

# Contraction Theory for Robust Learning-based Control: Toward Aerospace and Robotic Autonomy

Thesis by  
Hiroyasu Tsukamoto

In Partial Fulfillment of the Requirements for the  
Degree of  
Doctor of Philosophy

The logo for the California Institute of Technology (Caltech), featuring the word "Caltech" in a bold, orange, sans-serif font.

CALIFORNIA INSTITUTE OF TECHNOLOGY  
Pasadena, California

2023  
Defended May 15, 2023

© 2023

Hiroyasu Tsukamoto  
ORCID: 0000-0002-6337-2667

All rights reserved except where otherwise noted

## ACKNOWLEDGEMENTS

First and foremost, I am extremely grateful to my supervisor, Dr. Soon-Jo Chung, for his guidance and advice during my Ph.D. His immense knowledge, passion, and experience in our field of study have continuously motivated me in all our research projects and inspired me to start my career as an assistant professor. I have truly enjoyed my time at Caltech thanks to his significant assistance at every stage of my Ph.D. Besides my advisor, I greatly appreciate the committee members of my Ph.D. defense, Dr. John Doyle, Dr. Richard Murray, Dr. Sergio Pellegrino, and Dr. Michael Watkins (in alphabetical order) for their invaluable support in further refining my Ph.D. work. My sincere thanks also go to Dr. Jean-Jacques Slotine, who provided a wonderful opportunity for research collaboration starting in 2020 even during Covid. It would not be possible to build the basis of our research field without his invaluable insights, his theoretical establishment of contraction analysis, or his significant contributions to systems and control theory.

I have a great many other people I should be thankful to, although I can only partially list them as follows. I would like to thank my mentors, Dr. Fred Hadaegh, Dr. Chuchu Fan, and Dr. Guanya Shi, and again Dr. Soon-Jo Chung and Dr. Jean-Jacques Slotine, for their invaluable academic insights, and for their irreplaceable assistance, advice, and strong recommendation letters when I was on the academic job market.

I would like to thank all my collaborators at NASA Jet Propulsion Laboratory (JPL), Dr. Amir Rahmani, Dr. Michel Ingham, Dr. Yashwanth Nakka, Dr. Changrak Choi, Dr. Julie Castillo-Rogez, Mr. Benjamin Donitz, Mr. Declan Mages, Dr. Damon Landau, Mr. Peter Dixon, and Dr. Stefano Campagnola, for their valuable inputs and feedback during our research collaboration.

I would like to thank my undergraduate advisors at Kyoto University, Dr. Kei Senda, Dr. Hiroshi Yamakawa, and Dr. Ralf Klessen, for their thoughtful and insightful guidance during the early stage of my academic career and for their considerable support in making it possible to pursue Ph.D. at Caltech.

I would like to thank the GALCIT and CMS departments at Caltech for providing an extremely supportive and welcoming learning environment, where I could truly enjoy and focus on my research with access to world-class courses, faculty members, and research facilities. I would especially like to thank Dr. Kostia Zuev,

Dr. Dale I. Pullin, Dr. Tim Colonius, Dr. Nadia Lapusta, Dr. Michael Ortiz, Dr. Guruswami Ravichandran, Dr. Scott Ploen, Dr. Stefano Campagnola, Dr. Martin Lo, Dr. Daniel Scharf, Dr. Andrew Klesh, Dr. Sergio Pellegrino, Dr. Soon-Jo Chung, Dr. Venkat Chandrasekaran, Dr. Richard Murray, Dr. Aaron Ames, Dr. Andrew Stuart, Dr. Joel Tropp, and Dr. Yisong Yue for their inspiring and excellent lectures, which significantly deepened the knowledge and understanding of my research field.

I would like to thank the funding agencies, NASA JPL, Raytheon, DARPA, and Caltech CAST, for financially supporting my research. I also would like to thank Funai Foundation and Honjo Foundation not just for their continuous financial support but also for the great opportunities for international networking.

I would like to thank all my labmates and GALCIT friends, Benjamin Rivière, Michael O'Connell, Kai Matsuka, Guanya Shi, SooJean Han, Ellande Tang, Yashwanth Nakka, Lu Gan, Sorina Lupu, Jimmy Ragan, Nikhil Ranganathan, Connor Lee, John Lathrop, Joshua Cho, Fengze Xie, Yuta Takahashi, Tommaso Ravanelli, Matt Anderson, James Preiss, Karena Cai, Xichen Shi, Kyunam Kim, Wolfgang Hoenig, Patrick Spieler, Rebecca Foust, Akshay Joshi, Alexander Wen, Avinash Chandra, Conar Martin, Emile Oshima, Harsha Narravula, Jean-Sebastian Spratt, Kanthasamy Ubamanyu, Kira Headrick, Lucas Benoit-Marechal, Mouâdh Bouayad, Shen Naijian Eric, and Wesley Yu, for making my life at Caltech memorable. My special thanks go to Yashwanth Nakka, Benjamin Rivière, Jimmy Ragan, and Sorina Lupu, who provided immense help in our research collaboration and in conducting experimental validation of my research projects, and to Lu Gan, who is now also a professor and who was there to help me during our stressful faculty search process together. I also would like to thank Dr. Yingying Li, Dr. Tomomi Karigo, Dr. Kenshiro Oguri, and Dr. Yorie Nakahira for their great help during my faculty search.

I would like to thank all my friends in the US and in Japan for being there for me all the time during my Ph.D. I cannot list all the names here unfortunately, but having a place where I can be myself and have fun without worrying too much about my work has helped me a lot, especially during the toughest times. I respect them in many ways and truly appreciate them always being by my side, and being an inspiration to me and to my research.

I would like to express my deepest gratitude to my family, Shinya, Naomi, Yumi, Yasuo, Chieko, Eiju, and Takako, especially to my parents, Shinya and Naomi, for their continuous support, guidance, and love from afar. I will never be able to thank

them enough for being there for me without a doubt. I am also indebted to Satoshi, Hitomi, and Risako for inspiring me to study abroad and pursue my dream in the United States. Finally, words cannot express my gratitude to my dearest friend, Saki, for always making me laugh, for believing in me more than I do in myself, and for making my Ph.D. journey wonderful.

## ABSTRACT

Machine learning and AI have been used for achieving autonomy in various aerospace and robotic systems. In next-generation research tasks, which could involve highly nonlinear, complicated, and large-scale decision-making problems in safety-critical situations, however, the existing performance guarantees of black-box AI approaches may not be sufficiently powerful. This thesis gives a mathematical overview of contraction theory, with some practical examples drawn from joint projects with NASA JPL, for enjoying formal guarantees of nonlinear control theory even with the use of machine learning-based and data-driven methods. This is not to argue that these methods are always better than conventional approaches, but to provide formal tools to investigate their performance for further discussion, so we can design and operate truly autonomous aerospace and robotic systems safely, robustly, adaptively, and intelligently in real-time.

Contraction theory is an analytical tool to study differential dynamics of a non-autonomous (i.e., time-varying) nonlinear system under a contraction metric defined with a uniformly positive definite matrix, the existence of which results in a necessary and sufficient characterization of incremental exponential stability of multiple solution trajectories with respect to each other. Its nonlinear stability analysis boils down to finding a suitable contraction metric that satisfies a stability condition expressed as a linear matrix inequality, resulting in many parallels drawn between linear systems theory and contraction theory for nonlinear systems. This yields much-needed safety and stability guarantees for neural network-based control and estimation schemes, without resorting to a more involved method of using uniform asymptotic stability for input-to-state stability. Such distinctive features permit the systematic construction of a contraction metric via convex optimization, thereby obtaining an explicit exponential bound on the distance between a time-varying target trajectory and solution trajectories perturbed externally due to disturbances and learning errors. The first two parts of this thesis are about a theoretical overview of contraction theory and its advantages, with an emphasis on deriving formal robustness and stability guarantees for deep learning-based 1) feedback control, 2) state estimation, 3) motion planning, 4) multi-agent collision avoidance and robust tracking augmentation, 5) adaptive control, 6) neural net-based system identification and control, for nonlinear systems perturbed externally by deterministic and stochastic disturbances. In particular, we provide a detailed review of techniques for finding

contraction metrics and associated control and estimation laws using deep neural networks.

In the third part of the thesis, we present several numerical simulations and empirical validation of our proposed approaches to assess the impact of our findings on realizing aerospace and robotic autonomy. We mainly focus on the two joint projects with NASA JPL: 1) Science-Infused Spacecraft Autonomy for Interstellar Object Exploration and 2) Constellation Autonomous Space Technology Demonstration of Orbital Reconfiguration (CASTOR), where we also perform hardware demonstrations of our methods using our thruster-based spacecraft simulators (M-STAR) and in high-conflict, distributed, intelligent UAV swarm reconfiguration with up to 20 UAVs (crazyflies).

## PUBLISHED CONTENT AND CONTRIBUTIONS

- [1] **H. Tsukamoto**, B. Rivière, C. Choi, A. Rahmani, and S.-J. Chung, “CART: Collision avoidance and robust tracking augmentation in learning-based motion planning for multi-agent systems,” in *IEEE Conference on Decision and Control*, under review, 2023,  
H.T. participated in the conception of the project, proposed the framework, formulated the theoretical proofs, implemented and analyzed the algorithm, and wrote the manuscript.
- [2] B. P. S. Donitz, D. Mages, **H. Tsukamoto**, P. Dixon, D. Landau, S.-J. Chung, E. Bufanda, M. Ingham, and J. Castillo-Rogez, “Interstellar object accessibility and mission design,” in *IEEE Aerospace Conference*, 2022,  
H.T. participated in the conception of the project and the writing of the manuscript.
- [3] **H. Tsukamoto**, S.-J. Chung, B. Donitz, M. Ingham, D. Mages, and Y. K. Nakka, “Neural-Rendezvous: Learning-based robust guidance and control to encounter interstellar objects,” *AIAA Journal of Guidance, Control, and Dynamics*, under revision, 2022,  
H.T. participated in the conception of the project, proposed the framework, formulated the theoretical proofs, implemented and analyzed the algorithm, and wrote the manuscript.
- [4] **H. Tsukamoto** and S.-J. Chung, “Neural contraction metrics for robust estimation and control: A convex optimization approach,” *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 211–216, 2021. doi: 10.1109/LCSYS.2020.3001646,  
H.T. proposed the framework, formulated the theoretical proofs, implemented and analyzed the algorithm, and wrote the manuscript.
- [5] **H. Tsukamoto** and S.-J. Chung, “Robust controller design for stochastic nonlinear systems via convex optimization,” *IEEE Transactions on Automatic Control*, vol. 66, no. 10, pp. 4731–4746, 2021. doi: 10.1109/TAC.2020.3038402,  
H.T. proposed the framework, formulated the theoretical proofs, implemented and analyzed the algorithm, and wrote the manuscript.
- [6] **H. Tsukamoto** and S.-J. Chung, “Learning-based robust motion planning with guaranteed stability: A contraction theory approach,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6164–6171, 2021. doi: 10.1109/LRA.2021.3091019,  
H.T. proposed the framework, formulated the theoretical proofs, implemented and analyzed the algorithm, and wrote the manuscript.
- [7] **H. Tsukamoto**, S.-J. Chung, and J.-J. E. Slotine, “Neural stochastic contraction metrics for learning-based control and estimation,” *IEEE Control*

*Systems Letters*, vol. 5, no. 5, pp. 1825–1830, 2021. doi: 10.1109/LCSYS.2020.3046529,

H.T. proposed the framework, formulated the theoretical proofs, implemented and analyzed the algorithm, and wrote the manuscript.

- [8] **H. Tsukamoto**, S.-J. Chung, and J.-J. E. Slotine, “Learning-based adaptive control using contraction theory,” in *IEEE Conference on Decision and Control*, 2021, pp. 2533–2538. doi: 10.1109/CDC45484.2021.9683435, H.T. proposed the framework, formulated the theoretical proofs, implemented and analyzed the algorithm, and wrote the manuscript.
- [9] **H. Tsukamoto**, S.-J. Chung, and J.-J. E. Slotine, “Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview,” *Annual Reviews in Control*, vol. 52, pp. 135–169, 2021, issn: 1367-5788. doi: 10.1016/j.arcontrol.2021.10.001, H.T. participated in the conception of the project, the formulation of the theoretical proofs, and the writing of the manuscript.
- [10] **H. Tsukamoto**, S.-J. Chung, J.-J. E. Slotine, and C. Fan, “A theoretical overview of neural contraction metrics for learning-based control with guaranteed stability,” in *IEEE Conference on Decision and Control*, 2021, pp. 2949–2954. doi: 10.1109/CDC45484.2021.9682859, H.T. participated in the conception of the project, proposed the framework, derived the theoretical proofs, and wrote the manuscript.
- [11] S. Singh, **H. Tsukamoto**, B. T. Lopez, S.-J. Chung, and J.-J. E. Slotine, “Safe motion planning with tubes and contraction metrics,” in *IEEE Conference on Decision and Control*, 2021, pp. 2943–2948. doi: 10.1109/CDC45484.2021.9682865, H.T. participated in the conception of the project and the writing of the manuscript.
- [12] **H. Tsukamoto** and S.-J. Chung, “Convex optimization-based controller design for stochastic nonlinear systems using contraction analysis,” in *IEEE Conference on Decision and Control*, 2019, pp. 8196–8203. doi: 10.1109/CDC40024.2019.9028942, H.T. proposed the framework, formulated the theoretical proofs, implemented and analyzed the algorithm, and wrote the manuscript.

## TABLE OF CONTENTS

Acknowledgements . . . . .	iii
Abstract . . . . .	vi
Published Content and Contributions . . . . .	viii
Table of Contents . . . . .	ix
List of Illustrations . . . . .	xii
List of Tables . . . . .	xvii
Nomenclature . . . . .	xviii
Chapter 1: Introduction . . . . .	1
1.1 Thesis Organization . . . . .	3
1.2 Related Work . . . . .	5
1.3 Notation . . . . .	14
Chapter 2: Contraction Theory . . . . .	27
2.1 Fundamentals . . . . .	27
2.2 Path-Length Integral and Robust Incremental Stability Analysis . . . . .	34
2.3 Finite-Gain Stability and Contraction of Hierarchically Combined Systems . . . . .	40
2.4 Contraction Theory for Discrete-time Systems . . . . .	42
Chapter 3: Robust Nonlinear Control and Estimation via Contraction Theory . . . . .	50
3.1 Overview of Nonlinear Control and Estimation . . . . .	52
3.2 LMI Conditions for Contraction Metrics . . . . .	56
3.3 Bounded Real Lemma in Contraction Theory . . . . .	61
3.4 KYP Lemma in Contraction Theory . . . . .	62
Chapter 4: Convex Optimality in Robust Nonlinear Control and Estimation via Contraction Theory . . . . .	67
4.1 CV-STEM Control . . . . .	67
4.2 CV-STEM Estimation . . . . .	69
4.3 Control Contraction Metrics (CCMs) . . . . .	74
4.4 Remarks in CV-STEM Implementation . . . . .	76
Chapter 5: Contraction Theory for Learning-based Control . . . . .	81
5.1 Problem Formulation . . . . .	81
5.2 Formal Stability Guarantees via Contraction Theory . . . . .	84
Chapter 6: Learning-based Safe and Robust Control and Estimation . . . . .	92
6.1 Stability of NCM-based Control and Estimation . . . . .	94
6.2 NCMs as Lyapunov Functions . . . . .	98
6.3 Remarks in NCM Implementation . . . . .	102
Chapter 7: Learning-based Safe and Robust Motion Planning . . . . .	109
7.1 Overview of Existing Motion Planners . . . . .	110
7.2 Stability Guarantees of LAG-ROS . . . . .	110
7.3 Tube-based State Constraint Satisfaction . . . . .	113

7.4	Safe Exploration and Learning of Disturbances . . . . .	116
7.5	Collision Avoidance and Robust Tracking Augmentation . . . . .	116
7.6	Theoretical Overview of CART . . . . .	122
Chapter 8: Learning-based Adaptive Control and Contraction Theory for Learned Models . . . . .		
	Learned Models . . . . .	134
8.1	Learning-based Adaptive Control . . . . .	134
8.2	Contraction Theory for Learned Models . . . . .	143
Chapter 9: Interstellar Object Exploration . . . . .		
9.1	Technical Challenges in ISO Exploration . . . . .	156
9.2	Autonomous Guidance via Dynamical System-Based Deep Learning	158
9.3	Deep learning-Based Optimal Tracking Control . . . . .	165
9.4	Neural-Rendezvous: Learning-based Robust Guidance and Control to Encounter ISOs . . . . .	172
9.5	Extensions . . . . .	181
Chapter 10: Additional Numerical Simulations and Empirical Validation . . . . .		
10.1	Numerical Simulations for ISO Exploration . . . . .	190
10.2	Empirical Validation for ISO Exploration . . . . .	198
10.3	Numerical Simulations for CASTOR . . . . .	204
Chapter 11: Concluding Remarks . . . . .		
11.1	Summary of Presented Methods . . . . .	213
11.2	Future Plans . . . . .	215

## LIST OF ILLUSTRATIONS

<i>Number</i>	<i>Page</i>
1.1 Overview of our learning-based control with formal guarantees (in the context of nonlinear control theory). . . . .	4
1.2 Interstellar objects. Above: Artist’s illustration II/‘Oumuamua (credit: NASA, ESA, and STScI). Below: 2I/Borisov near and at perihelion (credit: NASA, ESA, and D. Jewitt (UCLA)). Click the picture to watch our YouTube video ( <a href="https://youtu.be/8h60B_p1fyQ">https://youtu.be/8h60B_p1fyQ</a> ). . . . .	12
1.3 Illustration of cruise and terminal GNC ( $t$ : time; $\alpha(t)$ : ISO state; $x(t)$ : spacecraft state relative to $\alpha(t)$ ; $y(t)$ : state measurement of the ISO and spacecraft; $u$ : control input; $\hat{\alpha}(t)$ : estimated ISO state; and $\hat{x}(t)$ : estimated spacecraft relative state). Neural-Rendezvous enables obtaining a verifiable delivery error bound even under the large ISO state uncertainty and high-velocity challenges. . . . .	13
1.4 Visualizations of our numerical and experimental validation to be discussed in Chapter 10.2. . . . .	14
2.1 Illustration of contraction theory, where $V$ is a differential Lyapunov function $V = \delta x^\top M(x, t)\delta x$ , $\delta z = \Theta(x, t)\delta x$ , and $M(x, t) = \Theta(x, t)\Theta(x, t)^\top > 0$ defines a contraction metric (see Theorem 2.1). . . . .	28
4.1 Block diagram of CV-STEM. . . . .	69
6.1 Illustration of NCM ( $x$ : system state; $M$ : positive definite matrix that defines optimal contraction metric; $x_i$ and $M_i$ : sampled $x$ and $M$ ; $\hat{x}$ : estimated system state; $y$ : measurement; and $u$ : system control input). Note that the target trajectory $(x_d, u_d)$ is omitted in the figure for simplicity. . . . .	94
6.2 Lorenz oscillator state estimation error smoothed using a 15-point moving average filter in Example 6.1 (left), and spacecraft positions $(p_x, p_y)$ on a planar field in Example 6.2 (right). . . . .	99
6.3 Rocket model (angle of attack $\varphi$ , pitch rate $q$ ). . . . .	104
6.4 State estimation and tracking errors in Example 6.5 ( $x = [\varphi, q]^T$ ). . . . .	106
7.1 Block diagram of tube-based planning using contraction theory. . . . .	110

7.2	Cart-pole balancing: $x = [p, \theta, \dot{p}, \dot{\theta}]^\top$ , $x$ and $x_d$ are given in (6.1) and (6.3), (a) – (c) are given in Table 7.1, and $r_\ell$ is given in (7.4). The shaded area denotes the standard deviation ( $+1\sigma$ and $-0.5\sigma$ ). . . . .	112
7.3	Illustration of state sampling in a robust bounded error tube. . . . .	114
7.4	Block diagram of state sampling in a robust bounded error tube. . . . .	115
7.5	Trajectories for the learning-based planner (a), robust tube-based planner (b), LAG-ROS (c), and offline centralized solution (d) (○: start, ●: goal). . . . .	116
7.6	Detailed block diagram of machine learning-based control using contraction theory, including LAG-ROS, where Fig. 4.1, Fig. 7.1, and Fig. 7.4 are utilized as building blocks. . . . .	117
7.7	Conceptual illustration of CART, showing the hierarchical combination of our safety filter and robust filter, where $\alpha_h, \alpha_V > 0$ , $h \geq 0$ is a safety function given, $S$ is a safe set given, $S_\varepsilon$ is some fictitious set containing learned trajectory $x_\ell$ with learning error $\varepsilon > 0$ , $x_d$ is a safe trajectory, $x_r$ is a reference trajectory given by global motion planner (7.8), $x$ is the actual state trajectory subject to disturbance, and $V$ is an incremental Lyapunov function for safe and robust trajectory tracking. Note that we use a log-barrier formulation in CART for its distributed implementation and analytical solution, but this figure uses $h$ for the simplicity of our concept description. . . . .	118
7.8	Different sources of robustness against external disturbance, where our safety filter is robust due to stability of a safe set and our robust filter is robust due to incremental stability of the closed-loop system with respect to a target trajectory. . . . .	119
8.1	Illustration of aNCM ( $M$ : positive definite matrix that defines aNCM; $\hat{\theta}$ : estimated parameter; $Y$ : error signal, see (8.10); $x$ : system state; $u$ : system control input; and $(x_d, u_d)$ : target state and control input trajectory). . . . .	140
9.1	Interstellar objects. Above: Artist’s illustration 1I/‘Oumuamua (credit: NASA, ESA, and STScI). Below: 2I/Borisov near and at perihelion (credit: NASA, ESA, and D. Jewitt (UCLA)). Click the picture to watch our YouTube summary of the ISO exploration project ( <a href="https://youtu.be/8h60B_p1fyQ">https://youtu.be/8h60B_p1fyQ</a> ). . . . .	152

9.2	Illustration of cruise and terminal GNC ( $t$ : time; $\alpha(t)$ : ISO state; $x(t)$ : spacecraft state relative to $\alpha(t)$ ; $y(t)$ : state measurement of the ISO and spacecraft; $u$ : control input; $\hat{\alpha}(t)$ : estimated ISO state; and $\hat{x}(t)$ : estimated spacecraft relative state). Neural-Rendezvous enables obtaining a verifiable delivery error bound even under the large ISO state uncertainty and high-velocity challenges. . . . .	154
9.3	Neural net loss functions. Left: SN-DNN trained to imitate a desired control input (first term of (9.8)); right: SN-DNN trained to imitate a desired state trajectory (second term of (9.8)). . . . .	161
9.4	Illustration of the optimality gap (9.11) in Lemma 9.1, where $\mathcal{S}_{\text{train}}$ denotes the training set in which training data $(\bar{x}_i, \bar{\alpha}_i, \bar{t}_i, \bar{\rho}_i) \in \Pi_{\text{train}}$ is sampled. . . . .	164
9.5	Illustration of the desired trajectory discussed in Sec. 9.3.I, where $v_f = C_{s2v}\varphi_\ell^{t_f}(\hat{x}(t_d), \hat{\alpha}(t_d), t_d)$ as in (9.13). Spacecraft obtains the desired trajectory $x_d$ of (9.14) as a result of backward integration (see Remark 9.6). . . . .	167
9.6	Illustration of the expected position tracking error bound (9.27) of Theorem 9.1, where $\zeta^t(Z_s, t_s)$ is assumed to be a non-increasing function in time. . . . .	178
9.7	Mission timeline for encountering an ISO, where $y(t)$ is the state measurement as in Fig. 9.2 and the other notation follows that of Lemma 9.1 and Theorem 9.1. Terminal G&C (Neural-Rendezvous) is performed using the SN-DNN guidance and min-norm control policies online as in Algorithm 1, thereby enabling fast response autonomous operations under the large ISO state uncertainty and high-velocity challenges. Note that the SN-DNN is trained offline. . .	179
10.1	Control performances versus weight ratio of the SN-DNN loss. The shaded area denotes the standard deviations ( $\pm 2.5 \times 10^{-1}\sigma$ for the spacecraft delivery error and $\pm 5 \times 10^{-2}\sigma$ for the control effort). . . .	193
10.2	Control performances versus ISOs in the test set (400-499), where $\bar{e}_f$ is the right-hand side of bound (9.27) in Theorem 9.1, the admissible delta-V (2-norm S/C velocity increase during the terminal guidance) is assumed to be 0.6 km/s, and each result shown above is the average of 10 simulations performed for each ISO. . . . .	194

10.3	Control performances versus control time interval, where each result shown above is the average of 10 simulations performed for all 100 ISOs in the test set (400-499). The shaded area denotes the standard deviations ( $\pm 1\sigma$ both for the spacecraft delivery error and control effort). . . . .	195
10.4	Control performances versus ISOs in the test set (400-499). Each result shown above is the average of 10 simulations performed for each ISO. . . . .	196
10.5	Performance comparison between our learning-based approaches and MPC. Each result shown above is the average of all ISOs in Fig. 10.2 and Fig. 10.4. . . . .	196
10.6	Visualized Neural-Rendezvous trajectories for ISO exploration, where yellow curves represent ISO trajectories and blue curves represent spacecraft trajectories. More details can be found <a href="#">here</a> . . . . .	197
10.7	Robotic platforms. Left: Multi-Spacecraft Testbed for Autonomy Research (M-STAR). Right: Crazyflie, a versatile open-source flying development platform that only weighs 27 g. . . . .	199
10.8	Demonstrated Neural-Rendezvous trajectories of M-STAR, where the shaded yellow circle indicates the expected delivery error bound (the left figure is a magnified view of the right figure). . . . .	201
10.9	Demonstrated Neural-Rendezvous trajectories of M-STAR observed in the actual environment, illustrated using their light trails. This movie can be found <a href="#">here</a> . . . . .	202
10.10	Crazyflie nominal position trajectories. First row: global solution trajectories of 18 crazyflies, found by solving the nonlinear motion planning problem by sequential convex programming. Second row: naive fictitious solution trajectories obtained with the Buffered Voronoi Cells method [14]. Initial and terminal positions are consistent with those of Fig. 10.11. . . . .	203
10.11	Demonstrated Neural-Rendezvous trajectories of 18 crazyflies, where the circles are their initial positions and the crosses are their target positions. The first three figures represent randomized experiments without artificial disturbance, 4th figure represents a randomized simulation with artificial disturbance, and 5th figure represents an experiment with the ISO model (see Fig. 10.12). . . . .	204

10.12	Demonstrated Neural-Rendezvous trajectories of 20 crazyflies observed in the actual environment, illustrated using their light trails. This movie can be found <a href="#">here</a> . . . . .	205
10.13	Position trajectories of the nonlinear system (7.7) with (10.2) with small and large disturbance, $\bar{d}^i = \bar{\gamma}^i = 2.0 \times 10^{-5}$ and $\bar{d}^i = \bar{\gamma}^i = 2.0 \times 10^{-2}$ in (7.7), respectively, which capture the trade-off discussed in (a) and (b) of Sec. 7.5.I-B. . . . .	206
10.14	Position trajectories of the spacecraft simulator system [11] with various disturbances (small: $\bar{d}^i = \bar{\gamma}^i = 3.0 \times 10^{-4}$ , medium: $\bar{d}^i = \bar{\gamma}^i = 5.0 \times 10^{-3}$ , and large: $\bar{d}^i = \bar{\gamma}^i = 5.0 \times 10^{-2}$ in (7.7)), which capture the trade-off discussed in Sec. 7.5.I. . . . .	207
10.15	Position trajectories for the multi-spacecraft reconfiguration task in LEO [2] with various disturbances (small: $\bar{d}^i = \bar{\gamma}^i = 1.0 \times 10^{-2}$ and large: $\bar{d}^i = \bar{\gamma}^i = 5.0 \times 10^{-2}$ in (7.6)). . . . .	208
10.16	Position trajectories for the swarm motion planning and control of UAVs with one rotor failure [20], [21] with disturbance $\bar{d}^i = \bar{\gamma}^i = 2.0 \times 10^{-2}$ in (7.7). . . . .	210
11.1	Overview of our learning-based control with formal guarantees, revisited. . . . .	214

## LIST OF TABLES

<i>Number</i>	<i>Page</i>
1.1 Differences between contraction theory and Lyapunov theory. . . . .	2
1.2 Additional notations to be used in this thesis. . . . .	13
3.1 Comparison between the SDC and CCM formulation (note that $\gamma(\mu = 0, t) = x_d$ and $\gamma(\mu = 1, t) = x$ ). . . . .	52
6.1 Notations in Example 6.5. . . . .	105
7.1 Comparison of LAG-ROS [1] with the learning-based and robust tube-based motion planners. . . . .	109
9.1 Notations in Lemma 9.1. . . . .	162
9.2 Summary of the ISO state flow and spacecraft state processes in Definitions 9.1 and 9.2, where $u_\ell$ and $u_{\text{mpc}}$ are the SN-DNN and MPC guidance policies, respectively. Note that the dependence on the terminal position $\rho$ is omitted as explained in Remark 9.5. . . . .	166
9.3 Notations in Sec. 9.3.III. . . . .	170
9.4 Notations in Theorem 9.1. . . . .	173
10.1 Computational time of each method for the ISO encounter averaged over 100 evaluations. . . . .	197
10.2 Scales of the state and control in each dynamics. . . . .	199
10.3 Control performances for the multi-spacecraft reconfiguration in LEO averaged over 50 simulations, where $J = \int_0^{t_f} \sum_{i=1}^N \ u^i\ ^2 d\tau$ . . . . .	209

## NOMENCLATURE

- AI.** Artificial Intelligence.
- aNCM.** Adaptive Neural Contraction Metric.
- CART.** Collision Avoidance and Robust Tracking.
- CASTOR.** Constellation Autonomous Space Technology Demonstration of Orbital Reconfiguration. This is one of Caltech-JPL's joint projects.
- CBF.** Control Barrier Function.
- CCM.** Control Contraction Metric.
- CLF.** Control Lyapunov Function.
- Crazyflie.** A versatile open-source flying development platform to perform various types of aerial robotics research (a micro-UAV). Its product description can be found here <https://www.bitcraze.io/products/crazyflie-2-1/>.
- CV-STEM.** ConVex optimization-based Steady-state Tracking Error Minimization.
- DNN.** Deep Neural Network.
- EKF.** Extended Kalman Filter.
- G&C.** Guidance and Control.
- GNC.** Guidance, Navigation, and Control.
- ISO.** InterStellar Object. ISOs, also known as extrasolar objects, are celestial bodies that originate from outside our solar system and travel through interstellar space. These objects are believed to have been ejected from their home star systems and are passing through our solar system as they travel through the vastness of space.
- ISS.** Input-to-State Stability.
- JPL.** Jet Propulsion Laboratory. JPL is a research facility and center for space exploration federally funded by NASA and managed by Caltech. Located in Pasadena, California, JPL is a leading institution in the field of planetary science and space technology.
- KYP lemma.** Kalman–Yakubovich–Popov lemma.
- LAG-ROS.** Learning-based Autonomous Guidance with RObustness and Stability.
- LMI.** Linear Matrix Inequality.

- LTV.** Linear Time-Varying system.
- M-STAR.** Multi-Spacecraft Testbed for Autonomy Research. It is a thruster-based spacecraft simulator we have at Caltech (see [here](#)).
- MPC.** Model Predictive Control.
- NCM.** Neural Contraction Metric.
- NSCM.** Neural Stochastic Contraction Metric.
- PDE.** Partial Differential Equation.
- SDC.** State-Dependent Coefficient.
- SDRE.** State-Dependent Riccati Equation.
- SLAM.** Simultaneous Localization And Mapping.
- SN-DNN.** Spectrally-Normalized Deep Neural Network.
- SOS.** Sum Of Squares.
- TCM.** Trajectory Correction Maneuver.
- UAV.** Unmanned, Uninhabited, or Uncrewed Aerial Vehicles.

*Chapter 1*

## INTRODUCTION

Nonlinear dynamical systems describe the time evolution of various real-world phenomena. As next-generation research objectives have become more complicated and large-scale, the Guidance, Navigation, and Control (GNC) problems can easily become computationally and/or theoretically intractable for conventional nonlinear control approaches, which has led to the emergence of learning-based control designs with, e.g., reinforcement learning [1]–[4], imitation learning [5]–[9], or neural networks [10]–[12]. In highly autonomous and safety-critical tasks, such as developing fully autonomous robotic explorers operating remotely on land, in water, and in deep space, however, the existing performance guarantees of learning-based approaches may not be sufficiently powerful.

The major focus of this thesis is to step back and provide mathematically rigorous ways to enjoy the formal guarantees of nonlinear control theory even with the use of learning-based autonomous GNC. Our motivation here is not just to develop learning-based methods with formal guarantees that work better than conventional control theoretical approaches, but to provide formal tools to assess whether we absolutely need machine learning or not in a given problem setting [13]. We can thereby confidently determine the most appropriate tools to design our truly autonomous systems safely, robustly, adaptively, and intelligently in real-time, without jumping to seemingly performant solutions given by some black-box AI approaches. Such analytical tools are the key ingredients for establishing the groundwork of truly autonomous GNC, where anyone can follow their curiosity and turn their imagination into reality to the full extent and beyond, regardless of the physical, intellectual, and temporal constraints inherent to humans. In particular, we use a nonlinear control theoretical tool called contraction theory, which can be understood in the context of Lyapunov theory in nonlinear systems theory.

Lyapunov theory is one of the most widely-used approaches to stability analysis of a nonlinear system [14], [16]–[20], which provides a condition for stability with respect to an equilibrium point, a target trajectory, or an invariant set. Contraction theory [15], [21]–[24] rewrites suitable Lyapunov stability conditions using a quadratic Lyapunov function of the differential states, defined by a Riemannian

Table 1.1: Differences between contraction theory and Lyapunov theory.

	Contraction theory (which constructs a positive definite matrix $M(x, t)$ that defines a contraction metric)	Lyapunov direct method (which constructs a Lyapunov function $V(x, t)$ )
1. Lyapunov function	Always a quadratic function of the differential state $\delta x$ ( $V = \delta x^T M(x, t) \delta x$ )	Any function of $x$ , including $V(x, t) = x^T M(x, t)x$
2. Stability condition	Exponential stability of trajectories including points and invariant sets*	Asymptotic or exponential stability of points and invariant sets
3. Incremental stability	Incremental stability of trajectories with differential displacements ( $\lim_{t \rightarrow \infty} \delta x = 0$ )	Incremental stability via stability of points ( $\lim_{t \rightarrow \infty} (x - x_d) = 0$ for given $x_d$ )
4. Non-autonomous system	Same theory as for autonomous systems	Additional conditions required for non-autonomous stability analysis
5. Robustness analysis	Intuitive both for ISS and finite-gain $\mathcal{L}_p$ stability due to the extensive use of exponential stability	Same as contraction theory if exponentially stable; more involved if uniformly asymptotically stable
6. Analogy to linear systems	LTV-like differential dynamics for global convergence	Indirect methods use linearization for local stability (direct methods use motion integrals)
7. $\mathcal{L}_2$ stability condition	Reduces to LMI conditions in terms of a contraction metric defined by a positive definite matrix $M$	Hamilton-Jacobi inequality (PDE) in terms of a Lyapunov function $V$ [14, p. 211]
8. Modular stability	Differential analysis handles hierarchical, feedback, and parallel combinations [15]	Passivity is not intuitive for hierarchical combinations

\*A semi-contracting system with a negative semi-definite generalized Jacobian matrix can be used to analyze asymptotic stability (see Chapter 8).

contraction metric and its uniformly positive definite matrix, thereby characterizing a necessary and sufficient condition for incremental exponential convergence of the multiple nonlinear system trajectories to one single trajectory. It can be regarded as a generalization of Krasovskii’s theorem [19, p. 83] applied to nonlinear incremental stability analysis [21], [25], where the differential formulation permits a pure differential coordinate change with a non-constant metric for simplifying its stability proofs [21].

The differential nature of contraction theory implies we can exploit the Linear Time-Varying (LTV) systems-type techniques for nonlinear stability analysis and control/estimation synthesis [26]–[30] (see Table 1.1). We emphasize that some of these methodological simplifications in contraction theory are accomplished by its extensive use of exponential stability along with the comparison lemma [14, pp. 102–103, pp. 350–353], in lieu of Input-to-State Stability (ISS) or uniform asymptotic stability which often renders nonlinear stability analysis more involved [14], [16]–[20]. Several studies related to the notion of contraction, although not based on direct differential analysis, can be traced back to [31]–[33].

The first two parts of this thesis are for elucidating how contraction theory may be utilized as a method of providing provable incremental exponential robustness and stability guarantees of learning-based and data-driven automatic control techniques (see Fig. 1.1). In pursuit of this goal, we also provide an overview of the advantages of contraction theory and present a systematic convex optimization formulation to explicitly construct an optimal contraction metric and a differential Lyapunov function for general nonlinear deterministic and stochastic systems. The third part of this thesis is dedicated to the real-world applications of our proposed approaches in our joint-projects with NASA Jet Propulsion Laboratory (JPL). It also includes empirical validation of some of our algorithms using robotic hardware such as thruster-based spacecraft simulators called M-STAR [34] and micro-UAVs called crazyflies (<https://www.bitcraze.io/products/crazyflie-2-1/>).

