Figure 7.9. Closed-loop trajectories from several initial conditions (black dots) for the system in Ex. 7.5 using $\mathbf{k}^{\text{MPC+DCBF}}$ (blue) and using the DCBF safety filter (6.4) (red). Here we see that $\mathbf{k}^{\text{MPC+DCBF}}$ is far better at avoiding the undesired equilibrium point and reaching the goal at (0,0) shown as a yellow star.

Improvements to Undesirable Equilibrium

Although the DCBF-based safety filters introduced so far in this thesis provide valuable safety guarantees and stochastic robustness properties, they handle performance only indirectly by modifying a nominal controller as little as possible to satisfy the safety constraint. While this safety-filtering approach is powerful, it can lead to undesired behavior when the nominal input is unsafe, often resulting in convergence to safe but undesirable equilibrium points [16]. Since we seek closed-loop systems that are both safe *and* performant, it is beneficial to jointly optimize for performance while enforcing safety, as done in the MPC+DCBF controller.

These undesirable stable equilibria can be seen in the discrete-time variant of the example from [16, Ex. 5.4]. In the following, we demonstrate how the $\mathbf{k}^{\text{MPC+DCBF}}$ controller can mitigate such equilibria by optimizing for performance metrics alongside safety.

Example 7.5. *Consider a two dimensional single integrator system with the dynamics* $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta_t \mathbf{u}_k$*, nominal controller* $\mathbf{k}_{\text{nom}}(\mathbf{x}) = -\mathbf{x}$*, and safety defined as* $h(\mathbf{x}) = -b^4 + \|\mathbf{x} - \mathbf{r}_1\|^2\|\mathbf{x} - \mathbf{r}_2\|^2$ *for* $\alpha = e^{-1\Delta_t}$*,* $\mathbf{r}_1 = \begin{bmatrix} a & c_2 \end{bmatrix}^\top$*, and* $\mathbf{r}_2 = \begin{bmatrix} a & -c_2 \end{bmatrix}^\top$ *with* $a = 3$*,* $b = 1.05a$*,* $c_2 = 4$*, and* $\Delta_t = 0.05$*.*

We implement the DCBF safety filter (6.4) *by linearizing the constraint at each step and we implement the* $\mathbf{k}^{\text{MPC+DCBF}}$ *controller using sequential quadratic programming where the first solution is initialized to either a semicircle on the left or the right*

of the obstacle depending on the sign of the x component of $\mathbf{x}_0 + \boldsymbol{\rho}$ where $\boldsymbol{\rho} \sim$ unif$(-0.01, 0.01)$ is sampled from a 2D normal distribution. The results are shown in Fig. 7.9.

We find that all trajectories generated by the $\mathbf{k}^{\text{MPC+DCBF}}$ controller are safe and reach the goal location at $(0, 0)$ while avoiding the undesirable equilibrium point that captures many of the trajectories generated by the DCBF safety filter.

We note that although the DCBF safety filter maintains safety, many trajectories stabilize to a point on the boundary of $\mathcal{C}$, leaving no robustness margin. This marginally safe behavior often leads to safety failures on real-world systems, where the robot settles in states with little to no tolerance for disturbances. This issue will be illustrated in the following quadruped and quadrotor examples.

Improved Feasibility Guarantees

Next, we begin our exploration of the robustness properties of the MPC and MPC+DCBF controllers by considering the problem of controlled invariance. Trivially, if h is a DCBF for system (6.1) under the input constraints $\mathbf{u} \in \mathcal{U}$, then the corresponding safe set $\mathcal{C}$ is control invariant. In this case, both the MPC controller (with $\mathcal{C}_N = \mathcal{C}$) and the MPC+DCBF controller are recursively feasible[6]. However, as discussed in Chapter 3, generating control invariant sets is a difficult problem. Therefore, we will now explore the recursive feasibility of the MPC and MPC+DCBF controllers when $\mathcal{C}$ is *not* control invariant.

Since the DCBF constraint (6.3) is a tightening of the standard MPC state constraint when $\mathbf{x} \in \mathcal{C}$, there are less states in $\mathcal{C}$ for which $\mathbf{k}^{\text{MPC+DCBF}}$ is feasible than for which $\mathbf{k}^{\text{MPC}}$ is feasible. Where other works have sought to improve this pointwise feasibility by allowing α to vary [47] or by only enforcing the DCBF constraint on the first step of the horizon [230], we instead take an entirely different approach and show that the reduced pointwise feasibility can actually improve closed-loop feasibility.

To do this we consider the case where, for all $\mathbf{x} \in \mathcal{C}$ we assume that:

$$\forall \mathbf{u} \in \mathcal{U}, \quad \Delta h(\mathbf{x}, \mathbf{u}) \geq -\delta \tag{7.34}$$

$$\forall \zeta \in [\epsilon, \delta], \ \exists \mathbf{u} \in \mathcal{U} \text{ s.t.} \quad \Delta h(\mathbf{x}, \mathbf{u}) = -\zeta \tag{7.35}$$

[6]The recursive feasibility of the MPC problem follows from [5, Thm. 12.1], and that of the MPC+DCBF follows from the infinite-horizon guarantees of Thm. 6.4.

where $\delta \geq \epsilon > 0$. The first assumption (7.34) captures the idea that, given bounded inputs and dynamics, the system can only decrease its safety by at most $-\delta$ within a single step. The second assumption (7.35) captures the idea that the system can only improve its safety degradation by so much over a single step. Notably, proving infinite-horizon safety or feasibility guarantees using these assumptions is impossible so we instead seek to guarantee feasibility and safety over the longest possible finite horizon.

To this end we have the following feasibility guarantee for the $\mathbf{k}^{\text{MPC}}$ controller:

Proposition 7.6. *If the closed loop system* (6.2) *satisfies* (7.34) *and* (7.35)*, its initial safety is $h(\mathbf{x}_0) > 0$, and it attempts to enforce constraint* (6.3) *with $\alpha = 0$ over a horizon of length $N \in \mathbb{N}$ for $\mathbf{u} \in \mathcal{U}$, then the FTOCP will be feasible for all:*

$$k \leq \frac{h(\mathbf{x}_0) - N\epsilon}{\delta} + 1. \tag{7.36}$$

The proof of Proposition 7.6 is provided in [60, Appx. A] .

Next, we present a closed loop feasibility guarantee for $\mathbf{k}^{\text{MPC+DCBF}}$ that can provide a *longer* guarantee of feasibility:

Proposition 7.7. *If the closed loop system* (6.2) *satisfies* (7.34) *and* (7.35)*, its initial safety is $h(\mathbf{x}_0) > 0$, and it attempts to enforce the constraint* (6.3) *for $\alpha \in (0, 1)$ over a horizon of length $N \in \mathbb{N}$ for $\mathbf{u} \in \mathcal{U}$ and the parameters of the system satisfy:*

$$\frac{1}{1-\alpha} \geq \frac{\delta + (N-2)\epsilon}{\delta + \epsilon}, \tag{7.37}$$

then the FTOCP+DCBF will be feasible $\forall k \leq k_\delta + k_\epsilon$ where:

$$k_\delta = \max\left\{\left\lfloor \frac{h(\mathbf{x}_0)}{\delta} - \frac{1}{1-\alpha} \right\rfloor, 0\right\}, k_\epsilon = \log_\alpha\left(\frac{\epsilon\left(\frac{1}{1-\alpha} + N - 1\right)}{h(\mathbf{x}_0) - k_\delta\delta}\right) + 1. \tag{7.38}$$

The proof of Proposition 7.7 is provided in [60, Appx. B].

Figure 7.10 illustrates the relative feasibility guarantees of Propositions 7.6 and 7.7 across various values of ϵ and α. As the system's ability to maintain safety improves (i.e., $\epsilon \to 0$), higher α values reduce pointwise feasibility but simultaneously extend guaranteed closed-loop feasibility. This mirrors the trade-off seen in tube-based MPC, where sacrificing pointwise feasibility can yield stronger recursive feasibility guarantees [258]. Even when $\mathcal{C}$ is not a control-invariant set, the decay condition imposed by the DCBF constraint can make the system *safer for longer*.

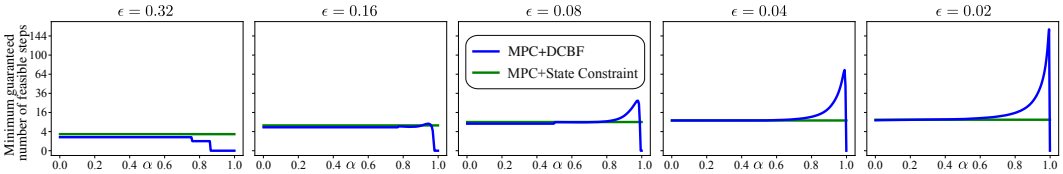


Figure 7.10. Plots demonstrating the minimum number of guaranteed feasible solutions for the closed-loop application of the MPC+DCBF controller for varying α according to Prop. 7.6 ($\alpha = 0$) and Thm. 7.7 ($\alpha \in (0, 1)$). The y axis is shown using a squared scale to better capture the range of possible outputs. The minimum number of feasible steps is plotted for the state constraint (i.e., $\alpha = 0$) in green and for the MPC+DCBF controller for varying $\alpha \in (0, 1)$ in blue. The other parameters used to generate this plot include: $h(\mathbf{x}_0) = 10, \delta = 1$, and $N = 25$. The plots from left to right show the minimum number of feasible steps for varying minimum safety decay values ϵ. As the minimum safety decay value goes to zero, the minimum number of guaranteed feasible steps generated by the MPC+DCBF can dramatically outperform the state constraint-based feasibility guarantee despite the pointwise reduction in feasibility with respect to state. This is due to the closed-loop properties of applying the more conservative MPC+DCBF controller.

MPC+DCBF Deterministic Robustness

Next, we explore the benefits of the unified $\mathbf{k}^{\text{MPC+DCBF}}$ controller under bounded dynamics uncertainty.

To perform this analysis, we now consider the discrete-time dynamical system (6.1) subject to additive dynamics uncertainty:

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{d}_k, \quad \forall k \in \mathbb{N}. \tag{7.39}$$

This uncertainty is represented by $\mathbf{d}_k$ which we assume is bounded by some $\overline{\delta} \geq \|\mathbf{d}_k\| \geq 0$ for all $k \geq 0$.

As with the undisturbed system (6.1), a controller can be added to generate the closed loop dynamical system:

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{k}(\mathbf{x}_k)) + \mathbf{d}_k, \quad \forall k \in \mathbb{N}. \tag{7.40}$$

Next we consider the robust safety properties of this closed-loop system.

In achieving robust safety, the feedback of the current safety value used in the DCBF constraint (6.3) results in the ISSf property which was introduced in Proposition 6.26 and which is dependent on the $\alpha \in [0, 1)$ parameter.

The ISSf property ensures that, under a bounded additive disturbance, the DCBF condition (6.3) for $\alpha \in [0, 1)$ can still generate guarantees of set invariance with respect to some larger set $\mathcal{C}_\delta \supset \mathcal{C}$ even when the original safe set $\mathcal{C}$ is not invariant. This robustness result differs from those generated by tube MPC [30], [107] approaches since the controller design does not require an *a priori* knowledge of the disturbance size.

Given this understanding of ISSf, we now compare the robustness of the $\mathbf{k}^{\text{MPC+DCBF}}$ controller for varying $\alpha \in [0, 1)$, where $\alpha = 0$ encodes the typical MPC state constraint. Importantly, α has the following effect on the system safety depending on whether $\mathbf{x}$ is inside or outside of $\mathcal{C}$:

- $\mathbf{x} \in \mathcal{C}$, larger α results in a *tighter* constraint that bounds the maximum rate that $h(\mathbf{x})$ can decrease to 0,
- $\mathbf{x} \notin \mathcal{C}$, larger α results in a *looser* constraint that bounds the minimum rate at which $h(\mathbf{x})$ must increase to 0.

Notably, the size of the expanded safe set $\mathcal{C}_\delta$ (as defined in Prop. 6.26) increases monotonically with α. However, $\alpha > 0$ may still be desirable since it gracefully brings the system back towards $\mathcal{C}$ via geometric decay of $h(\mathbf{x})$ whereas $\alpha = 0$ will force the system to return to $\mathcal{C}$ in a single step, a behavior that will likely result in infeasibility or overly aggressive behavior as can be seen in the following example:

Example 7.8. *To demonstrate the effect of α on the ISSf property and the associated inputs, we consider a one-dimensional, discrete-time double integrator system with $\Delta_t = 10^{-3}$:*

$$\underbrace{\begin{bmatrix} x_{k+1} \\ v_{k+1} \end{bmatrix}}_{\mathbf{x}_{k+1}} = \begin{bmatrix} 1 & \Delta_t \\ 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} x_k \\ v_k \end{bmatrix}}_{\mathbf{x}_k} + \begin{bmatrix} \frac{1}{2}\Delta_t \\ \Delta_t \end{bmatrix} \underbrace{\begin{bmatrix} a_k \end{bmatrix}}_{\mathbf{u}_k}, \tag{7.41}$$

with a safety condition that conflicts with the goal location:

$$h(\mathbf{x}) = 1 - x, \qquad \mathbf{x}_{\text{goal}} = \begin{bmatrix} 2, & 0 \end{bmatrix}^\top. \tag{7.42}$$

We simulate the system with both $\alpha = 0$ and $\alpha = e^{-\Delta_t}$ for both a constant disturbance and an impulsive disturbance:

$$\mathbf{d}_{\text{constant}} = \begin{bmatrix} 0.1\Delta_t \end{bmatrix} \qquad \mathbf{d}_{\text{impulse}} = \begin{bmatrix} 10\Delta_t s \end{bmatrix}. \tag{7.43}$$

The results of these simulations can be seen in Fig. 7.11.