## 1.1 Thesis Organization

This thesis is organized into the following three groups of chapters.

### **Part I: Nonlinear Stability Analysis (Chapters 2 – 4)**

In Chapter 2, we present the fundamental results of contraction theory for nonlinear robustness and stability analysis. Chapter 3 and Chapter 4 consider nonlinear optimal feedback control and estimation problems from the perspective of contraction

### Overview of Learning-based Control with Formal Guarantees (in the context of nonlinear control theory)

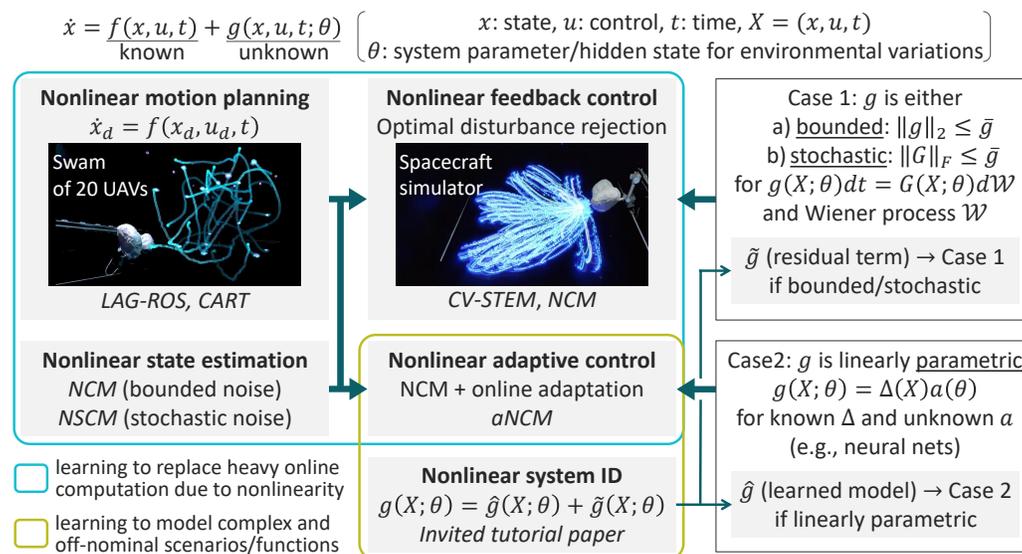


Figure 1.1: Overview of our learning-based control with formal guarantees (in the context of nonlinear control theory).

theory, deriving and delineating a convex optimization-based method for constructing contraction metrics. Chapter 3 also presents some new results on relating contraction theory to the bounded real lemma [35] and Kalman–Yakubovich–Popov (KYP) lemma [36, p. 218].

## Part II: Learning-based Control (Chapters 5 – 8)

In Chapter 5, we derive several theorems which form the basis of learning-based control using contraction theory. Chapter 6 and Chapter 7 present frameworks for learning-based control, estimation, and motion planning via contraction theory using deep neural networks for designing contraction metrics, and Chapter 8 extends these results to parametric uncertain nonlinear systems with adaptive control techniques. Also in Chapter 8, we propose model-free versions of contraction theory for learning-based and data-driven control.

## Part III: Real-World Applications (Chapters 9 – 10)

In Chapter 9, we present the real-world application of some of our approaches in exploring Interstellar Objects (ISOs). Chapter 10 then presents several numerical simulations and empirical validation of our proposed approaches to assess the impact of our findings on realizing aerospace and robotic autonomy. We mainly focus on the two joint projects with NASA JPL: 1) Science-Infused Spacecraft Autonomy for ISO Exploration [37], [38] and 2) Constellation Autonomous Space Technology Demon-

stration of Orbital Reconfiguration (CASTOR) [39], where we also perform hardware demonstrations of our methods using our thruster-based spacecraft simulators (M-STAR) [34] and in high-conflict, distributed, intelligent UAV swarm reconfiguration with up to 20 UAVs (crazyflies, <https://www.bitcraze.io/products/crazyflie-2-1/>).

## 1.2 Related Work

In the remainder, we give an overview of each chapter (Chapters 2 – 10) as well as a survey of related work.

### Contraction Theory (Chapter 2)

According to contraction theory, all the solution trajectories of a given nonlinear system converge to one single trajectory incrementally and exponentially, regardless of the initial conditions, if the system has a contraction metric and its associated quadratic Lyapunov function of the differential state [21] (see 1 – 3 of Table 1.1). This thesis primarily considers this generalized notion of stability, called incremental exponential stability, which enables systematic learning-based and data-driven control synthesis with formal robustness and stability guarantees. The purpose of this chapter is not for proposing that other notions of stability, such as traditional Lyapunov-based stability or incremental asymptotic stability for semi-contraction systems, should be replaced by incremental exponential stability, but for clarifying its advantages to help determine which of these concepts is the best fit when analyzing nonlinear robustness and stability (see Chapter 2 for the illustrative examples).

In keeping with the use of the comparison lemma [14, pp. 102-103, pp. 350-353], incremental exponential stability naturally holds for non-autonomous nonlinear systems without any additional conditions or modifications unlike Lyapunov techniques (see, e.g., the examples and theorems in [40]). Such aspects of contraction theory, including the extensive use of exponential stability, result in intuitive proofs on ISS and finite-gain  $\mathcal{L}_p$  stability both for autonomous and non-autonomous nonlinear systems, without resorting to uniform asymptotic stability which makes stability analysis much more involved than necessary [14], [16]–[20]. In particular, perturbed systems with a time-varying target trajectory are non-autonomous, and thus contraction theory allows us to easily obtain an explicit exponential bound on its tracking error [21], [26], [29], [30], [41], leveraging incremental stability of the perturbed system trajectories with respect to the target trajectory (see 4 of Table 1.1).

Having such analytical bounds on the tracking error is almost essential for the safe and robust implementation of automatic control schemes in real-world scenarios.

Contraction theory also simplifies input-output stability analysis, such as  $\mathcal{L}_p$  gain analysis of nonlinear systems including the  $\mathcal{H}_\infty$  nonlinear optimal control problem [42]–[49]. For example, in Lyapunov theory, the problem of finding a suitable Lyapunov function with the smallest  $\mathcal{L}_2$  gain boils down to solving a Partial Differential Equation (PDE) called the Hamilton-Jacobi inequality [14, p. 211], [42] in terms of its associated Lyapunov function. In essence, since contraction theory utilizes a quadratic Lyapunov function of the differential state for stability analysis, the problem could be solved with a Linear Matrix Inequality (LMI) constraint [35] analogous to the KYP lemma [36, p. 218] in LTV systems theory [26], [29], [30], [50], as shall be shown in Chapter 3 – 4 (see 5 – 7 of Table 1.1). There exist stochastic analogues of these stability results for nonlinear systems with stochastic perturbations [26], [29], [30], [51], [52], as shall be outlined also in this thesis.

Another notable feature of contraction theory is modularity, which preserves contraction through parallel, feedback, and hierarchical combinations [15], [23], specific time-delayed feedback communications [53], synchronized coupled oscillations [24], [54], and synchronized networks [15], [41], [54]–[57], expanding the results obtainable with the passivity formalism of Lyapunov theory [58], [14, p. 227], [19, p. 132] (see 8 of Table 1.1). Due to all these useful properties, extensions of contraction theory have been considered in many different settings. These include, but are not limited to, stochastic contraction (Gaussian white noise [26], [29], [30], [51], Poisson shot noise and Lévy noise [52]), contraction for discrete and hybrid nonlinear systems [21], [22], [26], [30], [54], [59], [60], partial contraction [24], transverse contraction [61], contraction in terms of semi-norms [62], incremental stability analysis of nonlinear estimation (the Extended Kalman Filter (EKF) [63], nonlinear observers [29], [64], Simultaneous Localization And Mapping (SLAM) [65]), generalized gradient descent based on geodesical convexity [66], contraction on Finsler and Riemannian manifolds [67]–[69], contraction on Banach and Hilbert spaces for PDEs [70]–[72], non-Euclidean contraction [73], contracting learning with piecewise-linear basis functions [74], incremental quadratic stability analysis [75], contraction after small transients [76], immersion and invariance stabilizing controller design [77], [78], and Lipschitz-bounded neural networks for robustness and stability guarantees [79]–[82].

### Construction of Contraction Metrics (Chapters 3 – 4)

The benefits of contraction theory reviewed so far naturally lead to a discussion on how to design a contraction metric and corresponding Lyapunov function. There are some cases in which we can analytically find them using special structures of systems in question [14], [16], [19]. Among these are Lagrangian systems [19, p. 392], where one easy choice of positive definite matrices that define a contraction metric is the inertia matrix, or feedback linearizable systems [83]–[87], where we could solve the Riccati equation for a contraction metric as in LTV systems. This is also the case in the context of state estimation (e.g., the nonlinear SLAM problem can be reformulated as an LTV estimation problem using virtual synthetic measurements [29], [65]). Once we find a contraction metric and Lyapunov function of a nominal nonlinear system for the sake of stability, they could be used as a Control Lyapunov Function (CLF) to attain stabilizing feedback control [14], [88], [89] or could be augmented with an integral control law called adaptive backstepping to recursively design a Lyapunov function for strict- and output-feedback systems [90]–[93]. However, deriving an analytical form of contraction metrics for general nonlinear systems is challenging, and thus several search algorithms have been developed for finding them at least numerically using the LMI nature of the contraction condition.

The simplest of these techniques is the method of State-Dependent Riccati Equation (SDRE) [94]–[97], which uses the State-Dependent Coefficient (SDC) parameterization (also known as extended linearization) of nonlinear systems for feedback control and state estimation synthesis. Motivating optimization-based approaches to design a contraction metric, it is proposed in [26], [29], [30] that the Hamilton-Jacobi inequality for the finite-gain  $\mathcal{L}_2$  stability condition can be expressed as an LMI when contraction theory is equipped with the extended linearity of the SDC formulation. Specifically, in [26]–[28], [30], a convex optimization-based framework for robust feedback control and state estimation, named ConVex optimization-based Steady-state Tracking Error Minimization (CV-STEM), is derived to find a contraction metric that minimizes an upper bound of the steady-state distance between perturbed and unperturbed system trajectories. In this context, we could utilize Control Contraction Metrics (CCMs) [50], [98]–[102] for extending contraction theory to the systematic design of differential feedback control  $\delta u = k(x, \delta x, u, t)$  via convex optimization, achieving greater generality at the expense of computational efficiency in obtaining  $u$ . Applications of the CCM to estimation, adaptive control, and motion planning are discussed in [103], [104]–[106], and [99], [107]–

[110], respectively, using geodesic distances between trajectories [68]. It is also worth noting that the objective function of CV-STEM has the condition number of a positive definite matrix for a contraction metric as one of its arguments, rendering it applicable and effective even to machine learning-based automatic control frameworks as shall be seen in Chapters 5 – 8.

### **Contraction Theory for Learning-based Control (Chapter 5)**

One drawback of these numerical schemes is that they require solving optimization problems or nonlinear systems of equations at each time instant online, which is not necessarily realistic in practice. In Lyapunov theory, approximating functions in a given hypothesis space has therefore been a standard technique [111]–[121], where examples of its function classes include piecewise quadratic functions [111], linearly parameterized non-quadratic functions [114], a linear combination of radial basis functions [115], Sum-Of-Squares (SOS) functions [116], and neural networks [113], [117], [118]. In [86], [98], the SOS approximation is investigated for the case of contraction theory, showing that the contraction condition can be relaxed to SOS conditions for dynamics with polynomial or rational vector fields. Although computationally tractable, it still has some limitations in that the problem size grows exponentially with the number of variables and basis functions [122]. Learning-based and data-driven control using contraction theory [27], [28], [123], [124] has been developed to refine these ideas, using the high representational power of Deep Neural Networks (DNNs) [10]–[12] and their scalable training realized by stochastic gradient descent [66], [125].

The major advantage of using contraction theory for learning-based and data-driven control is that, by regarding its internal learning error as an external disturbance, we can ensure the distance between the target and learned trajectories to be bounded exponentially with time as in the CV-STEM results [27], [28], [123], [124], with its steady-state upper bound proportional to the learning error. Such robustness and incremental stability guarantees are useful for formally evaluating the performance of machine learning techniques such as reinforcement learning [1]–[4], imitation learning [5]–[9], or neural networks [10]–[12]. This implies contraction theory could be utilized as a central tool in realizing safe and robust operations of learning-based and data-driven control, estimation, and motion planning schemes in real-world scenarios. We especially focus on the following areas of research.

### **Learning-based Robust Control and Estimation (Chapter 6)**

In order to achieve real-time computation of a contraction metric, mathematical models based on a Deep Neural Network (DNN) called a Neural Contraction Metric (NCM) [27] and Neural Stochastic Contraction Metric (NSCM) [28] are derived to compute optimal CV-STEM contraction metrics for nonlinear systems perturbed by deterministic and stochastic disturbances, respectively. It can be proven that the NCM and NSCM still yield robustness and optimality associated with the CV-STEM framework despite having non-zero modeling errors [126]. These metrics could also be synthesized and learned simultaneously with their feedback control laws directly by DNNs [110], [126], [127] at the expense of the convex property in the CV-STEM formulation.

### **Learning-based Safe and Robust Motion Planning (Chapter 7)**

In [99], [128], contraction theory is leveraged to develop a tracking feedback controller with an optimized control invariant tube, solving the problem of robust motion planning under bounded external disturbances. This problem is also considered for systems with changing operating conditions [107] and parametric uncertainty [108]–[110] for its broader use in practice. As these methods still require online computation of a target trajectory, Learning-based Autonomous Guidance with RObustness and Stability (LAG-ROS) [124] is developed to model such robust control laws including the CV-STEM by a DNN, without explicitly requiring the target or desired trajectory as its input. While this considers motion planning algorithms only implicitly to avoid solving them in real-time, it is shown that contraction theory still allows us to assure a property of robustness against deterministic and stochastic disturbances following the same argument as in the NCM and NSCM work [124]. Note that LAG-ROS using contraction theory is not intended to derive new learning-based motion planning, but rather to augment any existing motion planner with a real-time method of guaranteeing formal incremental robustness and stability. It is thus still applicable to other methods such as tube-based robust Model Predictive Control (MPC) [99], [128]–[132], its dynamic and adaptive counterparts [107]–[110], and CCM-based learning certified control [127].

We also present CART (which is named after Collision Avoidance and Robust Tracking) [39], an analytical method to augment a learning-based, distributed motion planning policy of a nonlinear multi-agent system with real-time collision avoidance and robust tracking guarantees, independently of the performance of the learning approaches used in designing the learned policy. Although our approach can be

extended further to handle general notions of safety, CART focuses on collision-free operation as the objective of safety.

### **Learning-based Adaptive Control and Contraction Theory for Learned Models (Chapter 8)**

Adaptive control using contraction theory is studied in [93] for parametric strict-feedback nonlinear systems, and recently generalized to deal with systems with unmatched parametric uncertainty by means of parameter-dependent CCM feedback control [104], [105]. This method is further explored to develop an adaptive Neural Contraction Metric (aNCM) [123], a parameter-dependent DNN model of the adaptive CV-STEM contraction metric. As the name suggests, the aNCM control makes adaptive control of [104], [105] implementable in real-time for asymptotic stabilization, while maintaining the learning-based robustness and CV-STEM-type optimality of the NCM. Although it is designed to avoid the computation for evaluating integrals involving geodesics unlike [104], [105], these differential state feedback schemes could still be considered, trading off added computational cost for generality. It is demonstrated in [123] that the aNCM is applicable to many types of systems such as robotics systems [19, p. 392], spacecraft high-fidelity dynamics [133], [134], and systems modeled by basis function approximation and DNNs [135], [136]. Discrete changes could be incorporated in this framework using [106], [137].

Furthermore, recent applications of machine learning often consider challenging scenarios in the field of systems and control theory, where we only have access to system trajectory data generated by unknown underlying dynamics, and the assumptions in the aforementioned adaptive control techniques are no longer valid. For situations where the data is used for system identification of full/residual dynamics [79], [82], [100], [138], [139], we can show by contraction theory that the model-based approaches (e.g., CV-STEM and NCM) are still utilizable to guarantee robustness against dynamics modeling errors and external disturbances. Note that it is also proposed in [140] that we could directly learn certificate functions such as contraction metrics and their associated Lyapunov functions using trajectory data. Also, some of the theoretical results on gradient descent algorithms, essential in the field of data-driven machine learning, can be replaced by more general ones based on contraction and geodesical convexity [66].

## **Interstellar Object Exploration (Chapter 9)**

The methods outlined so far are also shown to be useful in real-world applications such as ISO exploration. ISOs are active or inert objects passing through our solar system on an unbound hyperbolic trajectory about the Sun, which could sample planetesimals and primitive materials that provide vectors to compare our solar system with neighboring exoplanetary star systems [141]. To date, two such objects have been identified and observed: 1I/‘Oumuamua [142] discovered in 2017 and 2I/Borisov [143] discovered in 2019 (see Fig. 1.2). In 2022, the United States Department of Defense confirmed that a third ISO impacted Earth in 2014, three years before the identification of ‘Oumuamua. ISOs are physical laboratories that can enable the study of exosolar systems in-situ rather than remotely using telescopes such as the Hubble or James Webb Space Telescopes.

While remote observation can aid in constraining the level of activity, shape, and spectral signature of ISOs, limiting scientists to Earth-based telescopic observation prevents some of the most impactful science that can be performed. Using a dedicated spacecraft to flyby an ISO opens the doors to high-resolution imaging, mass or dust spectroscopy, and a larger number of vantage points than Earth observation. It could also resolve the target’s nucleus shape and spin, characterize the volatiles being shed from an active body, reveal fresh surface material using an impactor, and more [144].

The discovery and exploration of ISOs are “once in a lifetime” or maybe “once in a civilization” opportunities, and their exploration is challenging for three main reasons: 1) they are often not discovered until they are close to Earth, meaning that launches to encounter them often require high launch energy; 2) their orbital properties are poorly constrained at launch, generally leading to significant on-board resources to encounter; and 3) the encounter speeds are typically high ( $> 10$  of km/s) requiring fast response autonomous operations. Chapter 9 is for presenting an offline deep learning-based robust nonlinear guidance and control (G&C) approach, called Neural-Rendezvous [37], to autonomously encounter them even in the presence of such large state uncertainty and high-velocity challenges. As outlined in Fig. 1.3, the guidance, navigation, and control (GNC) of spacecraft for the ISO encounter are split into two segments: 1) the cruise phase, where the spacecraft utilizes state estimation obtained by ground-based telescopes and navigates via ground-in-the-loop operations, and; 2) the terminal phase, where it switches to fully autonomous operation with existing onboard navigation frameworks. The current state of practice

and performance of autonomous navigation systems for small bodies like ISOs are discussed in [145], [146]. Neural-Rendezvous is for performing the second phase of the autonomous terminal G&C with the on-board state estimates and is built upon the spectrally-normalized deep neural network (SN-DNN), a neural network that has been successfully used for providing stability guarantees of learning-based feedback controllers [79], [147], [148].



Figure 1.2: Interstellar objects. Above: Artist’s illustration 1I/ʻOumuamua (credit: NASA, ESA, and STScI). Below: 2I/Borisov near and at perihelion (credit: NASA, ESA, and D. Jewitt (UCLA)). Click the picture to watch our YouTube video ([https://youtu.be/8h60B\\_p1fyQ](https://youtu.be/8h60B_p1fyQ)).

### **Additional Numerical Simulations and Empirical Validation (Chapter 10)**

Applying deep learning-based algorithms to real-world systems always involves unmodeled uncertainties not just in its state, but also in the dynamics, environment, and control actuation, which could all lead to destabilizing behaviors different from what is learned and observed in numerical simulations. Our learning-based approaches to be discussed in this thesis are shown to be provably robust against these uncertainties, thereby helping to reproduce the simulation results seamlessly in real-world hardware experiments. We assess the validity of this argument based on our formal guarantees using the thruster-based spacecraft simulators (M-STAR) [34] for the ISO exploration demonstration and UAV swarms (crazyflies, <https://www.bitcraze.io/products/crazyflie-2-1/>) for achieving high-conflict, distributed, intelligent swarm reconfiguration with up to 20 UAVs as

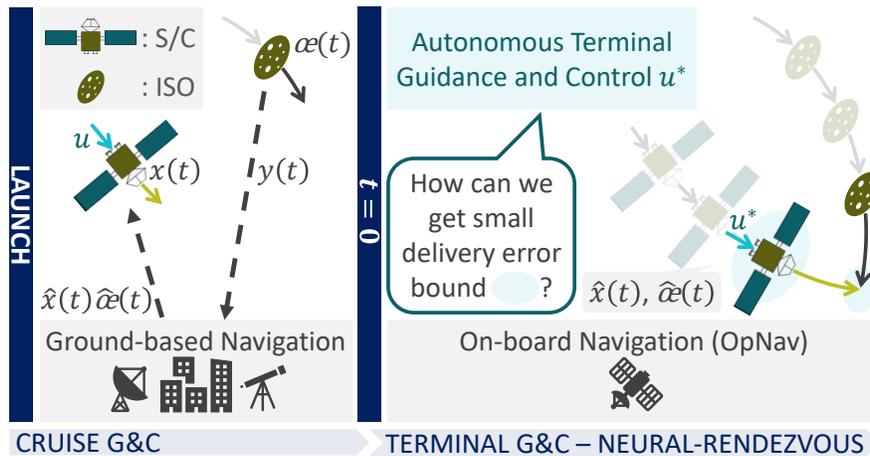
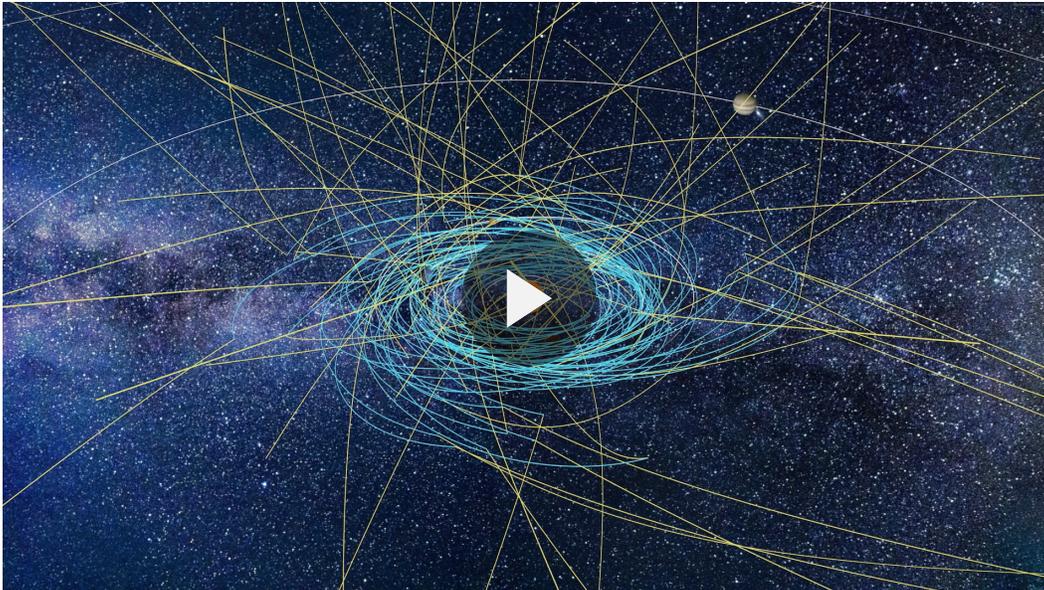


Figure 1.3: Illustration of cruise and terminal GNC ( $t$ : time;  $\alpha(t)$ : ISO state;  $x(t)$ : spacecraft state relative to  $\alpha(t)$ ;  $y(t)$ : state measurement of the ISO and spacecraft;  $u$ : control input;  $\hat{\alpha}(t)$ : estimated ISO state; and  $\hat{x}(t)$ : estimated spacecraft relative state). Neural-Rendezvous enables obtaining a verifiable delivery error bound even under the large ISO state uncertainty and high-velocity challenges.

Table 1.2: Additional notations to be used in this thesis.

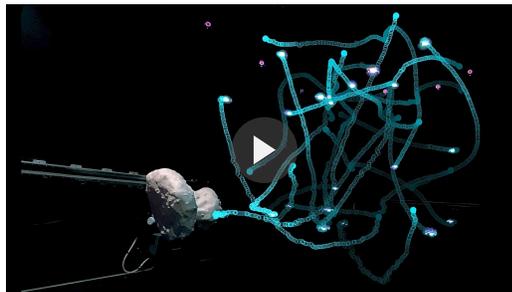
Notation	Description
$\ x\ $	Euclidean norm of $x \in \mathbb{R}^n$
$\delta x$	Differential displacement of $x \in \mathbb{R}^n$
$\ A\ $	Induced 2-norm of $A \in \mathbb{R}^{n \times m}$
$\ A\ _F$	Frobenius norm of $A \in \mathbb{R}^{n \times m}$
$\text{sym}(A)$	Symmetric part of $A \in \mathbb{R}^{n \times n}$ , i.e., $(A + A^T)/2$
$\lambda_{\min}(A)$	Minimum eigenvalue of $A \in \mathbb{R}^{n \times n}$
$\lambda_{\max}(A)$	Maximum eigenvalue of $A \in \mathbb{R}^{n \times n}$
$\mathbf{I}$	Identity matrix of appropriate dimensions
$\mathbb{E}$	Expected value operator
$\mathbb{P}$	Probability measure
$\mathbb{R}_{>0}$	Set of positive reals, i.e., $\{a \in \mathbb{R}   a \in (0, \infty)\}$
$\mathbb{R}_{\geq 0}$	Set of non-negative reals, i.e., $\{a \in \mathbb{R}   a \in [0, \infty)\}$



(a) Numerical simulations for ISO exploration



(b) Spacecraft simulator rendezvous



(c) UAV swarm reconfiguration

Figure 1.4: Visualizations of our numerical and experimental validation to be discussed in Chapter 10.2.

part of the CASTOR project. We also present some recent numerical simulation results in [39] (see Fig. 1.4).

### 1.3 Notation

For a square matrix  $A^{n \times n}$ , we use the notation  $A > 0$ ,  $A \geq 0$ ,  $A < 0$ , and  $A \leq 0$  for the positive definite, positive semi-definite, negative definite, negative semi-definite matrices, respectively. The  $\mathcal{L}_p$  norm in the extended space  $\mathcal{L}_{pe}$  [14, pp. 196-197],  $p \in [1, \infty]$ , is defined as  $\|(y)_\tau\|_{\mathcal{L}_p} = \left( \int_0^\tau \|y(t)\|^p \right)^{1/p} < \infty$  for  $p \in [1, \infty)$  and  $\|(y)_\tau\|_{\mathcal{L}_\infty} = \sup_{t \geq 0} \|(y(t))_\tau\| < \infty$  for  $p = \infty$ , where  $(y(t))_\tau$  is a truncation of  $y(t)$ , i.e.,  $(y(t))_\tau = 0$  for  $t > \tau$  and  $(y(t))_\tau = y(t)$  for  $0 \leq t \leq \tau$  with  $\tau \in \mathbb{R}_{\geq 0}$ . Furthermore, we use  $f_x = \partial f / \partial x$ ,  $M_{x_i} = \partial M / \partial x_i$ , and  $M_{x_i x_j} = \partial^2 M / (\partial x_i \partial x_j)$ ,

where  $x_i$  and  $x_j$  are the  $i$ th and  $j$ th elements of  $x \in \mathbb{R}^n$ , for describing partial derivatives in a limited space. The other notations are given in Table 1.2.

## References

- [1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998, ISBN: 0-262-19398-1.
- [2] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.
- [3] M. Everett, Y. F. Chen, and J. P. How, “Motion planning among dynamic, decision-making agents with deep reinforcement learning,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2018, pp. 3052–3059.
- [4] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, “Safe model-based reinforcement learning with stability guarantees,” in *Adv. Neural Inf. Process. Syst.*, vol. 30, Curran Associates, Inc., 2017, pp. 908–918.
- [5] J. Carius, F. Farshidian, and M. Hutter, “MPC-Net: A First Principles Guided Policy Search,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2897–2904, 2020.
- [6] B. Rivière, W. Hönig, Y. Yue, and S.-J. Chung, “GLAS: Global-to-local safe autonomy synthesis for multi-robot motion planning with end-to-end learning,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4249–4256, 2020.
- [7] J. Ho and S. Ermon, “Generative adversarial imitation learning,” in *Adv. Neural Inf. Process. Syst.*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, Curran Associates, Inc., 2016, pp. 4565–4573.
- [8] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, “Social GAN: Socially acceptable trajectories with generative adversarial networks,” in *IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 2255–2264.
- [9] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, “Imitating driver behavior with generative adversarial networks,” in *IEEE Intell. Vehicles Symp.*, 2017, pp. 204–211.
- [10] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989, ISSN: 0893-6080.
- [11] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Math. Control Signals Syst.*, vol. 2, no. 4, pp. 303–314, Dec. 1989, ISSN: 1435-568X.
- [12] K.-I. Funahashi, “On the approximate realization of continuous mappings by neural networks,” *Neural Netw.*, vol. 2, no. 3, pp. 183–192, 1989, ISSN: 0893-6080.

- [13] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview,” *Annu. Rev. Control*, vol. 52, pp. 135–169, 2021, ISSN: 1367-5788.
- [14] H. K. Khalil, *Nonlinear Systems*, 3rd. Upper Saddle River, NJ: Prentice-Hall, 2002.
- [15] J.-J. E. Slotine, “Modular stability tools for distributed computation and control,” *Int. J. Adapt. Control Signal Process.*, vol. 17, no. 6, pp. 397–416, 2003.
- [16] A. Isidori, *Nonlinear Control Systems*, 3rd. Berlin, Heidelberg: Springer-Verlag, 1995, ISBN: 3540199160.
- [17] R. Marino and P. Tomei, *Nonlinear Control Design: Geometric, Adaptive, and Robust* (Prentice-Hall information and system sciences series). Prentice Hall, 1995, ISBN: 9780133426359.
- [18] M. Vidyasagar, *Nonlinear Systems Analysis*, Second. Society for Industrial and Applied Mathematics, 2002.
- [19] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Upper Saddle River, NJ: Pearson, 1991.
- [20] H. Nijmeijer and A. Van der Schaft, *Nonlinear Dynamical Control Systems*. Berlin, Heidelberg: Springer-Verlag, 1990, ISBN: 038797234X.
- [21] W. Lohmiller and J.-J. E. Slotine, “On contraction analysis for nonlinear systems,” *Automatica*, vol. 34, no. 6, pp. 683–696, 1998, ISSN: 0005-1098.
- [22] W. Lohmiller and J.-J. E. Slotine, “Nonlinear process control using contraction theory,” *AIChE Journal*, vol. 46, pp. 588–596, Mar. 2000.
- [23] J.-J. E. Slotine and W. Lohmiller, “Modularity, evolution, and the binding problem: A view from stability theory,” *Neural Netw.*, vol. 14, no. 2, pp. 137–145, 2001.
- [24] W. Wang and J.-J. E. Slotine, “On partial contraction analysis for coupled nonlinear oscillators,” *Biol. Cybern.*, vol. 92, no. 1, pp. 38–53, Jan. 2005, ISSN: 0340-1200.
- [25] D. Angeli, “A Lyapunov approach to incremental stability properties,” *IEEE Trans. Autom. Control*, vol. 47, no. 3, pp. 410–421, Mar. 2002.
- [26] H. Tsukamoto and S.-J. Chung, “Robust controller design for stochastic nonlinear systems via convex optimization,” *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4731–4746, 2021.
- [27] H. Tsukamoto and S.-J. Chung, “Neural contraction metrics for robust estimation and control: A convex optimization approach,” *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 211–216, 2021.

- [28] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Neural stochastic contraction metrics for learning-based control and estimation,” *IEEE Control Syst. Lett.*, vol. 5, no. 5, pp. 1825–1830, 2021.
- [29] A. P. Dani, S.-J. Chung, and S. Hutchinson, “Observer design for stochastic nonlinear systems via contraction-based incremental stability,” *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 700–714, Mar. 2015.
- [30] H. Tsukamoto and S.-J. Chung, “Convex optimization-based controller design for stochastic nonlinear systems using contraction analysis,” in *IEEE Conf. Decis. Control*, Dec. 2019, pp. 8196–8203.
- [31] P. Hartman, *Ordinary Differential Equations*, Second. Soc. Ind. Appl. Math., 2002.
- [32] B. P. Demidovich, “Dissipativity of a system of nonlinear differential equations in the large,” in *Uspekhi Mat. Nauk*, vol. 16, 1961, p. 216.
- [33] D. C. Lewis, “Metric properties of differential equations,” *Amer. J. Math.*, vol. 71, no. 2, pp. 294–312, 1949.
- [34] Y. Nakka, R. Foust, E. Lupu, *et al.*, “A six degree-of-freedom spacecraft dynamics simulator for formation control research,” in *Astrodyn. Special Conf.*, 2018.
- [35] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory* (Studies in Applied Mathematics). Philadelphia, PA: SIAM, Jun. 1994, vol. 15, ISBN: 0-89871-334-X.
- [36] G. E. Dullerud and F. Paganini, *A Course in Robust Control Theory, A Convex Approach* (Texts in Applied Mathematics), 1st. Springer, 2000, ISBN: 978-0-387-98945-7.
- [37] H. Tsukamoto, S.-J. Chung, B. Donitz, M. Ingham, D. Mages, and Y. K. Nakka, “Neural-Rendezvous: Learning-based robust guidance and control to encounter interstellar objects,” *AIAA J. Guid. Control Dyn.*, *under review*, Aug. 2022.
- [38] B. P. S. Donitz, D. Mages, H. Tsukamoto, *et al.*, “Interstellar object accessibility and mission design,” in *IEEE Aerosp. Conf.*, 2022.
- [39] H. Tsukamoto, B. Rivière, C. Choi, A. Rahmani, and S.-J. Chung, “CART: Collision avoidance and robust tracking augmentation in learning-based motion planning for multi-agent systems,” in *IEEE Conf. Decis. Control*, *under review*, 2023.
- [40] A. Rantzer, “A dual to Lyapunov’s stability theorem,” *Syst. Control Lett.*, vol. 42, no. 3, pp. 161–168, 2001.
- [41] S.-J. Chung, S. Bandyopadhyay, I. Chang, and F. Y. Hadaegh, “Phase synchronization control of complex networks of Lagrangian systems on adaptive digraphs,” *Automatica*, vol. 49, no. 5, pp. 1148–1161, 2013.

- [42] A. J. Van der Schaft, “ $\mathcal{L}_2$ -gain analysis of nonlinear systems and nonlinear state-feedback  $\mathcal{H}_\infty$  control,” *IEEE Trans. Autom. Control*, vol. 37, no. 6, pp. 770–784, Jun. 1992.
- [43] A. Isidori and A. Astolfi, “Disturbance attenuation and  $\mathcal{H}_\infty$ -control via measurement feedback in nonlinear systems,” *IEEE Trans. Autom. Control*, vol. 37, no. 9, pp. 1283–1293, Sep. 1992.
- [44] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, “State-space solutions to standard  $\mathcal{H}_2$  and  $\mathcal{H}_\infty$  control problems,” *IEEE Trans. Autom. Control*, vol. 34, no. 8, pp. 831–847, Aug. 1989.
- [45] P. P. Khargonekar and M. A. Rotea, “Mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  control: A convex optimization approach,” *IEEE Trans. Autom. Control*, vol. 36, no. 7, pp. 824–837, Jul. 1991.
- [46] L. Xie and E. de Souza Carlos, “Robust  $\mathcal{H}_\infty$  control for linear systems with norm-bounded time-varying uncertainty,” *IEEE Trans. Autom. Control*, vol. 37, no. 8, pp. 1188–1191, Aug. 1992.
- [47] C. Scherer, P. Gahinet, and M. Chilali, “Multiobjective output-feedback control via LMI optimization,” *IEEE Trans. Autom. Control*, vol. 42, no. 7, pp. 896–911, Jul. 1997.
- [48] D. Hinrichsen and A. Pritchard, “Stochastic  $\mathcal{H}_\infty$ ,” *SIAM J. Control Optim.*, vol. 36, no. 5, pp. 1504–1538, 1998.
- [49] B.-S. Chen and W. Zhang, “Stochastic  $\mathcal{H}_2/\mathcal{H}_\infty$  control with state-dependent noise,” *IEEE Trans. Autom. Control*, vol. 49, no. 1, pp. 45–57, Jan. 2004, ISSN: 0018-9286.
- [50] I. R. Manchester and J.-J. E. Slotine, “Robust control contraction metrics: A convex approach to nonlinear state-feedback  $\mathcal{H}^\infty$  control,” *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 333–338, 2018.
- [51] Q. Pham, N. Tabareau, and J.-J. E. Slotine, “A contraction theory approach to stochastic incremental stability,” *IEEE Trans. Autom. Control*, vol. 54, no. 4, pp. 816–820, Apr. 2009.
- [52] S. Han and S.-J. Chung, *Incremental nonlinear stability analysis for stochastic systems perturbed by Lévy noise*, arXiv:2103.13338, Mar. 2021.
- [53] W. Wang and J.-J. E. Slotine, “Contraction analysis of time-delayed communications and group cooperation,” *IEEE Trans. Autom. Control*, vol. 51, no. 4, pp. 712–717, 2006.
- [54] J.-J. E. Slotine, W. Wang, and K. El Rifai, “Contraction analysis of synchronization in networks of nonlinearly coupled oscillators,” in *Int. Symp. Math. Theory Netw. Syst.*, Jul. 2004.
- [55] Q.-C. Pham and J.-J. E. Slotine, “Stable concurrent synchronization in dynamic system networks,” *Neural Netw.*, vol. 20, no. 1, pp. 62–77, 2007.