Although $\alpha > 0$ may lead to larger safety violations, it also results in much smoother trajectories and requires significantly smaller inputs (by approximately an order of magnitude) because constraint (6.3) *enforces geometric convergence back to the set whereas the state constraint ($\alpha = 0$) requires the system to return to $\mathcal{C}$ in a single step.*

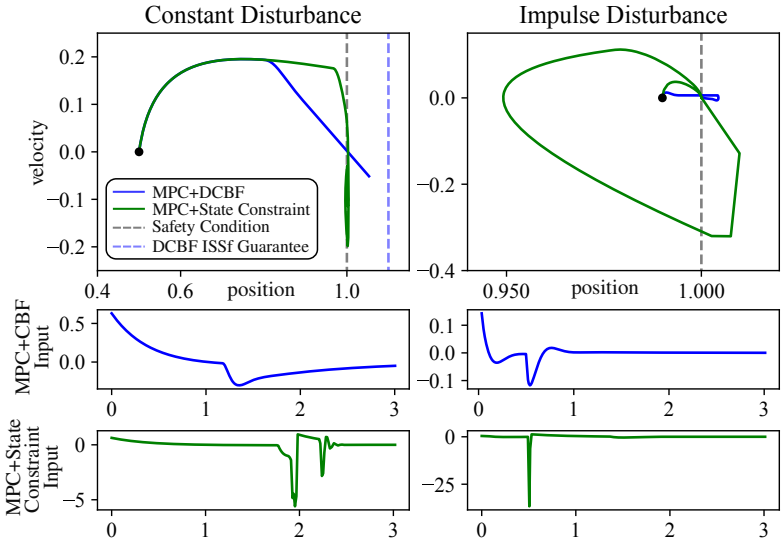


Figure 7.11. Plots displaying the state (x, v_x) trajectories for the double integrator system (top), MPC+DCBF inputs (a_x) for $\alpha = e^{-\Delta_t}$ (middle) and the state constrained problem where $\alpha = 0$ (bottom). On the left, trajectories are shown for a constant additive disturbance of $\mathbf{d}_{\text{constant}} = \begin{bmatrix} 0.1\Delta_t, & 0 \end{bmatrix}^\top$ which occurs at every step with $\Delta = 10^{-3}$. In this case both trajectories cross the safety boundary (gray dashed line). The trajectory with $\alpha > 0$ has a larger violation of safety but still achieves safety of $\mathcal{C}_\delta$ whose boundary is shown as the blue dashed line. On the right, a single impulse disturbance of $\mathbf{d}_{\text{impulse}} = \begin{bmatrix} 10\Delta_t, & 0 \end{bmatrix}^\top$ occurs at $k = 500$ causing both trajectories to violate safety. Importantly, although the state constraint may result in smaller violations in some cases, it requires very large inputs, approximately an order of magnitude larger than those for the CBF, which may cause problem infeasibility when $\mathcal{U}$ is bounded.

Here we find that the DCBF inequality (6.3) results in a smooth degradation of safety as the disturbance size increases and that, if the system becomes unsafe, the DCBF facilitates a graceful recovery of safety through its decay-based constraint. Alternatively, the step-wise safety requirement of the MPC controller requires the system to become safe again immediately, which can lead to overly aggressive behavior and infeasible safety requirements. In practice, we see this manifest on hardware when the true system and the model used in the FTOCP differ, which can lead to safety failures when either the FTOCP becomes infeasible or requires inputs which exceed the real-world bounds. These phenomena will be seen in both experimental demonstrations in this section.

Quadruped Experiments

To demonstrate the utility of the MPC+DCBF method and compare it to the effectiveness of other methods, we apply the proposed control algorithms to a dynamic collision avoidance scenario using a Unitree Go2 quadruped.

Example 7.9 (Quadruped Experiments)**.** *For the deployment of the MPC+DCBF*

method on the Unitree Go2 quadruped, we leverage a reduced-order model (ROM) hierarchical control framework [48] based on a two-dimensional single-integrator high-level control interface enabling the assignment of safe translational velocity commands without modification of the low-level locomotion controller.

To generate the function h defining safety, we first perceive the experimental space using a fixed overhead RGB camera, which provides a persistent global image stream of the robot's 2D environment at 60 fps. This video stream is passed to the efficient Track-Anything-Model (efficientTAM) [259] image segmenter, a high-speed distillation of the Meta SAM2 segmentation model [260]. By segmenting the environment to detect predefined obstacles, we build a 2D occupancy map of the space. The occupancy map is buffered by the physical geometry of the Go2 quadruped, enabling safety of the robot to be defined via its centroid. Next, to produce h we use the Poisson-based algorithm developed in [106]. This method yields a single continuous h for the entire experimental environment, which can be queried during autonomous operation. The velocity of the obstacle is estimated using optical flow [261] on the segmented images and incorporated as a time-varying component in h. Additionally, an overhead OptiTrack motion capture system is used to estimate the translational and rotational states of the robot.

We employ this function h in the $\mathbf{k}^{\text{MPC+DCBF}}$ controller, producing safe velocities which the quadrupedal system tracks. The result of the model mismatch between the true quadrupedal system and the single integrator dynamics used by $\mathbf{k}^{\text{MPC+DCBF}}$ controller can be modeled as a disturbance to the system and the recursive feasibility of the system can be analyzed in a way similar to Propositions 7.6 and 7.7 or through the lens of ISSf, in which case the $\mathbf{k}^{\text{MPC+DCBF}}$ controller may produce extended periods of recursive feasibility and graceful safety degradation whereas the MPC controller may result in earlier and more catastrophic safety failures.

To highlight safety-critical performance, the robot task is to hold a fixed reference coordinate $(1.75\,m,\ 2.75\,m)$ while staying in the safe set. We then roll a dodgeball into the environment along a path that requires the robot to move in order to maintain safety using a fixed-height ramp to produce a repeatable dynamic collision avoidance scenario. Across trials, safety was enforced via the three aforementioned methods: 1) $\mathbf{k}^{\text{MPC+DCBF}}$, 2) state-constrained $\mathbf{k}^{\text{MPC}}$ with $\alpha = 0$ – the naive MPC approach, and 3) the DCBF safety filter (6.4) *using a proportional nominal controller. The resulting data for a single set of comparison experiments can be seen in Fig. 7.12.*

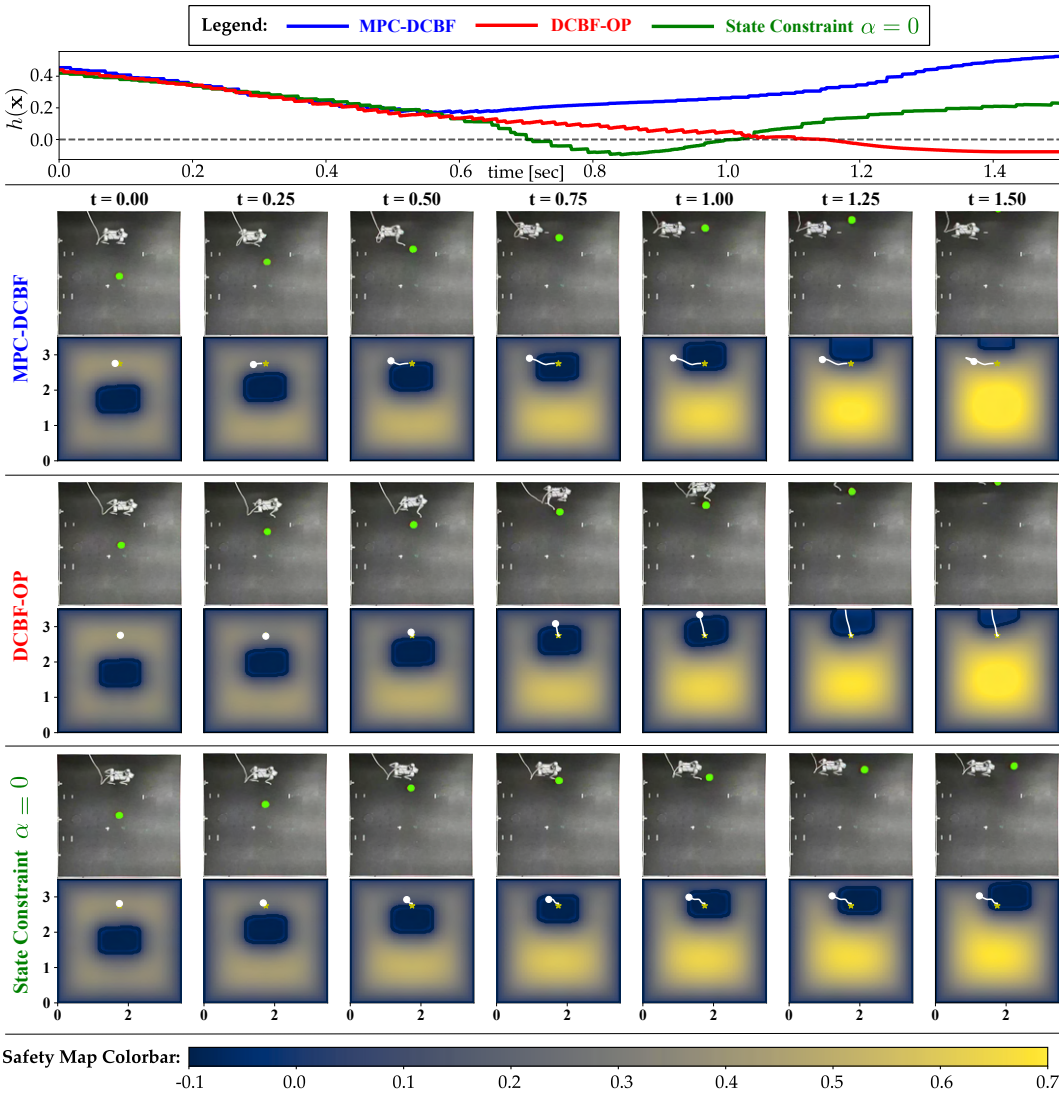


Figure 7.12. Quadrupedal robot dynamic obstacle avoidance experiments for the 6.4 in red, state-constrained MPC (i.e., MPC+DCBF with $\alpha = 0$) in green, and the MPC+DCBF controller in blue. **(Top)** Time series plots of the safety value $h(\mathbf{x})$ for each controller. The MPC+DCBF successfully maintains safety during the experiment while the state-constrained MPC and the 6.4 both result in safety failures. **(2nd Row)** Overhead images with the dynamic obstacle highlighted in green and contour plots of $h(\mathbf{x})$ through time (left to right: $t = 0$ to $t = 1.5$) for the MPC+DCBF experiments. The quadruped successfully moves out of the way to avoid the dynamic obstacle. **(3rd Row)** Overhead images and safety contour plots for the 6.4 controller which is unable to plan around the obstacle and gets squeezed in between the wall and the dynamic obstacle until a failure occurs and the quadruped steps out of the safe region at $t = 1.0$ second. **(4th Row)** Overhead images and safety contour plots for the MPC controller which reacts too late and a safety failure and collision occur at approximately $t = 0.75$ sec. **(Bottom)** A colorbar showing the meaning of the colors in the $h(\mathbf{x})$ contour plots.

By examining the top plot in the figure, it is immediately apparent that the MPC +DCBF method successfully enforces safety throughout the duration of the experiment. Meanwhile, state-constrained MPC and the DCBF safety filter both result

in safety violations. Furthermore, the overhead camera images highlight key differences in how the quadruped attempts to avoid the dynamic obstacle. Due to its increased robustness and the incorporation of a planning horizon, the MPC+DCBF controller begins to command its avoidance maneuver significantly earlier than the other two methods. Although the decay of h for all three methods appears the same until 0.55 seconds, the MPC+DCBF begins to command motion before 0.25 seconds, moving along level sets of h to attain a more optimal position for future actions. This is a direct result of the tightening of the constraints of the underlying optimization problem since $\alpha > 0$, which forces the DCBF constraint to activate sooner. Conversely, the state-constrained MPC controller reacts too late, commanding a large input to the single-integrator ROM which could not be tracked by the low-level locomotion controller. This inevitably led to a safety failure ($t = 0.75$ sec). Similarly, the small lateral gradients of $h(\mathbf{x})$ in the x direction cause the DCBF safety filter (6.4) *to be unable to effectively "flow" around the obstacle given the real-time sampeld-data nature of the hardware experiment, eventually causing a safety failure as quadruped left the rectangular safe region ($t = 1.05$ seconds).*

These experimental results were highly repeatable, as can be seen in the videos at the link in [250]. In fact, under these particular experimental conditions, the MPC+DCBF method maintained a 100% success rate, while the state-constrained MPC and 6.4 methods each had 0% success rates.

Probabilistic Safety with Dynamics Uncertainty

With that introduction to the benefits of the MPC+DCBF formulation, we now return to the context of stochastic uncertainty and show its stochastic robustness properties.

Critically, we note that the stochastic guarantees of Thms. 6.13 and 6.24 rely on $\alpha \in (0, 1)$ and cannot be used to provide guarantees when $\alpha = 0$. The self-referential, safety-feedback property of the DCBF constraint is critical in creating the necessary supermartingale relationship to invoke the concentration inequalities, and when α increases toward 1, these bounds guarantee a lower risk of safety failure. Alternatively, when $\alpha = 0$, as in the typical state-constraint formulation in MPC, these methods can no longer be used to make probabilistic guarantees and return vacuous probability bounds. Thus, the methods of the Chapter 6 can be used to produce probabilistic safety guarantees for a expectation-based MPC+DCBF controller, but *not* for the standard MPC implementation.

As in the deterministic case, to benefit from the horizon based planning of MPC and

the inherent robustness properties of stochastic DCBFs, we propose unifying them in the form of a stochastic Model-Aware Risk-Informed Optimization (MARIO) optimization problem:

$$\begin{aligned}
\min_{\substack{\boldsymbol{\xi}_{0:N}\in\mathbb{R}^{n_x}\\ \boldsymbol{\nu}_{0:N-1}\in\mathbb{R}^{n_u}}} \quad & \mathbb{E}\left[\sum_{i=0}^{N-1} c(\boldsymbol{\xi}_i,\boldsymbol{\nu}_i)+V(\boldsymbol{\xi}_N)\middle|\mathscr{F}_k\right] && \text{(MARIO)}\\
\text{s.t.} \quad & \boldsymbol{\xi}_{i+1}=\mathbf{F}(\boldsymbol{\xi}_i,\boldsymbol{\nu}_i,\mathbf{d}_i), \quad \forall i\in\{0,\dots,N-1\}\\
& \mathbb{E}[h(\boldsymbol{\xi}_{i+1})|\mathscr{F}_k]\geq \alpha h(\boldsymbol{\xi}_i), \quad \forall i\in\{0,\dots,N-1\}\\
& \boldsymbol{\nu}_k\in\mathcal{U}, \quad \forall i\in\{0,\dots,N-1\}\\
& \mathbf{d}_i\sim\mathcal{D}(\mathbf{x}_{k:0})\\
& \boldsymbol{\xi}_0=\mathbf{x}_k
\end{aligned}$$

with the MARIO controller $\mathbf{k}^{\text{MARIO}}(\mathbf{x}_k)=[\boldsymbol{\nu}_0^*(\mathbf{x}_k)]$.

In the reformulation of FTOCP+DCBF to MARIO, we replaced the deterministic DCBF constraint (6.3) with the expectation-based condition (6.16) across the horizon:

$$\mathbb{E}[h(\mathbf{F}(\boldsymbol{\xi}_i,\boldsymbol{\nu}_i,\mathbf{d}_i))|\mathscr{F}_k]\geq\alpha h(\boldsymbol{\xi}_i),\ \forall i\in\{0,\dots,N\}. \tag{7.44}$$

Importantly this reformulation is always conditioned on $\mathscr{F}_k$ for all prediction steps i. Thus, the planner's understanding of the uncertainty distribution at each step is only dependent on the current state history, $\mathcal{D}(\mathbf{x}_{k:0})$, making this controller causal and realizable.