- [56] G. Russo, M. di Bernardo, and E. D. Sontag, “A contraction approach to the hierarchical analysis and design of networked systems,” *IEEE Trans. Autom. Control*, vol. 58, no. 5, pp. 1328–1331, 2013.
- [57] S.-J. Chung and J.-J. E. Slotine, “Cooperative robot control and concurrent synchronization of Lagrangian systems,” *IEEE Trans. Robot.*, vol. 25, no. 3, pp. 686–700, Jun. 2009.
- [58] V. M. Popov and R. Georgescu, *Hyperstability of Control Systems*. Berlin, Heidelberg: Springer-Verlag, 1973, ISBN: 0387063730.
- [59] Q. Pham, “Analysis of discrete and hybrid stochastic systems by nonlinear contraction theory,” in *Int. Conf. Control Automat. Robot. Vision*, Dec. 2008, pp. 1054–1059.
- [60] R. Wang, R. Tóth, and I. R. Manchester, *Discrete-time contraction-based control of nonlinear systems with parametric uncertainties using neural networks*, arXiv:2105.05432, May 2021.
- [61] I. R. Manchester and J.-J. E. Slotine, “Transverse contraction criteria for existence, stability, and robustness of a limit cycle,” *Syst. Control Lett.*, vol. 63, pp. 32–38, 2014, ISSN: 0167-6911.
- [62] F. Bullo, *Contraction Theory for Dynamical Systems*, 1.1. Kindle Direct Publishing, 2023.
- [63] S. Bonnabel and J.-J. E. Slotine, “A contraction theory-based analysis of the stability of the deterministic extended Kalman filter,” *IEEE Trans. Autom. Control*, vol. 60, no. 2, pp. 565–569, Feb. 2015.
- [64] Y. Zhao and J.-J. E. Slotine, “Discrete nonlinear observers for inertial navigation,” *Syst. Control Lett.*, vol. 54, no. 9, pp. 887–898, 2005.
- [65] F. Tan, W. Lohmiller, and J.-J. E. Slotine, “Analytical SLAM without linearization,” *Int. J. Robot. Res.*, vol. 36, no. 13-14, pp. 1554–1578, 2017.
- [66] P. M. Wensing and J.-J. E. Slotine, “Beyond convexity – Contraction and global convergence of gradient descent,” *PLOS ONE*, vol. 15, pp. 1–29, Aug. 2020.
- [67] F. Forni and R. Sepulchre, “A differential Lyapunov framework for contraction analysis,” *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 614–628, 2014.
- [68] Q. C. Pham and J.-J. E. Slotine, *Stochastic contraction in Riemannian metrics*, arXiv:1304.0340, Apr. 2013.
- [69] J. W. Simpson-Porco and F. Bullo, “Contraction theory on Riemannian manifolds,” *Syst. Control Lett.*, vol. 65, pp. 74–80, 2014, ISSN: 0167-6911.
- [70] S. Y. Shafi, Z. Aminzare, M. Arcak, and E. D. Sontag, “Spatial uniformity in diffusively-coupled systems using weighted  $L^2$  norm contractions,” in *Proc. Amer. Control Conf.*, 2013, pp. 5619–5624.

- [71] S. Bonnabel, A. Astolfi, and R. Sepulchre, “Contraction and observer design on cones,” in *IEEE Conf. Decis. Control*, 2011, pp. 7147–7151.
- [72] P. Cisneros-Velarde, S. Jafarpour, and F. Bullo, *Contraction theory for dynamical systems on Hilbert spaces*, arXiv:2010.01219, 2020.
- [73] A. Davydov, S. Jafarpour, and F. Bullo, *Non-Euclidean contraction theory via semi-inner products*, arXiv:2103.12263, 2021.
- [74] W. Lohmiller, P. Gassert, and J.-J. E. Slotine, *Notes on stable learning with piecewise-linear basis functions*, arXiv:1804.10085, 2018.
- [75] M. C. Luis D’Alto, “Incremental quadratic stability,” *Numer. Algebr. Control Optim.*, vol. 3, no. 1, pp. 175–201, 2013.
- [76] M. Margaliot, E. D. Sontag, and T. Tuller, “Contraction after small transients,” *Automatica*, vol. 67, pp. 178–184, 2016, ISSN: 0005-1098.
- [77] L. Wang, F. Forni, R. Ortega, Z. Liu, and H. Su, “Immersion and invariance stabilization of nonlinear systems via virtual and horizontal contraction,” *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4017–4022, 2017.
- [78] A. Astolfi and R. Ortega, “Immersion and invariance: A new tool for stabilization and adaptive control of nonlinear systems,” *IEEE Trans. Autom. Control*, vol. 48, no. 4, pp. 590–606, 2003.
- [79] G. Shi, X. Shi, M. O’Connell, *et al.*, “Neural lander: Stable drone landing control using learned dynamics,” in *IEEE Int. Conf. Robot. Automat.*, May 2019, pp. 9784–9790.
- [80] M. Revay, R. Wang, and I. R. Manchester, *Lipschitz bounded equilibrium networks*, arXiv:2010.01732, Oct. 2020.
- [81] M. Revay, R. Wang, and I. R. Manchester, *Recurrent equilibrium networks: Flexible dynamic models with guaranteed stability and robustness*, arXiv:2104.05942, Apr. 2021.
- [82] I. R. Manchester, M. Revay, and R. Wang, “Contraction-based methods for stable identification and robust machine learning: A tutorial,” in *IEEE Conf. Decis. Control*, 2021, pp. 2955–2962.
- [83] R. A. Freeman and P. V. Kokotovic, “Optimal nonlinear controllers for feedback linearizable systems,” in *Proc. Amer. Control Conf.*, vol. 4, Jun. 1995, pp. 2722–2726.
- [84] J. A. Primbs, V. Nevistic, and J. C. Doyle, “Nonlinear optimal control: A control Lyapunov function and receding horizon perspective,” *Asian J. Control*, vol. 1, pp. 14–24, 1999.
- [85] S. Prajna, A. Papachristodoulou, and F. Wu, “Nonlinear control synthesis by sum of squares optimization: A Lyapunov-based approach,” in *Proc. Asian Control Conf.*, vol. 1, Jul. 2004, pp. 157–165.

- [86] E. M. Aylward, P. A. Parrilo, and J.-J. E. Slotine, “Stability and robustness analysis of nonlinear systems via contraction metrics and SOS programming,” *Automatica*, vol. 44, no. 8, pp. 2163–2170, 2008, ISSN: 0005-1098.
- [87] G. Chesi, “LMI techniques for optimization over polynomials in control: A Survey,” *IEEE Trans. Autom. Control*, vol. 55, no. 11, pp. 2500–2510, Nov. 2010.
- [88] E. Sontag, “A Lyapunov-like characterization of asymptotic controllability,” *SIAM J. Control Optim.*, vol. 21, no. 3, pp. 462–471, 1983.
- [89] E. D. Sontag, “A ‘universal’ construction of Artstein’s theorem on nonlinear stabilization,” *Syst. Control Lett.*, vol. 13, no. 2, pp. 117–123, 1989.
- [90] M. Krstic, P. V. Kokotovic, and I. Kanellakopoulos, *Nonlinear and Adaptive Control Design*, 1st. USA: John Wiley & Sons, Inc., 1995, ISBN: 0471127329.
- [91] H. Deng and M. Krstic, “Stochastic nonlinear stabilization – I: A backstepping design,” *Syst. Control Lett.*, vol. 32, no. 3, pp. 143–150, 1997, ISSN: 0167-6911.
- [92] H. Deng and M. Krstic, “Output-feedback stochastic nonlinear stabilization,” *IEEE Trans. Autom. Control*, vol. 44, no. 2, pp. 328–333, Feb. 1999.
- [93] B. B. Sharma and I. N. Kar, “Contraction based adaptive control of a class of nonlinear systems,” in *Proc. Amer. Control Conf.*, 2009, pp. 808–813.
- [94] J. R. Cloutier, “State-dependent Riccati equation techniques: An overview,” in *Proc. Amer. Control Conf.*, vol. 2, 1997, pp. 932–936.
- [95] H. T. Banks, B. M. Lewis, and H. T. Tran, “Nonlinear feedback controllers and compensators: A state-dependent Riccati equation approach,” *Comput. Optim. Appl.*, vol. 37, no. 2, pp. 177–218, Jun. 2007, ISSN: 1573-2894.
- [96] T. Çimen, “Survey of state-dependent Riccati equation in nonlinear optimal feedback control synthesis,” *AIAA J. Guid. Control Dyn.*, vol. 35, no. 4, pp. 1025–1047, Jul. 2012.
- [97] I. Chang and S.-J. Chung, “Exponential stability region estimates for the state-dependent Riccati Equation controllers,” in *IEEE Conf. Decis. Control*, 2009, pp. 1974–1979.
- [98] I. R. Manchester and J.-J. E. Slotine, “Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design,” *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 3046–3053, Jun. 2017.
- [99] S. Singh, A. Majumdar, J.-J. E. Slotine, and M. Pavone, “Robust online motion planning via contraction theory and convex optimization,” in *IEEE Int. Conf. Robot. Automat.*, May 2017, pp. 5883–5890.

- [100] S. Singh, S. M. Richards, V. Sindhvani, J.-J. E. Slotine, and M. Pavone, “Learning stabilizable nonlinear dynamics with contraction-based regularization,” *Int. J. Robot. Res.*, Aug. 2020.
- [101] R. Wang, R. Tóth, and I. R. Manchester, “A comparison of LPV gain scheduling and control contraction metrics for nonlinear control,” in *3rd IFAC Workshop on LPVS*, vol. 52, 2019, pp. 44–49.
- [102] R. Wang, R. Tóth, and I. R. Manchester, *Virtual CCMs: Convex nonlinear feedback design via behavioral embedding*, arXiv:2003.08513, Mar. 2020.
- [103] I. R. Manchester, “Contracting nonlinear observers: Convex optimization and learning from data,” in *Proc. Amer. Control Conf.*, 2018, pp. 1873–1880.
- [104] B. T. Lopez and J.-J. E. Slotine, “Adaptive nonlinear control with contraction metrics,” *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 205–210, 2021.
- [105] B. T. Lopez and J.-J. E. Slotine, “Universal adaptive control of nonlinear systems,” *IEEE Control Syst. Lett.*, vol. 6, pp. 1826–1830, 2022.
- [106] N. M. Boffi, S. Tu, and J.-J. E. Slotine, *Regret bounds for adaptive nonlinear control*, arXiv:2011.13101, 2020.
- [107] B. T. Lopez, J.-J. E. Slotine, and J. P. How, “Dynamic tube MPC for nonlinear systems,” in *Proc. Amer. Control Conf.*, 2019, pp. 1655–1662.
- [108] A. Lakshmanan, A. Gahlawat, and N. Hovakimyan, “Safe feedback motion planning: A contraction theory and  $\mathcal{L}_1$ -adaptive control based approach,” in *IEEE Conf. Decis. Control*, 2020, pp. 1578–1583.
- [109] A. Gahlawat, A. Lakshmanan, L. Song, *et al.*, “Contraction  $\mathcal{L}_1$ -adaptive control using Gaussian processes,” in *Proc. Conf. Learn. Dyn. Control*, ser. Proc. Mach. Learn. Res. Vol. 144, Jun. 2021, pp. 1027–1040.
- [110] D. Sun, M. J. Khojasteh, S. Shekhar, and C. Fan, *Uncertainty-aware safe exploratory planning using Gaussian process and neural control contraction metric*, arXiv:2105.06567, 2021.
- [111] M. Johansson and A. Rantzer, “Computation of piecewise quadratic Lyapunov functions for hybrid systems,” *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 555–559, Apr. 1998.
- [112] A. Chakrabarty, K. Berntorp, and S. Di Cairano, “Learning-based parameter-adaptive reference governors,” in *Amer. Control Conf.*, 2020, pp. 956–961.
- [113] N. Gaby, F. Zhang, and X. Ye, *Lyapunov-net: A deep neural network architecture for Lyapunov function approximation*, arXiv:2109.13359, Sep. 2021.
- [114] T. A. Johansen, “Computation of Lyapunov functions for smooth nonlinear systems using convex optimization,” *Automatica*, vol. 36, no. 11, pp. 1617–1626, 2000, ISSN: 0005-1098.

- [115] P. Giesl, *Construction of Global Lyapunov Functions Using Radial Basis Functions* (Lecture Notes in Mathematics). Springer-Verlag Berlin Heidelberg, 2007, vol. 1904, ISBN: 978-3-540-69907-1.
- [116] A. Papachristodoulou and S. Prajna, “On the construction of Lyapunov functions using the sum of squares decomposition,” in *IEEE Conf. Decis. Control*, vol. 3, Dec. 2002, pp. 3482–3487.
- [117] S. M. Richards, F. Berkenkamp, and A. Krause, “The Lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems,” in *Conf. Robot. Learn.*, vol. 87, Oct. 2018, pp. 466–476.
- [118] Y.-C. Chang, N. Roohi, and S. Gao, “Neural Lyapunov control,” in *Adv. Neural Inf. Process. Syst.* 2019, pp. 3245–3254.
- [119] D. Han and D. Panagou, “Chebyshev approximation and higher order derivatives of Lyapunov functions for estimating the domain of attraction,” in *IEEE Conf. Decis. Control*, 2017, pp. 1181–1186.
- [120] Z. Wang and R. M. Jungers, “Scenario-based set invariance verification for black-box nonlinear systems,” *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 193–198, 2021.
- [121] A. Chakrabarty, C. Danielson, S. Di Cairano, and A. Raghunathan, “Active learning for estimating reachable sets for systems with unknown dynamics,” *IEEE Trans. Cybern.*, pp. 1–12, 2020.
- [122] W. Tan, “Nonlinear control analysis and synthesis using sum-of-squares programming,” Ph.D. dissertation, Univ. California, Berkeley, 2006.
- [123] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Learning-based Adaptive Control using Contraction Theory,” in *IEEE Conf. Decis. Control*, 2021, pp. 2533–2538.
- [124] H. Tsukamoto and S.-J. Chung, “Learning-based robust motion planning with guaranteed stability: A contraction theory approach,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6164–6171, 2021.
- [125] H. Robbins and S. Monro, “A stochastic approximation method,” *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 1951.
- [126] H. Tsukamoto, S.-J. Chung, J.-J. Slotine, and C. Fan, “A theoretical overview of neural contraction metrics for learning-based control with guaranteed stability,” in *IEEE Conf. Decis. Control*, 2021, pp. 2949–2954.
- [127] D. Sun, S. Jha, and C. Fan, *Learning certified control using contraction metric*, arXiv:2011.12569, Nov. 2020.
- [128] P. Zhao, A. Lakshmanan, K. Ackerman, A. Gahlawat, M. Pavone, and N. Hovakimyan, *Tube-certified trajectory tracking for nonlinear systems with robust control contraction metrics*, arXiv:2109.04453, Sep. 2021.

- [129] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi, “Tube-based robust nonlinear model predictive control,” *Int. J. Robust Nonlinear Control*, vol. 21, no. 11, pp. 1341–1353, 2011.
- [130] D. Mayne, M. Seron, and S. Raković, “Robust model predictive control of constrained linear systems with bounded disturbances,” *Automatica*, vol. 41, no. 2, pp. 219–224, 2005, ISSN: 0005-1098.
- [131] A. Bemporad and M. Morari, “Robust model predictive control: A survey,” in *Robustness in identification and control*, London: Springer London, 1999, pp. 207–226.
- [132] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, “Learning-based model predictive control: Toward safe learning in control,” *Annu. Rev. Control Robot. Auton. Syst.*, vol. 3, no. 1, pp. 269–296, 2020.
- [133] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics*. AIAA Education Series, 1999.
- [134] D. Morgan, S.-J. Chung, L. Blackmore, B. Acikmese, D. Bayard, and F. Y. Hadaegh, “Swarm-keeping strategies for spacecraft under  $J_2$  and atmospheric drag perturbations,” *AIAA J. Guid. Control Dyn.*, vol. 35, no. 5, pp. 1492–1506, 2012.
- [135] O. Nelles, *Nonlinear Dynamic System Identification*. Springer Berlin Heidelberg, 2001, pp. 547–577.
- [136] R. M. Sanner and J.-J. E. Slotine, “Gaussian networks for direct adaptive control,” *IEEE Trans. Neur. Netw.*, vol. 3, no. 6, pp. 837–863, Nov. 1992, ISSN: 1045-9227.
- [137] K. El Rifai, O. El Rifai, and K. Youcef-Toumi, “On robust adaptive switched control,” in *Proc. Amer. Control Conf.*, vol. 1, 2005, pp. 18–23.
- [138] C. Blocher, M. Saveriano, and D. Lee, “Learning stable dynamical systems using contraction theory,” in *IEEE URAI*, 2017, pp. 124–129.
- [139] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” in *Int. Conf. Learn. Representations*, 2018.
- [140] N. M. Boffi, S. Tu, N. Matni, J.-J. E. Slotine, and V. Sindhvani, *Learning stability certificates from data*, arXiv:2008.05952, 2020.
- [141] S. N. Raymond, P. J. Armitage, D. Veras, E. V. Quintana, and T. Barclay, “Implications of the interstellar object 1I/Oumuamua for planetary dynamics and planetesimal formation,” *Monthly Notices Royal Astron. Soc.*, vol. 476, no. 3, pp. 3031–3038, Feb. 2018, ISSN: 0035-8711.
- [142] K. J. Meech, R. Weryk, M. Micheli, *et al.*, “A brief visit from a red and extremely elongated interstellar asteroid,” *Nature*, vol. 552, no. 7685, pp. 378–381, Dec. 2017, ISSN: 1476-4687.

- [143] P. Guzik, M. Drahus, K. Rusek, *et al.*, “Initial characterization of interstellar comet 2I/Borisov,” *Nature Astron.*, vol. 4, 2020.
- [144] J. Castillo-Rogez and K. Meech, “Mission design study for objects on oort cloud comet Orbits,” in *Community White Papers Planet. Sci. Decadal Surv.*, 2020.
- [145] J. Riedel *et al.*, “Autonomous optical navigation (AutoNav) DS1 technology validation report,” in *NASA JPL, Caltech*, 2000.
- [146] D. Mages, D. Landau, B. Donitz, and S. Bhaskaran, “Navigation evaluation for fast interstellar object flybys,” *Acta Astronaut.*, vol. 191, pp. 359–373, 2022, ISSN: 0094-5765.
- [147] G. Shi, W. Hönig, X. Shi, Y. Yue, and S.-J. Chung, “Neural-swarm2: Planning and control of heterogeneous multirotor swarms using learned interactions,” *IEEE Trans. Robot.*, pp. 1–17, 2021.
- [148] M. O’Connell, G. Shi, X. Shi, *et al.*, “Neural-Fly enables rapid learning for agile flight in strong winds,” *Sci. Robot.*, vol. 7, no. 66, eabm6597, 2022.

## **Part I: Nonlinear Stability Analysis**

## Chapter 2

### CONTRACTION THEORY

- [1] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview,” *Annu. Rev. Control*, vol. 52, pp. 135–169, 2021, ISSN: 1367-5788.
- [2] H. Tsukamoto and S.-J. Chung, “Robust controller design for stochastic nonlinear systems via convex optimization,” *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4731–4746, 2021.

In this chapter, we present a brief review of the results from [1]–[8]. They will be extensively used to provide formal robustness and stability guarantees for a variety of systems in the subsequent chapters, simplifying and generalizing Lyapunov theory.

#### 2.1 Fundamentals

Consider the following smooth non-autonomous (i.e., time-varying) nonlinear system:

$$\dot{x}(t) = f(x(t), t) \tag{2.1}$$

where  $t \in \mathbb{R}_{\geq 0}$  is time,  $x : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  the system state, and  $f : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  is a smooth function. Note that the smoothness of  $f(x, t)$  guarantees the existence and uniqueness of the solution to (2.1) for a given  $x(0) = x_0$  at least locally [9, pp. 88-95].

**Definition 2.1.** *A differential displacement,  $\delta x$ , is defined as an infinitesimal displacement at a fixed time as used in the calculus of variation [10, p. 107], and (2.1) yields the following differential dynamics:*

$$\delta \dot{x}(t) = \frac{\partial f}{\partial x}(x(t), t) \delta x(t) \tag{2.2}$$

where  $f(x(t), t)$  is given in (2.1).

Let us first present a special case of the comparison lemma [9, pp. 102-103, pp. 350-353] to be used extensively throughout this thesis.

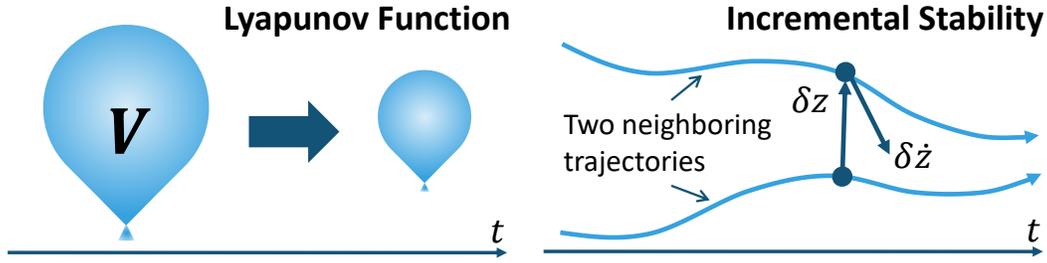


Figure 2.1: Illustration of contraction theory, where  $V$  is a differential Lyapunov function  $V = \delta x^\top M(x, t)\delta x$ ,  $\delta z = \Theta(x, t)\delta x$ , and  $M(x, t) = \Theta(x, t)\Theta(x, t)^\top > 0$  defines a contraction metric (see Theorem 2.1).

**Lemma 2.1.** *Suppose that a continuously differentiable function  $v \in \mathbb{R}_{\geq 0} \mapsto \mathbb{R}$  satisfies the following differential inequality:*

$$\dot{v}(t) \leq -\gamma v(t) + c, \quad v(0) = v_0, \quad \forall t \in \mathbb{R}_{\geq 0}$$

where  $\gamma \in \mathbb{R}_{> 0}$ ,  $c \in \mathbb{R}$ , and  $v_0 \in \mathbb{R}$ . Then we have

$$v(t) \leq v_0 e^{-\gamma t} + \frac{c}{\gamma}(1 - e^{-\gamma t}), \quad \forall t \in \mathbb{R}_{\geq 0}.$$

*Proof.* See [9, pp. 659-660]. □

### 2.1.I Contraction Theory and Contraction Metric

In Lyapunov theory, nonlinear stability of (2.1) is studied by constructing a Lyapunov function  $V(x, t)$ , one example of which is  $V = x^\top P(x, t)x$ . However, finding  $V(x, t)$  for general nonlinear systems is challenging as  $V(x, t)$  can be any scalar function of  $x$  (e.g., a candidate  $V(x, t)$  can be obtained by solving a PDE [9, p. 211]). In contrast, as summarized in Table 1.1, contraction theory uses a differential Lyapunov function that is always a quadratic function of  $\delta x$ , i.e.,  $V(x, \delta x, t) = \delta x^\top M(x, t)\delta x$ , thereby characterizing a necessary and sufficient condition for incremental exponential convergence of the multiple nonlinear system trajectories to one single trajectory. Thus, the problem of finding  $V$  for stability analysis boils down to finding a finite-dimensional positive-definite matrix  $M$ , as illustrated in Figure 2.1 [1]. These properties to be derived in Theorem 2.1, which hold both for autonomous (i.e. time-invariant) and non-autonomous systems, epitomize significant methodological simplifications of stability analysis in contraction theory.

**Theorem 2.1.** *If there exists a uniformly positive definite matrix given as  $M(x, t) = \Theta(x, t)^\top \Theta(x, t) > 0$ ,  $\forall x, t$ , where  $\Theta(x, t)$  defines a smooth coordinate transforma-*

tion of  $\delta x$ , i.e.,  $\delta z = \Theta(x, t)\delta x$ , s.t. either of the following equivalent conditions holds for  $\exists \alpha \in \mathbb{R}_{>0}$ ,  $\forall x, t$ :

$$\lambda_{\max}(F(x, t)) = \lambda_{\max}\left(\left(\dot{\Theta} + \Theta \frac{\partial f}{\partial x}\right) \Theta^{-1}\right) \leq -\alpha \quad (2.3)$$

$$\dot{M} + M \frac{\partial f}{\partial x} + \frac{\partial f^\top}{\partial x} M \leq -2\alpha M \quad (2.4)$$

where the arguments  $(x, t)$  of  $M(x, t)$  and  $\Theta(x, t)$  are omitted for notational simplicity, then all the solution trajectories of (2.1) converge to a single trajectory exponentially fast regardless of their initial conditions (i.e., contracting, see Definition 2.3), with an exponential convergence rate  $\alpha$ . The converse also holds.

*Proof.* The proof of this theorem can be found in [1], but here we emphasize the use of the comparison lemma given in Lemma 2.1. Taking the time-derivative of a differential Lyapunov function of  $\delta x$  (or  $\delta z$ ),  $V = \delta z^\top \delta z = \delta x^\top M(x, t)\delta x$ , using the differential dynamics (2.2), we have

$$\begin{aligned} \dot{V}(x, \delta x, t) &= 2\delta z^\top F \delta z = \delta x^\top \left( \dot{M} + \frac{\partial f^\top}{\partial x} M + M \frac{\partial f}{\partial x} \right) \delta x \\ &\leq -2\alpha \delta z^\top \delta z = -2\alpha \delta x^\top M(x, t)\delta x \end{aligned}$$

where the conditions (2.3) and (2.4) are used with the generalized Jacobian  $F$  in (2.3) obtained from  $\delta \dot{z} = \dot{\Theta}\delta x + \Theta\dot{\delta x} = F\delta z$ . We get  $d\|\delta z\|/dt \leq -\alpha\|\delta z\|$  by  $d\|\delta z\|^2/dt = 2\|\delta z\|d\|\delta z\|/dt$ , which then yields  $\|\delta z(t)\| \leq \|\delta z_0\|e^{-\alpha t}$  by the comparison lemma of Lemma 2.1. Hence, any infinitesimal length  $\|\delta z(t)\|$  and  $\|\delta x(t)\|$ , as well as  $\delta z$  and  $\delta x$ , tend to zero exponentially fast. By path integration (see Definition 2.2 and Theorem 2.3), this immediately implies that the length of any finite path converges exponentially to zero from any initial conditions.

Conversely, consider an exponentially convergent system, which implies the following for  $\exists \beta > 0$  and  $\exists k \geq 1$ :

$$\|\delta x(t)\|^2 \leq -k\|\delta x(0)\|^2 e^{-2\beta t} \quad (2.5)$$

and define a matrix-valued function  $\Xi(x(t), t) \in \mathbb{R}^{n \times n}$  (not necessarily  $\Xi > 0$ ) as

$$\dot{\Xi} = -2\beta\Xi - \Xi \frac{\partial f}{\partial x} - \frac{\partial f^\top}{\partial x} \Xi, \quad \Xi(x(0), 0) = kI. \quad (2.6)$$

Note that, for  $V = \delta x^\top \Xi \delta x$ , (2.6) gives  $\dot{V} = -2\beta V$ , resulting in  $V = -k\|\delta x(0)\|^2 e^{-2\beta t}$ . Substituting this into (2.5) yields

$$\|\delta x(t)\|^2 = \delta x(t)^\top \delta x(t) \leq V = \delta x(t)^\top \Xi(x(t), t)\delta x(t) \quad (2.7)$$

which indeed implies that  $\Xi \geq I > 0$  as (2.7) holds for any  $\delta x$ . Thus,  $\Xi$  satisfies the contraction condition (2.4), i.e.,  $\Xi$  defines a contraction metric (see Definition 2.3) [1].  $\square$

**Remark 2.1.** *Since  $M$  of Theorem 2.1 is positive definite, i.e.,  $v^\top M v = \|\Theta v\|^2 \geq 0$ ,  $\forall v \in \mathbb{R}^n$ , and  $\|\Theta v\|^2 = 0$  if and only if  $v = 0$ , the equation  $\Theta v = 0$  only has a trivial solution  $v = 0$ . This implies that  $\Theta$  is always non-singular (i.e.,  $\Theta(x, t)^{-1}$  always exists).*

**Definition 2.2.** *Let  $\xi_0(t)$  and  $\xi_1(t)$  denote some solution trajectories of (2.1). We say that (2.1) is incrementally exponentially stable if  $\exists C, \alpha > 0$  s.t. the following holds [11]:*

$$\|\xi_1(t) - \xi_0(t)\| \leq C e^{-\alpha t} \|\xi_0(0) - \xi_1(0)\|$$

for any  $\xi_0(t)$  and  $\xi_1(t)$ . Note that, since we have  $\|\xi_1(t) - \xi_0(t)\| = \left\| \int_{\xi_0}^{\xi_1} \delta x \right\|$  (see Theorem 2.3), Theorem 2.1 implies incremental stability of the system (2.1).

**Definition 2.3.** *The system (2.1) satisfying the conditions in Theorem 2.1 is said to be contracting, and a uniformly positive definite matrix  $M$  that satisfies (2.4) defines a contraction metric. As to be discussed in Theorem 2.3 of Sec. 2.2, a contracting system is incrementally exponentially stable in the sense of Definition 2.2.*

**Example 2.1.** *One of the distinct features of contraction theory in Theorem 2.1 is incremental stability with exponential convergence. Consider the example given in [12]:*

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -1 & x_1 \\ -x_1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \quad (2.8)$$

A Lyapunov function  $V = \|x\|^2/2$  for (2.8), where  $x = [x_1, x_2]^\top$ , yields  $\dot{V} \leq -2V$ . Thus, (2.8) is exponentially stable with respect to  $x = 0$ . The differential dynamics of (2.8) is given as

$$\frac{d}{dt} \begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix} = \begin{bmatrix} -1 + x_2 & x_1 \\ -2x_1 & -1 \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix}, \quad (2.9)$$

and the contraction condition (2.4) for (2.9) can no longer be proven by  $V = \|\delta x\|^2/2$ , due to the lack of the skew-symmetric property of (2.8) in (2.9). This difficulty illustrates the difference between Lyapunov theory and contraction theory, where the former considers stability of (2.8) with respect to the equilibrium point, while the latter analyzes exponential convergence of any couple of trajectories in (2.8) with respect to each other (i.e., incremental stability in Definition 2.2) [1], [12].

**Example 2.2.** *Contraction defined by Theorem 2.1 guarantees incremental stability of their solution trajectories but does not require the existence of stable fixed points. Let us consider the following system for  $x : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}$ :*

$$\dot{x} = -x + e^t. \quad (2.10)$$

*Using the constraint (2.4) of Theorem 2.1, we can easily verify that (2.10) is contracting as  $M = \mathbf{I}$  defines its contraction metric with the contraction rate  $\alpha = 1$ . However, since (2.10) has  $x(t) = e^t/2 + (x(0) - 1/2)e^{-t}$  as its unique solution, it is not stable with respect to any fixed point.*

**Example 2.3.** *Consider a Linear Time-Invariant (LTI) system,  $\dot{x} = f(x) = Ax$ . Lyapunov theory states that the origin is globally exponentially stable if and only if there exists a constant positive-definite matrix  $P \in \mathbb{R}^{n \times n}$  s.t. [13, pp. 67-68]*

$$\exists \epsilon > 0 \text{ s.t. } PA + A^\top P \leq -\epsilon \mathbf{I}. \quad (2.11)$$

*Now, let  $\bar{p} = \|P\|$ . Since  $-\mathbf{I} \leq -P/\bar{p}$ , (2.11) implies that  $PA + A^\top P \leq -(\epsilon/\bar{p})P$ , which shows that  $M = P$  with  $\alpha = \epsilon/(2\bar{p})$  satisfies (2.4) due to the relation  $\partial f/\partial x = A$ .*

*The contraction condition (2.4) can thus be viewed as a generalization of the Lyapunov stability condition (2.11) (see the generalized Krasovskii's theorem [14, pp. 83-86]) in a nonlinear non-autonomous system setting, expressed in the differential formulation that permits a non-constant metric and pure differential coordinate change [1]. Furthermore, if  $f(x, t) = A(t)x$  and  $M = \mathbf{I}$ , (2.4) results in  $A(t) + A(t)^\top \leq -2\alpha \mathbf{I}$  (i.e., all the eigenvalues of the symmetric matrix  $A(t) + A(t)^\top$  remain strictly in the left-half complex plane), which is a known sufficient condition for stability of Linear Time-Varying (LTV) systems [14, pp. 114-115].*

**Example 2.4.** *Most of the learning-based techniques involving neural networks are based on optimizing their hyperparameters by gradient descent [15]. Contraction theory provides a generalized view on the analysis of such continuous-time gradient-based optimization algorithms [16].*

*Let us consider a twice differentiable scalar output function  $f : \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}$ , a matrix-valued function  $M : \mathbb{R}^n \mapsto \mathbb{R}^{n \times n}$  with  $M(x) > 0, \forall x \in \mathbb{R}^n$ , and the following natural gradient system [17]:*

$$\dot{x} = h(x, t) = -M(x)^{-1} \nabla_x f(x, t). \quad (2.12)$$

Then,  $f$  is geodesically  $\alpha$ -strongly convex for each  $t$  in the metric defined by  $M(x)$  (i.e.,  $H(x) \geq \alpha M(x)$  with  $H(x)$  being the Riemannian Hessian matrix of  $f$  with respect to  $M$  [18]), if and only if (2.12) is contracting with rate  $\alpha$  in the metric defined by  $M$  as in (2.4) of Theorem 2.1, where  $A = \partial h / \partial x$ . More specifically, the Riemannian Hessian verifies  $H(x) = -(\dot{M} + MA + A^\top M)/2$ . See [16] for details.

**Remark 2.2.** Theorem 2.1 can be applied to other vector norms of  $\|\delta z\|_p$  with, e.g.,  $p = 1$  or  $p = \infty$  [1]. It can also be shown that for a contracting autonomous system of the form  $\dot{x} = f(x)$ , all trajectories converge to an equilibrium point exponentially fast.

### 2.1.II Partial Contraction

Although satisfying the condition (2.4) of Theorem 2.1 guarantees exponential convergence of any couple of trajectories in (2.1), proving their incremental stability with respect to a subset of these trajectories possessing a specific property could be sufficient for some cases [6], [8], [19], [20], leading to the concept of partial contraction [3].

**Theorem 2.2.** Consider the following nonlinear system with the state  $x \in \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  and the auxiliary or virtual system with the state  $q \in \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$ :

$$\dot{x}(t) = g(x(t), x(t), t) \tag{2.13}$$

$$\dot{q}(t) = g(q(t), x(t), t) \tag{2.14}$$

where  $g : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$  is a smooth function. Suppose that (2.14) is contracting with respect to  $q$ . If a particular solution of (2.14) verifies a smooth specific property, then all trajectories of (2.13) verify this property exponentially.

*Proof.* The theorem statement follows from Theorem 2.1 and the fact that  $q = x$  and a trajectory with the specific property are particular solutions of (2.14) (see [3] for details).  $\square$

The importance of this theorem lies in the fact that we can analyze contraction of some specific parts of the system (2.13) while treating the rest as a function of the time-varying parameter  $x(t)$ . Strictly speaking, the system (2.13) is said to be partially contracting, but we will not distinguish partial contraction from contraction of Definition 2.3 in this thesis for simplicity. Instead, we will use the variable  $q$  when referring to partial contraction of Theorem 2.2.

Examples of a trajectory with a specific property include the target trajectory  $x_d$  of feedback control in (3.3), or the trajectory of actual dynamics  $x$  for state estimation in (4.3) and (4.4) to be discussed in the subsequent chapters. Note that contraction can be regarded as a particular case of partial contraction.

**Example 2.5.** *Let us illustrate the role of partial contraction using the following nonlinear system [12]:*

$$\dot{x} = -D(x)x + u, \quad \dot{x}_d = -D(x_d)x_d$$

where  $x$  is the system state,  $x_d$  is the target state,  $u$  is the control input designed as  $u = -K(x)(x - x_d) + (D(x) - D(x_d))x_d$ , and  $D(x) + K(x) > 0$ . We could define a virtual system, which has  $q = x$  and  $q = x_d$  as its particular solutions, as follows:

$$\dot{q} = -(D(x) + K(x))(q - x_d) - D(x_d)x_d. \quad (2.15)$$

Since we have  $\delta\dot{q} = -(D(x) + K(x))\delta q$  and  $D(x) + K(x) > 0$ , (2.15) is contracting with  $M = I$  in (2.4). However, if we consider the following virtual system:

$$\dot{q} = -(D(q) + K(q))(q - x_d) - D(x_d)x_d \quad (2.16)$$

which also has  $q = x$  and  $q = x_d$  as its particular solutions, proving contraction is no longer straightforward because of the terms  $\partial D/\partial q_i$  and  $\partial K/\partial q_i$  in the differential dynamics of (2.16). This is due to the fact that, in contrast to (2.15), (2.16) has particular solutions nonlinear in  $q$  in addition to  $q = x$  and  $q = x_d$ , and the condition (2.4) becomes more involved for (2.16) as it is for guaranteeing exponential convergence of any couple of these particular solution trajectories [12].

**Example 2.6.** *As one of the key applications of partial contraction given in Theorem 2.2, let us consider the following closed-loop Lagrangian system [14, p. 392]:*

$$\mathcal{H}(q)\ddot{q} + C(q, \dot{q})\dot{q} + \mathcal{G}(q) = u(q, \dot{q}, t) \quad (2.17)$$

$$u(q, \dot{q}, t) = \mathcal{H}(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + \mathcal{G}(q) - \mathcal{K}(t)(\dot{q} - \dot{q}_r) \quad (2.18)$$

where  $q, \dot{q} \in \mathbb{R}^n$ ,  $\dot{q}_r = \dot{q}_d(t) - \Lambda(t)(q - q_d(t))$ ,  $\mathcal{H} : \mathbb{R}^n \mapsto \mathbb{R}^{n \times n}$ ,  $C : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^{n \times n}$ ,  $\mathcal{G} : \mathbb{R}^n \mapsto \mathbb{R}^n$ ,  $\mathcal{K} : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{n \times n}$ ,  $\Lambda : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{n \times n}$ , and  $(q_d, \dot{q}_d)$  is the target trajectory of the state  $(q, \dot{q})$ . Note that  $\mathcal{K}, \Lambda > 0$  are control gain matrices (design parameters), and  $\dot{\mathcal{H}} - 2C$  is skew-symmetric with  $\mathcal{H} > 0$  by construction.

By comparing with (2.17) and (2.18), we define the following virtual observer-like system of  $q$  (not  $\mathbf{q}$ ) that has  $q = \dot{q}$  and  $q = \dot{q}_r$  as its particular solutions:

$$\mathcal{H}(q)\dot{q} + C(q, \dot{q})q + \mathcal{K}(t)(q - \dot{q}) + \mathcal{G}(q) = u(q, \dot{q}, t) \quad (2.19)$$

which gives  $\mathcal{H}(\mathbf{q})\delta\dot{\mathbf{q}} + (C(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{K}(t))\delta\mathbf{q} = 0$ . We thus have that

$$\frac{d}{dt}(\delta\mathbf{q}^\top \mathcal{H}(\mathbf{q})\delta\mathbf{q}) = \delta\mathbf{q}^\top (\dot{\mathcal{H}} - 2C - 2\mathcal{K})\delta\mathbf{q} = -2\delta\mathbf{q}^\top \mathcal{K}\delta\mathbf{q} \quad (2.20)$$

where the skew-symmetry of  $\dot{\mathcal{H}} - 2C$  is used to obtain the second equality. Since  $\mathcal{K} > 0$ , (2.20) indicates that the virtual system (2.19) is partially contracting in  $q$  with  $\mathcal{H}$  defining its contraction metric. Contraction of the full state  $(\mathbf{q}, \dot{\mathbf{q}})$  will be discussed in Example 2.7.

Note that if we treat the arguments  $(\mathbf{q}, \dot{\mathbf{q}})$  of  $\mathcal{H}$  and  $C$  also as the virtual state  $q$ , we end up having additional terms such as  $\partial\mathcal{H}/\partial q_i$ , which makes proving contraction analytically more involved as in Example 2.5.

As can be seen from Examples 2.5 and 2.6, the role of partial contraction in theorem 2.2 is to provide some insight on stability even for cases where it is difficult to prove contraction for all solution trajectories as in Theorem 2.1. Although finding a contraction metric analytically for general nonlinear systems is challenging, we will see in Chapter 3 and Chapter 4 that the convex nature of the contraction condition (2.4) helps us find it numerically.

## 2.2 Path-Length Integral and Robust Incremental Stability Analysis

Theorem 2.1 can also be proven by using the transformed squared length integrated over two arbitrary solutions of (2.1) [1], [5]–[8], which enables formalizing its connection to incremental stability discussed in Definition 2.3. Note that the integral forms (2.21) and (2.22) to be given in Theorem 2.3 are useful for handling perturbed systems with external disturbances as shall be seen in Theorems 2.4 – 2.8.

**Theorem 2.3.** *Let  $\xi_0$  and  $\xi_1$  be the two arbitrary solutions of (2.1), and define the transformed squared length with  $M(x, t)$  of Theorem 2.1 as follows:*

$$V_{sl}(x, \delta x, t) = \int_{\xi_0}^{\xi_1} \|\delta z\|^2 = \int_0^1 \frac{\partial x^\top}{\partial \mu} M(x, t) \frac{\partial x}{\partial \mu} d\mu \quad (2.21)$$

where  $x$  is a smooth path parameterized as  $x(\mu = 0, t) = \xi_0(t)$  and  $x(\mu = 1, t) = \xi_1(t)$  by  $\mu \in [0, 1]$ . Also, define the path integral with the transformation  $\Theta(x, t)$  for  $M(x, t) = \Theta(x, t)^\top \Theta(x, t)$  as follows:

$$V_\ell(x, \delta x, t) = \int_{\xi_0}^{\xi_1} \|\delta z\| = \int_{\xi_0}^{\xi_1} \|\Theta(x, t)\delta x\|. \quad (2.22)$$

Then (2.21) and (2.22) are related as

$$\|\xi_1 - \xi_0\| = \left\| \int_{\xi_0}^{\xi_1} \delta x \right\| \leq \frac{V_\ell}{\sqrt{\underline{m}}} \leq \sqrt{\frac{V_{s\ell}}{\underline{m}}} \quad (2.23)$$

where  $M(x, t) \geq \underline{m}I$ ,  $\forall x, t$  for  $\exists \underline{m} \in \mathbb{R}_{>0}$ , and Theorem 2.1 can also be proven by using (2.21) and (2.22) as a Lyapunov-like function, resulting in incremental exponential stability of the system (2.1) (see Definition 2.2). Note that the shortest path integral  $V_\ell$  of (2.22) with a parameterized state  $x$  (i.e.,  $\inf V_\ell = \sqrt{\inf V_{s\ell}}$ ) defines the Riemannian distance and the path integral of a minimizing geodesic [21].

*Proof.* Using  $M(x, t) \geq \underline{m}I$  which gives  $\sqrt{\underline{m}}\|\xi_1 - \xi_0\| \leq V_\ell$ , we have  $\|\xi_1 - \xi_0\| = \|\int_{\xi_0}^{\xi_1} \delta x\| \leq V_\ell/\sqrt{\underline{m}}$  of (2.23). The inequality  $V_\ell \leq \sqrt{V_{s\ell}}$  of (2.23) can be proven by applying the Cauchy–Schwarz inequality [22, p. 316] to the functions  $\psi_1(\mu) = \|\Theta(x, t)(\partial x/\partial \mu)\|$  and  $\psi_2(\mu) = 1$ .

We can also see that computing  $\dot{V}_{s\ell}$  of (2.21) using the differential dynamics of (2.2) yields

$$\dot{V}_{s\ell} = \int_0^1 \frac{\partial x^\top}{\partial \mu} \left( \dot{M} + M \frac{\partial f}{\partial x} + \frac{\partial f^\top}{\partial x} M \right) \frac{\partial x}{\partial \mu} d\mu$$

to have  $\dot{V}_{s\ell} \leq -2\alpha V_{s\ell}$  and  $\dot{V}_\ell \leq -\alpha V_\ell$  by the contraction conditions (2.3) and (2.4). Since these hold for any  $\xi_0$  and  $\xi_1$ , the incremental exponential stability results in Theorem 2.1 follow from the comparison lemma of Lemma 2.1 (see also [6]–[8], and [21], [23] for the discussion on the geodesic).  $\square$

## 2.2.I Deterministic Perturbation

Let  $\xi_0(t)$  be a solution of the system (2.1). It is now perturbed as

$$\dot{x} = f(x, t) + d(x, t) \quad (2.24)$$

and let  $\xi_1(t)$  denote a trajectory of (2.24). Then a virtual system of a smooth path  $q(\mu, t)$  parameterized by  $\mu \in [0, 1]$ , which has  $q(\mu = 0, t) = \xi_0$  and  $q(\mu = 1, t) = \xi_1$  as its particular solutions is given as follows:

$$\dot{q}(\mu, t) = f(q(\mu, t), t) + d_\mu(\mu, \xi_1, t) \quad (2.25)$$

where  $d_\mu(\mu, \xi_1, t) = \mu d(\xi_1, t)$ . Since contraction means exponential convergence, a contracting system exhibits a superior property of robustness [1], [5].

**Theorem 2.4.** *If the system (2.1) satisfies (2.3) and (2.4) of Theorem 2.1 (i.e., the system (2.1) is contracting), then the path integral  $V_\ell(q, \delta q, t) = \int_{\xi_0}^{\xi_1} \|\Theta(q, t)\delta q\|$  of (2.22), where  $\xi_0$  is a solution of the contracting system (2.1),  $\xi_1$  is a solution of the perturbed system (2.24), and  $q$  is the virtual state of (2.25), exponentially converges to a bounded error ball as long as  $\Theta d \in \mathcal{L}_\infty$  (i.e.,  $\sup_{x,t} \|\Theta d\| < \infty$ ). Specifically, if  $\exists \underline{m}, \bar{m} \in \mathbb{R}_{>0}$  and  $\exists \bar{d} \in \mathbb{R}_{\geq 0}$  s.t.  $\bar{d} = \sup_{x,t} \|d(x, t)\|$  and*

$$\underline{m}\mathbf{I} \leq M(x, t) \leq \bar{m}\mathbf{I}, \quad \forall x, t, \quad (2.26)$$

then we have the following relation:

$$\|\xi_1(t) - \xi_0(t)\| \leq \frac{V_\ell(0)}{\sqrt{\underline{m}}} e^{-\alpha t} + \frac{\bar{d}}{\alpha} \sqrt{\frac{\bar{m}}{\underline{m}}} (1 - e^{-\alpha t}) \quad (2.27)$$

where  $V_\ell(t) = V_\ell(q(t), \delta q(t), t)$  for notational simplicity.

*Proof.* Using the contraction condition (2.4), we have for  $M = \Theta^\top \Theta$  given in Theorem 2.1 that

$$\begin{aligned} \frac{d}{dt} \|\Theta(q, t)\partial_\mu q\| &= (2\|\Theta(q, t)\partial_\mu q\|)^{-1} \frac{d}{dt} \partial_\mu q^\top M(q, t)\partial_\mu q \\ &\leq -\alpha \|\Theta(q, t)\partial_\mu q\| + \|\Theta(q, t)\partial_\mu d_\mu\| \end{aligned}$$

where  $\partial_\mu q = \partial q / \partial \mu$  and  $\partial_\mu d_\mu = \partial d_\mu / \partial \mu = d(\xi_1, t)$ . Taking the integral with respect to  $\mu$  gives

$$\frac{d}{dt} \int_0^1 \|\Theta \partial_\mu q\| d\mu \leq \int_0^1 -\alpha \|\Theta \partial_\mu q\| + \|\Theta d(\xi_1, t)\| d\mu$$

which implies  $\dot{V}_\ell(t) \leq -\alpha V_\ell(t) + \sup_{q, \xi_1, t} \|\Theta(q, t)d(\xi_1, t)\|$  for  $V_\ell$  in (2.22) of Theorem 2.3. Thus, applying the comparison lemma (see Lemma 2.1) results in

$$V_\ell(t) \leq e^{-\alpha t} V_\ell(0) + \sup_{q, \xi_1, t} \|\Theta(q, t)d(\xi_1, t)\| \frac{1 - e^{-\alpha t}}{\alpha}. \quad (2.28)$$

By using  $\xi_1 - \xi_0 = \int_{\xi_0}^{\xi_1} \delta x$  and  $\|\xi_1 - \xi_0\| = \left\| \int_{\xi_0}^{\xi_1} \delta x \right\| \leq \int_{\xi_0}^{\xi_1} \|\delta x\| \leq \int_{\xi_0}^{\xi_1} \|\Theta^{-1}\| \|\delta z\|$ , we obtain  $\sqrt{\inf_t \lambda_{\min}(M)} \|\xi_1 - \xi_0\| \leq V_\ell = \int_{\xi_0}^{\xi_1} \|\delta z\|$ , and thus

$$\|\xi_1 - \xi_0\| \leq \frac{e^{-\alpha t} V_\ell(0)}{\sqrt{\inf_t \lambda_{\min}(M)}} + \frac{\sup_{q, \xi_1, t} \|\Theta d\|}{\sqrt{\inf_t \lambda_{\min}(M)}} \frac{1 - e^{-\alpha t}}{\alpha}.$$

This relation with the bounds on  $M$  and  $d$  gives (2.27).  $\square$

## 2.2.II Stochastic Perturbation

Next, consider the following dynamical system modeled by the Itô stochastic differential equation:

$$dx = f(x, t)dt + G(x, t)d\mathcal{W}(t) \quad (2.29)$$

where  $G : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n \times w}$  is a matrix-valued function and  $\mathcal{W} : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^w$  is a  $w$ -dimensional Wiener process [24, p. 100] (see also [24, p. xii] for the notations used). For the sake of existence and uniqueness of the solution, we assume in (2.29) that

$$\begin{aligned} \exists L \in \mathbb{R}_{\geq 0} \text{ s.t. } \|f(x, t) - f(x', t)\| + \|G(x, t) - G(x', t)\|_F \\ \leq L\|x - x'\|, \quad \forall t \in \mathbb{R}_{\geq 0}, \forall x, x' \in \mathbb{R}^n \end{aligned} \quad (2.30)$$

$$\exists \bar{L} \in \mathbb{R}_{\geq 0} \text{ s.t. } \|f(x, t)\|^2 + \|G(x, t)\|_F^2 \leq \bar{L}(1 + \|x\|^2), \quad \forall t \in \mathbb{R}_{\geq 0}, \forall x \in \mathbb{R}^n. \quad (2.31)$$

In order to analyze the incremental stability property of (2.29) as in Theorem 2.3, we consider two trajectories  $\xi_0(t)$  and  $\xi_1(t)$  of stochastic nonlinear systems with Gaussian white noise, driven by two independent Wiener processes  $\mathcal{W}_0(t)$  and  $\mathcal{W}_1(t)$ :

$$d\xi_i = f(\xi_i, t)dt + G_i(\xi_i, t)d\mathcal{W}_i(t), \quad i = 0, 1. \quad (2.32)$$

One can show that (2.29) has a unique solution  $x(t)$  which is continuous with probability one under the conditions (2.30) and (2.31) (see [24, p. 105] and [4, 7]), leading to the following lemma as in the comparison lemma of Lemma 2.1.

**Lemma 2.2.** *Suppose that  $V_{s\ell}$  of (2.21) satisfies the following inequality:*

$$\mathcal{L}V_{s\ell} \leq -\gamma V_{s\ell} + c \quad (2.33)$$

where  $\gamma \in \mathbb{R}_{>0}$ ,  $c \in \mathbb{R}_{\geq 0}$ , and  $\mathcal{L}$  denotes the infinitesimal differential generator of the Itô process given in [25, p. 15]. Then we have the following bound [4]:

$$\mathbb{E} [\|\xi_1(t) - \xi_0(t)\|^2] \leq \frac{1}{\underline{m}} \left( \mathbb{E}[V_{s\ell}(0)]e^{-\gamma t} + \frac{c}{\gamma} \right) \quad (2.34)$$

where  $V_{s\ell}(0) = V_{s\ell}(x(0), \delta x(0), 0)$  for  $V_{s\ell}$  in (2.21),  $\underline{m}$  is given in (2.26),  $\xi_0$  and  $\xi_1$  are given in (2.32), and  $\mathbb{E}$  denotes the expected value operator. Furthermore, the probability that  $\|\xi_1 - \xi_0\|$  is greater than or equal to  $\varepsilon \in \mathbb{R}_{>0}$  is given as

$$\mathbb{P} [\|\xi_1(t) - \xi_0(t)\| \geq \varepsilon] \leq \frac{1}{\varepsilon^2 \underline{m}} \left( \mathbb{E}[V_{s\ell}(0)]e^{-\gamma t} + \frac{c}{\gamma} \right). \quad (2.35)$$

*Proof.* The bound (2.34) follows from Theorem 2 of [4] (see also [6]–[8], [26], [27] and [25, p. 10] (Dynkin’s formula)). The probability tracking error bound (2.35) then follows from Markov’s inequality [28, pp. 311-312].  $\square$

**Remark 2.3.** *Although Lemma 2.2 considers the second moment of  $\|\xi_1(t) - \xi_0(t)\|$ , i.e.,  $\mathbb{E}[\|\xi_1(t) - \xi_0(t)\|^2]$ , it can be readily generalized to the  $p$ -th moment of  $\|\xi_1(t) - \xi_0(t)\|$ , i.e.,  $\mathbb{E}[\|\xi_1(t) - \xi_0(t)\|^p]$ , applying the Lyapunov-based technique proposed in [29].*

A virtual system of a smooth path  $q(\mu, t)$  parameterized by  $\mu \in [0, 1]$ , which has  $q(\mu = 0, t) = \xi_0$  and  $q(\mu = 1, t) = \xi_1$  of (2.32) as its particular solutions, is given as follows:

$$dq(\mu, t) = f(q(\mu, t), t)dt + G(\mu, \xi_0, \xi_1, t)d\mathcal{W}(t) \quad (2.36)$$

where  $G(\mu, \xi_0, \xi_1, t) = [(1 - \mu)G_0(\xi_0, t), \mu G_1(\xi_1, t)]$  and  $\mathcal{W} = [\mathcal{W}_0^\top, \mathcal{W}_1^\top]^\top$ . As a consequence of Lemma 2.2, showing stochastic incremental stability between  $\xi_0$  and  $\xi_1$  of (2.32) reduces to proving the relation (2.33), similar to the deterministic case in Theorems 2.1 and 2.4.

**Theorem 2.5.** *Suppose that  $\exists \bar{g}_0 \in \mathbb{R}_{\geq 0}$  and  $\exists \bar{g}_1 \in \mathbb{R}_{\geq 0}$  s.t.  $\sup_{x,t} \|G_1(x, t)\|_F = \bar{g}_0$  and  $\sup_{x,t} \|G_1(x, t)\|_F = \bar{g}_1$  in (2.32). Suppose also that there exists  $M(x, t) > 0$ ,  $\forall x, t$ , s.t.  $M_{x_i} = \partial M / \partial x_i$  is Lipschitz with respect to  $x$  for all  $i = 1, \dots, n$ , i.e.,  $\exists L_m \in \mathbb{R}_{\geq 0}$  s.t.*

$$\|M_{x_i}(x, t) - M_{x_i}(x', t)\| \leq L_m \|x - x'\|, \quad \forall x, x', t, i. \quad (2.37)$$

*Also, suppose that  $M$  of (2.37) satisfies (2.26) and (2.4) with its right-hand side replaced by  $-2\alpha M - \alpha_s \mathbf{I}$  for  $\alpha_s = L_m(\bar{g}_0^2 + \bar{g}_1^2)(\alpha_G + 1/2)$ , i.e.,*

$$\dot{M} + M \frac{\partial f}{\partial x} + \frac{\partial f^\top}{\partial x} M \leq -2\alpha M - \alpha_s \mathbf{I} \quad (2.38)$$

*where  $\alpha_G \in \mathbb{R}_{> 0}$  is an arbitrary constant (see (2.42)). Then, the following error bound of incremental stability holds:*

$$\mathbb{E}[\|\xi_1(t) - \xi_0(t)\|^2] \leq \frac{\mathbb{E}[V_{s\ell}(0)]}{\underline{m}} e^{-2\alpha t} + \frac{C \bar{m}}{2\alpha \underline{m}} \quad (2.39)$$

*where  $\xi_0$  and  $\xi_1$  are the trajectories given in (2.32),  $V_{s\ell}(t) = V_{s\ell}(q(t), \delta q(t), t) = \int_{\xi_0}^{\xi_1} \delta q^\top M(q, t) \delta q$  is given in (2.21) with the virtual state  $q$  of (2.36),  $\underline{m}$  and  $\bar{m}$  are given in (2.26),  $C = (\bar{g}_0^2 + \bar{g}_1^2)(2\alpha_G^{-1} + 1)$ , and  $\mathbb{E}$  denotes the expected value*

operator. Furthermore, the probability that  $\|\xi_1 - \xi_0\|$  is greater than or equal to  $\varepsilon \in \mathbb{R}_{>0}$  is given as

$$\mathbb{P} [\|\xi_1(t) - \xi_0(t)\| \geq \varepsilon] \leq \frac{1}{\varepsilon^2} \left( \frac{\mathbb{E}[V_{s\ell}(0)]}{\underline{m}} e^{-2\alpha t} + \frac{C \bar{m}}{2\alpha \underline{m}} \right). \quad (2.40)$$

*Proof.* By definition of the infinitesimal differential generator given in Lemma 2.2 [25, p. 15], we have [6], [8]

$$\begin{aligned} \mathcal{L}V_{s\ell} = & \int_0^1 V_t + \sum_{i=1}^n \left( V_{q_i} f_i + V_{\partial_\mu q_i} \left( \frac{\partial f}{\partial q} \partial_\mu q \right)_i \right) \\ & + \frac{1}{2} \sum_{i,j=1}^n \left( V_{q_i q_j} (GG^\top)_{ij} + 2V_{q_i \partial_\mu q_j} (G \partial_\mu G^\top)_{ij} + V_{\partial_\mu q_i \partial_\mu q_j} (\partial_\mu G \partial_\mu G^\top)_{ij} \right) d\mu \end{aligned} \quad (2.41)$$

where  $V = \partial_\mu q^\top M(q, t) \partial_\mu q$ ,  $\partial_\mu q = \partial q / \partial \mu$ ,  $\partial_\mu G = \partial G / \partial \mu$ ,  $V_p = \partial V / \partial p$ , and  $V_{p_1 p_2} = \partial^2 V / (\partial p_1 \partial p_2)$ .

Since  $M_{x_i}$  is Lipschitz as in (2.37), we have  $\|M_{x_i x_j}\| \leq L_m$  and  $\|M_{x_i}\| \leq \sqrt{2L_m \bar{m}}$  using (2.26) as derived [20]. Computing  $\mathcal{L}V_{s\ell}$  of (2.41) using these bounds, the bounds of  $\|G_0\|_F$  and  $\|G_1\|_F$ , and  $\partial_\mu G = \partial G / \partial \mu = [-G_0, G_1]$  as in [6], [8] yields

$$\begin{aligned} \mathcal{L}V_{s\ell} \leq & \int_0^1 \partial_\mu q^\top (\dot{M} + 2 \text{sym}(M f_x)) \partial_\mu q d\mu \\ & + (\bar{g}_0^2 + \bar{g}_1^2) (L_m \|\partial_\mu q\|^2 / 2 + 2\sqrt{2L_m \bar{m}} \|\partial_\mu q\| + \bar{m}) \\ \leq & \int_0^1 \partial_\mu q^\top (\dot{M} + 2 \text{sym}(M f_x) + \alpha_s I) \partial_\mu q d\mu + C \bar{m} \end{aligned} \quad (2.42)$$

where  $\alpha_s = L_m(\bar{g}_0^2 + \bar{g}_1^2)(\alpha_G + 1/2)$ ,  $C = (\bar{g}_0^2 + \bar{g}_1^2)(2\alpha_G^{-1} + 1)$ , and the relation  $2ab \leq \alpha_G^{-1} a^2 + \alpha_G b^2$ , which holds for any  $a, b \in \mathbb{R}$  and  $\alpha_G \in \mathbb{R}_{>0}$ , is used with  $a = \sqrt{2\bar{m}}$  and  $b = \sqrt{L_m} \|\partial_\mu q\|$  to get the second inequality. This reduces to  $\mathcal{L}V_{s\ell} \leq -2\alpha V_{s\ell} + \underline{m}C$  under the condition (2.38), resulting in (2.39) and (2.40) as a result of (2.34) and (2.35) in Lemma 2.2.  $\square$

**Remark 2.4.** Although we consider the Gaussian white noise stochastic differential equation (2.32) when referring to stochastic systems in this thesis, other types of stochastic noises, including compound Poisson shot noise and bounded-measure Lévy noise, could be considered as in Theorem 2.5 using contraction theory [30].

### 2.3 Finite-Gain Stability and Contraction of Hierarchically Combined Systems

Due to the simpler stability analysis of Theorem 2.1 when compared with Lyapunov theory, Input-to-State Stability (ISS) and input-output stability in the sense of finite-gain  $\mathcal{L}_p$  stability can be easily studied using contraction theory [5].

**Theorem 2.6.** *If (2.24) is perturbed by  $d(x, t) \in \mathcal{L}_{pe}$  (i.e.,  $\|(d)_\tau\|_{\mathcal{L}_p} < \infty$  for  $\tau \in \mathbb{R}_{\geq 0}$  and  $p \in [1, \infty]$ , see Sec. 1.3) and Theorem 2.1 holds, then (2.24) is finite-gain  $\mathcal{L}_p$  stable with  $p \in [1, \infty]$  for an output  $y = h(x, d, t)$  with  $\int_{Y_0}^{Y_1} \|\delta y\| \leq \eta_0 \int_{\xi_0}^{\xi_1} \|\delta x\| + \eta_1 \|d\|$ ,  $\exists \eta_0, \eta_1 \geq 0$ , i.e.,  $\forall \tau \in \mathbb{R}_{\geq 0}$  [5]*

$$\left\| \left( \int_{Y_0}^{Y_1} \|\delta y\| \right)_\tau \right\|_{\mathcal{L}_p} \leq \left( \frac{\eta_0}{\alpha} + \eta_1 \right) \frac{\|(\Theta d)_\tau\|_{\mathcal{L}_p}}{\sqrt{\underline{m}}} + \frac{\eta_0 \zeta V_\ell(0)}{\sqrt{\underline{m}}} \quad (2.43)$$

where  $Y_0$  and  $Y_1$  denote the output trajectories of the original contracting system (2.1) and its perturbed system (2.24), respectively,  $\underline{m}$  is defined as  $M(x, t) \geq \underline{m}\mathbf{I}$ ,  $\forall x, t$ , as in (2.26), and  $V_\ell(0) = V_\ell(x(0), \delta x(0), 0)$  for  $V_\ell$  in (2.22). Also,  $\zeta = 1$  if  $p = \infty$  and  $\zeta = 1/(\alpha p)^{1/p}$  if  $p \in [1, \infty)$ . The perturbed system (2.24) also exhibits ISS.

*Proof.* In keeping with Theorem 5.1 of [9], (2.28) also implies the following relation for  $M = \Theta^\top \Theta$ :

$$\|(V_\ell)_\tau\|_{\mathcal{L}_p} \leq V_\ell(0) \|(e^{-\alpha t})_\tau\|_{\mathcal{L}_p} + \|e^{-\alpha t}\|_{\mathcal{L}_1} \|(\Theta d)_\tau\|_{\mathcal{L}_p} \leq V_\ell(0)\zeta + \|(\Theta d)_\tau\|_{\mathcal{L}_p} / \alpha. \quad (2.44)$$

Since we have  $\|\Theta^{-1}\| \leq 1/\sqrt{\underline{m}}$  for  $M = \Theta^\top \Theta$ , (2.43) can be obtained by using both (2.44) and the known bound of  $\|\delta y\|$ , thereby yielding a finite  $\mathcal{L}_p$  gain independently of  $\tau$ . ISS can be guaranteed by Lemma 4.6 of [9, p. 176], which states that exponential stability of an unperturbed system results in ISS.  $\square$

Theorems 2.1 – 2.6 are also applicable to the hierarchically combined system of two contracting dynamics due to Theorem 2.7 [2], [31].

**Theorem 2.7.** *Consider the following hierarchically combined system of two contracting dynamics:*

$$\frac{d}{dt} \begin{bmatrix} \delta z_0 \\ \delta z_1 \end{bmatrix} = \begin{bmatrix} F_{00} & 0 \\ F_{10} & F_{11} \end{bmatrix} \begin{bmatrix} \delta z_0 \\ \delta z_1 \end{bmatrix} \quad (2.45)$$

where  $F = F_{00}$  and  $F = F_{11}$  both satisfy the contraction condition (2.3), and suppose that it is subject to perturbation  $[d_0, d_1]^\top$ . Then the path length integral  $V_{\ell,i}(t) = \int_{\xi_0}^{\xi_1} \|\delta z_i\|$  with  $i = 0, 1$  between the original and perturbed dynamics trajectories,  $\xi_0$  and  $\xi_1$ , respectively, verifies [1]

$$\dot{V}_{\ell,0} + \alpha_0 V_{\ell,0} \leq \|\Theta_0 d_0\| \quad (2.46)$$

$$\dot{V}_{\ell,1} + \alpha_1 V_{\ell,1} \leq \|\Theta_1 d_1\| + \int_{\xi_0}^{\xi_1} \|F_{10}\| \|\delta z_1\| \quad (2.47)$$

where  $\alpha_i = \sup_{x,t} |\lambda_{\max}(F_{ii})|$ ,  $i = 0, 1$ . Hence, the error bounds of  $V_{\ell,0}(t)$  and  $V_{\ell,1}(t)$  can be obtained using Theorem 2.4 if  $\|F_{10}\|$  is bounded. In particular, if  $\|\Theta_i d_i\| \leq \sqrt{m_i} \bar{d}_i$ ,  $i = 0, 1$ , the relations (2.46) and (2.47) yield  $V_{\ell,0}(t) \leq V_{\ell,0}(0)e^{-\alpha_0 t} + (\sqrt{m_0} \bar{d}_0 / \alpha_0)(1 - e^{-\alpha_0 t})$  as in (2.27), and

$$V_{\ell,1}(t) \leq V_{\ell,1}(0)e^{-\alpha_1 t} + \frac{\sqrt{m_1} \bar{d}_1 + \bar{f}_{10} \bar{V}_0}{\alpha_1} (1 - e^{-\alpha_1 t}) \quad (2.48)$$

where  $\bar{f}_{10} = \sup_{t \in \mathbb{R}_{\geq 0}} \|F_{10}\|$  and  $\bar{V}_0 = V_{\ell,0}(0) + \sqrt{m_0} \bar{d}_0 / \alpha_0$ .

Also, similar to Theorem 2.6, we have  $\|(V_{\ell,0})_\tau\|_{\mathcal{L}_p} \leq V_{\ell,0}(0)\zeta_0 + \|(\Theta_0 d_0)_\tau\|_{\mathcal{L}_p} / \alpha_0$ , and thus a hierarchical connection for finite-gain  $\mathcal{L}_p$  stability can be established as follows [5]:

$$\|(V_{\ell,1})_\tau\|_{\mathcal{L}_p} \leq V_{\ell,1}(0)\zeta_1 + \frac{\|(\Theta_1 d_1)_\tau\|_{\mathcal{L}_p} + \bar{f}_{10} \|(V_{\ell,0})_\tau\|_{\mathcal{L}_p}}{\alpha_1} \quad (2.49)$$

where  $\zeta_i = 1$  if  $p = \infty$  and  $\zeta_i = 1/(\alpha_i p)^{1/p}$  if  $p \in [1, \infty)$  for  $i = 0, 1$ . By recursion, this result can be extended to an arbitrary number of hierarchically combined groups.

*Proof.* Applying the comparison lemma of Lemma 2.1 to (2.46) and (2.47), which follow from the differential dynamics (2.45) with the condition (2.3), we get (2.48) as in Theorem 2.4. Similarly, (2.49) follows from obtaining  $\|(V_{\ell,0})_\tau\|_{\mathcal{L}_p}$  by (2.44) using (2.46), and then recursively obtaining  $\|(V_{\ell,1})_\tau\|_{\mathcal{L}_p}$  using (2.47) as in Theorem 2.6 [5].  $\square$

**Example 2.7.** As demonstrated in Example 2.6, the Lagrangian virtual system (2.19) is contracting with respect to  $q$ , having  $q = \dot{\mathbf{q}}$  and  $q = \dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d - \Lambda(\mathbf{q} - \mathbf{q}_d)$  as its particular solutions. Let  $q_0 = q$  for such  $q$ . The virtual system of  $q_1$  which has  $[q_0^\top, q_1^\top]^\top = [\dot{\mathbf{q}}^\top, \mathbf{q}^\top]^\top$  and  $[\dot{\mathbf{q}}_r^\top, \mathbf{q}_d^\top]^\top$  as its particular solutions is given as

$$\dot{q}_1 = q_0 - \Lambda(q_1 - q)$$

resulting in  $\delta\dot{q}_1 = \text{I}\delta q_0 - \Lambda\delta q_1$ . Since the virtual system of  $q_0$  is contracting as in Example 2.6 and the virtual system  $\delta\dot{q}_1 = -\Lambda\delta q_1$  is contracting in  $q_1$  due to  $\Lambda > 0$ , (2.48) of Theorem 2.7 implies that the whole system (2.17) for  $[q_0^\top, q_1^\top]^\top$  is hierarchically contracting and robust against perturbation in the sense of Theorem 2.7 ( $F_{10} = \text{I}$  in this case). Also, see [5], [32] for the hierarchical multi-timescale separation of tracking and synchronization control for multiple Lagrangian systems.

## 2.4 Contraction Theory for Discrete-time Systems

The results presented so far can be readily extended to those for discrete-time nonlinear systems.

### 2.4.I Deterministic Perturbation

Let us consider the following nonlinear system with bounded deterministic perturbation  $d_k : \mathbb{R}^n \times \mathbb{N} \mapsto \mathbb{R}^n$  with  $\bar{d} \in \mathbb{R}_{\geq 0}$  s.t.  $\bar{d} = \sup_{x,k} \|d_k(x, k)\|$ :

$$x(k+1) = f_k(x(k), k) + d_k(x(k), k) \quad (2.50)$$

where  $k \in \mathbb{N}$ ,  $x : \mathbb{N} \mapsto \mathbb{R}^n$  is the discrete system state, and  $f_k : \mathbb{R}^n \times \mathbb{N} \mapsto \mathbb{R}^n$  is a smooth function. Although this thesis focuses mainly on continuous-time nonlinear systems, let us briefly discuss contraction theory for (2.50) to imply that the techniques in the subsequent chapters are applicable also to discrete-time nonlinear systems.

Let  $\xi_0(k)$  and  $\xi_1(k)$  be solution trajectories of (2.50) with  $d_k = 0$  and  $d_k \neq 0$ , respectively. Then a virtual system of  $q(\mu, k)$  parameterized by  $\mu \in [0, 1]$ , which has  $q(\mu = 0, k) = \xi_0(k)$  and  $q(\mu = 1, k) = \xi_1(k)$  as its particular solutions, can be expressed as follows:

$$q(\mu, k+1) = f_k(q(\mu, k), k) + \mu d_k(\xi_1(k), k). \quad (2.51)$$

The discrete version of robust contraction in Theorem 2.4 is given in the following theorem.

**Theorem 2.8.** *Let  $x_k = x(k)$  and  $q_k = q(\mu, k)$  for any  $k \in \mathbb{N}$ . If there exists a uniformly positive definite matrix  $M_k(x_k, k) = \Theta_k(x_k, k)^\top \Theta_k(x_k, k) > 0$ ,  $\forall x_k, k$ , where  $\Theta_k$  defines a smooth coordinate transformation of  $\delta x_k$ , i.e.,  $\delta z_k = \Theta_k(x_k, k)\delta x_k$ , s.t. either of the following equivalent conditions holds for  $\exists \alpha \in (0, 1)$ ,  $\forall x_k, k$ :*

$$\left\| \Theta_{k+1}(x_{k+1}, k+1) \frac{\partial f_k}{\partial x_k} \Theta_k(x_k, k)^{-1} \right\| \leq \alpha \quad (2.52)$$

$$\frac{\partial f_k^\top}{\partial x_k} M_{k+1}(x_{k+1}, k+1) \frac{\partial f_k}{\partial x_k} \leq \alpha^2 M_k(x_k, k), \quad (2.53)$$

then we have the following bound as long as we have  $\underline{m}\mathbf{I} \leq M_x(x_k, k) \leq \bar{m}\mathbf{I}$ ,  $\forall x_k, k$ , as in (2.26):

$$\|\xi_1(k) - \xi_0(k)\| \leq \frac{V_\ell(0)}{\sqrt{\underline{m}}}\alpha^k + \frac{\bar{d}(1 - \alpha^k)}{1 - \alpha} \sqrt{\frac{\bar{m}}{m}} \quad (2.54)$$

where  $V_\ell(k) = \int_{\xi_0}^{\xi_1} \|\Theta_k(q_k, k)\delta q_k\|$  as in (2.22) for the unperturbed trajectory  $\xi_0$ , perturbed trajectory  $\xi_1$ , and virtual state  $q_k = q(k)$  given in (2.51).