Since this controller enforces the expectation-based DCBF constraint (6.16) at the first step, the closed-loop system (6.7) under this controller satisfies the expectation-based DCBF condition required for Thms. 6.13 and 6.24. Thus, the $\mathbf{k}^{\text{MARIO}}$ controller, which can be considered as a stochastic formalization of the $\mathbf{k}^{\text{MPC+DCBF}}$, immediately benefits from inherent robustness guarantees of Thms. 6.13 or 6.24 which do not apply to the stochastic reformulation of the $\mathbf{k}^{\text{MPC}}$ controller with $\alpha=0$. Instead, most stochastic MPC methods rely on a quantile-based chance constraint [251], [262] which can require significantly more distribution information than the supermartingale methods which are based on the first-moment. Additionally, because the supermartingale methods rely on an inequality on the expectation, they can be thought of as distributionally robust, since the guarantees hold for all

distributions which satisfy the first-moment property. This robustness is achieved by sacrificing tightness of the probability bound.

Notably, enforcing the expectation-based DCBF constraint (6.16) may require additional consideration since the dynamics and safety function h may reshape the disturbance distribution. In this case, sampling-based methods may be used to approximate the expectation constraint, or the Jensen-gap bounding methods of Section 6.3 and Theorem 6.15 can be applied.

We recognize that practically implementing the $\mathbf{k}^{\text{MARIO}}$ controller may be difficult, as it requires propagation of compounding uncertainty through the dynamics, cost, and the DCBF h. We note that, since only the first constraint along the planning horizon of the MARIO FTOCP is leveraged at each time-step to produce the trajectory long guarantees of Thms. 6.13 and 6.24, further simplifications can be made for $i > 1$ to ease the computational burden. While this may reduce the optimality of the $\mathbf{k}^{\text{MARIO}}$ controller, it will still maintain its probabilistic guarantees. Additionally, the following discussion will consider a computationally tractable version of the $\mathbf{k}^{\text{MARIO}}$ controller that also incorporates state uncertainty.

Probabilistic Safety with Dynamics and State Uncertainty

We now consider systems with stochastic dynamic uncertainty as in (6.7) where we do not have direct access to the state. Instead we only have indirect access through noisy measurements:

$$\mathbf{y}_k = \mathbf{M}(\mathbf{x}_k, \mathbf{v}_k), \qquad \mathbf{v}_k \sim \mathcal{V}(\mathbf{x}_{k:0}) \tag{7.45}$$

where, $\mathbf{y}_k \in \mathbb{R}^{n_y}$ is a system measurement, $\mathbf{v}_k$ is a $\mathscr{F}_{k+1}$-measurable random variable taking values in $\mathbb{R}^{n_v}$ that represents the measurement noise, and $\mathbf{M} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_y}$ is the system's measurement function that obtains measurement $\mathbf{y}_k$ given the state $\mathbf{x}_k$; for example, $\mathbf{M}$ could be a camera that produces a noisy image $\mathbf{y}_k$ given the current position $\mathbf{x}_k$ and the noise $\mathbf{v}_k$.

In the context of real-world robotics and control systems, we never have access to the true state of the system due to uncertainties in our measurements. Because of this inability to access the true state, the state evolution of $\mathbf{x}_k$ becomes a partially observable Markov decision process (POMDP), and it is common to seek guarantees on the belief-state distribution instead of guarantees on the true state [21], [200]. In this work we will first seek a bound on the belief space safety and then extend this to the true state through a union bounding method.

Since we are considering systems where we do not know the true state $\mathbf{x}_k$, we now discuss a filtration generated by the $\sigma-$algebra over only the observations, $\mathbf{y}_k$; that is, we consider $\mathscr{G}_k = \sigma(\mathbf{y}_{k:0})$ which is contained in the filtration $\mathscr{F}_k$, i.e., $\mathscr{G}_k \subset \mathscr{F}_k$. Since the inputs are selected based on the system measurements, the input vector $\mathbf{u}_k$ is also $\mathscr{G}_k$-measurable. On the other hand, since the true state of the system $\mathbf{x}_k$ is not $\mathscr{G}_k$-measurable, we turn our attention to the expectation of the state conditioned on the measurement-based filtration $\mathscr{G}_k$. We will use the previously introduced martingale constructions to make safety guarantees with respect to the expected value of the belief state:

$$\widehat{\mathbf{x}}_{k|k-1} \triangleq \mathbb{E}[\, \mathbf{x}_k \mid \mathscr{G}_{k-1} \,], \qquad \widehat{\mathbf{x}}_{k|k} \triangleq \mathbb{E}[\, \mathbf{x}_k \mid \mathscr{G}_k \,], \tag{7.46}$$

where, as in a Kalman filter [263], $\widehat{\mathbf{x}}_{k|k-1}$ is the expected value of the predicted belief state at the next step and $\widehat{\mathbf{x}}_{k|k}$ is the expected value of the updated belief state after a new measurement has been taken.

Next, we seek to produce risk-based bounds on the K-step exit probability of the expected value of the belief state:

$$P_u(K, \widehat{\mathbf{x}}_{0|0}) \triangleq \mathbb{P}\{\, \widehat{\mathbf{x}}_{k|k} \notin \mathcal{C} \text{ for some } k \leq K \,\}, \tag{7.47}$$

where the dynamics for $\widehat{\mathbf{x}}_{k|k}$ include both the system dynamics for the prediction propagation to $\widehat{\mathbf{x}}_{k+1|k}$ and the system measurement update step to obtain $\widehat{\mathbf{x}}_{k+1|k+1}$ once $\mathbf{y}_{k+1}$ is measured.

To bound (7.47) we consider the following condition on $h(\widehat{\mathbf{x}}_{k+1|k+1})$ in order to generate stochastic guarantees:

$$\mathbb{E}[\, h(\widehat{\mathbf{x}}_{k+1|k+1}) \mid \mathscr{G}_k \,] \geq \alpha h(\widehat{\mathbf{x}}_{k|k}), \tag{7.48}$$

for some $\alpha \in (0, 1)$ where the expectation-based DTCBF condition in (6.16) is now applied to the expected value of the updated belief state.

To practically implement constraint (7.48) to generate probabilistic safety guarantees, we assume that the dynamics and measurements are linear:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{d}_k \tag{7.49}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \tag{7.50}$$

and that the disturbance $\mathbf{d}_k$ and noise $\mathbf{v}_k$ are sampled from zero-mean distributions[7].

[7] To prove Thm. 7.10 we assume zero mean, but any bias can also be accounted for by modeling it and including it as a part of the nominal dynamics and measurement model.

To construct an FTOCP for this system, instead of constraining the initial state in the plan to be the current state $\boldsymbol{\xi}_0 = \mathbf{x}_k$, which we do not have access to, we constrain it to the current expected belief state $\boldsymbol{\xi}_0 = \widehat{\mathbf{x}}_{k|k}$. Using this adjustment and the linear dynamics assumption we have the State-Uncertain Probabilistic Model-Aware Risk Informed Optimization (SUP-MARIO) optimization problem:

$$\begin{aligned} \min_{\substack{\boldsymbol{\xi}_{0:N}\in\mathbb{R}^{n_x} \\ \boldsymbol{\nu}_{0:N-1}\in\mathbb{R}^{n_u}}} \quad & \sum_{i=0}^{N-1} c(\boldsymbol{\xi}_i, \boldsymbol{\nu}_i) + V(\boldsymbol{\xi}_N) \qquad \text{(SUP-MARIO)} \\ \text{s.t.} \quad & \boldsymbol{\xi}_{i+1} = \mathbf{A}\boldsymbol{\xi}_i + \mathbf{B}\boldsymbol{\nu}_i, \quad \forall i \in \{0,\dots,N-1\} \\ & h(\boldsymbol{\xi}_{i+1}) \geq \alpha h(\boldsymbol{\xi}_i), \quad \forall i \in \{0,\dots,N-1\} \\ & \boldsymbol{\nu}_k \in \mathcal{U}, \quad \forall i \in \{0,\dots,N-1\} \\ & \boldsymbol{\xi}_0 = \widehat{\mathbf{x}}_{k|k} \end{aligned}$$

for some $\alpha \in (0,1)$ which can be used as before to define the SUP-MARIO controller $\mathbf{k}^{\text{SUP-MARIO}}(\mathbf{x}_k) = [\boldsymbol{\nu}_0^*(\mathbf{x}_k)]$.

Notably, whereas $\mathbf{k}^{\text{MARIO}}$ may be difficult to implement, the implementation of $\mathbf{k}^{\text{SUP-MARIO}}$ is straightforward. This is because $\mathbf{k}^{\text{SUP-MARIO}}$ relies on the expected-value of the state to implement the FTOCP+DCBF problem[8]. It therefore does not require the uncertainty distributions be propagated across the planning horizon.

Next, we show that this simple-to-implement controller still satisfies the DCBF condition in expectation (7.48). In particular, the $\mathbf{k}^{\text{SUP-MARIO}}$ controller satisfies the desired DCBF constraint on the safety of the belief state (7.48) which allows it to leverage Thm. 6.13 and/or Thm. 6.24 to provide bounds on the $K-$step exit probability of the belief state (7.47).

Theorem 7.10. *For systems with linear dynamics* (7.49)*, linear measurements* (7.50)*, and zero-mean disturbance* $\mathbf{d}_k$ *and measurement noise* $\mathbf{v}_k$*, if* $h : \mathbb{R}^{n_x} \to \mathbb{R}$ *is convex*[9]*, then the closed-loop system* (6.7) *with the* $\mathbf{k}^{\text{SUP-MARIO}}$ *controller satisfies:*

$$\mathbb{E}[\, h(\widehat{\mathbf{x}}_{k+1|k+1}) \mid \mathscr{G}_k \,] \geq \alpha h(\widehat{\mathbf{x}}_{k|k}) \tag{7.51}$$

A proof of Thm. 7.10 can be found in [60, Appx. C].

[8]The relationship between the FTOCP+DCBF and SUP-MARIO is similar to that between a linear quadratic regulator (LQR) and a linear quadratic guassian (LQG) controller.

[9]When h is concave, the method in Theorem 6.15 can be used. Alternatively, sampling-based methods can be used to approximate $\mathbb{E}[h(\mathbf{x})]$ from $h(\mathbb{E}[\mathbf{x}])$ for general h.

Since the $\mathbf{k}^{\text{SUP-MARIO}}$ satisfies condition (7.48), Thms. 6.13 and 6.24 can be used to guarantee bounds on the K-step exit probability of $\widehat{\mathbf{x}}_{k|k}$ when their respective hypotheses regarding bounds on h or step-wise and predictable quadratic variation (PQV) bounds are satisfied.

Finally, to analyze the safety achieved by the $\mathbf{k}^{\text{SUP-MARIO}}$ controller with respect to the true state $\mathbf{x}$, we can leverage tail-bounding methods like Cantelli's inequality [206] (one-sided Chebychev's inequality) in conjunction with the union bound (Boole's inequality) to extend beyond the $K-$step exit probability bounds on $\widehat{\mathbf{x}}_{k|k}$.

Using Cantelli's inequality we can extend a safety guarantee on the expected belief state $\widehat{\mathbf{x}}_{k|k}$ to a safety guarantee on the true state $\mathbf{x}_k$.

Theorem 7.11. *Assume that the variance of safety is bounded as* $\text{Var}(h(\mathbf{x}_k)) \leq \sigma_h$ *for some* $\sigma_h > 0$*, all* $\mathbf{x}_k$*, and all* $k \leq K$ *and that* h *is convex. If the system achieves the* K*-step exit probability* $P_u(K, \widehat{\mathbf{x}}_0) \leq \epsilon$ *for the belief state* $\widehat{\mathbf{x}}_{k|k}$*, then failure probability for the* $\mathcal{C}_{\delta_C} = \{\mathbf{x} \in \mathbb{R}^{n_x} | h(\mathbf{x}) \geq -\delta_C\}$ *for the true state* $\mathbf{x}$ *and some* $\delta_C \geq 0$ *is bounded as:*

$$\mathbb{P}\{h(\mathbf{x}_k) < -\delta_C \text{ for some } k \leq K\} \leq \epsilon + (1+K)\left(\frac{\sigma_h^2}{\sigma_h^2 + \delta_C^2}\right).$$

This final theorem allows us to place theoretical guarantees on the probability that the true state of the system will be safe despite indirect knowledge of $\mathbf{x}$ due to noisy measurements. Its proof can be found in Appx. [60, Appx. D].

Quadrotor Experiments

Next we apply the $\mathbf{k}^{\text{SUP-MARIO}}$ controller to a quadrotor robot to achieve dynamic obstacle avoidance. To do this we consider the same quadrotor model as in (7.11).

To control the quadrotor robot we use a hierarchical control scheme that consists of three layers. At the lowest layer we use an opensource Betaflight controller to track commanded thrust and angle rates at 8 kHz. At the mid-layer, we implement the geometric tracking controller presented in [94] at 800 Hz to generate thrust and angle rate commands based on desired position trajectories. Finally, at the highest-layer we generate twice continuously differentiable position outputs using the $\mathbf{k}^{\text{SUP-MARIO}}$ controller at 20 Hz, where the linear model used in the SUP-MARIO FTOCP is:

$$\underbrace{\begin{bmatrix}\mathbf{p}_{k+1}\\ \mathbf{v}_{k+1}\end{bmatrix}}_{\boldsymbol{\xi}_{k+1}} = \begin{bmatrix}\mathbf{I} & \Delta_t\mathbf{I}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\underbrace{\begin{bmatrix}\mathbf{p}_k\\ \mathbf{v}_k\end{bmatrix}}_{\boldsymbol{\xi}_k} + \begin{bmatrix}\frac{\Delta_t^2}{2}\mathbf{I}\\ \Delta_t\end{bmatrix}\underbrace{\begin{bmatrix}\mathbf{a}_k\end{bmatrix}}_{\boldsymbol{\nu}_k}. \tag{7.52}$$

To limit the angle rate commands produced by the tracking controller and ensure smoother flight, we add a constraint on the system jerk by bounding the difference between the current and next acceleration inputs in the FTOCP and SUP+MARIO problems. Furthermore, to avoid infeasibility during flight, we implement these finite-difference-based jerk bounds as soft constraints with slack variables.

The geometric tracking controller can be used to establish the differential flatness of the quadrotor system [237] that ensures the (almost everywhere) tracking of the desired trajectories and any remaining error in the model that occurs transiently due to initial condition error, angle-rate convergence, or lack of smoothness between solution updates can be analyzed through the deterministic and stochastic robustness frameworks presented in this section.

For safety, we consider collision avoidance between our quadrotor drone and a dynamic projectile obstacle. Mathematically we define this safety using the function:

$$h_0(\mathbf{x}) = \|\tilde{\mathbf{p}}_{x:y}\| - r \tag{7.53}$$

where $\tilde{\mathbf{p}}$ represents the relative position of the quadrotor with respect to the obstacle, the subscript indices indicate the extraction of the first two elements of $\tilde{\mathbf{p}}$, and r is the radius of the obstacle, which accounts for the maximum dimension of the quadrotor. The 0-superlevel set of h_0 functionally defines safety for our quadrotor system as staying outside the planar (x, y) region containing the obstacle. To implement the horizon-based planning of the MPC and SUP-MARIO controllers, we use a constant-velocity model of the dynamic obstacle.