*Proof.* If (2.52) or (2.53) holds, we have that

$$\begin{aligned} V_\ell(k+1) &\leq \int_0^1 \|\Theta_{k+1}(\partial_{q_k} f_k(q_k, k)\partial_\mu q_k + d_k(x_k, k))\| d\mu \\ &\leq \alpha \int_0^1 \|\Theta_k(q_k, k)\partial_\mu q_k\| d\mu + \bar{d}\sqrt{\bar{m}} = \alpha V_\ell(k) + \bar{d}\sqrt{\bar{m}} \end{aligned}$$

where  $\Theta_{k+1} = \Theta_{k+1}(q_{k+1}, k+1)$ ,  $\partial_{q_k} f_k(q_k, k) = \partial f_k / \partial q_k$ , and  $\partial_\mu q_k = \partial q_k / \partial \mu$ . Applying this inequality iteratively results in (2.54).  $\square$

Theorem 2.8 can be used with Theorem 2.4 for stability analysis of hybrid nonlinear systems [33]–[35], or with Theorem 2.5 for stability analysis of discrete-time stochastic nonlinear systems [6], [8], [35]. For example, it is shown in [6] that if the time interval in discretizing (2.1) as (2.50) is sufficiently small, contraction of discrete-time systems with stochastic perturbation reduces to that of continuous-time systems as follows.

## 2.4.II Stochastic Perturbation

Let us also present a discrete-time version of Theorem 2.5, which can be extensively used for proving the stability of discrete-time and hybrid stochastic nonlinear systems, along with known results for deterministic systems [33], [34]. Consider the discrete-time nonlinear system with stochastic perturbation modeled by the stochastic difference equation

$$x(k+1) = f_k(x(k), k) + G_k(x(k), k)w(k) \quad (2.55)$$

where  $G_k : \mathbb{R}^n \times \mathbb{N} \rightarrow \mathbb{R}^{n \times d}$  is a matrix-valued function and  $w(k)$  is a  $d$ -dimensional sequence of zero mean uncorrelated normalized Gaussian random variables. Consider the following two systems with trajectories  $\xi_0(k)$  and  $\xi_1(k)$  driven by two independent stochastic perturbation  $w_0(k)$  and  $w_1(k)$ :

$$\xi_i(k+1) = f_k(\xi_i(k), k) + G_{i,k}(\xi_i(k), k)w_i(k), \quad i = 0, 1, \quad (2.56)$$

Similar to (2.36), a virtual system of  $q(\mu, k)$  parameterized by  $\mu \in [0, 1]$ , which has  $q(\mu = 0, k) = \xi_0(k)$  and  $q(\mu = 1, k) = \xi_1(k)$  as its particular solutions, can be given as follows:

$$q(\mu, k+1) = f_k(q(\mu, k), k) + G_k(\mu, \xi_0(k), \xi_1(k), k)w(k) \quad (2.57)$$

where  $G_k(\mu, \xi_0(k), \xi_1(k), k) = [(1-\mu)G_{0,k}(\xi_0(k), k), \mu G_{1,k}(\xi_1(k), k)]$  and  $w(k) = [w_0(k)^\top, w_1(k)^\top]^\top$ . The following theorem analyzes stochastic incremental stability for discrete-time nonlinear systems (2.56), which is different from [26], [35] in that the stability is studied in a differential sense and its Riemannian metric is state- and time-dependent.

**Theorem 2.9.** *Suppose that (2.53) holds for the discrete-time deterministic system (2.56) with  $\alpha^2 = 1 - \gamma_d$  and that  $\exists \underline{m}, \bar{m} \in \mathbb{R}_{>0}$  and  $\bar{g}_{0d}, \bar{g}_{1d} \in \mathbb{R}_{\geq 0}$  s.t.  $\underline{m}I \leq M_k(x, k) \leq \bar{m}I$ ,  $\forall x, k$ ,  $\sup_{x,k} \|G_{1,k}(x, k)\|_F = \bar{g}_{0d}$ , and  $\sup_{x,k} \|G_{2,k}(x, k)\|_F = \bar{g}_{1d}$ . Suppose also that  $\exists \gamma_2 \in (0, 1)$  s.t.  $\gamma_2 \leq 1 - (\bar{m}/\underline{m})(1 - \gamma_d)$ , where  $\gamma_d$  is the contraction rate. Consider the generalized squared length with respect to a Riemannian metric  $M_k(q(\mu, k), k)$  defined as*

$$V_{s\ell}(q, \delta q, k) = \int_{\xi_0}^{\xi_1} \delta q^\top M_k(q(\mu, k), k) \delta q = \int_0^1 \frac{\partial q^\top}{\partial \mu} M_k(q(\mu, k), k) \frac{\partial q}{\partial \mu} d\mu \quad (2.58)$$

s.t.  $V_k(q, \delta q, k) \geq \underline{m} \|\xi_1(k) - \xi_0(k)\|^2$ . Then the mean squared distance between the two trajectories of the system (2.56) is bounded as follows:

$$\mathbb{E} [\|\xi_1(k) - \xi_0(k)\|^2] \leq \frac{1 - \tilde{\gamma}_d^k}{1 - \tilde{\gamma}_d} C_d + \frac{\tilde{\gamma}_d^k}{\underline{m}} E[V_{s\ell}(0)]. \quad (2.59)$$

where  $V_{s\ell}(0) = V_{s\ell}(q(0), \delta q(0), 0)$ ,  $C_d = (\bar{m}/\underline{m})(\bar{g}_{0d}^2 + \bar{g}_{1d}^2)$ , and  $\tilde{\gamma}_d = 1 - \gamma_2 \in (0, 1)$ .

*Proof.* Let  $q_k = q(\mu, k)$ ,  $w_k = w(k)$ ,  $V_k = V_{s\ell}(q(\mu, k), \delta q(\mu, k), k)$ , and  $M_k = M_k(q(\mu, k), k)$  for any  $k \in \mathbb{N}$  for notational simplicity. Using the assumed bounds along with (2.53) ( $\alpha^2 = 1 - \gamma_d$ ) and (2.57), we have, for  $\ell \in \mathbb{N}$ , that

$$\begin{aligned} V_{\ell+1} &\leq \bar{m} \int_0^1 \left\| \frac{\partial f_\ell}{\partial q_\ell} \frac{\partial q_\ell}{\partial \mu} + \frac{\partial G_\ell}{\partial \mu} w_\ell \right\|^2 d\mu \\ &\leq \frac{\bar{m}}{\underline{m}} (1 - \gamma_d) \int_0^1 \frac{\partial q_\ell^\top}{\partial \mu} M_\ell \frac{\partial q_\ell}{\partial \mu} d\mu \\ &\quad + \bar{m} \int_0^1 \left( 2 \frac{\partial q_\ell^\top}{\partial \mu} \frac{\partial f_\ell^\top}{\partial q_\ell} \frac{\partial G_\ell}{\partial \mu} w_\ell + w_\ell^\top \frac{\partial G_\ell^\top}{\partial \mu} \frac{\partial G_\ell}{\partial \mu} w_\ell \right) d\mu. \end{aligned} \quad (2.60)$$

Taking the conditional expected value of (2.60) when  $q_\ell$ ,  $\delta q_\ell$ , and  $\ell$  are given, we have that (see also: Theorem 2 of [26])

$$\begin{aligned}\mathbb{E}_{\zeta_\ell}[V_{\ell+1}] &\leq \gamma_m V_\ell + \overline{m}\mathbb{E}_{\zeta_\ell} \left[ \int_0^1 w_\ell^\top \frac{\partial G_\ell^\top}{\partial \mu} \frac{\partial G_\ell}{\partial \mu} w_\ell d\mu \right] \\ &\leq \gamma_m V_\ell + \sum_{i=1,2} \overline{m}\mathbb{E}_{\zeta_\ell} \left[ \text{Tr} \left( w_{i,\ell} w_{i,\ell}^\top G_{i,\ell}^\top G_{i,\ell} \right) \right] \\ &\leq \gamma_m V_\ell + \overline{m} \sum_{i=1,2} \text{Tr} \left( G_{i,\ell}^\top G_{i,\ell} \right) \leq \tilde{\gamma}_d V_\ell + \underline{m}C_d,\end{aligned}\tag{2.61}$$

where  $\gamma_m = \overline{m}/\underline{m}(1 - \gamma_d)$ , and  $q_\ell$ ,  $\delta q_\ell$ , and  $\ell$  are denoted as  $\zeta_\ell$ . Here, we used the condition:  $\exists \gamma_2 \in (0, 1)$  s.t.  $\gamma_m \leq 1 - \gamma_2 = \tilde{\gamma}_d$ . Taking expectation over  $\zeta_{\ell-1}$  in (2.61) with the tower rule  $\mathbb{E}_{\zeta_{\ell-1}}[V_{\ell+1}] = \mathbb{E}_{\zeta_{\ell-1}}[\mathbb{E}_{\zeta_\ell}[V_{\ell+1}]]$  gives us that

$$\mathbb{E}_{\zeta_{\ell-1}}[V_{\ell+1}] \leq \tilde{\gamma}_d^2 V_{\ell-1} + \underline{m}C_d + \underline{m}C_d \tilde{\gamma}_d$$

where  $\tilde{\gamma}_d$  is defined as  $\tilde{\gamma}_d = 1 - \gamma_2$ . Continuing this operation with the relation  $\underline{m}\mathbb{E}_{\zeta_0} [\|\xi_{1,\ell+1} - \xi_{2,\ell+1}\|^2] \leq \mathbb{E}_{\zeta_0} [V_{\ell+1}]$  yields

$$\mathbb{E}_{\zeta_0} [\|\xi_{1,k} - \xi_{2,k}\|^2] - \frac{\tilde{\gamma}_d^k}{\underline{m}} V_0 \leq C_d \sum_{i=0}^{k-1} \tilde{\gamma}_d^i = \frac{1 - \tilde{\gamma}_d^k}{1 - \tilde{\gamma}_d} C_d$$

where  $k = \ell + 1$ . Taking expectation over  $\zeta_0$  and rearranging terms result in (2.59).  $\square$

### 2.4.III Connection between Continuous and Discrete Stochastic Contraction Theory

Let us now consider the case where the time interval  $\Delta t = t_{k+1} - t_k$  for discretization is sufficiently small, i.e.,  $\Delta t \gg (\Delta t)^2$ . Then the continuous-time stochastic system (2.29) can be discretized as

$$\begin{aligned}x(k+1) &= x(k) + \int_{t_k}^{t_{k+1}} f(x(t), t) dt + G(x(t), t) d\mathcal{W}(t) \\ &= x(k) + f(x(k), t_k) \Delta t + G(x(k), t_k) \Delta \mathcal{W}(k) + \mathcal{O}(\Delta t^2)\end{aligned}$$

where  $x(k) = x(t_k)$ ,  $\Delta \mathcal{W}(k) = \sqrt{\Delta t} w(k)$ , and  $w(k)$  is a  $d$ -dimensional sequence of zero mean uncorrelated normalized Gaussian random variables. When  $\Delta t \gg (\Delta t)^2$ ,  $f_k(x(k), k)$  and  $G_k(x(k), k)$  in (2.55) can be approximated as  $f_k(x(k), k) \simeq x(k) + f(x(k), t_k) \Delta t$  and  $G_k(x(k), k) \simeq \sqrt{\Delta t} G(x(k), t_k)$ . In this situation, we have the following theorem that connects the stochastic incremental stability of discrete-time systems with that of continuous-time systems.

**Theorem 2.10.** Suppose that (2.61) in Theorem 2.9 holds with  $\tilde{\gamma}_d = 1 - \gamma_2 \in (0, 1)$ . Then the expected value of  $V_{k+1}$  up to first order in  $\Delta t$  is given as  $\mathbb{E}_{\zeta_k}[V_{k+1}] = V_k + \Delta t \mathcal{L}V_k$ , where  $V_k = V_{s\ell}(q(\mu, k), \delta q(\mu, k), k)$  for  $V_{s\ell}$  of (2.58) and  $\mathcal{L}$  is the infinitesimal differential generator. Furthermore, the following inequality holds:

$$\mathcal{L}V_{s\ell}(q_k, \delta q_k, t_k) \leq -\frac{\gamma_2}{\Delta t} V_{s\ell}(q, \delta q_k, t_k) + \underline{m}\tilde{C}_c \quad (2.62)$$

where  $q_k = q(\mu, k)$   $\tilde{C}_c$  is a positive constant given as

$$\tilde{C}_c = \frac{C_d}{\Delta t} = \frac{\bar{m}}{m\Delta t}(\bar{g}_{0d}^2 + \bar{g}_{1d}^2) = \frac{\bar{m}}{m}(\bar{g}_0^2 + \bar{g}_1^2) \quad (2.63)$$

with  $\bar{g}_0$  and  $\bar{g}_1$  defined in Theorem 2.5.

*Proof.* Let  $M_k = M_k(q(\mu, k), k)$ .  $M_{k+1}$  up to first order in  $\Delta t$  is written as

$$\begin{aligned} M_{k+1} &= \frac{\partial M_k}{\partial t_k} \Delta t + \sum_{i=1}^n \frac{\partial M_k}{\partial (q_k)_i} (f_{c,k} \Delta t + G_{c,k} \Delta \mathcal{W}_k)_i \\ &\quad + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2 M_k}{\partial (q_k)_i \partial (q_k)_j} (G_{c,k} \Delta \mathcal{W}_k)_i (G_{c,k} \Delta \mathcal{W}_k)_j + M_k + \mathcal{O}(\Delta t^2) \end{aligned} \quad (2.64)$$

where  $f_{c,k}$  and  $G_{c,k}$  are defined as  $f_{c,k} = f(q_k, t_k)$  and  $G_{c,k} = G(q_k, t_k)$  for notational simplicity. The subscripts  $i$  and  $j$  denote the corresponding vectors'  $i$ th and  $j$ th elements. Similarly,  $\partial q_{k+1} / \partial \mu$  up to first order in  $\Delta t$  can be computed as

$$\frac{\partial q_{k+1}}{\partial \mu} = \frac{\partial q_k}{\partial \mu} + \frac{\partial f_{c,k}}{\partial q_k} \frac{\partial q_k}{\partial \mu} \Delta t + \frac{\partial G_{c,k}}{\partial \mu} \Delta \mathcal{W}_k + \mathcal{O}(\Delta t^2). \quad (2.65)$$

Substituting (2.64) and (2.65) into  $\mathbb{E}_{\zeta_k}[V_{k+1}]$  yields

$$\begin{aligned} \mathbb{E}_{\zeta_k}[V_{k+1}] &= \mathbb{E}_{\zeta_k} \left[ \int_0^1 \frac{\partial q_{k+1}}{\partial \mu}^\top M_{k+1} \frac{\partial q_{k+1}}{\partial \mu} d\mu \right] \\ &= V_k + (dV_{d,k} + dV_{s,k}) \Delta t + \mathcal{O}(\Delta t^{3/2}) \end{aligned}$$

where  $dV_{d,k}$  and  $dV_{s,k}$  are given by

$$dV_{d,k} = \int_0^1 \frac{\partial q_k}{\partial \mu}^\top \left( \frac{\partial f_{c,k}}{\partial q_k} M_k + \dot{M}_k + M_k \frac{\partial f_{c,k}}{\partial q_k} \right) \frac{\partial q_k}{\partial \mu} d\mu$$

with  $\dot{M}_k = \partial M_k / \partial t_k + \sum_{i=1}^n (\partial M_k / \partial (q_k)_i) f_{c,k}$  and

$$\begin{aligned} dV_{s,k} &= \int_0^1 \left[ \sum_{i=1}^n \sum_{j=1}^n (M_k)_{ij} \left( \frac{\partial G_{c,k}}{\partial \mu} \frac{\partial G_{c,k}}{\partial \mu}^\top \right)_{ij} + 2 \frac{\partial (M_k)_i}{\partial (q_k)_j} \frac{\partial q_k}{\partial \mu} \left( G_{c,k} \frac{\partial G_{c,k}}{\partial \mu}^\top \right)_{ij} \right. \\ &\quad \left. + \frac{1}{2} \frac{\partial q_k}{\partial \mu}^\top \frac{\partial^2 M_k}{\partial (q_k)_i \partial (q_k)_j} \frac{\partial q_k}{\partial \mu} (G_{c,k} G_{c,k}^\top)_{ij} \right] d\mu. \end{aligned}$$

We note that the properties of  $w(k)$  as a  $d$ -dimensional sequence of zero mean uncorrelated normalized Gaussian random variables are used to derive these relations. Since  $dV_{d,k} + dV_{s,k} = \mathcal{L}V_k$  where  $\mathcal{L}$  is the infinitesimal differential generator, we have  $\mathbb{E}_{\zeta_k}[V_{k+1}] = V_k + \Delta t \mathcal{L}V_k$ . Thus, the condition  $\mathbb{E}_{\zeta_k}[V_{k+1}] \leq (1 - \gamma_2)V_k + \underline{m}C_d$  given by (2.61) in Theorem 2.9 reduces to the following inequality:

$$\mathcal{L}V_k(q_k, \partial_\mu q_k, t_k) \leq -\frac{\gamma_2}{\Delta t}V_k(q_k, \partial_\mu q_k, t_k) + \underline{m}\frac{C_d}{\Delta t}. \quad (2.66)$$

Finally, (2.66) with the relations  $\tilde{C}_c = C_d/\Delta t$  and  $G_k(q_k, k) = \sqrt{\Delta t}G(q_k, t_k)$  results in (2.62) and (2.63).  $\square$

For example, in practical control applications, we use the same control input at  $t = t_k$  for a finite time interval  $t \in [t_k, t_{k+1})$ . Theorems 2.5 and 2.10 indicate that if  $\Delta t$  is sufficiently small, a discrete-time stochastic controller can be viewed as a continuous-time counterpart with contraction rate  $2\gamma_1 = \gamma_2/\Delta t$ . We will illustrate how to select the sampling period  $\Delta t$  large enough without deteriorating the control performance as demonstrated in [6].

We finally remark that the steady-state upper bounds of (2.27) in Theorem 2.4, (2.39) in Theorem 2.5, and (2.54) in Theorem 2.8 are all functions of  $\bar{m}/\underline{m}$ . This property is to be used extensively in Chapter 4 for designing a convex optimization-based control and estimation synthesis algorithm via contraction theory.

## References

- [1] W. Lohmiller and J.-J. E. Slotine, "On contraction analysis for nonlinear systems," *Automatica*, vol. 34, no. 6, pp. 683–696, 1998, ISSN: 0005-1098.
- [2] J.-J. E. Slotine, "Modular stability tools for distributed computation and control," *Int. J. Adapt. Control Signal Process.*, vol. 17, no. 6, pp. 397–416, 2003.
- [3] W. Wang and J.-J. E. Slotine, "On partial contraction analysis for coupled nonlinear oscillators," *Biol. Cybern.*, vol. 92, no. 1, pp. 38–53, Jan. 2005, ISSN: 0340-1200.
- [4] Q. Pham, N. Tabareau, and J.-J. E. Slotine, "A contraction theory approach to stochastic incremental stability," *IEEE Trans. Autom. Control*, vol. 54, no. 4, pp. 816–820, Apr. 2009.
- [5] S.-J. Chung, S. Bandyopadhyay, I. Chang, and F. Y. Hadaegh, "Phase synchronization control of complex networks of Lagrangian systems on adaptive digraphs," *Automatica*, vol. 49, no. 5, pp. 1148–1161, 2013.

- [6] H. Tsukamoto and S.-J. Chung, “Robust controller design for stochastic nonlinear systems via convex optimization,” *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4731–4746, 2021.
- [7] A. P. Dani, S.-J. Chung, and S. Hutchinson, “Observer design for stochastic nonlinear systems via contraction-based incremental stability,” *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 700–714, Mar. 2015.
- [8] H. Tsukamoto and S.-J. Chung, “Convex optimization-based controller design for stochastic nonlinear systems using contraction analysis,” in *IEEE Conf. Decis. Control*, Dec. 2019, pp. 8196–8203.
- [9] H. K. Khalil, *Nonlinear Systems*, 3rd. Upper Saddle River, NJ: Prentice-Hall, 2002.
- [10] D. E. Kirk, *Optimal Control Theory: An Introduction*. Dover Publications, Apr. 2004, ISBN: 0486434842.
- [11] D. Angeli, “A Lyapunov approach to incremental stability properties,” *IEEE Trans. Autom. Control*, vol. 47, no. 3, pp. 410–421, Mar. 2002.
- [12] J. Jouffroy and J.-J. E. Slotine, “Methodological remarks on contraction theory,” in *IEEE Conf. Decis. Control*, vol. 3, Dec. 2004, pp. 2537–2543.
- [13] W. J. Rugh, *Linear Systems Theory*. USA: Prentice-Hall, Inc., 1996, ISBN: 0134412052.
- [14] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Upper Saddle River, NJ: Pearson, 1991.
- [15] H. Robbins and S. Monro, “A stochastic approximation method,” *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 1951.
- [16] P. M. Wensing and J.-J. E. Slotine, “Beyond convexity – Contraction and global convergence of gradient descent,” *PLOS ONE*, vol. 15, pp. 1–29, Aug. 2020.
- [17] S.-I. Amari, “Natural gradient works efficiently in learning,” *Neural Comput.*, vol. 10, no. 2, pp. 251–276, 1998.
- [18] C. Udriste, *Convex functions and optimization methods on Riemannian manifolds*. Springer Science & Business Media, 1994, vol. 297.
- [19] H. Tsukamoto and S.-J. Chung, “Neural contraction metrics for robust estimation and control: A convex optimization approach,” *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 211–216, 2021.
- [20] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Neural stochastic contraction metrics for learning-based control and estimation,” *IEEE Control Syst. Lett.*, vol. 5, no. 5, pp. 1825–1830, 2021.
- [21] I. R. Manchester and J.-J. E. Slotine, “Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design,” *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 3046–3053, Jun. 2017.

- [22] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd. Cambridge University Press, 2012, ISBN: 0521548233.
- [23] S. Singh, A. Majumdar, J.-J. E. Slotine, and M. Pavone, “Robust online motion planning via contraction theory and convex optimization,” in *IEEE Int. Conf. Robot. Automat.*, May 2017, pp. 5883–5890.
- [24] L. Arnold, *Stochastic Differential Equations: Theory and Applications*. Wiley, 1974.
- [25] H. J. Kushner, *Stochastic Stability and Control*, English. Academic Press New York, 1967, xiv, 161 p.
- [26] T.-J. Tarn and Y. Rasis, “Observers for nonlinear stochastic systems,” *IEEE Trans. Autom. Control*, vol. 21, no. 4, pp. 441–448, Aug. 1976.
- [27] M. Zakai, “On the ultimate boundedness of moments associated with solutions of stochastic differential equations,” *SIAM J. Control*, vol. 5, no. 4, pp. 588–593, 1967.
- [28] D. S. Geoffrey R. Grimmett, *Probability and Random Processes*, 3rd. United Kingdom: Oxford University Press, 2001.
- [29] E. Mazumdar, T. Westenbroek, M. I. Jordan, and S. Shankar Sastry, “High confidence sets for trajectories of stochastic time-varying nonlinear systems,” in *IEEE Conf. Decis. Control*, 2020, pp. 4275–4280.
- [30] S. Han and S.-J. Chung, *Incremental nonlinear stability analysis for stochastic systems perturbed by Lévy noise*, arXiv:2103.13338, Mar. 2021.
- [31] J.-J. E. Slotine and W. Lohmiller, “Modularity, evolution, and the binding problem: A view from stability theory,” *Neural Netw.*, vol. 14, no. 2, pp. 137–145, 2001.
- [32] S.-J. Chung and J.-J. E. Slotine, “Cooperative robot control and concurrent synchronization of Lagrangian systems,” *IEEE Trans. Robot.*, vol. 25, no. 3, pp. 686–700, Jun. 2009.
- [33] J.-J. E. Slotine, W. Wang, and K. El Rifai, “Contraction analysis of synchronization in networks of nonlinearly coupled oscillators,” in *Int. Symp. Math. Theory Netw. Syst.*, Jul. 2004.
- [34] W. Lohmiller and J.-J. E. Slotine, “Nonlinear process control using contraction theory,” *AIChE Journal*, vol. 46, pp. 588–596, Mar. 2000.
- [35] Q. Pham, “Analysis of discrete and hybrid stochastic systems by nonlinear contraction theory,” in *Int. Conf. Control Automat. Robot. Vision*, Dec. 2008, pp. 1054–1059.

## ROBUST NONLINEAR CONTROL AND ESTIMATION VIA CONTRACTION THEORY

- [1] H. Tsukamoto and S.-J. Chung, “Robust controller design for stochastic nonlinear systems via convex optimization,” *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4731–4746, 2021.
- [2] H. Tsukamoto and S.-J. Chung, “Convex optimization-based controller design for stochastic nonlinear systems using contraction analysis,” in *IEEE Conf. Decis. Control*, Dec. 2019, pp. 8196–8203.

As shown in Theorem 2.4 for deterministic disturbance and in Theorem 2.5 for stochastic disturbance, contraction theory provides explicit bounds on the distance of any couple of perturbed system trajectories. This property is useful in designing robust and optimal feedback controllers for a nonlinear system such as  $\mathcal{H}_\infty$  control [1]–[11], which attempts to minimize the system  $\mathcal{L}_2$  gain for optimal disturbance attenuation.

Most of such feedback control and estimation schemes are, however, based on the assumption that we know a Lyapunov function candidate. This chapter thus delineates one approach to solve a nonlinear optimal feedback control problem via contraction theory [12], [13], thereby proposing one explicit way to construct a Lyapunov function and contraction metric for general nonlinear systems for the sake of robustness. This approach is also utilizable for optimal state estimation problems as shall be seen in Chapter 4.

We consider the following smooth nonlinear system, perturbed by bounded deterministic disturbances  $d_c(x, t)$  with  $\sup_{x,t} \|d_c(x, t)\| = \bar{d}_c \in \mathbb{R}_{\geq 0}$  or by Gaussian white noise, driven by a Wiener process  $\mathcal{W}(t)$  with  $\sup_{x,t} \|G_c(x, t)\|_F = \bar{g}_c \in \mathbb{R}_{\geq 0}$ :

$$\dot{x} = f(x, t) + B(x, t)u + d_c(x, t) \quad (3.1)$$

$$dx = (f(x, t) + B(x, t)u)dt + G_c(x, t)d\mathcal{W}(t) \quad (3.2)$$

$$\dot{x}_d = f(x_d, t) + B(x_d, t)u_d \quad (3.3)$$

where  $x : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  is the system state,  $u \in \mathbb{R}^m$  is the system control input,  $f : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  and  $B : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{n \times m}$  are known smooth functions,

$d_c : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  and  $G_c : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{n \times w}$  are unknown bounded functions for external disturbances, and  $\mathcal{W} : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^w$  is a  $w$ -dimensional Wiener process. Also, for (3.3),  $x_d : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  and  $u_d : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^m$  denote the desired target state and control input trajectories, respectively.

**Remark 3.1.** *We consider control-affine nonlinear systems (3.1)–(3.3) in Chapter 3, 4, and 6 – 8.1. This is primarily because the controller design techniques for control-affine nonlinear systems are less complicated than those for control non-affine systems (which often result in  $u$  given implicitly by  $u = k(x, u, t)$  [14], [15]), but still utilizable even for the latter, e.g., by treating  $\dot{u}$  as another control input (see Example 3.1), or by solving the implicit equation  $u = k(x, u, t)$  iteratively with a discrete-time controller (see Example 3.2 and Remark 3.3).*

**Example 3.1.** *By using  $\dot{u}$  instead of  $u$  in (3.1) and (3.2), a control non-affine system,  $\dot{x} = f(x, u, t)$ , can be rewritten as*

$$\frac{d}{dt} \begin{bmatrix} x \\ u \end{bmatrix} = \begin{bmatrix} f(x, u, t) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \dot{u}$$

*which can be viewed as a control-affine nonlinear system with the state  $[x^\top, u^\top]^\top$  and control  $\dot{u}$ .*

**Example 3.2.** *One drawback of the technique in Example 3.1 is that we have to control  $\dot{u}$  instead of  $u$ , which could be difficult in practice. In this case, we can utilize the following control non-affine nonlinear system decomposed into control-affine and non-affine parts:*

$$\dot{x} = f(x, u, t) = f_a(x, t) + B_a(x, t)u + r(x, u, t)$$

*where  $r(x, u, t) = f(x, u, t) - f_a(x, t) - B_a(x, t)u$ . The controller  $u$  can now be designed implicitly as*

$$B_a(x, t)u = B_a(x, t)u^* - r(x, u, t) \tag{3.4}$$

*where  $u^*$  is a stabilizing controller for the control-affine system  $\dot{x} = f_a(x, t) + B_a(x, t)u^*$ . Since solving such an implicit equation in (3.4) in real-time could be unrealistic in practice, we will derive a learning-based approach to solve it iteratively for unknown  $r(x, u, t)$ , without deteriorating its stability performance (see Lemma 8.2 and Theorem 8.4 of Chapter 8).*

### 3.1 Overview of Nonlinear Control and Estimation

We briefly summarize the advantages and disadvantages of existing nonlinear feedback control and state estimation schemes, so that one can identify which strategy is appropriate for their study and refer to the relevant parts of this thesis.

Table 3.1: Comparison between the SDC and CCM formulation (note that  $\gamma(\mu = 0, t) = x_d$  and  $\gamma(\mu = 1, t) = x$ ).

	SDC (Theorem 4.2) [12], [13], [16]–[18]	CCM (Theorem 4.6) [19], [20]
Control law	$u = u_d - K(x, x_d, u_d, t)(x - x_d)$ or $u_d - K(x, t)(x - x_d)$	$u = u_d + \int_0^1 k(\gamma(\mu, t), \partial_\mu \gamma(\mu, t), u, t) d\mu$
Computation	Evaluates $K(x, x_d, u_d, t)$ for given $(x, x_d, u_d, t)$ as in LTV systems	Computes geodesics $\gamma$ for given $(x, x_d, t)$ and integrates $k$
Generality	Captures nonlinearity by (multiple) SDC matrices	Handles general differential dynamics
Contraction	Depends on $(x, x_d, u_d, t)$ or $(x, t)$ (partial contraction)	Depends on $(x, t)$ (contraction)

#### 3.1.I Systems with Known Lyapunov Functions

As discussed in Sec. 1.2, there are several nonlinear systems equipped with a known contraction metric/Lyapunov function, such as Lagrangian systems [21, p. 392], whose inertia matrix  $\mathcal{H}(q)$  defines its contraction metric (see Example 2.6), or the nonlinear SLAM problem [18], [22] with virtual synthetic measurements, which can be reduced to an LTV estimation problem [22]. Once we have a contraction metric/Lyapunov function, stabilizing control and estimation laws can be easily derived by using, e.g., [23]–[25]. Thus, those dealing primarily with such nonlinear systems should skip this chapter and proceed to Part II of this thesis (Chapter 5 – 8) on learning-based and data-driven control using contraction theory. Note that these known contraction metrics are not necessarily optimal, and the techniques to be derived in Chapter 3 and Chapter 4 are for obtaining contraction metrics with an optimal disturbance attenuation property [12], [13].

#### 3.1.II Linearization of Nonlinear Systems

If a contraction metric of a given nonlinear system is unknown, we could linearize it to apply methodologies inspired by LTV systems theory such as  $\mathcal{H}_\infty$  control [6]–[11], iterative Linear Quadratic Regulator (iLQR) [26], [27], or Extended Kalman Filter (EKF). Their stability is typically analyzed by decomposing  $f(x, t)$  as  $f(x, t) =$

$Ax + (f(x, t) - Ax)$  assuming that the nonlinear part  $f(x, t) - Ax$  is bounded, or by finding a local contraction region for the sake of local exponential stability as in [16], [28]. Since the decomposition  $f(x, t) = Ax + (f(x, t) - Ax)$  allows applying the result of Theorem 2.4, we could exploit the techniques in Chapter 3 and Chapter 4 for providing formal robustness and optimality guarantees for the LTV systems-type approaches. For systems whose nonlinear part  $f(x, t) - Ax$  is not necessarily bounded, Sec. 8.2.II elucidates how contraction theory can be used to stabilize them with the learned dynamics for control synthesis.

### 3.1.III State-Dependent Coefficient (SDC) Formulation

It is shown in [12], [13], [16]–[18] that the SDC-based control and estimation [29]–[32], which capture nonlinearity using a state-dependent matrix  $A(x, t)$  s.t.  $f(x, t) = A(x, t)x$  (e.g., we have  $A(x, t) = \cos x$  for  $f(x, t) = x \cos x$ ), result in exponential boundedness of system trajectories both for deterministic and stochastic systems due to Theorems 2.4 and 2.5 [16]. Because of the extended linear form of SDC (see Table 3.1), the results to be presented in Chapter 3 – 4 based on the SDC formulation are applicable to linearized dynamics that can be viewed as an LTV system with some modifications (see Remark 3.2).

This idea is slightly generalized in [17] to explicitly consider incremental stability with respect to a target trajectory (e.g.,  $x_d$  for control and  $x$  for estimation) instead of using  $A(x, t)x = f(x, t)$ . Let us derive the following lemma for this purpose [12], [13], [17], [18], [32]. Let us derive the following lemma for this purpose [12], [13], [17], [18], [32].

**Lemma 3.1.** *Let  $f : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  and  $B : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{n \times m}$  be piecewise continuously differentiable functions. Then there exists a matrix-valued function  $A : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{n \times n}$  s.t.,  $\forall s \in \mathbb{R}^n, \bar{s} \in \mathbb{R}^n, \bar{u} \in \mathbb{R}^m$ , and  $t \in \mathbb{R}_{\geq 0}$ ,*

$$A(s, \bar{s}, \bar{u}, t)\mathbf{e} = f(s, t) + B(s, t)\bar{u} - f(\bar{s}, t) - B(\bar{s}, t)\bar{u}$$

where  $\mathbf{e} = s - \bar{s}$ , and one such  $A$  is given as follows:

$$A(s, \bar{s}, \bar{u}, t) = \int_0^1 \frac{\partial \bar{f}}{\partial s}(cs + (1-c)\bar{s}, \bar{u}, t)dc \quad (3.5)$$

where  $\bar{f}(s, \bar{u}, t) = f(s, t) + B(s, t)\bar{u}$ . We call  $A$  an SDC matrix if it is constructed to satisfy the controllability (or observability for estimation) condition. Furthermore, the choice of  $A$  is not unique for  $n \geq 2$ , where  $n$  is the number of states, and

the convex combination of such non-unique SDC matrices also verifies extended linearization as follows:

$$\begin{aligned} & f(s, t) + B(s, t)\bar{u} - f(\bar{s}, t) - B(\bar{s}, t)\bar{u} \\ &= A(\varrho, s, \bar{s}, \bar{u}, t)(s - \bar{s}) = \sum_{i=1}^{s_A} \varrho_i A_i(s, \bar{s}, \bar{u}, t)(s - \bar{s}) \end{aligned} \quad (3.6)$$

where  $\varrho = (\varrho_1, \dots, \varrho_{s_A})$ ,  $\sum_{i=1}^{s_A} \varrho_i = 1$ ,  $\varrho_i \geq 0$ , and each  $A_i$  satisfies the relation  $\bar{f}(s, \bar{u}, t) - \bar{f}(\bar{s}, \bar{u}, t) = A_i(s, \bar{s}, \bar{u}, t)(s - \bar{s})$ .

*Proof.* The first statement on (3.5) follows from the integral relation given as

$$\int_0^1 \frac{d\bar{f}}{dc}(cs + (1-c)\bar{s}, \bar{u}, t)dc = \bar{f}(s, \bar{u}, t) - \bar{f}(\bar{s}, \bar{u}, t).$$

If there are multiple SDC matrices  $A_i$ , we clearly have  $\varrho_i A_i(s, \bar{s}, \bar{u}, t)(s - \bar{s}) = \varrho_i(\bar{f}(s, \bar{u}, t) - \bar{f}(\bar{s}, \bar{u}, t))$ ,  $\forall i$ , and therefore, the relation  $\sum_{i=1}^{s_A} \varrho_i = 1$ ,  $\varrho_i \geq 0$  gives (3.6).  $\square$

**Example 3.3.** Let us illustrate how Lemma 3.1 can be used in practice taking the following nonlinear system as an example:

$$\dot{x} = [x_2, -x_1 x_2]^\top + [0, \cos x_1]^\top u \quad (3.7)$$

where  $x = [x_1, x_2]^\top$ . If we use  $(s, \bar{s}, \bar{u}) = (x, x_d, u_d)$  in Lemma 3.1 for a given target trajectory  $(x_d, u_d)$  that satisfies (3.7), evaluating the integral of (3.5) gives

$$A_1(x, x_d, u_d, t) = - \begin{bmatrix} 0 & 1 \\ \frac{x_2 + x_{2d}}{2} - \frac{u_d(\cos x_1 - \cos x_{d1})}{x_1 - x_{d1}} & \frac{x_1 + x_{1d}}{2} \end{bmatrix} \quad (3.8)$$

due to the relation  $\partial \bar{f} / \partial s = \begin{bmatrix} 0 & 1 \\ -s_2 & -s_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -u_d \sin s_1 & 0 \end{bmatrix}$  for  $\bar{f}(s, u_d, t) = f(s, t) + B(s, t)u_d$ , where  $x_d = [x_{1d}, x_{2d}]^\top$ . Note that we have

$$\frac{(\cos x_1 - \cos x_{d1})}{x_1 - x_{d1}} = -\sin\left(\frac{x_1 + x_{1d}}{2}\right) \operatorname{sinc}\left(\frac{x_1 - x_{1d}}{2}\right)$$

and thus  $A(x, x_d, u_d, t)$  is defined for all  $x, x_d, u_d$ , and  $t$ . The SDC matrix (3.8) indeed verifies  $A_1(x, x_d, u_d, t)(x - x_d) = \bar{f}(x, t) - \bar{f}(x_d, t)$ .

We can see that the following is also an SDC matrix of the nonlinear system (3.7):

$$A_2(x, x_d, u_d, t) = - \begin{bmatrix} 0 & 1 \\ x_2 - \frac{u_d(\cos x_1 - \cos x_{d1})}{x_1 - x_{d1}} & x_{1d} \end{bmatrix}. \quad (3.9)$$

Therefore, the convex combination of  $A_1$  in (3.8) and  $A_2$  in (3.9),  $A = \varrho_1 A_1 + \varrho_2 A_2$  with  $\varrho_1 + \varrho_2 = 1$ ,  $\varrho_1, \varrho_2 \geq 0$ , is also an SDC matrix due to Lemma 3.1.

The major advantage of the formalism in Lemma 3.1 lies in its systematic connection to LTV systems based on uniform controllability and observability, adequately accounting for the nonlinear nature of underlying dynamics through  $A(\varrho, x, x_d, u_d, t)$  for global stability, as shall be seen in Chapter 3 and Chapter 4. Since  $A$  depends also on  $(x_d, u_d)$  in this case unlike the original SDC matrix, we could consider contraction metrics using a positive definite matrix  $M(x, x_d, u_d, t)$  instead of  $M(x, t)$  in Definition 2.3, to improve the representation power of  $M$  at the expense of computational efficiency. Another interesting point is that the non-uniqueness of  $A$  in Lemma 3.1 for  $n \geq 2$  creates additional degrees of freedom for selecting the coefficients  $\varrho$ , which can also be treated as decision variables in constructing optimal contraction metrics as proposed in [12], [13], [18].

We focus mostly on the generalized SDC formulation in Chapter 3 and Chapter 4, as it yields optimal control and estimation laws with global stability [17] while keeping the analysis simple enough to be understood as in LTV systems theory.

**Remark 3.2.** *This does not mean that contraction theory works only for the SDC parameterized nonlinear systems but implies that it can be used with the other techniques discussed in Sec. 3.1. For example, due to the extended linear form given in Table 3.1, the results to be presented in Chapter 3 and in Chapter 4 based on the SDC formulation are applicable to linearized dynamics that can be viewed as an LTV system with some modifications, regarding the dynamics modeling error term as an external disturbance as in Sec. 3.1.II. Also, the original SDC formulation with respect to a fixed point (e.g.,  $(s, \bar{s}, \bar{u}) = (x, 0, 0)$  in Lemma 3.1) can still be used to obtain contraction conditions independent of a target trajectory  $(x_d, u_d)$  (see Theorem 3.2 for details).*

### 3.1.IV Control Contraction Metric (CCM) Formulation

We could also consider using the partial derivative of  $f$  of the dynamical system directly for control synthesis through differential state feedback  $\delta u = k(x, \delta x, u, t)$ . This idea, formulated as the concept of a CCM [3], [14], [15], [19], [20], [33], constructs contraction metrics with global stability guarantees independently of target trajectories, achieving greater generality while requiring added computation in evaluating integrals involving minimizing geodesics. Similar to the CCM, we could design a state estimator using a general formulation based on geodesics distances between trajectories [34], [35]. These approaches are well compatible with the

convex optimization-based schemes in Chapter 4, and hence will be discussed in Sec. 4.3.

The differences between the SDC and CCM formulation are summarized in Table 3.1. Considering such trade-offs would help determine which form of the control law is the best fit when using contraction theory for nonlinear stabilization.

**Remark 3.3.** *For control non-affine nonlinear systems, we could find  $f(x, u, t) - f(x_d, u_d, t) = A(x, x_d, u, u_d, t)(x - x_d) + B(x, x_d, u, u_d, t)(u - u_d)$  by Lemma 3.1 on the SDC formulation and use it in Theorem 4.2, although (3.10) has to be solved implicitly as  $B$  depends on  $u$  in this case. A similar approach for the CCM formulation can be found in [14], [15]. As discussed in Example 3.2, designing such implicit control laws will be discussed in Lemma 8.2 and Theorem 8.4 of Sec. 8.2.II.*

### 3.2 LMI Conditions for Contraction Metrics

We design a nonlinear feedback tracking control law parameterized by a matrix-valued function  $M(x, x_d, u_d, t)$  (or  $M(x, t)$ , see Theorem 3.2) as follows:

$$\begin{aligned} u &= u_d - K(x, x_d, u_d, t)(x - x_d) \\ &= u_d - R(x, x_d, u_d, t)^{-1} B(x, t)^\top M(x, x_d, u_d, t)(x - x_d) \end{aligned} \quad (3.10)$$

where  $R(x, x_d, u_d, t) \succ 0$  is a weight matrix on the input  $u$  and  $M(x, x_d, u_d, t) \succ 0$  is a positive definite matrix (which satisfies the matrix inequality constraints for a contraction metric, to be given in Theorem 3.1). As discussed in Sec. 3.1.III, the extended linear form of the tracking control (3.10) enables LTV systems-type approaches to Lyapunov function construction, while being general enough to capture the nonlinearity of the underlying dynamics due to Lemma 3.2 [36].

**Lemma 3.2.** *Consider a general feedback controller  $u$  defined as  $u = k(x, x_d, u_d, t)$  with  $k(x_d, x_d, u_d, t) = u_d$ , where  $k : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^m$ . If  $k$  is piecewise continuously differentiable, then  $\exists K : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{m \times n}$  s.t.  $u = k(x, x_d, u_d, t) = u_d - K(x, x_d, u_d, t)(x - x_d)$ .*

*Proof.* Using  $k(x_d, x_d, u_d, t) = u_d$ ,  $u$  can be decomposed as  $u = u_d + (k(x, x_d, u_d, t) - k(x_d, x_d, u_d, t))$ . Since we have  $k(x, x_d, u_d, t) - k(x_d, x_d, u_d, t) = \int_0^1 (dk(cx + (1 - c)x_d, x_d, u_d, t)/dc)dc$ , selecting  $K$  as

$$K = - \int_0^1 \frac{\partial k}{\partial x}(cx + (1 - c)x_d, x_d, u_d, t)dc$$

gives the desired relation [36]. □

**Remark 3.4.** Lemma 3.2 implies that designing optimal  $k$  of  $u = k(x, x_d, u_d, t)$  reduces to designing the optimal gain  $K(x, x_d, u_d, t)$  of  $u = u_d - K(x, x_d, u_d, t)(x - x_d)$ . We could also generalize this idea further using the CCM-based differential feedback controller  $\delta u = k(x, \delta x, u, t)$  [3], [14], [15], [19], [20], [33] (see Theorem 4.6).

Substituting (3.10) into (3.1) and (3.2) yields the following virtual system of a smooth path  $q(\mu, t)$ , parameterized by  $\mu \in [0, 1]$  to have  $q(0, \mu) = x_d$  and  $q(1, t) = x$ , for partial contraction in Theorem 2.2:

$$\dot{q}(\mu, t) = \zeta(q(\mu, t), x, x_d, u_d, t) + d(\mu, x, t) \quad (3.11)$$

$$dq(\mu, t) = \zeta(q(\mu, t), x, x_d, u_d, t)dt + G(\mu, x, t)d\mathcal{W}(t) \quad (3.12)$$

where  $d(\mu, x, t) = \mu d_c(x, t)$ ,  $G(\mu, x, t) = \mu G_c(x, t)$ , and  $\zeta(q, x, x_d, u_d, t)$  is defined as

$$\zeta = (A(q, x, x_d, u_d, t) - B(x, t)K(x, x_d, u_d, t))(q - x_d) + f(x_d, t) + B(x_d, t)u_d \quad (3.13)$$

where  $A$  is the SDC matrix of Lemma 3.1 with  $(s, \bar{s}, \bar{u}) = (x, x_d, u_d)$ . Setting  $\mu = 1$  in (3.11) and (3.12) results in (3.1) and (3.2), respectively, and setting  $\mu = 0$  simply results in (3.3). Consequently, both  $q = x$  and  $q = x_d$  are particular solutions of (3.11) and (3.12). If there is no disturbance acting on the dynamics (3.1) and (3.2), the differential dynamics of (3.11) and (3.12) for  $\partial_\mu q = \partial q / \partial \mu$  is given as

$$\partial_\mu \dot{q} = (A(q, x, x_d, u_d, t) - B(x, t)K(x, x_d, u_d, t)) \partial_\mu q. \quad (3.14)$$

In [12], [13], [16], [17], it is proposed that the contraction conditions of Theorems 2.1 and 2.5 for the closed-loop dynamics (3.11) and (3.12) can be expressed as convex constraints as summarized in Theorem 3.1.

**Theorem 3.1.** Let  $\beta$  be defined as  $\beta = 0$  for deterministic systems (3.1) and

$$\beta = \alpha_s = L_m \bar{g}_c^2 (\alpha_G + 1/2)$$

for stochastic systems (3.2), respectively, where  $\bar{g}_c$  is given in (3.2),  $L_m$  is the Lipschitz constant of  $\partial M / \partial x_i$  for  $M$  of (3.10), and  $\alpha_G \in \mathbb{R}_{>0}$  is an arbitrary constant as in Theorem 2.5. Also, let  $W = M(x, x_d, u_d, t)^{-1}$  (or  $W = M(x, t)^{-1}$ , see

Theorem 3.2),  $\bar{W} = \nu W$ , and  $\nu = \bar{m}$ . Then the following three matrix inequalities are equivalent:

$$\dot{M} + M \frac{\partial \zeta}{\partial q} + \frac{\partial \zeta^\top}{\partial q} M \leq -2\alpha M - \beta \mathbf{I}, \forall \mu \in [0, 1] \quad (3.15)$$

$$\dot{M} + 2 \text{sym}(MA) - 2MBR^{-1}B^\top M \leq -2\alpha M - \beta \mathbf{I} \quad (3.16)$$

$$-\dot{\bar{W}} + 2 \text{sym}(A\bar{W}) - 2\nu BR^{-1}B^\top \leq -2\alpha \bar{W} - \frac{\beta}{\nu} \bar{W}^2 \quad (3.17)$$

where  $\zeta$  is as defined in (3.13). For stochastic systems with  $\beta = \alpha_s > 0$ , these inequalities are also equivalent to

$$\begin{bmatrix} -\dot{\bar{W}} + 2 \text{sym}(A\bar{W}) - 2\nu BR^{-1}B^\top + 2\alpha \bar{W} & \bar{W} \\ \bar{W} & -\frac{\nu}{\beta} \mathbf{I} \end{bmatrix} \leq 0. \quad (3.18)$$

Note that  $\nu$  and  $\bar{W}$  are required for (3.17) and (3.18) and the arguments  $(x, x_d, u_d, t)$  for each matrix are suppressed for notational simplicity.

Furthermore, under these equivalent contraction conditions, Theorems 2.4 and 2.5 hold for the virtual systems (3.11) and (3.12), respectively. In particular, if  $\underline{m}\mathbf{I} \leq M \leq \bar{m}\mathbf{I}$  of (2.26) holds, or equivalently

$$\mathbf{I} \leq \bar{W}(x, x_d, u_d, t) \leq \chi \mathbf{I} \quad (3.19)$$

holds for  $\chi = \bar{m}/\underline{m}$ , then we have the following bounds:

$$\|x(t) - x_d(t)\| \leq \frac{V_\ell(0)}{\sqrt{\underline{m}}} e^{-\alpha t} + \frac{\bar{d}_c}{\alpha} \sqrt{\chi} (1 - e^{-\alpha t}) \quad (3.20)$$

$$\mathbb{E} [\|x(t) - x_d(t)\|^2] \leq \frac{\mathbb{E}[V_{s\ell}(0)]}{\underline{m}} e^{-2\alpha t} + \frac{C_C}{2\alpha} \chi \quad (3.21)$$

where  $V_{s\ell} = \int_{x_d}^x \delta q^\top M \delta q$  and  $V_\ell = \int_{x_d}^x \|\Theta \delta q\|$  are as given in Theorem 2.3 with  $M = \Theta^\top \Theta$ , the disturbance bounds  $\bar{d}_c$  and  $\bar{g}_c$  are given in (3.1) and (3.2), respectively, and  $C_C = \bar{g}_c^2(2\alpha_G^{-1} + 1)$ . Note that for stochastic systems, the probability that  $\|x - x_d\|$  is greater than or equal to  $\varepsilon \in \mathbb{R}_{>0}$  is given as

$$\mathbb{P} [\|x(t) - x_d(t)\| \geq \varepsilon] \leq \frac{1}{\varepsilon^2} \left( \frac{\mathbb{E}[V_{s\ell}(0)]}{\underline{m}} e^{-2\alpha t} + \frac{C_C}{2\alpha} \chi \right). \quad (3.22)$$

*Proof.* Substituting (3.13) into (3.15) gives (3.16). Since  $\nu > 0$  and  $W > 0$ , multiplying (3.16) by  $\nu$  and then by  $W$  from both sides preserves matrix definiteness. Also, the resultant inequalities are equivalent to the original ones [37, p. 114]. These operations performed on (3.16) yield (3.17). If  $\beta = \alpha_s > 0$  for stochastic systems,

applying Schur's complement lemma [37, p. 7] to (3.17) results in the Linear Matrix Inequality (LMI) constraint (3.18) in terms of  $\bar{W}$  and  $\nu$ . Therefore, (3.15) – (3.18) are indeed equivalent.

Also, since we have  $\|\partial_\mu d(\mu, x, t)\| \leq \bar{d}_c$  for  $d$  in (3.11) and  $\|\partial_\mu G(\mu, x, t)\|_F^2 \leq \bar{g}_c^2$  for  $G$  in (3.12), the virtual systems in (3.11) and (3.12) clearly satisfy the conditions of Theorems 2.4 and 2.5 if it is equipped with (3.15), which is equivalent to (3.16) – (3.18). This implies the exponential bounds (3.20) – (3.22) rewritten using  $\chi = \bar{m}/m$ , following the proofs of Theorems 2.4 and 2.5.  $\square$

Because of the control and estimation duality in differential dynamics similar to that of the Kalman filter and Linear Quadratic Regulator (LQR) in LTV systems, we have an analogous robustness result for the contraction theory-based state estimator as to be derived in Sec. 4.2.

Although the conditions (3.15) – (3.18) depend on  $(x_d, u_d)$ , we could also use the SDC formulation with respect to a fixed point [12], [13] in Lemma 3.1 to make them independent of the target trajectory as in the following theorem.

**Theorem 3.2.** *Let  $(\bar{x}, \bar{u})$  be a fixed point selected arbitrarily in  $\mathbb{R}^n \times \mathbb{R}^m$ , e.g.,  $(\bar{x}, \bar{u}) = (0, 0)$ , and let  $A(x, t)$  be an SDC matrix constructed with  $(s, \bar{s}, \bar{u}) = (x, \bar{x}, \bar{u})$  in Lemma 3.1, i.e.,*

$$A(\varrho, x, t)(x - \bar{x}) = f(x, t) + B(x, t)\bar{u} - f(\bar{x}, t) - B(\bar{x}, t)\bar{u}. \quad (3.23)$$

*Suppose that the contraction metric of Theorem 3.1 is designed by  $M(x, t)$  with  $A$  of (3.23), independently of the target trajectory  $(x_d, u_d)$ , and that the systems (3.1) and (3.2) are controlled by*

$$u = u_d - R(x, t)^{-1}B(x, t)^\top M(x, t)(x - x_d) \quad (3.24)$$

*with such  $M(x, t)$ , where  $R(x, t) \succ 0$  is a weight matrix on  $u$ . If the function  $\phi(x, x_d, u_d, t) = A(\varrho, x, t)(x_d - \bar{x}) + B(x, t)(u_d - \bar{u})$  is Lipschitz in  $x$  with its Lipschitz constant  $\bar{L}$ , then Theorem 3.1 still holds with  $\alpha$  of the conditions (3.15) – (3.18) replaced by  $\alpha + \bar{L}\sqrt{\bar{m}/m}$ . The same argument holds for state estimation of Theorem 4.3 to be discussed in Sec. 4.2.*

*Proof.* The unperturbed virtual system of (3.1), (3.2), and (3.3) with  $A$  of (3.23) and  $u$  of (3.24) is given as follows:

$$\begin{aligned} \dot{q} = & (A(\varrho, x, t) - B(x, t)K(x, t))(q - x_d) + A(\varrho, q, t)(x_d - \bar{x}) + B(q, t)(u_d - \bar{u}) \\ & + f(\bar{x}, t) + B(\bar{x}, t)\bar{u} \end{aligned} \quad (3.25)$$

where  $K(x, t) = R(x, t)^{-1}B(x, t)^\top M(x, t)$ . Following the proof of Theorem 3.1, the computation of  $\dot{V}$ , where  $V = \delta q^\top M(x, t)\delta q$ , yields an extra term

$$2\delta q^\top M \frac{\partial \phi}{\partial q}(q, x_d, u_d, t)\delta q \leq 2\bar{L}\sqrt{\frac{\bar{m}}{\underline{m}}}\delta q^\top M\delta q \quad (3.26)$$

due to the Lipschitz condition on  $\phi$ , where  $\phi(q, x_d, u_d, t) = A(q, t)(x_d - \bar{x}) + B(q, t)(u_d - \bar{u})$ . This indeed implies that the system (3.25) is contracting as long as the conditions (3.15) – (3.18) hold with  $\alpha$  replaced by  $\alpha + \bar{L}\sqrt{\bar{m}/\underline{m}}$ . The last statement on state estimation follows from the nonlinear control and estimation duality to be discussed in Sec. 4.2.  $\square$

**Remark 3.5.** *As in [12], we could directly use the extra term  $2\delta q^\top M(\partial\phi/\partial q)\delta q$  of (3.26) in (3.15) – (3.18) without upper-bounding it, although now the constraints of Theorem 3.2 depend on  $(x, q, t)$  instead of  $(x, t)$ . Also, the following two inequalities given in [12] with  $\bar{\gamma} = \nu\gamma$ ,  $\gamma \in \mathbb{R}_{\geq 0}$ :*

$$\begin{aligned} -\dot{\bar{W}} + A\bar{W} + \bar{W}A^\top + \bar{\gamma}I - \nu BR^{-1}B^\top &\leq 0 \\ \begin{bmatrix} \bar{\gamma}I + \nu BR^{-1}B^\top - \bar{W}\phi^\top - \phi\bar{W} - 2\alpha\bar{W} & \bar{W} \\ \bar{W} & \frac{\nu}{2\alpha_s}I \end{bmatrix} &\geq 0. \end{aligned}$$

are combined as one LMI (3.18) in Theorems 3.1 and 3.2.

**Example 3.4.** *The inequalities in Theorem 3.1 can be interpreted as in the Riccati inequality in  $\mathcal{H}_\infty$  control. Consider the following system:*

$$\dot{x} = Ax + B_u u + B_w w, \quad z = C_z x \quad (3.27)$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $B_u \in \mathbb{R}^{n \times m}$ ,  $B_w \in \mathbb{R}^{n \times w}$ , and  $C_z \in \mathbb{R}^{o \times n}$  are constant matrices,  $w \in \mathbb{R}^w$  is an exogenous input, and  $z \in \mathbb{R}^o$  is a system output. As shown in [4] and [37, p. 109], there exists a state feedback gain  $K = R^{-1}B_u^\top P$  such that the  $\mathcal{L}_2$  gain of the closed-loop system (3.27),  $\sup_{\|w\| \neq 0} \|z\|/\|w\|$ , is less than or equal to  $\gamma$  if

$$2 \operatorname{sym}(PA) - 2PB_u R^{-1}B_u^\top P + \frac{PB_w B_w^\top P}{\gamma^2} + C_z^\top C_z \leq 0 \quad (3.28)$$

has a solution  $P > 0$ , where  $R > 0$  is a constant weight matrix on the input  $u$ . If we select  $B_w$  and  $C_z$  to have  $B_w B_w^\top \geq (P^{-1})^2$  and  $C_z^\top C_z \geq 2\alpha P$  for some  $\alpha > 0$ , the contraction condition (3.16) in Theorem 3.1 can be satisfied with  $M = P$ ,  $B = B_u$ , and  $\beta = 1/\gamma^2$  due to (3.28).

In Sec. 3.3 and Sec. 3.4, we will discuss the relationship to input-output stability theory as in Example 3.4, using the results of Theorem 3.1.

### 3.3 Bounded Real Lemma in Contraction Theory

The LMI (3.18) of Theorem 3.1 can be interpreted as the LMI condition for the bounded real lemma [38, p. 369]. Let us consider the following Hurwitz linear system with  $\varsigma(t) \in \mathcal{L}_{2e}$ , (i.e.,  $\|(\varsigma)_\tau\|_{\mathcal{L}_2} < \infty$  for  $\tau \in \mathbb{R}_{\geq 0}$ , see Sec. 1.3):

$$\dot{\kappa} = \mathcal{A}\kappa + \mathcal{B}\varsigma(t), \quad v = C\kappa + \mathcal{D}\varsigma(t). \quad (3.29)$$

Setting  $\kappa = \delta q$  and viewing  $\varsigma$  as external disturbance, this system can be interpreted as the differential closed-loop dynamics defined earlier in (3.14). The bounded real lemma states that this system is  $\mathcal{L}_2$  gain stable with its  $\mathcal{L}_2$  gain less than or equal to  $\gamma$ , i.e.,  $\|(v)_\tau\|_{\mathcal{L}_2} \leq \gamma\|(\varsigma)_\tau\|_{\mathcal{L}_2} + \text{const.}$ , or equivalently, the  $\mathcal{H}_\infty$  norm of the transfer function of (3.29) is less than or equal to  $\gamma$ , if the following LMI for  $\mathcal{P} > 0$  holds (see [38, p. 369]):

$$\begin{bmatrix} \dot{\mathcal{P}} + 2 \text{sym}(\mathcal{P}\mathcal{A}) + C^\top C & \mathcal{P}\mathcal{B} + C^\top \mathcal{D} \\ \mathcal{B}^\top \mathcal{P} + \mathcal{D}^\top C & \mathcal{D}^\top \mathcal{D} - \gamma^2 \mathbf{I} \end{bmatrix} \leq 0. \quad (3.30)$$

Theorem 3.3 introduces the bounded real lemma in the context of contraction theory.

**Theorem 3.3.** *Let  $\mathcal{A} = A - BR^{-1}B^\top M$ ,  $\mathcal{B} = M^{-1}$ ,  $C = \sqrt{2\alpha}\Theta$ , and  $\mathcal{D} = 0$  in (3.29), where  $M = \Theta^\top \Theta$ , and the other variables are as defined in Theorem 3.1. Then (3.30) with  $\mathcal{P} = M$  and  $\gamma = 1/\sqrt{\beta}$  is equivalent to (3.18), and thus (3.18) implies  $\mathcal{L}_2$  gain stability of (3.29) with its  $\mathcal{L}_2$  gain less than or equal to  $\gamma$ .*

*Proof.* Multiplying (3.18) by  $v^{-1}$  and then by  $\begin{bmatrix} M & 0 \\ 0 & \mathbf{I} \end{bmatrix}$  from both sides gives the following matrix inequality:

$$v \begin{bmatrix} \dot{M} + 2 \text{sym}(M(A - BR^{-1}B^\top M)) + 2\alpha M & \mathbf{I} \\ \mathbf{I} & -\frac{1}{\beta} \mathbf{I} \end{bmatrix} \leq 0.$$

This is indeed equivalent to (3.30) if  $\mathcal{A} = A - BR^{-1}B^\top M$ ,  $\mathcal{B} = M^{-1}$ ,  $C = \sqrt{2\alpha}\Theta$ ,  $\mathcal{D} = 0$ ,  $\mathcal{P} = M$ , and  $\gamma = 1/\sqrt{\beta}$ . Now, multiplying (3.30) by  $[\delta q^\top, \varsigma^\top]^\top = [\kappa^\top, \varsigma^\top]^\top$  for such  $(\mathcal{A}, \mathcal{B}, C, \mathcal{D})$  gives

$$\kappa^\top (\dot{\mathcal{P}} + 2\mathcal{P}\mathcal{A} + C^\top C)\kappa + 2\kappa^\top \mathcal{P}\mathcal{B}\varsigma - \gamma^2 \|\varsigma\|^2 \leq 0$$

resulting in  $\dot{V} + \|v\|^2 - \gamma^2 \|\varsigma\|^2 \leq 0$  for  $V = \kappa^\top \mathcal{P}\kappa = \delta q^\top M \delta q$ . This implies  $\mathcal{L}_2$  gain stability with its  $\mathcal{L}_2$  gain less than or equal to  $\gamma = 1/\sqrt{\beta}$  [25, p. 209].  $\square$

### 3.4 KYP Lemma in Contraction Theory

Analogously to Theorem 3.3, the LMI (3.18) of Theorem 3.1 can be understood using the Kalman-Yakubovich-Popov (KYP) lemma [39, p. 218]. Consider the quadratic Lyapunov function  $V = \kappa^\top Q \kappa$  with  $Q > 0$ , satisfying the following output strict passivity (dissipativity) condition:

$$\dot{V} - 2\zeta^\top v + (2/\gamma)v^\top v \leq 0. \quad (3.31)$$

This which can be expanded by completing the square to have

$$\begin{aligned} \dot{V} &\leq -\|\gamma(\zeta - v/\gamma)\|^2 + \gamma\|\zeta\|^2 - (1/\gamma)\|v\|^2 \\ &\leq \gamma\|\zeta\|^2 - (1/\gamma)\|v\|^2. \end{aligned} \quad (3.32)$$

This implies that we have  $\|(v)_\tau\|_{\mathcal{L}_2} \leq \gamma\|(\zeta)_\tau\|_{\mathcal{L}_2} + \sqrt{\gamma V(0)}$  by the comparison lemma [25, pp. 102-103, pp. 350-353], leading to  $\mathcal{L}_2$  gain stability with its  $\mathcal{L}_2$  gain less than or equal to  $\gamma$  [39, p. 218]. The condition (3.31) can be expressed equivalently as an LMI form as follows:

$$\begin{bmatrix} \dot{Q} + 2\text{sym}(Q\mathcal{A}) + \frac{2C^\top C}{\gamma} & Q\mathcal{B} + C^\top \left( \frac{2\mathcal{D}}{\gamma} - I \right) \\ \mathcal{B}^\top Q + \left( \frac{2\mathcal{D}^\top}{\gamma} - I \right) C & -(\mathcal{D}^\top + \mathcal{D}) + \frac{2\mathcal{D}^\top \mathcal{D}}{\gamma} \end{bmatrix} \leq 0 \quad (3.33)$$

where  $\mathcal{A}$ ,  $\mathcal{B}$ ,  $C$ , and  $\mathcal{D}$  are as defined in (3.29).

**Theorem 3.4.** *If (3.33) holds, the LMI (3.30) for the bounded real lemma holds with  $\mathcal{P} = \gamma Q$ , i.e., the system (3.29) is  $\mathcal{L}_2$  gain stable with its  $\mathcal{L}_2$  gain less than or equal to  $\gamma$ . Thus, for systems with  $\mathcal{A}$ ,  $\mathcal{B}$ ,  $C$ , and  $\mathcal{D}$  defined in Theorem 3.3, the condition (3.33) guarantees the contraction condition (3.18) of Theorem 3.1.*

*Proof.* Writing the inequality in (3.32) in a matrix form, we have that

$$\begin{aligned} &\begin{bmatrix} \dot{Q} + 2\text{sym}(Q\mathcal{A}) + \frac{2C^\top C}{\gamma} & Q\mathcal{B} + C^\top \left( \frac{2\mathcal{D}}{\gamma} - I \right) \\ \mathcal{B}^\top Q + \left( \frac{2\mathcal{D}^\top}{\gamma} - I \right) C & -(\mathcal{D}^\top + \mathcal{D}) + \frac{2\mathcal{D}^\top \mathcal{D}}{\gamma} \end{bmatrix} \\ &\geq \begin{bmatrix} \dot{Q} + 2\text{sym}(Q\mathcal{A}) & Q\mathcal{B} \\ \mathcal{B}^\top Q & 0 \end{bmatrix} + \begin{bmatrix} \frac{C^\top C}{\gamma} & \frac{C^\top \mathcal{D}}{\gamma} \\ \frac{\mathcal{D}^\top C}{\gamma} & \frac{\mathcal{D}^\top \mathcal{D}}{\gamma} \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & \gamma I \end{bmatrix} \\ &= \begin{bmatrix} \dot{Q} + 2\text{sym}(Q\mathcal{A}) + \frac{C^\top C}{\gamma} & Q\mathcal{B} + \frac{C^\top \mathcal{D}}{\gamma} \\ \mathcal{B}^\top Q + \frac{\mathcal{D}^\top C}{\gamma} & \frac{\mathcal{D}^\top \mathcal{D}}{\gamma} - \gamma I \end{bmatrix}. \end{aligned}$$

Therefore, a necessary condition of (3.33) reduces to (3.30) if  $Q = \mathcal{P}/\gamma$ . The rest follows from Theorem 3.3.  $\square$

**Example 3.5.** *Theorems 3.3 and 3.4 imply the following statements.*

- *If (3.30) holds, the system is finite-gain  $\mathcal{L}_2$  stable with  $\gamma$  as its  $\mathcal{L}_2$  gain.*
- *If (3.33) holds, the system is finite-gain  $\mathcal{L}_2$  stable with  $\gamma$  as its  $\mathcal{L}_2$  gain.*
- *(3.31) is equivalent to (3.33), and to the output strict passivity condition with dissipation  $1/\gamma$  [25, p. 231].*
- *If (3.33) holds (KYP lemma), (3.30) holds (bounded real lemma). The converse is not necessarily true.*

## References

- [1] T. Basar and P. Bernhard,  *$\mathcal{H}_\infty$  Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Birkhauser, 1995.
- [2] W. Zhang and B. Chen, “State feedback  $\mathcal{H}_\infty$  control for a class of nonlinear stochastic systems,” *SIAM J. Control Optim.*, vol. 44, no. 6, pp. 1973–1991, 2006.
- [3] I. R. Manchester and J.-J. E. Slotine, “Robust control contraction metrics: A convex approach to nonlinear state-feedback  $\mathcal{H}^\infty$  control,” *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 333–338, 2018.
- [4] A. J. Van der Schaft, “ $\mathcal{L}_2$ -gain analysis of nonlinear systems and nonlinear state-feedback  $\mathcal{H}_\infty$  control,” *IEEE Trans. Autom. Control*, vol. 37, no. 6, pp. 770–784, Jun. 1992.
- [5] A. Isidori and A. Astolfi, “Disturbance attenuation and  $\mathcal{H}_\infty$ -control via measurement feedback in nonlinear systems,” *IEEE Trans. Autom. Control*, vol. 37, no. 9, pp. 1283–1293, Sep. 1992.
- [6] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, “State-space solutions to standard  $\mathcal{H}_2$  and  $\mathcal{H}_\infty$  control problems,” *IEEE Trans. Autom. Control*, vol. 34, no. 8, pp. 831–847, Aug. 1989.
- [7] P. P. Khargonekar and M. A. Rotea, “Mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  control: A convex optimization approach,” *IEEE Trans. Autom. Control*, vol. 36, no. 7, pp. 824–837, Jul. 1991.
- [8] L. Xie and E. de Souza Carlos, “Robust  $\mathcal{H}_\infty$  control for linear systems with norm-bounded time-varying uncertainty,” *IEEE Trans. Autom. Control*, vol. 37, no. 8, pp. 1188–1191, Aug. 1992.
- [9] C. Scherer, P. Gahinet, and M. Chilali, “Multiobjective output-feedback control via LMI optimization,” *IEEE Trans. Autom. Control*, vol. 42, no. 7, pp. 896–911, Jul. 1997.
- [10] B.-S. Chen and W. Zhang, “Stochastic  $\mathcal{H}_2/\mathcal{H}_\infty$  control with state-dependent noise,” *IEEE Trans. Autom. Control*, vol. 49, no. 1, pp. 45–57, Jan. 2004, issn: 0018-9286.

- [11] D. Hinrichsen and A. Pritchard, “Stochastic  $\mathcal{H}_\infty$ ,” *SIAM J. Control Optim.*, vol. 36, no. 5, pp. 1504–1538, 1998.
- [12] H. Tsukamoto and S.-J. Chung, “Robust controller design for stochastic nonlinear systems via convex optimization,” *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4731–4746, 2021.
- [13] H. Tsukamoto and S.-J. Chung, “Convex optimization-based controller design for stochastic nonlinear systems using contraction analysis,” in *IEEE Conf. Decis. Control*, Dec. 2019, pp. 8196–8203.
- [14] R. Wang, R. Tóth, and I. R. Manchester, *Virtual CCMs: Convex nonlinear feedback design via behavioral embedding*, arXiv:2003.08513, Mar. 2020.
- [15] R. Wang, R. Tóth, and I. R. Manchester, “A comparison of LPV gain scheduling and control contraction metrics for nonlinear control,” in *3rd IFAC Workshop on LPVS*, vol. 52, 2019, pp. 44–49.
- [16] H. Tsukamoto and S.-J. Chung, “Neural contraction metrics for robust estimation and control: A convex optimization approach,” *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 211–216, 2021.
- [17] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Neural stochastic contraction metrics for learning-based control and estimation,” *IEEE Control Syst. Lett.*, vol. 5, no. 5, pp. 1825–1830, 2021.
- [18] A. P. Dani, S.-J. Chung, and S. Hutchinson, “Observer design for stochastic nonlinear systems via contraction-based incremental stability,” *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 700–714, Mar. 2015.
- [19] I. R. Manchester and J.-J. E. Slotine, “Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design,” *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 3046–3053, Jun. 2017.
- [20] S. Singh, A. Majumdar, J.-J. E. Slotine, and M. Pavone, “Robust online motion planning via contraction theory and convex optimization,” in *IEEE Int. Conf. Robot. Automat.*, May 2017, pp. 5883–5890.
- [21] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Upper Saddle River, NJ: Pearson, 1991.
- [22] F. Tan, W. Lohmiller, and J.-J. E. Slotine, “Analytical SLAM without linearization,” *Int. J. Robot. Res.*, vol. 36, no. 13-14, pp. 1554–1578, 2017.
- [23] E. Sontag, “A Lyapunov-like characterization of asymptotic controllability,” *SIAM J. Control Optim.*, vol. 21, no. 3, pp. 462–471, 1983.
- [24] E. D. Sontag, “A ‘universal’ construction of Artstein’s theorem on nonlinear stabilization,” *Syst. Control Lett.*, vol. 13, no. 2, pp. 117–123, 1989.
- [25] H. K. Khalil, *Nonlinear Systems*, 3rd. Upper Saddle River, NJ: Prentice-Hall, 2002.