While $h_0(\mathbf{x})$ is used in the implementations of the $\mathbf{k}^{\text{MPC}}$ with state constraints, we instead use a higher-order CBF (HOCBF) extension [75] to implement the $\mathbf{k}^{\text{DCBF-OP}}$ and $\mathbf{k}^{\text{SUP-MARIO}}$ controllers:

$$h(\mathbf{x}) = \underbrace{\frac{\tilde{\mathbf{p}}_{x:y}^\top}{\|\tilde{\mathbf{p}}_{x:y}\|}\dot{\tilde{\mathbf{p}}}_{x:y}}_{\dot{h}_0(\mathbf{x})} + \gamma h_0(\mathbf{x}) \tag{7.54}$$

for a $\gamma > 0$. This is a relative degree 1 DCBF designed for the double integrator model used in the $\mathbf{k}^{\text{SUP-MARIO}}$ controller. While the use of h as in (7.54) in place of h_0 is not theoretically necessary, and in fact moves from a convex h_0 to a non-convex h, we find that it provides better closed-loop system behavior.

An additional DCBF constraint was also implemented in the $\mathbf{k}^{\text{MPC}}$ and $\mathbf{k}^{\text{SUP-MARIO}}$ controllers to avoid collisions with the ground.

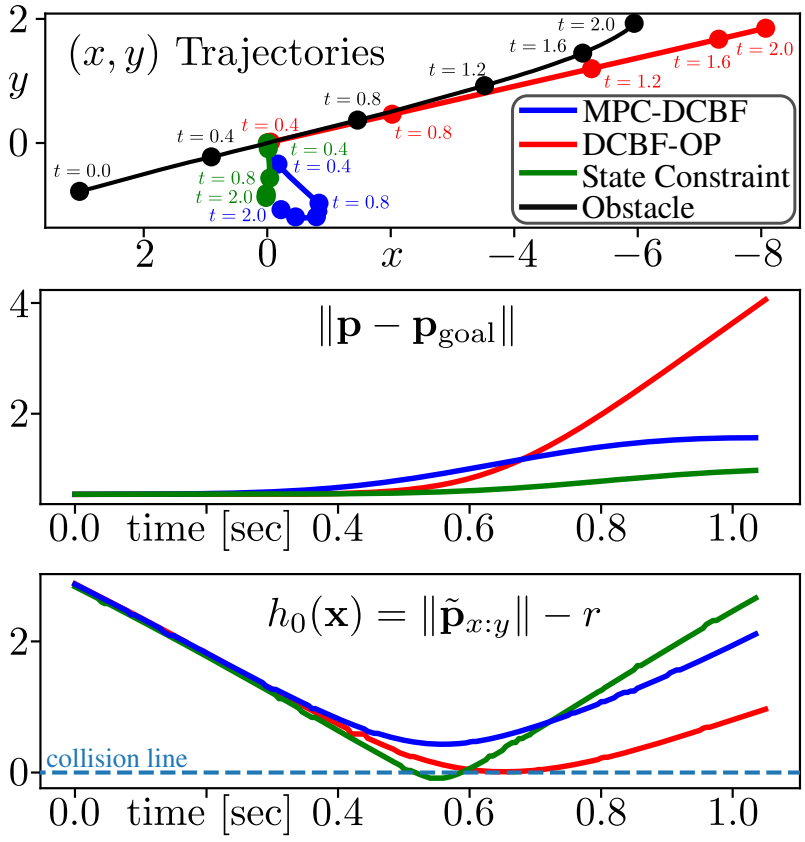


Figure 7.13. Simulated demonstrations of the $\mathbf{k}^{\text{MPC}}$, $\mathbf{k}^{\text{DCBF-OP}}$, and $\mathbf{k}^{\text{SUP-MARIO}}$ controllers performing a dynamical obstacle avoidance task. **(Top)** The planar (x, y) trajectories for the obstacle (black), DCBF-OP (red), MPC+DCBF (blue), and MPC with state constraints (green). **(Middle)** The distance to the goal through time for each controller. **(Bottom)** The signed-distance function representing collision between the drone and the obstacle. Aided by their planning horizons, the $\mathbf{k}^{\text{MPC}}$ controller with state constraints and the $\mathbf{k}^{\text{SUP-MARIO}}$ controller both produce trajectories with relatively small deviations away from the goal point $(0, 0)$. Meanwhile, the $\mathbf{k}^{\text{DCBF-OP}}$ and $\mathbf{k}^{\text{SUP-MARIO}}$ controllers manage to avoid collisions, but the $\mathbf{k}^{\text{MPC}}$ controller results in a safety failure due to a the model-mismatch and lack of robustness. Here the DCBF-OP achieves safety during the scenario, but its myopic, pointwise optimization does so by moving in the same direction as the obstacle whereas the $\mathbf{k}^{\text{SUP-MARIO}}$ controller optimizes performance while achieving safety by moving in a direction which is predominantly orthogonal to the obstacle's velocity. Videos of these simulations can be found at [250].

Simulation Experiments: In simulation we compare the effectiveness of the $\mathbf{k}^{\text{DCBF-OP}}$ from (6.4), $\mathbf{k}^{\text{MPC}}$, and $\mathbf{k}^{\text{SUP-MARIO}}$ controllers[10] at achieving dynamic collision avoidance for the quadrotor system (7.11). The results of this comparison simulation can be seen in Fig. 7.13.

To approximate real-world uncertainty, random noise was added to both the state and obstacle values and measurements and Kalman filters were implemented to estimate

[10] To practically implement the $\mathbf{k}^{\text{DCBF-OP}}$ we added the ISSf-CBF term [31] from Def. (2.24) to account for model uncertainty. This term was not added in the implmentation of the $\mathbf{k}^{\text{SUP-MARIO}}$ controller.

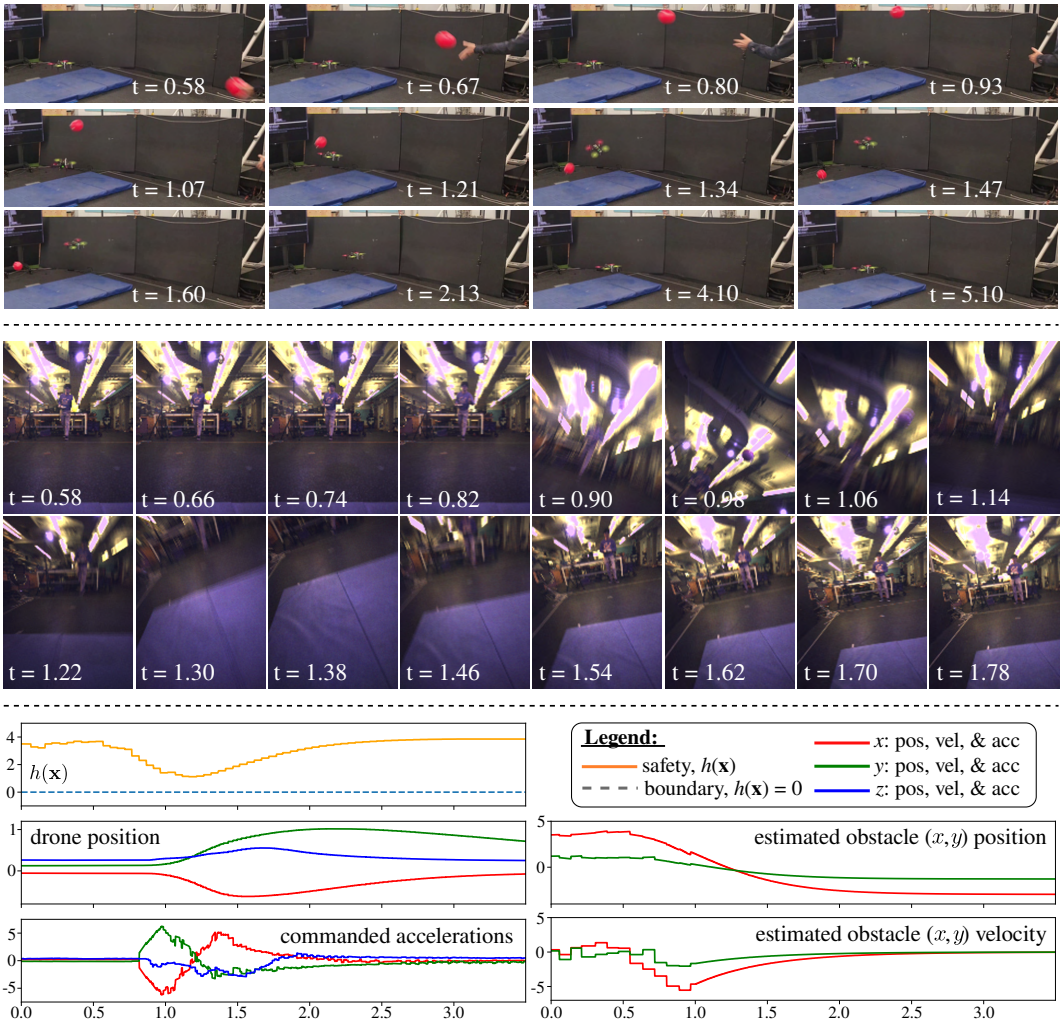


Figure 7.14. Experimental demonstration of the $\mathbf{k}^{\text{SUP-MARIO}}$ controller on a quadrotor drone with onboard dynamic obstacle detection, state estimation, and avoidance. **(Top)** The top two rows show the scene through time from the perspective of an external camera. The drone starts in the top left and a red ball moves towards it. The drone then moves back and to the side to avoid a collision. **(Middle)** The middle two rows show the scene from the robot's perspective. The obstacle mask, as seen by the robot, is shown in yellow and can be seen to have significant noise with an innacurate mask at $t = 0.58$, significant motion blur, and missing detections at $t = 0.90$ and 0.98. **(Bottom)** The bottom plots show the system's safety value $h(\mathbf{x})$ in orange (which never drops below zero), the drone position, the commanded acceleration, and the obstacle's estimated position and velocity. The different components of these vectors are shown as x in red, y in green, and z in blue. The maximum velocity of the obstacle during the experiment was measured using a motion capture system to be 6.24 m/s. The video of this experiment can be found at the link in [250].

both the robot and obstacle states. The goal position for each controller was $(0, 0)$ which the MPC and SUP-MARIO controllers tracked using a quadratic receding horizon cost. The DCBF controller tracked the goal position using a proportional-derivative nominal controller which was pointwise modified to enforce safety. The obstacle was a sphere with radius $r_{\text{obs}} = 0.15$ m that tracked a trajectory that passed

through $(0, 0)$ which was both the robot's initial and goal position.

While the $\mathbf{k}^{\text{DCBF-OP}}$ and $\mathbf{k}^{\text{SUP-MARIO}}$ controllers both achieve safety, the myopia of the DCBF-OP safety filter causes it to produce trajectories that are safe but highly suboptimal by moving in the same direction as the obstacle, resulting in significant departure from the goal position. The $\mathbf{k}^{\text{MPC}}$ controller with state constraints achieves performant behavior by planning a trajectory that moves perpendicular to the obstacle path; however, it lacks sufficient robustness resulting in a safety failure in simulation. Finally, the $\mathbf{k}^{\text{SUP-MARIO}}$ controller combines the benefits of both methods and achieves safe and performant behavior, effectively side-stepping with sufficient margin to avoid collision.

The simulations were performed in a ROS-based simulation environment which models the full-robot multi-layer control and communication system.

Hardware Setup: The quadrotor hardware platform is the same as in Section 7.2, but with the addition of an Intel RealSense D455 depth camera and with the Nvidia Jetson Tx2 replaced with the more powerful Nvidia Jetson Orin Nx onboard computer. We utilize the IMU and its internal Kalman filter for orientation state estimation, and we use an OptiTrack motion capture system for global position measurements from which velocities are also estimated via finite-difference and a low-pass filter. A diagram of the quadrotor is provided in Fig. 7.15.

The environment sensing system utilizes an RGBd video stream (4 channels: 3 color channels and 1 depth channel) generated by the Intel Realsense D455 stereo depth camera. To ensure time alignment between the color and depth images, the stereo image used to calculate the depth is simultaneously used as the RGB image, and the active infrared projector is disabled. To segment the obstacle within an image and track it between frames, we utilize the efficient Track-Anything-Model (efficientTAM) [259], a distillation of Meta's Segment-Anything-Model [260], which achieves faster than 11 Hz image segmentation and tracking on the Jetson Orin NX. To identify obstacles, we initialize the efficientTAM model with prompts (mouse clicks on an image) which identify key points in the image. For the obstacle we predominately used the red ball shown in Figs. 7.16 and 7.14, but the experimental video found at the link in [250] and Fig. 7.17 also show a demonstration of the SUP-MARIO controller dodging a green turtle shell.

Once a segmentation mask is obtained for an RGBd image, the intrinsic camera matrix and the geometry of the robot are used to generate 3D vectors representing

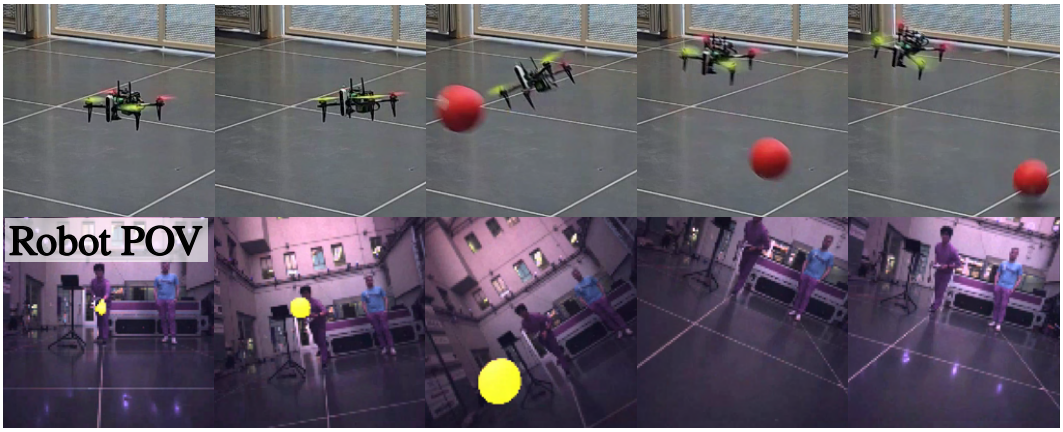


Figure 7.16. A dynamic avoidance maneuver generated by the MPC+DCBF controller. The bottome row shows the first-person-view (FPV) masked image used for state estimation. A video of these experiments can be found at the link in [250].

the relative position of the pixelized masked image contents from the robot body reference frame. We then perform a weighted averaging of those vectors based on their distance to the center of the mask to estimate the relative position of the obstacle centroid with respect to the camera frame. This relative position is converted to a global frame using the time-synchronized drone state. Given this position estimate, a Kalman filter for a double integrator system is used to estimate the relative obstacle position $\tilde{\mathbf{p}}$ and velocity $\dot{\tilde{\mathbf{p}}}$ which relate to system safety via h_0 in (7.53) and h in (7.54).

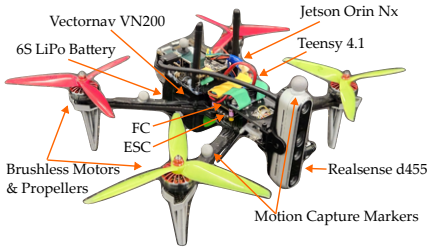


Figure 7.15. A diagram of the quadrotor robot used in the dynamical obstacle avoidance experiments and Figs. 7.16 and 7.14.

Finally, the $\mathbf{k}^{\text{SUP-MARIO}}$ controller is implemented at 20 Hz with a horizon length of $N = 20$ and a $\Delta_t = 0.05$ sec, for a total real-time horizon length of 1 second. To solve the non-convex optimization problem we use a sequential quadratic programming (SQP) method in a real-time iteration (RTI) implementation [264]. The DCBF-OP and MPC controllers were not demonstrated on hardware due to practical safety concerns with the simulated trajectories in Fig. 7.13. Results from these experiments can be seen in Figs. 7.16 and 7.14 and a video can be found at the link in [250] where the quadrotor drone successfully avoids dynamic projectile obstacles over several trials. For the experiment shown in Fig. 7.14, motion capture markers were added to the obstacle to provide ground truth state information, but this ground truth was *not used for real-time collision avoidance*.

Figure 7.17. The quadrotor robot avoiding a collision with the toy turtle shell.

These ground truth measurements only used to obtain the true obstacle velocities, and from this we know that the quadrotor robot successfully dodged obstacles moving at upwards of $6.24\ \mathrm{m/s}$, overcoming the significant uncertainty resulting from the noisy, low-frame-rate ($11\ \mathrm{Hz}$) environmental perception and from its uncertain, reduced-order-model of its dynamics.

Conclusion

In this section we studied the combination of two predominant control techniques: model predictive control (MPC) and control barrier function (CBF)-based safety filters. By combining the cost function and horizon-based planning of the MPC problem with the DCBF-based safety constraint, we found both practical and theoretical benefits in nominal operation, operation under bounded uncertainty, and operation under (potentially unbounded) stochastic state and dynamic uncertainty, that extend the capabilities beyond either of the individual methods. We show that the unified MPC+DCBF controller displays favorable safety, performance, and closed-loop feasibility properties, and we demonstrate the utility of this controller via quadrupedal and quadrotor experiments for dynamic obstacle avoidance.

7.5 Conclusion

This chapter provided several examples of DCBF methods achieving robust safety-critical control for robots operating under significant real-world uncertainty, with practical examples of (1) a quadrotor robot operating safely with chaotic disturbances caused by unmodeled masses attached with flexible cables in Section 7.2, (2) a humanoid robot safely navigating complex environments in Section 7.3, and (3) a quadruped and a quadrotor robot achieving highly dynamic obstacle avoidance in Section 7.4.

To realize these behaviors, Sections 7.2 and 7.3 leveraged generative modeling techniques to capture the effects of disturbances on the system and Section 7.4

improved its real-world safety and performance using horizon-based planning.

The contributions of this chapter open up several avenues for future work. Firstly, although we can create probabilistic guarantees of safety using real-time controllers, these probability bounds are very loose. Future work will examine extensions beyond martingale concentration, Cantelli's, and Boole's inequalities that improve theoretical guarantees without sacrificing real-time capabilities. Secondly, significant work should be invested in defining the functions h given sensor output, developing methods for converting sensor information to scene understanding and safety requirements while optimizing them to yield improved probability bounds. Thirdly, since the distributions used to implement the controllers[11] are not the *true* uncertainty distributions, the guarantees of Chapter 6 do not necessarily hold. Future work will explore how these guarantees can be retained in the context of learned models.

[11] The controllers of Sections 7.2 and 7.3 were implemented by using CVAEs to approximate the disturbance distribution and the controller in Section 7.4 was implemented assuming the standard zero-mean process and measurement noise of a Kalman filter.

Chapter 8

CONCLUSION

This thesis explored the problem of achieving dynamic safety for robots facing real-world uncertainties. Based on control-theoretic foundations, this thesis presented: proof techniques for achieving robust safety guarantees in the face of uncertainty, algorithms for improving the performance of these systems while achieving safety, and hardware demonstrations of these methods achieving safe and performant behaviors.

Two main theoretical paradigms for robust safety were considered, namely, worst-case safety bounds and probabilistic safety guarantees. In both cases, we considered various sources of uncertainty including both model mismatch and measurement/state uncertainty and develop practical control algorithms for achieving those guarantees that we deploy on hardware systems.

Contributions

The main contributions of this thesis are:

- **Safe-Set Synthesis Methods**: Chapter 3 tackled the challenge of synthesizing control invariant sets based on user-defined safety criteria and provided constructive methods for several highly relevant classes of systems that enable CBF synthesis.
- **Robustness Guarantees with Bounded Uncertainty**: Chapter 4 presented several techniques for achieving robust guarantees in the presence of a variety of bounded uncertainty including measurement uncertainty, dynamics uncertainty, and errors in imitation learning. The contributions of this chapter constituted the first theoretical analysis and hardware deployment of the MRCBF method with guaranteed robustness to measurement uncertainty.
- **Learning-based Performance Improvements**: Chapter 5 presented several methods for improving the closed-loop behavior of safety-critical systems by learning improved models of dynamics and perception systems or by learning intangible concepts like desired robustness and social responsibility. These learned components are critical for safety and enabled significant performance improvements. This chapter included the first hardware demonstrations of

CBFs on a bipedal robot and the first use of preference-based learning in conjunction with CBFs.

- **Stochastic Safety Guarantees**: Chapter 6 presented several novel proofs providing probabilistic guarantees of safety for DCBF-based control. It also provided practical considerations for realizing these controllers and novel martingale-based proofs of probabilistic safety.

- **Deployment of Stochastic Guarantees**: Chapter 7 presented novel frameworks for deploying the stochastic control methods of Chapter 6 by unifying them with generative modeling techniques, resulting in dynamic safety for a quadrotor and humanoid robot operating under significant uncertainty; these results constituted the first demonstrations of the martingale-based DCBF safety paradigm on hardware. This chapter also demonstrates the horizon-based performance improvements and the inherent robustness guarantees that come with DCBF-based methods.

Limitations and Directions for Future Work

The results of this thesis present several interesting directions for future work.

1. Improved Stochastic Guarantees: I believe that the stochastic results of Chapter 6 provide a solid foundation from which build improved guarantees. In particular, interesting directions for future work include bridging the continuous-time models of the first portion of this thesis with the distrete-time stochastic analysis of the second half by analyzing a sampled-data formulation of the SDEs and CBF-based guarantees in [110], [130], [193].

Additionally, while the stochastic guarantees of this thesis proved useful for the hardware deployment in Chapter 7, the use of a generative model in place of the true distribution results in a gap between the theoretical and practical deployment. I believe that there are fruitful research opportunities in analyzing the distribution shift between the true and learned uncertainties and the distributional robustness of the martingale-based DCBF methods considered in this thesis.

Furthermore, the represent here our notion of probabilistic safety (Def. 6.7) is based simply on the finite-horizon probability that $h(\mathbf{x})$ will taken on a negative value. An interesting direction for future work is to merge the analysis and controllers of Chapters 6 and 7 with more nuanced understanding of risk, such as coherent risk

measures like CVaR [265]. I believe that our martingale-based methods provide a natural paradigm from which we can generalize to coherent risk measures.

Finally, the stochastic guarantees considered here are quite loose. While the generality of the martingale framework allows these methods to be implemented in real-time with relatively little *a priori* distribution knowledge, it also results in poorly calibrated, highly conservative probability bounds whose utility is often more as a guiding tool in choosing α than in directly prescribing a value. Future research directions should consider methods for improving the calibration of these bounds, potentially by reconsidering the source of uncertainty in the model, considering multi-model safety constraints, and/or by applying sampling-based methods such as [266]–[268].

2. Safe Learning: While the learning methods introduced in this thesis in both Chapters 5 and 7 resulted in desirable closed-loop behavior, they compromised the theoretical guarantees and often required *a priori* data collection. Safe learning and particularly safe *online* learning [187], where safety guarantees are retained during learning and models are improved performed during deployment, represent an important future direction for this work with ultimate goal of achieving *safe lifelong learning*. A fruitful research area may be in combining the results of robust adaptive control with safety-aware learning [99] and sampling-based verification methods [266].

Additionally, I believe that the generative modeling of dynamics residuals in Chapter 7 also produces several interesting directions for future work in closing the sim-to-real gap. As demonstrated with SHIELD (Sec. 7.3), these techniques have utility in modeling uncertain dynamics to achieve improvements in both trajectory tracking and safety. This ability can further help close the sim-to-real gap by improving simulations of highly uncertain systems and by making real-world systems behave more like simulated ones, similar to the residual modeling in [269]. Furthermore, the CVAE methods shown here has difficulties representing truly multi-modal distributions [225] and, with improvements in computational capabilities, we may be able to use more capable generative modeling techniques such as diffusion models [221] in high-frequency components of the robot's control system.

3. Bridging Control-theoretic and Human Understandings of Safety: Chapter 1 motivated the study of safety with Asimov's Three Laws of Robotics, a human-interpretable description of dynamic safety. Meanwhile, the safety-critical control tools introduced in Chapter 2 and used throughout the rest of this thesis assume

the existence of predefined safety criteria. I believe that this difference reveals an important direction for future work that involves bridging this gap between human-interpretable safety descriptions and control theoretic ones. Extending the safe-set synthesis methods of Chapter 3, this work will additionally require methods that allow for the rapid synthesis of semantically meaningful *safety criteria*, replacing the need for a "user-defined safe set $\mathcal{C}$," potentially from logic specifications [270] or directly from data [271].

This also introduces interesting new directions that involve extending definitions of safety beyond collision avoidance to consider more nuanced understandings that cannot be neatly divided into binary safe/not-safe states. While the probabilistic safety guarantees of Chapter 6 begin to provide an understanding of the importance of the *value* of $h(\mathbf{x})$ beyond just its sign and rate of change, future directions should develop an understanding of relative degrees of safety and the importance of the value of $h(\mathbf{x})$ in how it can be used to modulate risk and how it relates to human understandings of safety. This more nuanced understand of safety and the value of $h(\mathbf{x})$ will enable the future co-optimization of safety and performance.

4. Computationally Accelerated Safe Set Synthesis: Although Chapter 3 presented several methods for rapid generation of CBFs or control invariant sets, it relied on several assumptions regarding the system structure. Reachability methods [4], [8] present a more complete solution to this problem, but have historically been limited by their computational complexity. As technology improves, I believe that this problem will likely be worth revisiting as we may be able to rapidly solve, or at least approximately solve, these reachability problems with improved compute or learning-based methods.

BIBLIOGRAPHY

[1] J. C. Knight, "Safety critical systems: Challenges and directions," in *International Conference on Software Engineering (ICSE)*, 2002, pp. 547–550.

[2] J. Guiochet, M. Machin, and H. Waeselynck, "Safety-critical advanced robots: A survey," *Robotics and Autonomous Systems*, vol. 94, pp. 43–52, 2017, ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2017.04.004.

[3] I. Asimov, *Handbook of Robotics, 56th Edition, 2058 A.D.* 1942.

[4] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-jacobi reachability: A brief overview and recent advances," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 2242–2253. DOI: 10.1109/CDC.2017.8263977.

[5] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[6] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3420–3431. DOI: 10.23919/ECC.2019.8796030.

[7] J.-P. Aubin, A. M. Bayen, and P. Saint-Pierre, *Viability theory: New directions*. Springer Science & Business Media, 2011.

[8] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust control barrier–value functions for safety-critical control," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 6814–6821. DOI: 10.1109/CDC45484.2021.9683085.

[9] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

[10] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 6271–6278. DOI: 10.1109/CDC.2014.7040372.

[11] Z. Artstein, "Stabilization with relaxed controls," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 7, no. 11, pp. 1163–1173, 1983.

[12] E. D. Sontag, "A 'universal' construction of artstein's theorem on nonlinear stabilization," *Systems & Control Letters*, vol. 13, no. 2, pp. 117–123, 1989, ISSN: 0167-6911. DOI: https://doi.org/10.1016/0167-6911(89)90028-5. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0167691189900285.

[13] M. Nagumo, "Über die lage der integralkurven gewöhnlicher differentialgleichungen," *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, vol. 24, pp. 551–559, 1942.

[14] P. Mestres and J. Cortés, "Converse theorems for certificates of safety and stability," *arXiv preprint arXiv:2406.14823*, 2024.

[15] T. Gurriet, A. Singletary, J. Reher, L. Ciarletta, E. Feron, and A. D. Ames, "Towards a framework for realizable safety critical control through active set invariance," in *International Conference on Cyber-Physical Systems (ICCPS)*, IEEE Press, 2018, pp. 98–106.

[16] P. Mestres, Y. Chen, E. Dall'anese, and J. Cortés, "Control barrier function-based safety filters: Characterization of undesired equilibria, unbounded trajectories, and limit cycles," *arXiv preprint arXiv:2501.09289*, 2025.

[17] L. Yang, B. Werner, R. K. Cosner, D. Fridovich-Keil, P. Culbertson, and A. Ames, "Shield: Safety on humanoids via cbfs in expectation on learned dynamics," *submitted to the 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025. [Online]. Available: `https://arxiv.org/pdf/2505.11494`,

[18] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017. DOI: `10.1109/TAC.2016.2638961`.

[19] A. Alan, A. J. Taylor, C. R. He, A. D. Ames, and G. Orosz, "Control barrier functions and input-to-state safety with application to automated vehicles," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 6, pp. 2744–2759, 2023.

[20] P. Glotfelter, J. Cortés, and M. Egerstedt, "Boolean composability of constraints and control synthesis for multi-robot systems via nonsmooth control barrier functions," in *Conference on Control Technology and Applications (CCTA)*, IEEE, 2018, pp. 897–902.

[21] M. Vahs, C. Pek, and J. Tumova, "Belief control barrier functions for risk-aware control," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 8565–8572, 2023. DOI: `10.1109/LRA.2023.3330662`.

[22] A. Singletary, A. Swann, Y. Chen, and A. D. Ames, "Onboard safety guarantees for racing drones: High-speed geofencing with control barrier functions," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2897–2904, 2022.

[23] R. M. Bena, C. Zhao, and Q. Nguyen, "Safety-aware perception for autonomous collision avoidance in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 7962–7969, 2023.

[24] A. Singletary, W. Guffey, T. G. Molnar, R. Sinnet, and A. D. Ames, "Safety-critical manipulation for collision-free food preparation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 954–10 961, 2022.

[25] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, "Multi-layered safety for legged robots via control barrier functions and model predictive control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 8352–8358.

[26] T. Gurriet, M. Mote, A. D. Ames, and E. Feron, "An online approach to active set invariance," in *Conference on Decision & Control (CDC)*, IEEE, 2018, pp. 3592–3599.