- [26] W. Li and E. Todorov, “Iterative linear quadratic regulator design for nonlinear biological movement systems,” in *Int. Conf. Inform. Control Automat. Robot.*, 2004, pp. 222–229.
- [27] W. Li and E. Todorov, “An iterative optimal control and estimation design for nonlinear stochastic system,” in *IEEE Conf. Decis. Control*, 2006, pp. 3242–3247.
- [28] S. Bonnabel and J.-J. E. Slotine, “A contraction theory-based analysis of the stability of the deterministic extended Kalman filter,” *IEEE Trans. Autom. Control*, vol. 60, no. 2, pp. 565–569, Feb. 2015.
- [29] J. R. Cloutier, “State-dependent Riccati equation techniques: An overview,” in *Proc. Amer. Control Conf.*, vol. 2, 1997, pp. 932–936.
- [30] H. T. Banks, B. M. Lewis, and H. T. Tran, “Nonlinear feedback controllers and compensators: A state-dependent Riccati equation approach,” *Comput. Optim. Appl.*, vol. 37, no. 2, pp. 177–218, Jun. 2007, ISSN: 1573-2894.
- [31] I. Chang and S.-J. Chung, “Exponential stability region estimates for the state-dependent Riccati Equation controllers,” in *IEEE Conf. Decis. Control*, 2009, pp. 1974–1979.
- [32] T. Çimen, “Survey of state-dependent Riccati equation in nonlinear optimal feedback control synthesis,” *AIAA J. Guid. Control Dyn.*, vol. 35, no. 4, pp. 1025–1047, Jul. 2012.
- [33] S. Singh, S. M. Richards, V. Sindhvani, J.-J. E. Slotine, and M. Pavone, “Learning stabilizable nonlinear dynamics with contraction-based regularization,” *Int. J. Robot. Res.*, Aug. 2020.
- [34] Q. C. Pham and J.-J. E. Slotine, *Stochastic contraction in Riemannian metrics*, arXiv:1304.0340, Apr. 2013.
- [35] I. R. Manchester, “Contracting nonlinear observers: Convex optimization and learning from data,” in *Proc. Amer. Control Conf.*, 2018, pp. 1873–1880.
- [36] H. Tsukamoto and S.-J. Chung, “Learning-based robust motion planning with guaranteed stability: A contraction theory approach,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6164–6171, 2021.
- [37] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory* (Studies in Applied Mathematics). Philadelphia, PA: SIAM, Jun. 1994, vol. 15, ISBN: 0-89871-334-X.
- [38] T. Başar and P. Bernhard,  *$\mathcal{H}_\infty$  Optimal Control and Related Minimax Design Problems, A Dynamic Game Approach* (Modern Birkhäuser Classics), 2nd. Birkhäuser Basel, 2008, ISBN: 978-0-387-98945-7.

- [39] G. E. Dullerud and F. Paganini, *A Course in Robust Control Theory, A Convex Approach* (Texts in Applied Mathematics), 1st. Springer, 2000, ISBN: 978-0-387-98945-7.

## CONVEX OPTIMALITY IN ROBUST NONLINEAR CONTROL AND ESTIMATION VIA CONTRACTION THEORY

- [1] H. Tsukamoto and S.-J. Chung, “Robust controller design for stochastic nonlinear systems via convex optimization,” *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4731–4746, 2021.
- [2] H. Tsukamoto and S.-J. Chung, “Convex optimization-based controller design for stochastic nonlinear systems using contraction analysis,” in *IEEE Conf. Decis. Control*, Dec. 2019, pp. 8196–8203.

Theorem 3.1 indicates that the problem of finding contraction metrics for general nonlinear systems could be formulated as a convex feasibility problem. This chapter thus delineates one approach, called the method of ConVex optimization-based Steady-state Tracking Error Minimization (CV-STEM) [1]–[4], to optimally design  $M$  of Theorem 3.1 that defines a contraction metric and minimizes an upper bound of the steady-state tracking error in Theorem 2.4 or in Theorem 2.5 via convex optimization. In particular, we present an overview of the SDC- and CCM-based CV-STEM frameworks for provably stable and optimal feedback control and state estimation of nonlinear systems, perturbed by deterministic and stochastic disturbances. It is worth noting that the steady-state bound is expressed as a function of the condition number  $\chi = \bar{m}/\underline{m}$  as to be seen in (4.1) and (4.17), which renders the CV-STEM applicable and effective even to learning-based and data-driven control frameworks as shall be seen in Chapter 5.

### 4.1 CV-STEM Control

As a result of Theorems 2.4, 2.5, and 3.1, the control law  $u = u_d - K(x - x_d)$  of (3.10) gives a convex steady-state upper bound of the Euclidean distance between the system trajectories, which can be used as an objective function for the CV-STEM control framework in Theorem 4.2 [1]–[4].

**Theorem 4.1.** *If one of the matrix inequalities of Theorem 3.1 holds, we have the following bound for  $\chi = \bar{m}/\underline{m}$ :*

$$\lim_{t \rightarrow \infty} \sqrt{\mathbb{E} [\|x - x_d\|^2]} \leq c_0(\alpha, \alpha_G) \sqrt{\chi} \leq c_0(\alpha, \alpha_G) \chi \quad (4.1)$$

where  $c_0(\alpha, \alpha_G) = \bar{g}_c \sqrt{(2\alpha_G^{-1} + 1)/(2\alpha)}$  for stochastic systems and  $c_0(\alpha, \alpha_G) = \bar{d}_c/\alpha$  for deterministic systems, with the variables given in Theorem 3.1.

*Proof.* Taking  $\lim_{t \rightarrow \infty}$  in the exponential tracking error bounds (3.20) and (3.21) of Theorem 3.1 gives the first inequality of (4.1). The second inequality follows from the relation  $1 \leq \sqrt{\bar{m}/\underline{m}} \leq \bar{m}/\underline{m} = \chi$  due to  $\underline{m} \leq \bar{m}$ .  $\square$

The convex optimization problem of minimizing (4.1), subject to the contraction constraint of Theorem 3.1, is given in Theorem 4.2, thereby introducing the CV-STEM control [1]–[4] for designing an optimal contraction metric and contracting control policy as depicted in Fig. 4.1.

**Theorem 4.2.** *Suppose that  $\alpha$ ,  $\alpha_G$ ,  $\bar{d}_c$ , and  $\bar{g}_c$  in (4.1) and the Lipschitz constant  $L_m$  of  $\partial M/\partial x_i$  in Theorem 3.1 are given. If the pair  $(A, B)$  is uniformly controllable, the non-convex optimization problem of minimizing the upper bound (4.1) is equivalent to the following convex optimization problem, with the convex contraction constraint (3.17) or (3.18) of Theorem 3.1 and  $I \leq \bar{W} \leq \chi I$  of (3.19):*

$$J_{CV}^* = \min_{\nu \in \mathbb{R}_{>0}, \chi \in \mathbb{R}, \bar{W} > 0} c_0 \chi + c_1 \nu + c_2 P(\chi, \nu, \bar{W}) \quad (4.2)$$

s.t. (3.17) and (3.19) for deterministic systems

(3.18) and (3.19) for stochastic systems

where  $W = M(x, x_d, u_d, t)^{-1}$  for Theorem 3.1 or  $W = M(x, t)^{-1}$  for Theorem 3.2,  $\bar{W} = \nu W$ ,  $\nu = \bar{m}$ ,  $\chi = \bar{m}/\underline{m}$ ,  $c_0$  is as defined in (4.1) of Theorem 4.1,  $c_1, c_2 \in \mathbb{R}_{\geq 0}$ , and  $P$  is a performance-based convex cost function.

The weight  $c_1$  for  $\nu$  can be viewed as a penalty on the 2-norm of the feedback control gain  $K$  of  $u = u_d - K(x - x_d)$  in (3.10). Using non-zero  $c_1$  and  $c_2$  thus enables finding contraction metrics optimal in a different sense. Furthermore, the coefficients of the SDC parameterizations  $\varrho$  in Lemma 3.1 (i.e.,  $A = \sum \varrho_i A_i$  in (3.6)) can also be treated as decision variables by convex relaxation, thereby adding a design flexibility to mitigate the effects of external disturbances while verifying the system controllability.

*Proof.* As proven in Theorems 3.1 and 4.1,  $\underline{m}I \leq M \leq \bar{m}I$  of (2.26), the contraction constraint (3.15), and the objective (4.1) reduce to (4.2) with  $c_1 = 0$  and  $c_2 = 0$ . Since the resultant constraints are convex and the objective is affine in terms of the decision variables  $\nu$ ,  $\chi$ , and  $\bar{W}$ , the problem (4.2) is indeed convex. Since we have

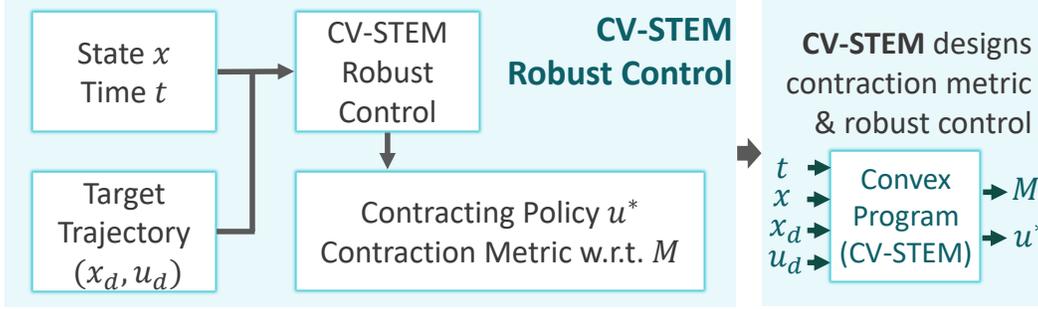


Figure 4.1: Block diagram of CV-STEM.

$\sup_{x,t} \|M\| \leq \bar{m} = \nu$ ,  $c_1$  can be used as a penalty to optimally adjust the induced 2-norm of the control gain (see Example 4.1). The SDC coefficients  $\varrho$  can also be utilized as decision variables for controllability due to Proposition 1 of [1].  $\square$

**Example 4.1.** *The weights  $c_0$  and  $c_1$  of the CV-STEM control of Theorem 4.2 establish an analogous trade-off to the case of the LQR with the cost weight matrices of  $Q$  for state and  $R$  for control, since the upper bound of the steady-state tracking error (4.1) is proportional to  $\chi$ , and an upper bound of  $\|K\|$  for the control gain  $K$  in (3.10),  $u = u_d - K(x - x_d)$ , is proportional to  $\nu$ . In particular [2], [3],*

- if  $c_1$  is much greater than  $c_0$  in (4.2) of Theorem 4.2, we get smaller control effort but with a large steady-state tracking error, and
- if  $c_1$  is much smaller than  $c_0$  in (4.2) of Theorem 4.2, we get a smaller steady-state state tracking error but with larger control effort.

This is also because the solution  $P > 0$  of the LQR Riccati equation [5, p. 89],  $-\dot{P} = PA + A^T P - PBR^{-1}B^T P + Q$ , can be viewed as a positive definite matrix  $M$  that defines a contraction metric as discussed in Example 2.3.

## 4.2 CV-STEM Estimation

We could also design an optimal state estimator analogously to the CV-STEM control of Theorem 4.2, due to the differential nature of contraction theory that enables LTV systems-type approaches to stability analysis. In particular, we exploit the estimation and control duality in differential dynamics similar to that of the Kalman filter and LQR in LTV systems.

Let us consider the following smooth nonlinear systems with a measurement  $y(t)$ , perturbed by deterministic disturbances  $d_{e0}(x, t)$  and  $d_{e1}(x, t)$  with  $\sup_{x,t} \|d_{e0}(x, t)\| = \bar{d}_{e0} \in \mathbb{R}_{\geq 0}$  and  $\sup_{x,t} \|d_{e1}(x, t)\| = \bar{d}_{e1} \in \mathbb{R}_{\geq 0}$ , or by Gaussian white noise, driven

by Wiener processes  $\mathcal{W}_0(t)$  and  $\mathcal{W}_1(t)$  with  $\sup_{x,t} \|G_{e0}(x,t)\|_F = \bar{g}_{e0} \in \mathbb{R}_{\geq 0}$  and  $\sup_{x,t} \|G_{e1}(x,t)\|_F = \bar{g}_{e1} \in \mathbb{R}_{\geq 0}$ :

$$\dot{x} = f(x,t) + d_{e0}(x,t), \quad y = h(x,t) + d_{e1}(x,t) \quad (4.3)$$

$$dx = f(x,t)dt + G_{e0}d\mathcal{W}_0, \quad ydt = h(x,t)dt + G_{e1}d\mathcal{W}_1 \quad (4.4)$$

where  $t \in \mathbb{R}_{\geq 0}$  is time,  $x : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  is the system state,  $y : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^m$  is the system measurement,  $f : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  and  $h : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^m$  are known smooth functions,  $d_{e0} : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$ ,  $d_{e1} : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^0$ ,  $G_{e0} : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{n \times w_0}$ , and  $G_{e1} : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{n \times w_1}$  are unknown bounded functions for external disturbances,  $\mathcal{W}_0 : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{w_0}$  and  $\mathcal{W}_1 : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{w_1}$  are two independent Wiener processes, and the arguments of  $G_{e0}(x,t)$  and  $G_{e1}(x,t)$  are suppressed for notational convenience. Let  $A(\varrho_a, x, \hat{x}, t)$  and  $C(\varrho_c, x, \hat{x}, t)$  be the SDC matrices given by Lemma 3.1 with  $(f, s, \bar{s}, \bar{u})$  replaced by  $(f, \hat{x}, x, 0)$  and  $(h, \hat{x}, x, 0)$ , respectively, i.e.

$$A(\varrho_a, x, \hat{x}, t)(\hat{x} - x) = f(\hat{x}, t) - f(x, t) \quad (4.5)$$

$$C(\varrho_c, x, \hat{x}, t)(\hat{x} - x) = h(\hat{x}, t) - h(x, t). \quad (4.6)$$

We design a nonlinear state estimation law parameterized by a matrix-valued function  $M(\hat{x}, t)$  as follows:

$$\begin{aligned} \dot{\hat{x}} &= f(\hat{x}, t) + L(\hat{x}, t)(y - h(\hat{x}, t)) \\ &= f(\hat{x}, t) + M(\hat{x}, t)\bar{C}(\varrho_c, \hat{x}, t)^\top R(\hat{x}, t)^{-1}(y - h(\hat{x}, t)) \end{aligned} \quad (4.7)$$

where  $\bar{C}(\varrho_c, \hat{x}, t) = C(\varrho_c, \hat{x}, \bar{x}, t)$  for a fixed trajectory  $\bar{x}$  (e.g.,  $\bar{x} = 0$ , see Theorem 3.2),  $R(\hat{x}, t) > 0$  is a weight matrix on the measurement  $y$ , and  $M(\hat{x}, t) > 0$  is a positive definite matrix (which satisfies the matrix inequality constraint for a contraction metric, to be given in (4.12) of Theorem 4.5). Note that we could use other forms of estimation laws such as the EKF [2], [6], [7], analytical SLAM [8], or SDC with respect to a fixed point [1], [4], [9], depending on the application of interest, which result in a similar stability analysis as in Theorem 3.2.

#### 4.2.I Nonlinear Stability Analysis of SDC-based State Estimation using Contraction Theory

Substituting (4.7) into (4.3) and (4.4) yields the following virtual system of a smooth path  $q(\mu, t)$ , parameterized by  $\mu \in [0, 1]$  to have  $q(\mu = 0, t) = x$  and  $q(\mu = 1, t) = \hat{x}$ :

$$\dot{q}(\mu, t) = \zeta(q(\mu, t), x, \hat{x}, t) + d(\mu, x, \hat{x}, t) \quad (4.8)$$

$$dq(\mu, t) = \zeta(q(\mu, t), x, \hat{x}, t)dt + G(\mu, x, \hat{x}, t)d\mathcal{W}(t) \quad (4.9)$$

where  $d(\mu, x, \hat{x}, t) = (1 - \mu)d_{e0}(x, t) + \mu L(\hat{x}, t)d_{e1}(x, t)$ ,  $G(\mu, x, \hat{x}, t) = [(1 - \mu)G_{e0}(x, t), \mu L(\hat{x}, t)G_{e1}(x, t)]$ ,  $\mathcal{W} = [\mathcal{W}_0^\top, \mathcal{W}_1^\top]^\top$ , and  $\zeta(q, x, x_d, u_d, t)$  is defined as

$$\zeta(q, x, \hat{x}, t) = (A(\varrho_a, x, \hat{x}, t) - L(\hat{x}, t)C(\varrho_c, x, \hat{x}, t))(q - x) + f(x, t). \quad (4.10)$$

Note that (4.10) is constructed to contain  $q = \hat{x}$  and  $q = x$  as its particular solutions of (4.8) and (4.9). If  $d = 0$  and  $\mathcal{W} = 0$ , the differential dynamics of (4.8) and (4.9) for  $\partial_\mu q = \partial q / \partial \mu$  is given as

$$\partial_\mu \dot{q} = (A(\varrho_a, x, \hat{x}, t) - L(\hat{x}, t)C(\varrho_c, x, \hat{x}, t)) \partial_\mu q. \quad (4.11)$$

The similarity between (3.14) ( $\partial_\mu \dot{q} = (A - BK)\partial_\mu q$ ) and (4.11) leads to the following theorem [1]–[4]. Again, note that we could also use the SDC formulation with respect to a fixed point as delineated in Theorem 3.2 and as demonstrated in [1], [4], [9].

**Theorem 4.3.** *Suppose  $\exists \bar{\rho}, \bar{c} \in \mathbb{R}_{\geq 0}$  s.t.  $\|R^{-1}(\hat{x}, t)\| \leq \bar{\rho}$ ,  $\|C(\varrho_c, x, \hat{x}, t)\| \leq \bar{c}$ ,  $\forall x, \hat{x}, t$ . Suppose also that  $\underline{m}I \leq M \leq \bar{m}I$  of (2.26) holds, or equivalently,  $I \leq \bar{W} \leq \chi I$  of (3.19) holds with  $W = M(\hat{x}, t)^{-1}$ ,  $\bar{W} = \nu W$ ,  $\nu = \bar{m}$ , and  $\chi = \bar{m}/\underline{m}$ . As in Theorem 3.1, let  $\beta$  be defined as  $\beta = 0$  for deterministic systems (4.3) and*

$$\beta = \alpha_s = \alpha_{e0} + \nu^2 \alpha_{e1} = L_m \bar{g}_{e0}^2 (\alpha_G + 1/2)/2 + \nu^2 L_m \bar{\rho}^2 \bar{c}^2 \bar{g}_{e1}^2 (\alpha_G + 1/2)/2$$

for stochastic systems (4.4), where  $2\alpha_{e0} = L_m \bar{g}_{e0}^2 (\alpha_G + 1/2)$ ,  $2\alpha_{e1} = L_m \bar{\rho}^2 \bar{c}^2 \bar{g}_{e1}^2 (\alpha_G + 1/2)$ ,  $L_m$  is the Lipschitz constant of  $\partial W / \partial x_i$ ,  $\bar{g}_{e0}$  and  $\bar{g}_{e1}$  are given in (4.4), and  $\exists \alpha_G \in \mathbb{R}_{>0}$  is an arbitrary constant as in Theorem 2.5.

If  $M(\hat{x}, t)$  in (4.7) is constructed to satisfy the following convex constraint for  $\exists \alpha \in \mathbb{R}_{>0}$ :

$$\dot{\bar{W}} + 2 \text{sym}(\bar{W}A - \nu \bar{C}^\top R^{-1}C) \leq -2\alpha \bar{W} - \nu \beta I \quad (4.12)$$

then Theorems 2.4 and 2.5 hold for the virtual systems (4.8) and (4.9), respectively, i.e., we have the following bounds for  $\mathbf{e} = \hat{x} - x$  with  $\nu = \bar{m}$  and  $\chi = \bar{m}/\underline{m}$ :

$$\|\mathbf{e}(t)\| \leq \sqrt{\bar{m}} V_\ell(0) e^{-\alpha t} + \frac{\bar{d}_{e0} \sqrt{\chi} + \bar{\rho} \bar{c} \bar{d}_{e1} \nu}{\alpha} (1 - e^{-\alpha t}) \quad (4.13)$$

$$\mathbb{E}[\|\mathbf{e}(t)\|^2] \leq \bar{m} \mathbb{E}[V_{s\ell}(0)] e^{-2\alpha t} + \frac{C_{e0} \chi + C_{e1} \chi \nu^2}{2\alpha} \quad (4.14)$$

where  $V_{s\ell} = \int_x^{\hat{x}} \delta q^\top W \delta q$  and  $V_\ell = \int_x^{\hat{x}} \|\Theta \delta q\|$  are given in Theorem 2.3 with  $W = M^{-1} = \Theta^\top \Theta$  defining a contraction metric, the disturbance bounds  $\bar{d}_{e0}$ ,  $\bar{d}_{e1}$ ,

$\bar{g}_{e0}$ , and  $\bar{g}_{e1}$  are given in (4.3) and (4.4), respectively,  $C_{e0} = \bar{g}_{e0}^2(2\alpha_G^{-1} + 1)$ , and  $C_{e1} = \bar{\rho}^2 \bar{c}^2 \bar{g}_{e1}^2(2\alpha_G^{-1} + 1)$ . Note that for stochastic systems, the probability that  $\|\mathbf{e}\|$  is greater than or equal to  $\varepsilon \in \mathbb{R}_{>0}$  is given as

$$\mathbb{P} [\|\mathbf{e}(t)\| \geq \varepsilon] \leq \frac{1}{\varepsilon^2} \left( \bar{m} \mathbb{E}[V_{st}(0)] e^{-2\alpha t} + \frac{C_E}{2\alpha} \right) \quad (4.15)$$

where  $C_E = C_{e0}\chi + C_{e1}\chi\nu^2$ .

*Proof.* Theorem 3.1 indicates that (4.12) is equivalent to

$$\dot{W} + 2 \text{sym} (WA - \bar{C}^\top R^{-1}C) \leq -2\alpha W - \beta I. \quad (4.16)$$

Computing the time derivative of a Lyapunov function  $V = \partial_\mu q^\top W \partial_\mu q$  with  $\partial_\mu q = \partial q / \partial \mu$  for the unperturbed virtual dynamics (4.11), we have using (4.16) that

$$\dot{V} = \partial_\mu q^\top W \partial_\mu q = \partial_\mu q^\top (\dot{W} + 2WA - 2\bar{C}^\top R^{-1}C) \partial_\mu q \leq -2\alpha V - \beta \|\partial_\mu q\|^2$$

which implies that  $W = M^{-1}$  defines a contraction metric. Since we have  $\bar{m}^{-1}I \leq W \leq \underline{m}^{-1}I$ ,  $V \geq \bar{m}^{-1} \|\partial_\mu q\|^2$ , and

$$\begin{aligned} \|\Theta(\hat{x}, t) \partial_\mu d\| &\leq \bar{d}_{e0} / \sqrt{\bar{m}} + \bar{d}_{e1} \bar{\rho} \bar{c} \sqrt{\bar{m}} \\ \|\partial_\mu G\|_F^2 &\leq \bar{g}_{e0}^2 + \bar{\rho}^2 \bar{c}^2 \bar{g}_{e1}^2 \bar{m}^2 \end{aligned}$$

for  $d$  in (4.8) and  $G$  in (4.9), the bounds (4.13) – (4.15) follow from the proofs of Theorems 2.4 and 2.5 [2], [3].  $\square$

**Remark 4.1.** Although (4.12) is not an LMI due to the nonlinear term  $-\nu\beta I$  on its right-hand side for stochastic systems (4.4), it is a convex constraint as  $-\nu\beta = -\nu\alpha_s = -\nu\alpha_{e0} - \nu^3\alpha_{e1}$  is a concave function for  $\nu \in \mathbb{R}_{>0}$  [3], [10].

#### 4.2.II CV-STEM Formulation for State Estimation

The estimator (4.7) gives a convex steady-state upper bound of the Euclidean distance between  $x$  and  $\hat{x}$  as in Theorem 4.1 [1]–[4].

**Theorem 4.4.** If (4.12) of Theorem 4.3 holds, then we have the following bound:

$$\lim_{t \rightarrow \infty} \sqrt{\mathbb{E} [\|\hat{x} - x\|^2]} \leq c_0(\alpha, \alpha_G)\chi + c_1(\alpha, \alpha_G)\nu^s \quad (4.17)$$

where  $c_0 = \bar{d}_{e0}/\alpha$ ,  $c_1 = \bar{\rho}\bar{c}\bar{d}_{e1}/\alpha$ ,  $s = 1$  for deterministic systems (4.8), and  $c_0 = \sqrt{C_{e0}/(2\alpha)}$ ,  $c_1 = C_{e1}/(2\sqrt{2\alpha}C_{e0})$ , and  $s = 2$  for stochastic systems (4.9), with  $C_{e0}$  and  $C_{e1}$  given as  $C_{e0} = \bar{g}_{e0}^2(2\alpha_G^{-1} + 1)$  and  $C_{e1} = \bar{\rho}^2 \bar{c}^2 \bar{g}_{e1}^2(2\alpha_G^{-1} + 1)$ .

*Proof.* The upper bound (4.17) for deterministic systems (4.8) follows from (4.13) with the relation  $1 \leq \sqrt{\chi} \leq \chi$  due to  $\underline{m} \leq \bar{m}$ . For stochastic systems, we have using (4.14) that

$$C_{e0}\chi + C_{e1}\nu^2\chi \leq C_{e0}(\chi + (C_{e1}/(2C_{e0}))\nu^2)^2$$

due to  $1 \leq \chi \leq \chi^2$  and  $\nu \in \mathbb{R}_{>0}$ . This gives (4.17) for stochastic systems (4.9).  $\square$

Finally, the CV-STEM estimation framework is summarized in Theorem 4.5 [1]–[4].

**Theorem 4.5.** *Suppose that  $\alpha$ ,  $\alpha_G$ ,  $\bar{d}_{e0}$ ,  $\bar{d}_{e1}$ ,  $\bar{g}_{e0}$ ,  $\bar{g}_{e1}$ , and  $L_m$  in (4.12) and (4.17) are given. If the pair  $(A, C)$  is uniformly observable, the non-convex optimization problem of minimizing the upper bound (4.17) is equivalent to the following convex optimization problem with the contraction constraint (4.12) and  $I \leq \bar{W} \leq \chi I$  of (3.19):*

$$J_{CV}^* = \min_{\nu \in \mathbb{R}_{>0}, \chi \in \mathbb{R}, \bar{W} > 0} c_0\chi + c_1\nu^s + c_2P(\chi, \nu, \bar{W}) \quad (4.18)$$

s.t. (4.12) and (3.19)

where  $c_0$ ,  $c_1$ , and  $s$  are as defined in (4.17) of Theorem 4.4,  $c_2 \in \mathbb{R}_{\geq 0}$ , and  $P$  is some performance-based cost function as in Theorem 4.2.

The weight  $c_1$  for  $\nu^s$  indicates how much we trust the measurement  $y(t)$ . Using non-zero  $c_2$  enables finding contraction metrics optimal in a different sense in terms of  $P$ . Furthermore, the coefficients of the SDC parameterizations  $\varrho_a$  and  $\varrho_c$  in Lemma 3.1 (i.e.,  $A = \sum \varrho_{a,i}A_i$  and  $C = \sum \varrho_{c,i}C_i$  in (4.5) and (4.6)) can also be treated as decision variables by convex relaxation [9], thereby adding a design flexibility to mitigate the effects of external disturbances while verifying the system observability.

*Proof.* The proposed optimization problem is convex as its objective and constraints are convex in terms of decision variables  $\chi$ ,  $\nu$ , and  $\bar{W}$  (see Remark 4.1). Also, larger  $\bar{d}_{e1}$  and  $\bar{g}_{e1}$  in (4.3) and (4.4) imply larger measurement uncertainty. Thus by definition of  $c_1$  in Theorem 4.4, the larger the weight of  $\nu$ , the less confident we are in  $y(t)$  (see Example 4.2). The last statement on the SDC coefficients for guaranteeing observability follows from Proposition 1 of [9] and Proposition 1 of [1].  $\square$

**Example 4.2.** *The weights  $c_0$  and  $c_1$  of the CV-STEM estimation of Theorem 4.5 has an analogous trade-off to the case of the Kalman filter with the process and*

sensor noise covariance matrices,  $Q$  and  $P$ , respectively, since the term  $c_0\chi$  in upper bound of the steady-state tracking error in (4.17) becomes dominant if measurement noise is much smaller than process noise ( $\bar{d}_{e0} \gg \bar{d}_{e1}$  or  $\bar{g}_{e0} \gg \bar{g}_{e1}$ ), and the term  $c_1v^s$  becomes dominant if measurement noise is much greater than process noise ( $\bar{d}_{e0} \ll \bar{d}_{e1}$  or  $\bar{g}_{e0} \ll \bar{g}_{e1}$ ). In particular [2], [3],

- if  $c_1$  is much greater than  $c_0$ , large measurement noise leads to state estimation that responds slowly to unexpected changes in the measurement  $y$  (i.e. small estimation gain due to  $\nu = \bar{m} \geq \|M\|$ ), and
- if  $c_1$  is much smaller than  $c_0$ , large process noise leads to state estimation that responds fast to changes in the measurement (i.e. large  $\nu = \bar{m} \geq \|M\|$ ).

This is also because the solution  $Q = P^{-1} > 0$  of the Kalman filter Riccati equation [7, p. 375],  $\dot{P} = AP + PA^\top - PC^\top R^{-1}CP + Q$ , can be viewed as a positive definite matrix that defines a contraction metric as discussed in Example 2.3.

### 4.3 Control Contraction Metrics (CCMs)

As briefly discussed in Remark 3.4, the concept of a CCM [11]–[16] is introduced to extend contraction theory to design differential feedback control laws for control-affine deterministic nonlinear systems (3.1),  $\dot{x} = f(x, t) + B(x, t)u$ . Let us show that the CCM formulation could also be considered in the CV-STEM control of Theorem 4.2, where similar ideas have been investigated in [12], [17], [18].

**Theorem 4.6.** *Suppose the CV-STEM control of Theorem 4.2 is designed with its contraction condition replaced by the following set of convex constraints along with  $I \leq \bar{W} \leq \chi I$  of (3.19):*

$$B_\perp^\top \left( -\frac{\partial \bar{W}}{\partial t} - \partial_f \bar{W} + 2 \operatorname{sym} \left( \frac{\partial f}{\partial x} \bar{W} \right) + 2\alpha \bar{W} \right) B_\perp < 0 \quad (4.19)$$

$$B_\perp^\top \left( \partial_{b_i} \bar{W} - 2 \operatorname{sym} \left( \frac{\partial b_i}{\partial x} \bar{W} \right) \right) B_\perp = 0, \forall x, t, i \quad (4.20)$$

where  $B_\perp(x, t)$  is a matrix whose columns span the cokernel of  $B(x, t)$  defined as  $\operatorname{coker}(B) = \{a \in \mathbb{R}^n \mid B^\top a = 0\}$  satisfying  $B^\top B_\perp = 0$ ,  $b_i(x, t)$  is the  $i$ th column of  $B(x, t)$ ,  $\bar{W}(x, t) = \nu W(x, t)$ ,  $W(x, t) = M(x, t)^{-1}$  for  $M(x, t)$  that defines a contraction metric,  $\nu = \bar{m}$ ,  $\chi = \bar{m}/\underline{m}$ , and  $\partial_p F = \sum_{k=1}^n (\partial F / \partial x_k) p_k$  for  $p(x, t) \in \mathbb{R}^n$  and  $F(x, t) \in \mathbb{R}^{n \times n}$ . Then the controlled system (3.1) with  $d_c = 0$ , i.e.,  $\dot{x} = f(x, t) + B(x, t)u$ , is

- 1) *universally exponentially open-loop controllable*
- 2) *universally exponentially stabilizable via sampled-data feedback with arbitrary sample times*
- 3) *universally exponentially stabilizable using continuous feedback defined almost everywhere, and everywhere in a neighborhood of the target trajectory*

all with rate  $\alpha$  and overshoot  $R = \sqrt{\bar{m}/\underline{m}}$  for  $\underline{m}$  and  $\bar{m}$  given in  $\underline{m}\mathbf{I} \leq M \leq \bar{m}\mathbf{I}$  of (2.26). Such a positive definite matrix  $M(x, t)$  defines a CCM.

Given a CCM, there exists a differential feedback controller  $\delta u = k(x, \delta x, u, t)$  that stabilizes the following differential dynamics of (3.1) with  $d_c = 0$  along all solutions (i.e., the closed-loop dynamics is contracting as in Definition 2.3):

$$\delta \dot{x} = A(x, u, t)\delta x + B(x, t)\delta u \quad (4.21)$$

where  $A = \partial f / \partial x + \sum_{i=1}^m (\partial b_i / \partial x) u_i$  and  $\delta u$  is a tangent vector to a smooth path of controls at  $u$ . Furthermore, it can be computed as follows [11], [12], [16]:

$$u(x(t), t) = u_d + \int_0^1 k(\gamma(\mu, t), \partial_\mu \gamma(\mu, t), u(\gamma(\mu, t), t), t) d\mu \quad (4.22)$$

where  $\partial_\mu \gamma = \partial \gamma / \partial \mu$ ,  $\gamma$  is the minimizing geodesic with  $\gamma(0, t) = x_d$  and  $\gamma(1, t) = x$  for  $\mu \in [0, 1]$ ,  $(x_d, u_d)$  is a given target trajectory in  $\dot{x}_d = f(x_d, t) + B(x_d, t)u_d$  of (3.3), and the computation of  $k$  is given in [11], [12] (see Remark 4.2). Furthermore, Theorem 2.4 for deterministic disturbance rejection still holds and the CCM controller (4.22) thus possesses the same sense of optimality as for the CV-STEM control in Theorem 4.2.

*Proof.* Since the columns of  $B_\perp(x, t)$  span the cokernel of  $B(x, t)$ , the constraints (4.19) and (4.20) can be equivalently written as  $B_\perp(x, t)$  replaced by  $a$  in  $\text{coker}(B) = \{a \in \mathbb{R}^n | B^\top a = 0\}$ . Let  $a = M\delta x = W^{-1}\delta x$ . Then multiplying (4.19) and (4.20) by  $v^{-1} > 0$  and rewriting them using  $a$  yields

$$\delta x^\top M B = 0 \implies \delta x^\top (\dot{M} + 2 \text{sym}(MA) + 2\alpha M)\delta x < 0 \quad (4.23)$$

where  $A = \partial f / \partial x + \sum_{i=1}^m (\partial b_i / \partial x) u_i$  for the differential dynamics (4.21). The relation (4.23) states that  $\delta x$  orthogonal to the span of actuated directions  $b_i$  is naturally contracting, thereby implying the stabilizability of the system (3.1) with  $d_c = 0$ , i.e.,  $\dot{x} = f(x, t) + B(x, t)u$ . See [11], [12], [16] (and [17] on the CV-STEM formulation) for the rest of the proof.  $\square$

**Example 4.3.** Let us consider a case where  $k$  of the differential feedback controller does not depend on  $u$ , and is defined explicitly by  $M(x, t) > 0$  as follows [17]:

$$u = u_d - \int_{x_d}^x R(q, t)^{-1} B(q, t)^\top M(q, t) \delta q \quad (4.24)$$

where  $R(x, t) > 0$  is a given weight matrix and  $q$  is a smooth path that connects  $x$  to  $x_d$ . Since (4.24) yields  $\delta u = -R(x, t)^{-1} B(x, t)^\top M(x, t) \delta x$ , the contraction conditions (4.19) and (4.20) could be simplified as

$$\begin{aligned} -\frac{\partial \bar{W}}{\partial t} - \partial_f \bar{W} + 2 \operatorname{sym} \left( \frac{\partial f}{\partial x} \bar{W} \right) - \nu B R^{-1} B^\top &\leq -2\alpha \bar{W} \\ -\partial_{b_i} \bar{W} + 2 \operatorname{sym} \left( \frac{\partial b_i}{\partial x} \bar{W} \right) &= 0 \end{aligned} \quad (4.25)$$

yielding the convex optimization-based control synthesis algorithm of Theorem 4.6 independently of  $(x_d, u_d)$  similar to that of Theorem 4.2 (see [17] for details).

**Example 4.4.** If  $B$  is of the form  $[0, I]^\top$  for the zero matrix  $0 \in \mathbb{R}^{n_1 \times m}$  and identity matrix  $I \in \mathbb{R}^{n_2 \times m}$  with  $n = n_1 + n_2$ , the condition (4.25) says that  $M$  should not depend on the last  $n_2$  state variables [11].

**Remark 4.2.** We could consider stochastic perturbation in Theorem 4.6 using Theorem 2.5, even with the differential control law of the form (4.22) or (4.24) as demonstrated in [17]. Also, although the relation (4.20) or (4.25) is not included as a constraint in Theorem 4.2 for simplicity of discussion, the dependence of  $\dot{\bar{W}}$  on  $u$  in Theorem 4.2 can be removed by using it in a similar way to [17].

As stated in (4.22), the computation of the differential feedback gain  $k(x, \delta x, u, t)$  and minimizing geodesics  $\gamma$  is elaborated in [11], [12]. For example, if  $M$  is state-independent, then geodesics are just straight lines.

Let us again emphasize that, as delineated in Sec. 3.1, the differences between the SDC- and CCM-based CV-STEM frameworks in Theorems 4.2 and 4.6 arise only from their different form of controllers in  $u = u_d - K(x - x_d)$  of (3.10) and  $u = u_d + \int_0^1 k d\mu$  of (4.22), leading to the trade-offs outlined in Table 3.1.

#### 4.4 Remarks in CV-STEM Implementation

We propose some useful techniques for the practical application of the CV-STEM in Theorems 4.2, 4.5, and 4.6. Note that the CV-STEM requires solving a convex optimization problem at each time instant, but its solution can be approximated with formal stability guarantees to enable faster computation using machine learning techniques as shall be seen in Chapter 6.