[27] N. Csomay-Shanklin, R. K. Cosner, M. Dai, A. J. Taylor, and A. D. Ames, "Episodic learning for safe bipedal locomotion with control barrier functions and projection-to-state safety," *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, vol. 144, pp. 1041–1053, 2021. [Online]. Available: https://proceedings.mlr.press/v144/csomay-shanklin21a.html,

[28] K. Zhou and J. C. Doyle, *Essentials of robust control*. Prentice Hall, 1998, vol. 104.

[29] E. D. Sontag, "Input to state stability: Basic concepts and results," in *Nonlinear and Optimal Control Theory*, Springer, 2008, pp. 163–220.

[30] W. Langson, I. Chryssochoos, S. Raković, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.

[31] S. Kolathaya and A. D. Ames, "Input-to-state safety with control barrier functions," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 108–113, 2019. DOI: 10.1109/LCSYS.2018.2853698.

[32] A. Alan, A. J. Taylor, C. R. He, G. Orosz, and A. D. Ames, "Safe controller synthesis with tunable input-to-state safe control barrier functions," *Control Systems Letters*, vol. 6, pp. 908–913, 2022.

[33] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[34] A. J. Taylor and A. D. Ames, "Adaptive safety with control barrier functions," in *American Control Conference (ACC)*, IEEE, 2020, pp. 1399–1405.

[35] R. E. Mortensen, "Stochastic stability and control," *SIAM Review*, vol. 10, no. 4, pp. 460–462, 1968, ISSN: 00361445. [Online]. Available: http://www.jstor.org/stable/2027204 (visited on 11/05/2024).

[36] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, eabk2822, 2022.

[37] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "Tossingbot: Learning to throw arbitrary objects with residual physics," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1307–1319, 2020. DOI: 10.1109/TRO.2020.2988642.

[38] Z. Zhuang, S. Yao, and H. Zhao, "Humanoid parkour learning," *arXiv preprint arXiv:2406.10759*, 2024.

[39] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, A. M. Bayen, A. Jadbabaie, G. Pappas, *et al.*, Eds., ser. Proceedings of Machine Learning Research, vol. 120, PMLR, 2020, pp. 708–717.

[40] G. Shi, X. Shi, M. O'Connell, *et al.*, "Neural lander: Stable drone landing control using learned dynamics," in *2019 International Conference on Robotics and Automation (ICRA)*, ISSN: 2577-087X, May 2019, pp. 9784–9790. DOI: 10.1109/ICRA.2019.8794351.

[41] A. Robey, H. Hu, L. Lindemann, *et al.*, "Learning control barrier functions from expert demonstrations," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 3717–3724. DOI: 10.1109/CDC42340.2020.9303785.

[42] J. Choi, F. Castañeda, C. Tomlin, and K. Sreenath, "Reinforcement learning for safety-critical control under model uncertainty, using control Lyapunov functions and control barrier functions," in *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, 2020. DOI: 10.15607/RSS.2020.XVI.088.

[43] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 3387–3395, 2019. DOI: 10.1609/aaai.v33i01.33013387.

[44] C. W. Warren, "Global path planning using artificial potential fields," in *1989 IEEE International Conference on Robotics and Automation*, IEEE Computer Society, 1989, pp. 316–317.

[45] K. Garg, R. K. Cosner, U. Rosolia, A. D. Ames, and D. Panagou, "Multi-rate control design under input constraints via fixed-time barrier functions," *IEEE Control Systems Letters*, vol. 6, pp. 608–613, 2022, ISSN: 2475-1456. DOI: 10.1109/LCSYS.2021.3084322,

[46] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *2021 American Control Conference (ACC)*, 2021, pp. 3882–3889. DOI: 10.23919/ACC50511.2021.9483029.

[47] J. Zeng, Z. Li, and K. Sreenath, "Enhancing feasibility and safety of nonlinear model predictive control with discrete-time control barrier functions," in *2021 60th IEEE Conference on Decision and Control (CDC)*, IEEE, 2021, pp. 6137–6144.

[48] T. G. Molnar, R. K. Cosner, A. W. Singletary, W. Ubellacker, and A. D. Ames, "Model-free safety-critical control for robotic systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 944–951, 2022, ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2021.3135569,

[49] M. H. Cohen, R. K. Cosner, and A. D. Ames, "Constructive safety-critical control: Synthesizing control barrier functions for partially feedback linearizable systems," *IEEE Control Systems Letters*, pp. 2229–2234, 2024. DOI: 10.1109/LCSYS.2024.3412003,

[50] G. Bahati, R. K. Cosner, M. H. Cohen, R. M. Bena, and A. D. Ames, "Control barrier function synthesis for nonlinear systems with dual relative degree," *submitted to the 2025 IEEE 64th Conference on Decision and Control (CDC)*, 2025. [Online]. Available: https://arxiv.org/pdf/2504.00397,

[51] S. Dean, A. J. Taylor, R. K. Cosner, B. Recht, and A. D. Ames, "Guaranteeing safety of learned perception modules via measurement-robust control barrier functions," *Proceedings of the 2020 Conference on Robot Learning*, vol. 155, pp. 654–670, 2021. [Online]. Available: https://proceedings.mlr.press/v155/dean21a.html,

[52] R. K. Cosner, A. W. Singletary, A. J. Taylor, T. G. Molnar, K. L. Bouman, and A. D. Ames, "Measurement-robust control barrier functions: certainty in safety with uncertainty in state," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021. DOI: 10.1109/IROS51168.2021.9636584,

[53] R. K. Cosner, M. Tucker, A. J. Taylor, K. Li, T. G. Molnar, W. Ubellacker, A. Alan, G. Orosz, Y. Yue, and A. D. Ames, "Safety-aware preference-based learning for safety-critical control," *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, Proceedings of Machine Learning Research, vol. 168, pp. 1020–1033, 2022. [Online]. Available: https://proceedings.mlr.press/v168/cosner22a.html,

[54] R. K. Cosner, Y. Yue, and A. D. Ames, "End-to-end imitation learning with safety guarantees using control barrier functions," *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 5316–5322, 2022. DOI: 10.1109/CDC51059.2022.9993193,

[55] R. K. Cosner, I. D. J. Rodriguez, T. G. Molnar, W. Ubellacker, Y. Yue, A. D. Ames, and K. L. Bouman, "Self-supervised online learning for safety-critical control using stereo vision," *2022 International Conference on Robotics and*

Automation (ICRA), pp. 11 487–11 493, 2022. DOI: 10.1109/ICRA46639.2022.9812183,

[56] R. K. Cosner, Y. Chen, K. Leung, and M. Pavone, "Learning responsibility allocations for safe human-robot interaction with applications to autonomous driving," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9757–9763, 2023. DOI: 10.1109/ICRA48891.2023.10161112,

[57] R. K. Cosner, P. Culbertson, A. J. Taylor, and A. D. Ames, "Robust safety under stochastic uncertainty with discrete-time control barrier functions," *Proceedings of Robotics: Science and Systems*, 2023. DOI: 10.15607/RSS.2023.XIX.084,

[58] R. K. Cosner, P. Culbertson, and A. D. Ames, "Bounding stochastic safety: Leveraging freedman's inequality with discrete-time control barrier functions," *IEEE Control Systems Letters*, pp. 193 7 –1942, 2024. DOI: 10.1109/LCSYS.2024.3409105,

[59] R. K. Cosner, I. Sadalski, J. K. Woo, P. Culbertson, and A. D. Ames, "Generative modeling of residuals for real-time risk-sensitive safety with discrete-time control barrier functions," *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024. DOI: 10.1109/ICRA57147.2024.10611355,

[60] R. K. Cosner, R. M. Bena, and A. D. Ames, "Unified mpc+cbf control for performant safety: Mutual benefits and inherent robustness properties," *submitted to IEEE Transactions on Robotics*, 2025. [Online]. Available: http://www.rkcosner.com/assets/files/dodgeball_paper.pdf,

[61] H. K. Khalil, *Nonlinear systems*, 3rd Edition. Upper Saddle River: Prentice Hall, 2002.

[62] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, "Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, 2014.

[63] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, 2018.

[64] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[65] E. D. Sontag and Y. Wang, "On characterizations of the input-to-state stability property," *Systems & Control Letters*, vol. 24, no. 5, pp. 351–359, 1995.

[66] E. D. Sontag and Y. Wang, "On characterizations of input-to-state stability with respect to compact sets," *IFAC Proceedings Volumes*, vol. 28, no. 14, pp. 203–208, 1995.

[67] A. J. Taylor, *Robust safety-critical control: A Lyapunov and barrier approach*. California Institute of Technology, 2023.

[68] F. Blanchini, S. Miani, *et al.*, *Set-theoretic methods in control*. Springer, 2008, vol. 78.

[69] J.-P. Aubin and H. Frankowska, *Set-valued analysis*. Springer Science & Business Media, 2009.

[70] R. Konda, A. D. Ames, and S. Coogan, "Characterizing safety: Minimal control barrier functions from scalar comparison systems," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 523–528, 2021. DOI: 10.1109/LCSYS.2020.3003887.

[71] X. Xu, J. W. Grizzle, P. Tabuada, and A. D. Ames, "Correctness guarantees for the composition of lane keeping and adaptive cruise control," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1216–1229, 2018. DOI: 10.1109/TASE.2017.2760863.

[72] P. Zhao, R. Ghabcheloo, Y. Cheng, H. Abdi, and N. Hovakimyan, "Convex synthesis of control barrier functions under input constraints," *IEEE Control Systems Letters*, vol. 7, pp. 3102–3107, 2023.

[73] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "Fastrack: A modular framework for fast and guaranteed safe motion planning," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 1517–1522. DOI: 10.1109/CDC.2017.8263867.

[74] G. Wu and K. Sreenath, "Safety-critical control of a planar quadrotor," in *2016 American Control Conference (ACC)*, 2016, pp. 2252–2258. DOI: 10.1109/ACC.2016.7525253.

[75] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*, 2016, pp. 322–328. DOI: 10.1109/ACC.2016.7524935.

[76] W. Xiao and C. Belta, "High-orderccontrol barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3655–3662, 2022. DOI: 10.1109/TAC.2021.3105491.

[77] X. Xu, "Constrained control of input–output linearizable systems using control sharing barrier functions," *Automatica*, vol. 87, pp. 195–201, 2018, ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2017.10.005.

[78] X. Tan, W. S. Cortez, and D. V. Dimarogonas, "High-order barrier functions: Robustness, safety, and performance-critical control," *IEEE Transactions on Automatic Control*, vol. 67, no. 6, pp. 3021–3028, 2022. DOI: 10.1109/TAC.2021.3089639.

[79] M. H. Cohen, P. Ong, G. Bahati, and A. D. Ames, "Characterizing smooth safety filters via the implicit function theorem," *IEEE Control Systems Letters*, vol. 7, pp. 3890–3895, 2023. DOI: 10.1109/LCSYS.2023.3341345.

[80] A. J. Taylor, P. Ong, T. G. Molnar, and A. D. Ames, "Safe backstepping with control barrier functions," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 5775–5782. DOI: 10.1109/CDC51059.2022.9992763.

[81] S. Singh, M. Chen, S. L. Herbert, C. J. Tomlin, and M. Pavone, "Robust tracking with model mismatch for fast and safe planning: An SOS optimization approach," in *International Workshop on the Algorithmic Foundations of Robotics*, M. Morales, L. Tapia, G. Sánchez-Ante, and S. Hutchinson, Eds., 2020, pp. 545–564. DOI: 10.1007/978-3-030-44051-0_32.

[82] L. Sabattini, C. Secchi, N. Chopra, and A. Gasparri, "Distributed control of multirobot systems with global connectivity maintenance," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1326–1332, 2013. DOI: 10.1109/TRO.2013.2267971.

[83] S. Zhao and Z. Sun, "Defend the practicality of single-integrator models in multi-robot coordination control," in *IEEE International Conference on Control Automation*, 2017, pp. 666–671. DOI: 10.1109/ICCA.2017.8003139.

[84] A. Singletary, K. Klingebiel, J. R. Bourne, A. Browning, P. Tokumaru, and A. D. Ames, "Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[85] A. De Luca, G. Oriolo, and M. Vendittelli, "Control of wheeled mobile robots: An experimental overview," in *Lecture Notes in Control and Information Sciences*, S. Nicosia, S. B., A. Bicchi, and P. Valigi, Eds., vol. 270, Berlin: Springer, 2001, pp. 181–226. DOI: 10.1007/3-540-45000-9_8.

[86] D. Koung, I. Fantoni, O. Kermorgant, and L. Belouaer, "Consensus-based formation control and obstacle avoidance for nonholonomic multi-robot system," in *International Conference on Control, Automation, Robotics and Vision*, 2020, pp. 92–97. DOI: 10.1109/ICARCV50220.2020.9305426.

[87] M. H. Cohen, T. G. Molnar, and A. D. Ames, "Safety-critical control for autonomous systems: Control barrier functions via reduced-order models," *Annual Reviews in Control*, vol. 57, p. 100 947, 2024, ISSN: 1367-5788. DOI: https://doi.org/10.1016/j.arcontrol.2024.100947. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1367578824000166.

[88] W. D. Compton, N. Csomay-Shanklin, C. Johnson, and A. D. Ames, *Dynamic tube mpc: Learning tube dynamics with massively parallel simulation*

for robust safety in practice, 2024. arXiv: 2411.15350 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2411.15350.

[89] A. Papachristodoulou and S. Prajna, "A tutorial on sum of squares techniques for systems analysis," in *Proceedings of the 2005, American Control Conference, 2005.*, 2005, 2686–2700 vol. 4. DOI: 10.1109/ACC.2005.1470374.

[90] K. J. Åström and R. Murray, *Feedback systems: An introduction for scientists and engineers*. Princeton University Press, 2021.

[91] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. New York: John Wiley and Sons, 2005.

[92] P. Glotfelter, J. Cortes, and M. Egerstedt, "A nonsmooth approach to controller synthesis for Boolean specifications," *IEEE Transactions on Automatic Control*, pp. 1–1, 2020. DOI: 10.1109/TAC.2020.3035467.

[93] Supplemental Video for "Model-free safety-critical control for robotic systems." https://youtu.be/_h8KTLsBGvw.

[94] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," en, in *49th IEEE Conference on Decision and Control (CDC)*, Atlanta, GA: IEEE, Dec. 2010, pp. 5420–5425. DOI: 10.1109/CDC.2010.5717652.

[95] J. Buchli, M. Kalakrishnan, M. Mistry, P. Pastor, and S. Schaal, "Compliant quadruped locomotion over rough terrain," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2009, pp. 814–820.

[96] W. Ubellacker, N. Csomay-Shanklin, T. G. Molnar, and A. D. Ames, "Verifying safe transitions between dynamic motion primitives on legged robots," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 8477–8484. DOI: 10.1109/IROS51168.2021.9636537.

[97] A. Isidori, *Nonlinear control systems*. Springer-Verlag London, 1995, vol. 3.

[98] Supplemental Video for "Constructive safety-critical control: synthesizing control barrier functions for partially feedback linearizable systems." https://youtu.be/04Cu8ChoMAo.

[99] M. H. Cohen and C. Belta, *Adaptive and learning-based control of safety-critical systems*. Springer Nature, 2023.

[100] Supplemental Video for "Control barrier function synthesis for nonlinear systems with dual relative degree." https://youtu.be/sOU5oED-9Y4.

[101] L. Doeser, P. Nilsson, A. D. Ames, and R. M. Murray, "Invariant sets for integrators and quadrotor obstacle avoidance," in *2020 American Control Conference (ACC)*, 2020, pp. 3814–3821. DOI: 10.23919/ACC45564.2020.9147872.

[102] D. Liberzon, *Switching in systems and control*. Birkhäuser Boston, 2003.

[103] T. Gurriet, M. Mote, A. Singletary, P. Nilsson, E. Feron, and A. D. Ames, "A scalable safety critical control framework for nonlinear systems," *IEEE Access*, vol. 8, pp. 187 249–187 275, 2020. DOI: 10.1109/ACCESS.2020.3025248.

[104] W. Tan and A. Packard, "Stability region analysis using sum of squares programming," in *2006 American Control Conference*, 2006, pp. 2297–2302. DOI: 10.1109/ACC.2006.1656562.

[105] D. E. J. Van Wijk, S. Coogan, T. G. Molnar, M. Majji, and K. L. Hobbs, "Disturbance-robust backup control barrier functions: Safety under uncertain dynamics," *IEEE Control Systems Letters*, 2024.

[106] G. Bahati, R. M. Bena, and A. D. Ames, "Dynamic safety in complex environments: Synthesizing safety filters with poisson's equation," in *Robotics: Science and Systems*, 2025.

[107] J. Sieber, S. Bennani, and M. N. Zeilinger, "A system level approach to tube-based model predictive control," *IEEE Control Systems Letters*, vol. 6, pp. 776–781, 2021.

[108] A. J. Taylor, A. Singletary, Y. Yue, and A. D. Ames, "A control barrier perspective on episodic learning via projection-to-state safety," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 1019–1024, 2021. DOI: 10.1109/LCSYS.2020.3009082.

[109] R. Freeman, "Global internal stabilizability does not imply global external stabilizability for small sensor disturbances," *IEEE Transactions on Automatic Control*, vol. 40, no. 12, pp. 2119–2122, 2002.

[110] A. Clark, "Control barrier functions for complete and incomplete information stochastic systems," in *2019 American Control Conference (ACC)*, 2019, pp. 2928–2935. DOI: 10.23919/ACC.2019.8814901.

[111] P. Nilsson and A. D. Ames, "Lyapunov-like conditions for tight exit probability bounds through comparison theorems for sdes," in *2020 American Control Conference (ACC)*, 2020, pp. 5175–5181. DOI: 10.23919/ACC45564.2020.9147414.

[112] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[113] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[114] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[115] F. Codevilla, M. Miiller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 1–9.

[116] Y. Pan, C.-A. Cheng, K. Saigol, *et al.*, "Agile autonomous driving using end-to-end deep imitation learning," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, 2018. DOI: 10.15607/RSS.2018.XIV.056.

[117] D. S. Brown, Y. Cui, and S. Niekum, "Risk-aware active inverse reinforcement learning," in *Conference on Robot Learning*, PMLR, 2018, pp. 362–372.

[118] J. Zhang and K. Cho, "Query-efficient imitation learning for end-to-end simulated driving," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.

[119] S. Yaghoubi, G. Fainekos, and S. Sankaranarayanan, "Training neural network controllers using control barrier functions in the presence of disturbances," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2020, pp. 1–6.

[120] A. Lambert, A. Shaban, A. Raj, Z. Liu, and B. Boots, "Deep forward and inverse perceptual models for tracking and prediction," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 675–682.

[121] S. Tang, V. Wüest, and V. Kumar, "Aggressive flight with suspended payloads using vision-based control," *Robotics and Automation Letters*, vol. 3, no. 2, pp. 1152–1159, 2018.

[122] Supplemental Video for "Guaranteeing safety of learned perception modules via measurement-robust control barrier functions." https://youtu.be/qMKKnhc6Je4.

[123] Supplemental Video for "Measurement-robust control barrier functions: certainty in safety with uncertainty in state." https://youtu.be/xw4yy2XQEHw.

[124] A. Domahidi, E. Chu, and S. Boyd, "Ecos: An socp solver for embedded systems," in *European Control Conference (ECC)*, IEEE, 2013, pp. 3071–3076.

[125] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[126] Supplemental Video for "Safety-aware preference-based learning for safety-critical control." https://youtu.be/fEYkCY17xtY.

[127] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, *et al.*, "Maximum entropy inverse reinforcement learning.," in *AAAI Conference on Artificial Intelligence*, vol. 8, 2008, pp. 1433–1438.

[128] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[129] J. Breeden, K. Garg, and D. Panagou, "Control barrier functions in sampled-data systems," *IEEE Control Systems Letters*, vol. 6, pp. 367–372, 2022. DOI: 10.1109/LCSYS.2021.3076127.

[130] O. So, A. Clark, and C. Fan, *Almost-sure safety guarantees of stochastic zero-control barrier functions do not hold*, 2023. arXiv: 2312.02430 [math.OC]. [Online]. Available: https://arxiv.org/abs/2312.02430.

[131] C. Santoyo, M. Dutreix, and S. Coogan, "A barrier function approach to finite-time stochastic system verification and control," *Automatica*, vol. 125, p. 109 439, 2021, ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2020.109439. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109820306415.

[132] U. Rosolia, A. Carvalho, and F. Borrelli, "Autonomous racing using learning model predictive control," in *2017 American Control Conference (ACC)*, IEEE, 2017, pp. 5115–5120.

[133] Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, and K. Sreenath, "3d dynamic walking on stepping stones with control barrier functions," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 2016, pp. 827–834.

[134] Supplemental Video for "Episodic learning for safe bipedal locomotion with control barrier functions and projection-to-state safety." https://youtu.be/NASRlnUIZ7U.

[135] A. J. Taylor, V. D. Dorobantu, H. M. Le, Y. Yue, and A. D. Ames, "Episodic learning with control lyapunov functions for uncertain robotic systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 6878–6884.

[136] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 42–56, 2003.

[137] E. Ambrose, W.-L. Ma, C. Hubicki, and A. D. Ames, "Toward benchmarking locomotion economy across design configurations on the modular robot: Amber-3m," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 2017, pp. 1270–1276.

[138] W.-L. Ma, H.-H. Zhao, S. Kolathaya, and A. D. Ames, "Human-inspired walking via unified pd and impedance control," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 5088–5094.

[139] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC Press, 2018.

[140] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018.

[141] E. Hazan, "Introduction to online convex optimization," *Foundations and trends in optimization*, vol. 2, no. 3-4, pp. 157–325, 2016, ISSN: 2167-3888. DOI: 10.1561/2400000013. [Online]. Available: http://dx.doi.org/10.1561/2400000013.

[142] J. H. Gillula and C. J. Tomlin, "Guaranteed safe online learning via reachability: Tracking a ground target using a quadrotor," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 2723–2730. DOI: 10.1109/ICRA.2012.6225136.

[143] B. Sofman, E. Lin, J. A. Bagnell, J. Cole, N. Vandapel, and A. Stentz, "Improving robot navigation through self-supervised online learning," *Journal of Field Robotics*, vol. 23, no. 11-12, pp. 1059–1075, 2006.

[144] A. Koppel, J. Fink, G. Warnell, E. Stump, and A. Ribeiro, "Online learning for characterizing unknown environments in ground robotic vehicle models," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 626–633. DOI: 10.1109/IROS.2016.7759118.

[145] Supplemental Video for "Self-supervised online learning for safety-critical control using stereo vision." https://youtu.be/k-9x6i_Z7fg.

[146] M. Perrollaz, A. Spalanzani, and D. Aubert, "Probabilistic representation of the uncertainty of stereo-vision and application to obstacle detection," in *2010 IEEE Intelligent Vehicles Symposium*, IEEE, 2010, pp. 313–318.

[147] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Asian Conference on Computer Vision*, Springer, 2010, pp. 25–38.

[148] R. Szeliski and D. Scharstein, "Symmetric sub-pixel stereo matching," in *European Conference on Computer Vision*, Springer, 2002, pp. 525–540.

[149] K. Liu, K. Ok, W. Vega-Brown, and N. Roy, "Deep inference for covariance estimation: learning Gaussian noise models for state estimation," en, in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD: IEEE, May 2018, pp. 1436–1443, ISBN: 978-1-5386-3081-5. DOI: 10.1109/ICRA.2018.8461047. (visited on 04/09/2021).

[150] A. Tao, K. Sapra, and B. Catanzaro, *Hierarchical multi-scale attention for semantic segmentation*, 2020. arXiv: 2005.10821 [cs.CV].

[151] Y. Yue, J. Broder, R. Kleinberg, and T. Joachims, "The k-armed dueling bandits problem," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1538–1556, 2012.

[152] P. Shivaswamy and T. Joachims, "Online structured prediction via coactive learning," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 2012, pp. 59–66.

[153] M. Tucker, E. Novoseller, C. Kann, *et al.*, "Preference-based learning for exoskeleton gait optimization," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 2351–2357.

[154] Y. Sui, A. Gotovos, J. Burdick, and A. Krause, "Safe exploration for optimization with 'Gaussian processes," in *International Conference on Machine Learning (ICML)*, 2015, pp. 997–1005.

[155] Y. Sui, V. Zhuang, J. Burdick, and Y. Yue, "Stagewise safe Bayesian optimization with Gaussian processes," in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause, Eds., ser. Proceedings of Machine Learning Research, vol. 80, PMLR, 2018, pp. 4781–4789.

[156] M. Tucker, M. Cheng, E. Novoseller, *et al.*, "Human preference-based learning for high-dimensional optimization of exoskeleton walking gaits," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 3423–3430.

[157] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," *arXiv preprint arXiv:1606.06565*, 2016.

[158] W. Chu and Z. Ghahramani, "Preference learning with Gaussian processes," in *International Conference on Machine Learning (ICML)*, 2005, pp. 137–144.

[159] W. Chu and Z. Ghahramani, "Gaussian processes for ordinal regression.," *Journal of Machine Learning Research*, vol. 6, no. 7, 2005.

[160] K. Li, M. Tucker, E. Bıyık, *et al.*, "ROIAL: Region of interest active learning for characterizing exoskeleton gait preference landscapes," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 3212–3218.

[161] O. Chapelle and L. Li, "An empirical evaluation of Thompson sampling," *Advances in Neural Information Processing Systems*, vol. 24, pp. 2249–2257, 2011.

[162] J. Kirschner, M. Mutný, N. Hiller, R. Ischebeck, and A. Krause, "Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces," in *International Conference on Machine Learning*, PMLR, 2019, pp. 3429–3438.

[163] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with Gaussian processes," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 491–496.

[164] "IEEE standard for assumptions in safety-related models for automated driving systems," *IEEE Std 2846-2022*, 2022.

[165] D. Nistér, H.-L. Lee, J. Ng, and Y. Wang, "The safety force field," *NVIDIA White Paper*, 2019.

[166] J. Usevitch and D. Panagou, "Adversarial resilience for sampled-data systems under high-relative-degree safety constraints," *IEEE Transactions on Automatic Control*, vol. 68, no. 3, pp. 1537–1552, 2023. DOI: 10.1109/TAC.2022.3157791.

[167] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *arXiv preprint arXiv:1708.06374*, 2017.

[168] H. Caesar, V. Bankiti, A. H. Lang, *et al.*, "nuScenes: A multimodal dataset for autonomous driving," in *IEEE / CVF Computer Vision and Pattern Recognition Conference*, 2020.

[169] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proceedings of the National Academy of Sciences*, vol. 116, no. 50, pp. 24 972–24 978, 2019.

[170] B. Toghi, R. Valiente, D. Sadigh, R. Pedarsani, and Y. P. Fallah, "Cooperative autonomous vehicles that sympathize with human drivers," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4517–4524. DOI: 10.1109/IROS51168.2021.9636151.

[171] Y. Lyu, W. Luo, and J. M. Dolan, "Responsibility-associated multi-agent collision avoidance with social preferences," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 3645–3651. DOI: 10.1109/ITSC55140.2022.9922565.

[172] K. Guo, D. Wang, T. Fan, and J. Pan, "Vr-orca: Variable responsibility optimal reciprocal collision avoidance," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4520–4527, 2021. DOI: 10.1109/LRA.2021.3067851.

[173] Y. Chen, A. Singletary, and A. D. Ames, "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 127–132, 2021. DOI: 10.1109/LCSYS.2020.3000748.

[174] C. Dawson, S. Gao, and C. Fan, "Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1749–1767, 2023. DOI: 10.1109/TRO.2022.3232542.

[175] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, "Learning safe multi-agent control with decentralized neural barrier certificates," *International Conference on Learning Representations*, 2021.

[176] Y. Lyu, W. Luo, and J. M. Dolan, "Adaptive safe merging control for heterogeneous autonomous vehicles using parametric control barrier functions," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 542–547. DOI: 10.1109/IV51971.2022.9827329.

[177] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physics Review E*, vol. 51, pp. 4282–4286, 5 1995. DOI: 10.1103/PhysRevE.51.4282. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.51.4282.

[178] D. R. Scobee and S. S. Sastry, "Maximum likelihood constraint inference for inverse reinforcement learning," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=BJliakStvH.

[179] N. Aghasadeghi and T. Bretl, "Maximum entropy inverse reinforcement learning in continuous state spaces with path integrals," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 1561–1566. DOI: 10.1109/IROS.2011.6094679.

[180] D. Wilkie, J. van den Berg, and D. Manocha, "Generalized velocity obstacles," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 5573–5578. DOI: 10.1109/IROS.2009.5354175.

[181] Y. Chen, M. Jankovic, M. Santillo, and A. D. Ames, "Backup control barrier functions: Formulation and comparative study," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 6835–6841. DOI: 10.1109/CDC45484.2021.9683111.

[182] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, 2004, pp. I–I. DOI: 10.1109/CVPR.2004.1315094.

[183] P. Glotfelter, I. Buckley, and M. Egerstedt, "Hybrid nonsmooth barrier functions with applications to provably safe and composable collision avoidance for robotic systems," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1303–1310, 2019. DOI: 10.1109/LRA.2019.2895125.

[184] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[185] D. Xu, Y. Chen, B. Ivanovic, and M. Pavone, "Bits: Bi-level imitation for traffic simulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 2929–2936. DOI: 10.1109/ICRA48891.2023.10161167.

[186] D. Park, Y. Hoshi, and C. C. Kemp, "A mltimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," en, *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1544–1551, Jul. 2018, ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2018.2801475. (visited on 08/30/2023).

[187] L. Brunke, M. Greeff, A. W. Hall, *et al.*, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. Volume 5, 2022, pp. 411–444, 2022, ISSN: 2573-5144. DOI: https://doi.org/10.1146/annurev-control-042920-020211. [Online]. Available: https://www.annualreviews.org/content/journals/10.1146/annurev-control-042920-020211.

[188] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems," *Automatica*, vol. 44, no. 11, pp. 2724–2734, 2008, ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2008.03.027. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109808002677.

[189] S. Jafarpour, Z. Liu, and Y. Chen, "Probabilistic reachability analysis of stochastic control systems," *IEEE Transactions on Automatic Control*, pp. 1–15, 2025. DOI: 10.1109/TAC.2025.3566983.

[190] M. P. Chapman, R. Bonalli, K. M. Smith, I. Yang, M. Pavone, and C. J. Tomlin, "Risk-sensitive safety analysis using conditional value-at-risk," *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6521–6536, 2022. DOI: 10.1109/TAC.2021.3131149.

[191] L. Lindemann, N. Matni, and G. J. Pappas, "Stl robustness risk over discrete-time stochastic processes," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 1329–1335. DOI: 10.1109/CDC45484.2021.9683305.

[192] J. Steinhardt and R. Tedrake, "Finite-time regional verification of stochastic non-linear systems," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 901–923, 2012. DOI: 10.1177/0278364912444146.

[193] M. Black, G. Fainekos, B. Hoxha, D. Prokhorov, and D. Panagou, "Safety under uncertainty: Tight bounds with risk-aware control barrier functions," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 12 686–12 692. DOI: 10.1109/ICRA48891.2023.10161379.

[194] S. Prajna, A. Jadbabaie, and G. J. Pappas, "Stochastic safety verification using barrier certificates," in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, vol. 1, 2004, 929–934 Vol.1. DOI: 10.1109/CDC.2004.1428804.

[195] C. Santoyo, M. Dutreix, and S. Coogan, "Verification and control for finite-time safety of stochastic systems via barrier functions," in *2019 IEEE Conference on Control Technology and Applications (CCTA)*, 2019, pp. 712–717. DOI: 10.1109/CCTA.2019.8920407.

[196] F. B. Mathiesen, L. Romao, S. C. Calvert, A. Abate, and L. Laurenti, "Inner approximations of stochastic programs for data-driven stochastic barrier function design," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 3073–3080. DOI: 10.1109/CDC49753.2023.10383306.

[197] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation," in *Proceedings of Robotics: Science and Systems*, Cambridge, Massachusetts, 2017. DOI: 10.15607/RSS.2017.XIII.073.

[198] S. Liu, J. Zeng, K. Sreenath, and C. A. Belta, "Iterative ccnvex optimization for model predictive control with discrete-time high-order control barrier functions," in *2023 American Control Conference (ACC)*, 2023, pp. 3368–3375. DOI: 10.23919/ACC55779.2023.10156532.

[199] A. G. Wills and W. P. Heath, "Barrier function based model predictive control," *Automatica*, vol. 40, no. 8, pp. 1415–1422, 2004.

[200] M. Ahmadi, A. Singletary, J. W. Burdick, and A. D. Ames, "Safe policy synthesis in multi-agent pomdps via discrete-time barrier functions," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, 2019, pp. 4797–4803.

[201] J. Ville, "Etude critique de la notion de collectif," 1939.

[202] D. A. Freedman, "On tail probabilities for martingales," *The Annals of Probability*, vol. 3, no. 1, pp. 100–118, 1975, ISSN: 00911798, 2168894X. [Online]. Available: http://www.jstor.org/stable/2959268 (visited on 11/05/2024).

[203] A. J. Taylor, V. D. Dorobantu, R. K. Cosner, Y. Yue, and A. D. Ames, "Safety of sampled-data systems with control barrier functions via approximate discrete time models," *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 7127–7134, 2022. DOI: 10.1109/CDC51059.2022.9993226,

[204] P. Culbertson, R. K. Cosner, and A. D. Ames, "Input-to-state stability in probability," *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 5796–5803, 2023. DOI: 10.1109/CDC49753.2023.10383579,

[205] M. Ahmadi, A. Singletary, J. W. Burdick, and A. D. Ames, "Barrier functions for multiagent-POMDPs with DTL specifications," in *2020 59th IEEE Conference on Decision and Control (CDC)*, ISSN: 2576-2370, Dec. 2020, pp. 1380–1385. DOI: 10.1109/CDC42340.2020.9304266.

[206] G. Grimmett and D. Stirzaker, *Probability and random processes*. Oxford university press, 2020.

[207] R. K. Cosner, P. Culbertson, and A. D. Ames, *Bounding stochastic safety: Leveraging freedman's inequality with discrete-time control barrier functions*, 2024. arXiv: 2403.05745 [eess.SY]. [Online]. Available: https://arxiv.org/abs/2403.05745.

[208] J. G. Liao and A. Berg, "Sharpening jensen's inequality," *The American Statistician*, vol. 73, no. 3, pp. 278–281, 2019. DOI: 10.1080/00031305.2017.1419145.

[209] R. A. Becker, "The variance drain and jensen's inequality," *CAEPR Working Paper*, no. 2012-004, 2012.

[210] Supplemental Video for "Robust safety under stochastic uncertainty with discrete-time control barrier functions." https://vimeo.com/829156256?share=copy.

[211] X. Fan, I. Grama, and Q. Liu, "Hoeffding's inequality for supermartingales," *Stochastic Processes and their Applications*, vol. 122, no. 10, pp. 3545–3559, 2012, ISSN: 0304-4149. DOI: https://doi.org/10.1016/j.spa.2012.06.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304414912001378.

[212] Code, *Repository for this work:*, https://github.com/rkcosner/freedman.git.

[213] X. Xiong and A. Ames, "3-d underactuated bipedal walking via h-lip based gait synthesis and stepping stabilization," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2405–2425, 2022. DOI: 10.1109/TRO.2022.3150219.

[214] S. X. Wei, A. Dixit, S. Tomar, and J. W. Burdick, "Moving obstacle avoidance: A data-driven risk-aware approach," *IEEE Control Systems Letters*, vol. 7, pp. 289–294, 2023. DOI: 10.1109/LCSYS.2022.3181191.

[215] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, Eds., ser. Proceedings of Machine Learning Research, vol. 37, Lille, France: PMLR, 2015, pp. 1530–1538.

[216] D. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, Curran Associates, Inc., 2021, pp. 21 696–21 707.

[217] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, 2022. arXiv: 1312.6114 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1312.6114.

[218] A. Ramesh, M. Pavlov, G. Goh, *et al.*, "Zero-shot text-to-image generation," in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 8821–8831.

[219] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[220] X. Huang, Z. Li, Y. Xiang, *et al.*, "Creating a dynamic quadrupedal robotic goalkeeper with reinforcement learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 2715–2722. DOI: 10.1109/IROS55552.2023.10341936.

[221] C. Chi, S. Feng, Y. Du, *et al.*, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, 2023. DOI: 10.15607/RSS.2023.XIX.026.

[222] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters, "Motion planning diffusion: Learning and planning of robot motions with diffusion models," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 1916–1923. DOI: 10.1109/IROS55552.2023.10342382.

[223] H. Ren and P. Ben-Tzvi, "Learning inverse kinematics and dynamics of a robotic manipulator using generative adversarial networks," en, *Robotics and Autonomous Systems*, vol. 124, p. 103 386, Feb. 2020, ISSN: 09218890. DOI: 10.1016/j.robot.2019.103386. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0921889019303501 (visited on 08/30/2023).

[224] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28, Curran Associates, Inc., 2015.

[225] B. Ivanovic, K. Leung, E. Schmerling, and M. Pavone, "Multimodal deep generative models for trajectory prediction: A conditional variational autoencoder approach," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 295–302, Apr. 2021, Conference Name: IEEE Robotics and Automation Letters, ISSN: 2377-3766. DOI: 10.1109/LRA.2020.3043163.

[226] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, 2023. DOI: 10.15607/RSS.2023.XIX.016.

[227] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter, "Perceptive locomotion through nonlinear model-predictive control," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3402–3421, 2023.

[228] U. Rosolia and F. Borrelli, "Learning how to autonomously race a car: A predictive control approach," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2713–2719, 2019.

[229] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, p. 109 597, 2021, ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2021.109597.

[230] H. Ma, X. Zhang, S. E. Li, Z. Lin, Y. Lyu, and S. Zheng, "Feasibility enhancement of constrained receding horizon control using generalized control barrier function," in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, IEEE, 2021, pp. 551–557.

[231] Supplemental Video for "Generative modeling of residuals for real-time risk-sensitive safety with discrete-time control barrier functions." https://youtu.be/QH6rO1KTXds.

[232] S. V. Rakovic and D. Q. Mayne, "Set robust control invariance for linear discrete time systems," in *Proceedings of the 44th IEEE Conference on Decision and Control*, IEEE, 2005, pp. 975–980.

[233] J. Lam, Z. Shu, S. Xu, and E.-K. Boukas, "Robust control of descriptor discrete-time markovian jump systems," *International Journal of Control*, vol. 80, no. 3, pp. 374–385, 2007.

[234] L. Xie, C. E. De Souza, and Y. Wang, "Robust control of discrete time uncertain dynamical systems," *Automatica*, vol. 29, no. 4, pp. 1133–1137, 1993.

[235] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019, ISSN: 1935-8237. DOI: 10.1561/2200000056. [Online]. Available: http://dx.doi.org/10.1561/2200000056.

[236] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 6840–6851.

[237] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 2520–2525.

[238] R. K. Cosner, I. Sadalski, J. K. Woo, P. Culbertson, and A. D. Ames, *Generative modeling of residuals for real-time risk-sensitive safety with discrete-time control barrier functions*, 2023. arXiv: 2311.05802 [eess.SY]. [Online]. Available: https://arxiv.org/abs/2311.05802.

[239] M. O'Connell, G. Shi, X. Shi, *et al.*, "Neural-fly enables rapid learning for agile flight in strong winds," en, *Science Robotics*, vol. 7, no. 66, May 2022, arXiv:2205.06908 [cs, eess], ISSN: 2470-9476. DOI: 10.1126/scirobotics.abm6597. [Online]. Available: http://arxiv.org/abs/2205.06908 (visited on 09/06/2023).

[240] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," *Science Robotics*, vol. 9, no. 89, eadi9579, 2024.

[241] Supplemental Video for "SHIELD: Safety on humanoids via CBFs in expectation on learned dynamics." https://youtu.be/dJj4GBtH6Gw.

[242] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, 2020.

[243] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, *Flow matching for generative modeling*, 2023. arXiv: 2210.02747 [cs.LG].

[244] E. Garone, S. Di Cairano, and I. Kolmanovsky, "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, vol. 75, pp. 306–328, 2017.

[245] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China: IEEE, 2011.

[246] U. Robotics, *Unitree sdk2*, Accessed: 2025-03-01, 2024. [Online]. Available: https://github.com/unitreerobotics/unitree_sdk2.

[247] M. Mittal, C. Yu, Q. Yu, *et al.*, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023. DOI: 10.1109/LRA.2023.3270034.

[248] X. Gu, Y.-J. Wang, X. Zhu, *et al.*, "Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning," in *Robotics: Science and Systems*, 2024.

[249] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968. DOI: 10.1109/tssc.1968.300136.

[250] Supplemental Video for "Unified MPC+CBF control for performant safety: mutual benefits and inherent robustness properties." https://youtu.be/VvBiguw0RMo.

[251] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 30–44, 2016.

[252] M. Cannon, B. Kouvaritakis, S. V. Raković, and Q. Cheng, "Stochastic tubes in model predictive control with probabilistic constraints," *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 194–200, 2011. DOI: 10.1109/TAC.2010.2086553.

[253] B. Kouvaritakis, M. Cannon, S. V. Raković, and Q. Cheng, "Explicit use of probabilistic distributions in linear predictive control," *Automatica*, vol. 46, no. 10, pp. 1719–1724, 2010, ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2010.06.034.

[254] S. X. Wei, *Data-Driven Safety-Critical Autonomy in Unknown, Unstructured, and Dynamic Environments*. California Institute of Technology, 2024.

[255] N. J. Sanket, C. M. Parameshwara, C. D. Singh, *et al.*, "EVDodgeNet: deep dynamic obstacle dodging with event cameras," en, *arXiv:1906.02919 [cs]*, Mar. 2020, arXiv: 1906.02919. [Online]. Available: http://arxiv.org/abs/1906.02919 (visited on 01/24/2022).

[256] D. Falanga, K. Kleber, and D. Scaramuzza, "Dynamic obstacle avoidance for quadrotors with event cameras," en, *Science Robotics*, vol. 5, no. 40, Mar. 2020, ISSN: 2470-9476. DOI: 10.1126/scirobotics.aaz9712. [Online]. Available: https://www.science.org/doi/10.1126/scirobotics.aaz9712 (visited on 01/24/2022).

[257] B. Lindqvist, S. S. Mansouri, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "Nonlinear MPC for collision avoidance and control of UAVs with dynamic obstacles," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6001–6008, Oct. 2020, Conference Name: IEEE Robotics and Automation Letters, ISSN: 2377-3766. DOI: 10.1109/LRA.2020.3010730.

[258] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[259] Y. Xiong, C. Zhou, X. Xiang, *et al.*, "Efficient track anything," *arXiv preprint arXiv:2411.18933*, 2024.

[260] N. Ravi, V. Gabeur, Y.-T. Hu, *et al.*, *Sam 2: Segment anything in images and videos*, 2024. arXiv: 2408.00714 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2408.00714.

[261] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis*, Springer Berlin Heidelberg, 2003, pp. 363–370.

[262] J. Schilliger, T. Lew, S. M. Richards, S. Hänggi, M. Pavone, and C. Onder, "Control barrier functions for cyber-physical systems and applications to nmpc," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8623–8630, 2021.

[263] S. Thrun, "Probabilistic robotics," en, *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, Mar. 2002, ISSN: 0001-0782, 1557-7317. DOI: 10.1145/504729.504754. (visited on 11/25/2022).

[264] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, "From linear to nonlinear mpc: Bridging the gap via the real-time iteration," *International Journal of Control*, vol. 93, no. 1, pp. 62–80, 2020.

[265] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *Journal of Risk*, vol. 2, pp. 21–42, 2000.

[266] L. Lindemann, M. Cleaveland, G. Shim, and G. J. Pappas, "Safe planning in dynamic environments using conformal prediction," *IEEE Robotics and Automation Letters*, 2023.

[267] A. Dixit, L. Lindemann, S. X. Wei, M. Cleaveland, G. J. Pappas, and J. W. Burdick, "Adaptive conformal prediction for motion planning among dynamic agents," in *Learning for Dynamics and Control Conference*, PMLR, 2023, pp. 300–314.

[268] P. Akella, A. Dixit, M. Ahmadi, J. W. Burdick, and A. D. Ames, "Sample-based bounds for coherent risk measures: Applications to policy synthesis and verification," *Artificial Intelligence*, vol. 336, p. 104 195, 2024.

[269] T. He, J. Gao, W. Xiao, *et al.*, "Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills," *arXiv preprint arXiv:2502.01143*, 2025.

[270] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 96–101, 2018.

[271] K. Nakamura, L. Peters, and A. Bajcsy, "Generalizing safety beyond collision-avoidance via latent-space reachability analysis," *arXiv preprint arXiv: 2502.00935*, 2025.