#### 4.4.I Performance-based Cost Function

Selecting  $c_2 = 0$  in Theorems 4.2, 4.5, and 4.6 yields a convex objective function that allows for a systematic interpretation of its weights as seen earlier in Examples 4.1 and 4.2. One could also select  $c_2 > 0$  to augment the CV-STEM with other control and estimation performances of interest, as long as  $P(\nu, \chi, \bar{W})$  in (4.2) and (4.18) is convex. For example, we could consider a steady-state tracking error bound of parametric uncertain systems as  $P$ , using the adaptive control technique to be discussed in Chapter 8.1 [19]. Such a modification results in a contraction metric optimal in a different sense. Also, the different assumption on the boundedness of  $M$  and its derivatives lead to different convex optimization as derived in [1], [4].

#### 4.4.II Selecting and Computing CV-STEM Parameters

The CV-STEM optimization problems derived in Theorems 4.2, 4.5, and 4.6 are convex if we assume that  $\alpha$ ,  $\alpha_G$ , and  $L_m$  are given. However, these parameters would also affect the optimality of resultant contraction metrics. In [2], [3], [12], a line search algorithm is performed to find optimal  $\alpha$  and  $\alpha_G$ , while the Lipschitz constraint given with  $L_m$  is guaranteed by spectrally-normalization [3], [20] as shall be seen in detail in Sec. 6.3 (see [21]–[23] for contraction theory-based techniques for obtaining Lipschitz bounds). Also, the CV-STEM can be formulated as a finite-dimensional problem by using backward difference approximation on  $\dot{\bar{W}}$ , where we can then use  $-\bar{W} \leq -I$  to get a sufficient condition of its constraints, or we could alternatively solve it along pre-computed trajectories  $\{x(t_i)\}_{i=0}^M$  as in [2]. In Chapter 6, we use a parameterized function such as neural networks [24]–[26] for approximating  $M$  to explicitly compute  $\dot{M}$ .

#### References

- [1] H. Tsukamoto and S.-J. Chung, “Robust controller design for stochastic nonlinear systems via convex optimization,” *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4731–4746, 2021.
- [2] H. Tsukamoto and S.-J. Chung, “Neural contraction metrics for robust estimation and control: A convex optimization approach,” *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 211–216, 2021.
- [3] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Neural stochastic contraction metrics for learning-based control and estimation,” *IEEE Control Syst. Lett.*, vol. 5, no. 5, pp. 1825–1830, 2021.

- [4] H. Tsukamoto and S.-J. Chung, “Convex optimization-based controller design for stochastic nonlinear systems using contraction analysis,” in *IEEE Conf. Decis. Control*, Dec. 2019, pp. 8196–8203.
- [5] W. J. Rugh, *Linear Systems Theory*. USA: Prentice-Hall, Inc., 1996, ISBN: 0134412052.
- [6] S. Bonnabel and J.-J. E. Slotine, “A contraction theory-based analysis of the stability of the deterministic extended Kalman filter,” *IEEE Trans. Autom. Control*, vol. 60, no. 2, pp. 565–569, Feb. 2015.
- [7] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering: with MATLAB Exercises and Solutions*, 4th. New York, NY: Wiley, 1997.
- [8] F. Tan, W. Lohmiller, and J.-J. E. Slotine, “Analytical SLAM without linearization,” *Int. J. Robot. Res.*, vol. 36, no. 13-14, pp. 1554–1578, 2017.
- [9] A. P. Dani, S.-J. Chung, and S. Hutchinson, “Observer design for stochastic nonlinear systems via contraction-based incremental stability,” *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 700–714, Mar. 2015.
- [10] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Mar. 2004.
- [11] I. R. Manchester and J.-J. E. Slotine, “Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design,” *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 3046–3053, Jun. 2017.
- [12] S. Singh, A. Majumdar, J.-J. E. Slotine, and M. Pavone, “Robust online motion planning via contraction theory and convex optimization,” in *IEEE Int. Conf. Robot. Automat.*, May 2017, pp. 5883–5890.
- [13] S. Singh, S. M. Richards, V. Sindhvani, J.-J. E. Slotine, and M. Pavone, “Learning stabilizable nonlinear dynamics with contraction-based regularization,” *Int. J. Robot. Res.*, Aug. 2020.
- [14] R. Wang, R. Tóth, and I. R. Manchester, “A comparison of LPV gain scheduling and control contraction metrics for nonlinear control,” in *3rd IFAC Workshop on LPVS*, vol. 52, 2019, pp. 44–49.
- [15] R. Wang, R. Tóth, and I. R. Manchester, *Virtual CCMs: Convex nonlinear feedback design via behavioral embedding*, arXiv:2003.08513, Mar. 2020.
- [16] I. R. Manchester and J.-J. E. Slotine, “Robust control contraction metrics: A convex approach to nonlinear state-feedback  $\mathcal{H}^\infty$  control,” *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 333–338, 2018.
- [17] H. Tsukamoto, S.-J. Chung, J.-J. Slotine, and C. Fan, “A theoretical overview of neural contraction metrics for learning-based control with guaranteed stability,” in *IEEE Conf. Decis. Control*, 2021, pp. 2949–2954.

- [18] P. Zhao, A. Lakshmanan, K. Ackerman, A. Gahlawat, M. Pavone, and N. Hovakimyan, *Tube-certified trajectory tracking for nonlinear systems with robust control contraction metrics*, arXiv:2109.04453, Sep. 2021.
- [19] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Learning-based Adaptive Control using Contraction Theory,” in *IEEE Conf. Decis. Control*, 2021, pp. 2533–2538.
- [20] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” in *Int. Conf. Learn. Representations*, 2018.
- [21] M. Revay, R. Wang, and I. R. Manchester, *Lipschitz bounded equilibrium networks*, arXiv:2010.01732, Oct. 2020.
- [22] M. Revay, R. Wang, and I. R. Manchester, *Recurrent equilibrium networks: Flexible dynamic models with guaranteed stability and robustness*, arXiv:2104.05942, Apr. 2021.
- [23] I. R. Manchester, M. Revay, and R. Wang, “Contraction-based methods for stable identification and robust machine learning: A tutorial,” in *IEEE Conf. Decis. Control*, 2021, pp. 2955–2962.
- [24] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989, ISSN: 0893-6080.
- [25] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Math. Control Signals Syst*, vol. 2, no. 4, pp. 303–314, Dec. 1989, ISSN: 1435-568X.
- [26] K.-I. Funahashi, “On the approximate realization of continuous mappings by neural networks,” *Neural Netw.*, vol. 2, no. 3, pp. 183–192, 1989, ISSN: 0893-6080.

## **Part II: Learning-based Control**

## CONTRACTION THEORY FOR LEARNING-BASED CONTROL

- [1] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview,” *Annu. Rev. Control*, vol. 52, pp. 135–169, 2021, ISSN: 1367-5788.

Machine learning techniques, e.g., reinforcement learning [1]–[4], imitation learning [5]–[9], and neural networks [10]–[12], have gained popularity due to their ability to achieve a large variety of innovative engineering and scientific tasks which have been impossible heretofore. Starting from this chapter, we will see how contraction theory enhances learning-based and data-driven automatic control frameworks providing them with formal optimality, stability, and robustness guarantees. In particular, we present Theorems 5.2 and 5.3 for obtaining robust exponential bounds on trajectory tracking errors of nonlinear systems in the presence of learning errors, whose steady-state terms are again written as a function of the condition number  $\chi = \bar{m}/\underline{m}$  of a positive definite matrix  $M$  that defines a contraction metric, consistently with the CV-STEM frameworks of Chapter 4.

### 5.1 Problem Formulation

Let us consider the following virtual nonlinear system as in Theorem 2.2:

$$\dot{q}(\mu, t) = g(q(\mu, t), \varpi, t) + \mu \Delta_L(\varpi, t) + d(\mu, \varpi, t) \quad (5.1)$$

where  $\mu \in [0, 1]$ ,  $q : [0, 1] \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  is a smooth path of the system states,  $\varpi : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^p$  is a time-varying parameter,  $g : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  is a known smooth function which renders  $\dot{q} = g(q, \varpi, t)$  contracting with respect to  $q$ ,  $d : [0, 1] \times \mathbb{R}^p \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  with  $\sup_{\mu, \varpi, t} \|\partial d / \partial \mu\| = \bar{d} \in \mathbb{R}_{\geq 0}$  is unknown external disturbance parameterized by  $\mu$  as in Theorem 2.4, and  $\Delta_L : \mathbb{R}^p \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  is the part to be learned with machine learning-based methodologies such as neural networks [10]–[12]. We can also formulate this in a stochastic setting as follows:

$$dq(\mu, t) = (g(q(\mu, t), \varpi, t) + \mu \Delta_L(\varpi, t))dt + G(\mu, \varpi, t)d\mathcal{W} \quad (5.2)$$

where  $\mathcal{W} : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^w$  is a Wiener process, and  $G : [0, 1] \times \mathbb{R}^p \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{n \times w}$  is a matrix-valued function parameterized by  $\mu$  with  $\sup_{\mu, \varpi, t} \|\partial G / \partial \mu\|_F = \bar{g} \in \mathbb{R}_{\geq 0}$  as

in Theorem 2.5. The objective of learning is to find  $\Delta_L$  which satisfies the following, assuming that  $\varpi \in \mathcal{S}_\varpi \subseteq \mathbb{R}^p$  and  $t \in \mathcal{S}_t \subseteq \mathbb{R}_{\geq 0}$  for some compact sets  $\mathcal{S}_\varpi$  and  $\mathcal{S}_t$ :

$$\|\Delta_L(\varpi, t)\| \leq \epsilon_{\ell 0} + \epsilon_{\ell 1} \|\xi_1 - \xi_0\|, \quad \forall (\varpi, t) \in \mathcal{S} \quad (5.3)$$

where  $\mathcal{S} = \mathcal{S}_\varpi \times \mathcal{S}_t$ ,  $\xi_0 = q(0, t)$  and  $\xi_1 = q(1, t)$  are particular solutions of (5.1) and (5.2), and  $\epsilon_{\ell 0}, \epsilon_{\ell 1} \in \mathbb{R}_{\geq 0}$  are given learning errors.

Examples 5.1 – 5.3 illustrate how the problem (5.3) with the nonlinear systems (5.1) and (5.2) can be used to describe several learning-based and data-driven control problems to be discussed in the subsequent chapters, regarding  $\varpi$  as  $x, x_d, \hat{x}, u_d$ , etc.

**Example 5.1** (Feedback tracking control). *Let us consider the problem of learning a computationally-expensive (or unknown) feedback control law  $u^*(x, x_d, u_d, t)$  that tracks a target trajectory  $(x_d, u_d)$  given by  $\dot{x}_d = f(x_d, t) + B(x_d, t)u_d$  as in (3.3), assuming that  $f$  and  $B$  are known. If the dynamics is perturbed by  $d_c(x, t)$  as in (3.1), we have that*

$$\dot{x} = f(x, t) + B(x, t)u_L + d_c(x, t) = f(x, t) + B(x, t)u^* + B(x, t)(u_L - u^*) + d_c(x, t)$$

where  $u_L(x, x_d, u_d, t)$  denotes a learned law that models  $u^*(x, x_d, u_d, t)$ . As long as  $u^*$  renders the closed-loop dynamics  $\dot{x} = f(x, t) + B(x, t)u^*(x, x_d, u_d, t)$  contracting with respect to  $x$  [13]–[16], we can define the functions of (5.1) as follows:

$$\begin{aligned} g(q, \varpi, t) &= f(q, t) + B(q, t)u^*(q, x_d, u_d, t) \\ \Delta_L(\varpi, t) &= B(x, t)(u_L(x, x_d, u_d, t) - u^*(x, x_d, u_d, t)) \end{aligned} \quad (5.4)$$

with  $d(\mu, \varpi, t) = \mu d_c(x, t)$ , where  $\varpi = [x^\top, x_d^\top, u_d^\top]^\top$  in this case. It can be easily verified that (5.1) indeed has  $q(\mu = 0, t) = x_d$  and  $q(\mu = 1, t) = x$  as particular solutions if  $u^*(x_d, x_d, u_d, t) = u_d$ . Similarly, we can use (5.2) with  $G = \mu G_c(x, t)$  if the dynamics is stochastically perturbed by  $G_c(x, t)d\mathcal{W}$  as in (3.2).

The learning objective here is to make  $\|u_L(x, x_d, u_d, t) - u^*(x, x_d, u_d, t)\|$  as small as possible, which aligns with the aforementioned objective in (5.3). Note that if  $\|B\|$  is bounded in the compact set  $\mathcal{S}$  of (5.3), we can bound  $\|\Delta_L\|$  of (5.4) with  $\epsilon_{\ell 0} \neq 0$  and  $\epsilon_{\ell 1} = 0$  as to be explained in Remark 5.1. See Theorems 6.1, 6.3, and 7.1 for details.

**Example 5.2** (State estimation). *Next, let us consider the problem of learning a computationally-expensive (or unknown) state estimator, approximating its estimation gain  $L(\hat{x}, t)$  by  $L_L(\hat{x}, t)$ . If there is no disturbance in (4.3) and (4.4), we have that*

$$\begin{aligned}\dot{\hat{x}} &= f(\hat{x}, t) + L_L(h(x, t) - h(\hat{x}, t)) \\ &= f(\hat{x}, t) + L(h(x, t) - h(\hat{x}, t)) + (L_L - L)(h(x, t) - h(\hat{x}, t))\end{aligned}$$

where  $\dot{x} = f(x, t)$  is the true system,  $y = h(x, t)$  is the measurement, and we assume that  $f$  and  $h$  are known. If  $L$  is designed to render  $\dot{q} = f(q, t) + L(\hat{x}, t)(h(x, t) - h(q, t))$  contracting with respect to  $q$  [13], [15], [17], [18], we could define the functions of (5.1) and (5.2) as follows:

$$g(q, \varpi, t) = f(q, t) + L(\hat{x}, t)(h(x, t) - h(q, t)) \quad (5.5)$$

$$\Delta_L(q, \varpi, t) = (L_L(\hat{x}, t) - L(\hat{x}, t))(h(x, t) - h(\hat{x}, t)) \quad (5.6)$$

where  $\varpi = [x^\top, \hat{x}^\top]^\top$ . It can be seen that (5.1) and (5.2) with the relations (5.5) and (5.6) indeed has  $q(\mu = 0, t) = x$  and  $q(\mu = 1, t) = \hat{x}$  as its particular solutions when perturbed by deterministic and stochastic disturbances as in (4.3) and (4.4), respectively. We can bound  $\|\Delta_L\|$  of (5.6) in the compact set of (5.3) with  $\epsilon_{\ell_0} = 0$  and  $\epsilon_{\ell_1} \neq 0$  if  $h$  is Lipschitz, and with  $\epsilon_{\ell_0} \neq 0$  and  $\epsilon_{\ell_1} = 0$  if  $h$  is bounded in  $\mathcal{S}$ , using the techniques to be discussed in Remark 5.1. See Theorems 6.2 and 6.3 for details.

**Example 5.3** (System identification). *We can use the problem (5.3) with the systems (5.1) and (5.2) also if  $f_{\text{true}}$  of the underlying dynamics  $\dot{x}^* = f_{\text{true}}(x^*, t)$  is unknown and learned by  $\dot{x} = f_L(x, t)$ . Since we have*

$$\dot{x}^* = f_{\text{true}}(x^*, t) = f_L(x^*, t) + (f_{\text{true}}(x^*, t) - f_L(x^*, t))$$

we could define  $g$  and  $\Delta_L$  of (5.1) and (5.2) as follows, to have  $q(\mu = 0, t) = x$  and  $q(\mu = 1, t) = x^*$  as its particular solutions:

$$g(q, \varpi, t) = f_L(q, t), \quad \Delta_L(\varpi, t) = f_{\text{true}}(x^*, t) - f_L(x^*, t) \quad (5.7)$$

where  $\varpi = x^*$ , as long as  $\dot{x} = f_L(x, t)$  is contracting with respect to  $x$  [19], [20]. Since  $\Delta_L$  of (5.7) is the learning error itself,  $\|\Delta_L\|$  can be bounded in  $\mathcal{S}$  using the techniques of Remark 5.1. See Theorems 8.3 and 8.4 for details.

**Remark 5.1.** *As seen in Examples 5.1 – 5.2,  $\Delta_L$  of (5.3) is typically given by a learning error,  $\phi(z) - \varphi(z)$ , multiplied by a bounded or Lipschitz continuous function,*

where  $z = (\varpi, t) \in \mathcal{S}$  for a compact set  $\mathcal{S}$ , and  $\phi$  is the true computationally-expensive/unknown function to be learned by  $\varphi$  as in (5.4), (5.6), (5.7), etc.

Let  $\mathcal{T}$  be a set of training data  $\{(z_i, \phi(z_i))\}_{i=1}^N$  sampled in  $\mathcal{S}$ . For systems with the true function  $\phi$  and its approximation  $\varphi$  being Lipschitz (e.g., by spectral normalization [21] to be discussed in Definition 6.3 or by using contraction theory as in [22]–[24]), we can analytically find a bound for  $\|\phi(z) - \varphi(z)\|$  as follows if target data samples  $z$  are in  $B(r) = \{z \in \mathcal{S} \mid \sup_{z' \in \mathcal{T}} \|z - z'\| \leq r\}$  for some  $r \in \mathbb{R}_{\geq 0}$ :

$$\sup_{z \in B(r)} \|\phi(z) - \varphi(z)\| \leq \sup_{z' \in \mathcal{T}} \|\phi(z') - \varphi(z')\| + (L_\phi + L_\varphi)r \quad (5.8)$$

where  $L_\phi$  and  $L_\varphi$  are the Lipschitz constants of  $\phi$  and  $\varphi$ , respectively. The term  $\sup_{z' \in \mathcal{T}} \|\phi(z') - \varphi(z')\|$  can then be bounded by a constant, e.g., by using a deep robust regression model as proven in [25], [26] with spectral normalization, under standard training data distribution assumptions.

Deep Neural Networks (DNNs) have been shown to generalize well to the set of unseen events that are from almost the same distribution as their training set [27]–[31], and consequently, obtaining a tighter and more general upper bound for the learning error as in (5.8) has been an active field of research [25], [26], [32], [33], where some recent examples include, but are not limited to, spectral normalization [21], [29], neural ordinary differential equations [34]–[41], robust implicit networks [42], [43], robust equilibrium networks [24], [44], and Lipschitz neural networks [45]–[47]. Thus, the condition (5.3) has become a common assumption in analyzing the performance of learning-based and data-driven control techniques [29], [48]–[55].

## 5.2 Formal Stability Guarantees via Contraction Theory

One drawback of naively using existing learning-based and data-driven control approaches for the perturbed nonlinear systems (5.1) and (5.2) without analyzing contraction is that, as shall be seen in the following theorem, we can only guarantee the trajectory error to be bounded by a function that increases exponentially with time.

**Theorem 5.1.** *Suppose that  $\Delta_L$  of (5.1) is learned to satisfy (5.3), and that  $g(q, \varpi, t)$  of (5.1) is Lipschitz with respect to  $q$  with its 2-norm Lipschitz constant  $L_g \in \mathbb{R}_{\geq 0}$ , i.e.,*

$$\|g(q, \varpi, t) - g(q', \varpi, t)\| \leq L_g \|q - q'\|, \quad \forall (\varpi, t) \in \mathcal{S}$$

where  $q, q' \in \mathbb{R}^n$  and  $\mathcal{S}$  is the compact set of (5.3). Then we have the following upper bound for all  $(\varpi, t) \in \mathcal{S}$ :

$$\|\mathbf{e}(t)\| \leq \|\mathbf{e}(0)\| e^{(L_g + \epsilon_{\ell 1})t} + \frac{\epsilon_{\ell 0} + \bar{d}}{L_g + \epsilon_{\ell 1}} (e^{(L_g + \epsilon_{\ell 1})t} - 1) \quad (5.9)$$

where  $\mathbf{e}(t) = \xi_1(t) - \xi_0(t)$ ,  $\xi_0(t) = q(0, t)$  and  $\xi_1(t) = q(1, t)$  for  $q$  of (5.1), and  $\bar{d} = \sup_{\mu, \varpi, t} \|\partial d / \partial \mu\|$  for  $d$  of (5.3).

*Proof.* See the Gronwall-Bellman inequality [56, p. 651] and Theorem 3.4 of [56, pp. 96-97].  $\square$

The bound obtained in Theorem 5.1 is useful in that it gives mathematical guarantees even for naive learning-based frameworks without a contracting property (e.g., it can be used to prove safety in the learning-based Model Predictive Control (MPC) framework [57]). However, the exponential term  $e^{(L_g + \epsilon_{\ell 1})t}$  in (5.9) causes the upper bound to diverge, which could result in more conservative automatic control designs than necessary.

In contrast, contraction theory gives an upper bound on the trajectory tracking error  $\|\mathbf{e}(t)\|$  which is exponentially bounded linearly in the learning error, even under the presence of external disturbances [15], [51]–[53].

**Theorem 5.2.** *Let us consider the virtual system of a smooth path  $q(\mu, t)$  in (5.1) and suppose that  $\Delta_L$  of (5.1) is learned to satisfy (5.3). If the condition  $\underline{m}\mathbf{I} \leq M \leq \bar{m}\mathbf{I}$  of (2.26) holds and the system (5.1) with  $\Delta_L = 0$  and  $d = 0$  is contracting, i.e.,*

$$\dot{M} + M(\partial g / \partial q) + (\partial g / \partial q)^\top M \leq -2\alpha M$$

*of Theorem 2.1 or 2.2 holds for  $M > 0$  that defines a contraction metric with the contraction rate  $\alpha$ , and if the learning error  $\epsilon_{\ell 1}$  of (5.3) is sufficiently small to satisfy*

$$\exists \alpha_\ell \in \mathbb{R}_{>0} \text{ s.t. } \alpha_\ell = \alpha - \epsilon_{\ell 1} \sqrt{\bar{m}/\underline{m}} > 0, \quad (5.10)$$

*then we have the following bound for all  $(\varpi, t) \in \mathcal{S}$ :*

$$\|\mathbf{e}(t)\| \leq \frac{V_\ell(0)}{\sqrt{\underline{m}}} e^{-\alpha_\ell t} + \frac{\epsilon_{\ell 0} + \bar{d}}{\alpha_\ell} \sqrt{\frac{\bar{m}}{\underline{m}}} (1 - e^{-\alpha_\ell t}) \quad (5.11)$$

where  $\mathbf{e}(t) = \xi_1(t) - \xi_0(t)$  with  $\xi_0(t) = q(0, t)$  and  $\xi_1(t) = q(1, t)$  for  $q$  of (5.1),  $\bar{d} = \sup_{\mu, \varpi, t} \|\partial d / \partial \mu\|$  as in (5.1),  $\mathcal{S}$  is the compact set of Theorem 5.1,  $\epsilon_{\ell 0}$  and  $\epsilon_{\ell 1}$  are the learning errors of (5.3), and  $V_\ell(t) = \int_{\xi_0}^{\xi_1} \|\Theta(q(t), t) \delta q(t)\|$  for  $M = \Theta^\top \Theta$  as in Theorem 2.3.

*Proof.* Since (5.1) with  $\Delta_L = 0$  and  $d = 0$  is contracting and we have that

$$\begin{aligned} \|\partial(\Delta_L + d)/\partial\mu\| &\leq \epsilon_{\ell 0} + \epsilon_{\ell 1}\|\xi_1 - \xi_0\| + \bar{d} \\ &\leq \epsilon_{\ell 0} + \frac{\epsilon_{\ell 1}}{\sqrt{\underline{m}}} \int_{\xi_0}^{\xi_1} \|\Theta\delta q\| + \bar{d} = \epsilon_{\ell 0} + \frac{\epsilon_{\ell 1}}{\sqrt{\underline{m}}} V_\ell + \bar{d} \end{aligned} \quad (5.12)$$

for all  $\mu \in [0, 1]$  and  $(\varpi, t) \in \mathcal{S}$  due to  $\sup_{\mu, \varpi, t} \|\partial d/\partial\mu\| = \bar{d}$ , the direct application of Theorem 2.4 to the system (5.1), along with the condition (5.10), yields (5.11).  $\square$

Using Theorem 2.5, we can easily derive a stochastic counterpart of Theorem 5.2 for the system (5.2) [53], [58].

**Theorem 5.3.** *Consider the virtual system of a smooth path  $q(\mu, t)$  in (5.2) and suppose  $\Delta_L$  of (5.2) is learned to satisfy (5.3). If  $\underline{m}\mathbf{I} \leq M \leq \bar{m}\mathbf{I}$  of (2.26) holds and the system (5.2) is contracting, i.e.,*

$$\dot{M} + M(\partial g/\partial q) + (\partial g/\partial q)^\top M \leq -2\alpha M - \alpha_s \mathbf{I}$$

of Theorem 2.5 holds, and if the learning error  $\epsilon_{\ell 1}$  of (5.3) and an arbitrary constant  $\alpha_d \in \mathbb{R}_{>0}$  (see (5.16)) are selected to satisfy

$$\exists \alpha_\ell \in \mathbb{R}_{>0} \text{ s.t. } \alpha_\ell = \alpha - (\alpha_d/2 + \epsilon_{\ell 1}\sqrt{\bar{m}/\underline{m}}) > 0, \quad (5.13)$$

then we have the following bound for all  $(\varpi, t) \in \mathcal{S}$ :

$$\mathbb{E} [\|\mathbf{e}(t)\|^2] \leq \frac{\mathbb{E}[V_{s\ell}(0)]}{\underline{m}} e^{-2\alpha_\ell t} + \frac{C}{2\alpha_\ell} \frac{\bar{m}}{\underline{m}} \quad (5.14)$$

where  $\mathbf{e}(t) = \xi_1(t) - \xi_0(t)$  with  $\xi_0(t) = q(0, t)$  and  $\xi_1(t) = q(1, t)$  for  $q$  of (5.2),  $\mathcal{S}$  is the compact set given in Theorem 5.1,  $\epsilon_{\ell 0}$  and  $\epsilon_{\ell 1}$  are the learning errors of (5.3),  $V_{s\ell}(t) = \int_{\xi_0}^{\xi_1} \|\Theta(q(t), t)\delta q(t)\|^2$  for  $M = \Theta^\top \Theta$  as given in Theorem 2.3, and  $C = \bar{g}^2(2\alpha_G^{-1} + 1) + \epsilon_{\ell 0}^2 \alpha_d^{-1}$  for  $\bar{g} = \sup_{\mu, \varpi, t} \|\partial G/\partial\mu\|_F$  of (5.2) with an arbitrary constant  $\alpha_G \in \mathbb{R}_{>0}$  as in Theorem 2.5. Furthermore, the probability that  $\|\mathbf{e}\|$  is greater than or equal to  $\varepsilon \in \mathbb{R}_{>0}$  is given as

$$\mathbb{P} [\|\mathbf{e}(t)\| \geq \varepsilon] \leq \frac{1}{\varepsilon^2} \left( \frac{\mathbb{E}[V_{s\ell}(0)]}{\underline{m}} e^{-2\alpha_\ell t} + \frac{C}{2\alpha_\ell} \frac{\bar{m}}{\underline{m}} \right). \quad (5.15)$$

*Proof.* Computing  $\mathcal{L}V_{s\ell}$  with the virtual system (5.2) as in the proof of Theorem 2.5, we get an additional term  $2 \int_0^1 \partial_\mu q^\top M \Delta_L$ . Using the learning error assumption (5.3) to have  $\|\Delta_L\| \leq \epsilon_{\ell 0} + (\epsilon_{\ell 1}/\sqrt{\underline{m}}) \int_0^1 \|\Theta\partial_\mu q\| d\mu$  as in (5.12), we have that

$$\begin{aligned} 2 \int_0^1 \partial_\mu q^\top M \Delta_L &\leq 2\sqrt{\bar{m}} V_\ell (\epsilon_{\ell 0} + (\epsilon_{\ell 1}/\sqrt{\underline{m}}) V_\ell) \\ &\leq \alpha_d^{-1} \epsilon_{\ell 0}^2 \bar{m} + (\alpha_d + 2\epsilon_{\ell 1}\sqrt{\bar{m}/\underline{m}}) V_\ell^2 \end{aligned} \quad (5.16)$$

where  $V_\ell = \int_0^1 \|\Theta \partial_\mu q\| d\mu$  as in Theorem 2.3, and the relation  $2ab \leq \alpha_d^{-1}a^2 + \alpha_d b^2$ , which holds for any  $a, b \in \mathbb{R}$  and  $\alpha_d \in \mathbb{R}_{>0}$  ( $a = \epsilon_{\ell 0} \sqrt{\bar{m}}$  and  $b = V_\ell$  in this case), is used to obtain the second inequality. Since we have  $V_\ell^2 \leq V_{sl}$  as proven in Theorem 2.3, selecting  $\alpha_d$  and  $\epsilon_{\ell 1}$  sufficiently small to satisfy (5.13) gives the desired relations (5.14) and (5.15) due to Theorem 2.5.  $\square$

As discussed in Theorems 5.2 and 5.3, using contraction theory for learning-based and data-driven control, we can formally guarantee the system trajectories to stay in a tube with an exponentially convergent bounded radius centered around the target trajectory, even with the external disturbances and learning errors  $\epsilon_{\ell 0}$  and  $\epsilon_{\ell 1}$ . The exponential bounds (5.11), (5.14), and (5.15) become tighter as we achieve smaller  $\epsilon_{\ell 0}$  and  $\epsilon_{\ell 1}$  using more training data for verifying (5.3) (see Remark 5.1). It is also worth noting that the steady-state bounds of (5.11) and (5.14) are again some functions of the condition number  $\chi = \bar{m}/\underline{m}$  as in Theorems 4.1 and 4.4, which renders the aforementioned CV-STEM approach valid and effective also in the learning-based frameworks. Theorems 5.2 and 5.3 play a central role in providing incremental exponential stability guarantees for learning-based and data-driven automatic control to be introduced in Chapter 6 – 8.1.

## References

- [1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998, ISBN: 0-262-19398-1.
- [2] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.
- [3] M. Everett, Y. F. Chen, and J. P. How, “Motion planning among dynamic, decision-making agents with deep reinforcement learning,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2018, pp. 3052–3059.
- [4] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, “Safe model-based reinforcement learning with stability guarantees,” in *Adv. Neural Inf. Process. Syst.*, vol. 30, Curran Associates, Inc., 2017, pp. 908–918.
- [5] J. Carius, F. Farshidian, and M. Hutter, “MPC-Net: A First Principles Guided Policy Search,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2897–2904, 2020.
- [6] B. Rivière, W. Höning, Y. Yue, and S.-J. Chung, “GLAS: Global-to-local safe autonomy synthesis for multi-robot motion planning with end-to-end learning,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4249–4256, 2020.

- [7] J. Ho and S. Ermon, “Generative adversarial imitation learning,” in *Adv. Neural Inf. Process. Syst.*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, Curran Associates, Inc., 2016, pp. 4565–4573.
- [8] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, “Social GAN: Socially acceptable trajectories with generative adversarial networks,” in *IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 2255–2264.
- [9] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, “Imitating driver behavior with generative adversarial networks,” in *IEEE Intell. Vehicles Symp.*, 2017, pp. 204–211.
- [10] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989, ISSN: 0893-6080.
- [11] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Math. Control Signals Syst*, vol. 2, no. 4, pp. 303–314, Dec. 1989, ISSN: 1435-568X.
- [12] K.-I. Funahashi, “On the approximate realization of continuous mappings by neural networks,” *Neural Netw.*, vol. 2, no. 3, pp. 183–192, 1989, ISSN: 0893-6080.
- [13] H. Tsukamoto and S.-J. Chung, “Robust controller design for stochastic nonlinear systems via convex optimization,” *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4731–4746, 2021.
- [14] I. R. Manchester and J.-J. E. Slotine, “Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design,” *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 3046–3053, Jun. 2017.
- [15] H. Tsukamoto and S.-J. Chung, “Neural contraction metrics for robust estimation and control: A convex optimization approach,” *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 211–216, 2021.
- [16] S. Singh, A. Majumdar, J.-J. E. Slotine, and M. Pavone, “Robust online motion planning via contraction theory and convex optimization,” in *IEEE Int. Conf. Robot. Automat.*, May 2017, pp. 5883–5890.
- [17] A. P. Dani, S.-J. Chung, and S. Hutchinson, “Observer design for stochastic nonlinear systems via contraction-based incremental stability,” *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 700–714, Mar. 2015.
- [18] S. Bonnabel and J.-J. E. Slotine, “A contraction theory-based analysis of the stability of the deterministic extended Kalman filter,” *IEEE Trans. Autom. Control*, vol. 60, no. 2, pp. 565–569, Feb. 2015.
- [19] S. Singh, S. M. Richards, V. Sindhvani, J.-J. E. Slotine, and M. Pavone, “Learning stabilizable nonlinear dynamics with contraction-based regularization,” *Int. J. Robot. Res.*, Aug. 2020.

- [20] N. M. Boffi, S. Tu, N. Matni, J.-J. E. Slotine, and V. Sindhvani, *Learning stability certificates from data*, arXiv:2008.05952, 2020.
- [21] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” in *Int. Conf. Learn. Representations*, 2018.
- [22] M. Revay, R. Wang, and I. R. Manchester, *Lipschitz bounded equilibrium networks*, arXiv:2010.01732, Oct. 2020.
- [23] M. Revay, R. Wang, and I. R. Manchester, *Recurrent equilibrium networks: Flexible dynamic models with guaranteed stability and robustness*, arXiv:2104.05942, Apr. 2021.
- [24] I. R. Manchester, M. Revay, and R. Wang, “Contraction-based methods for stable identification and robust machine learning: A tutorial,” in *IEEE Conf. Decis. Control*, 2021, pp. 2955–2962.
- [25] X. Chen, M. Monfort, A. Liu, and B. D. Ziebart, “Robust Covariate Shift Regression,” in *19th Int. Conf. Artif. Intell. Statist.*, A. Gretton and C. C. Robert, Eds., ser. Proc. Mach. Learn. Res. Vol. 51, Cadiz, Spain, May 2016, pp. 1270–1279.
- [26] A. Liu, G. Shi, S.-J. Chung, A. Anandkumar, and Y. Yue, “Robust regression for safe exploration in control,” in *Conf. Robot Learn.*, Nov. 2020.
- [27] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning requires rethinking generalization,” in *Int. Conf. Learn. Representations*, Toulon, France: OpenReview.net, Apr. 2017.
- [28] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 770–778.
- [29] G. Shi, X. Shi, M. O’Connell, *et al.*, “Neural lander: Stable drone landing control using learned dynamics,” in *IEEE Int. Conf. Robot. Automat.*, May 2019, pp. 9784–9790.
- [30] G. K. Dziugaite and D. M. Roy, “Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data,” in *33rd Conf. Uncertainty Artif. Intell.*, Sydney, Australia: AUAI Press, Aug. 2017.
- [31] B. Neyshabur, S. Bhojanapalli, D. Mcallester, and N. Srebro, “Exploring generalization in deep learning,” in *Adv. Neural Inf. Process. Syst.*, vol. 30, Curran Associates, Inc., 2017.
- [32] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, “Spectrally-normalized margin bounds for neural networks,” in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

- [33] B. Neyshabur, S. Bhojanapalli, and N. Srebro, “A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks,” in *Int. Conf. Learn. Representations*, 2018.
- [34] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, “Neural ordinary differential equations,” in *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.
- [35] A. Tuor, J. Drgona, and D. Vrabie, *Constrained neural ordinary differential equations with stability guarantees*, arXiv:2004.10883, Apr. 2020.
- [36] C. Dong, L. Liu, Z. Li, and J. Shang, “Towards adaptive residual network training: A neural-ODE perspective,” in *37th Int. Conf. Mach. Learn.*, ser. Proc. Mach. Learn. Res. Vol. 119, PMLR, Jul. 2020, pp. 2616–2626.
- [37] S. Grunbacher, R. Hasani, M. Lechner, J. Cyranka, S. A. Smolka, and R. Grosu, “On the verification of neural ODEs with stochastic guarantees,” *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 13, pp. 11 525–11 535, May 2021.
- [38] H. Chen, Z. Mo, Z. Yang, and X. Wang, “Theoretical investigation of generalization bound for residual networks,” in *Proc. Int. Joint Conf. Artif. Intell.*, Jul. 2019, pp. 2081–2087.
- [39] M. Biloš, J. Sommer, S. S. Rangapuram, T. Januschowski, and S. Günemann, *Neural flows: Efficient alternative to neural ODEs*, arXiv:2110.13040, Oct. 2021.
- [40] M. Zakwan, L. Xu, and G. Ferrari-Trecate, *On robust classification using contractive Hamiltonian neural ODEs*, arXiv:2203.11805, Mar. 2022.
- [41] I. D. J. Rodriguez, A. D. Ames, and Y. Yue, *LyaNet: A Lyapunov framework for training neural ODEs*, arXiv:2202.02526, Feb. 2022.
- [42] S. Jafarpour, A. Davydov, A. Proskurnikov, and F. Bullo, “Robust implicit networks via non-Euclidean contractions,” in *Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 9857–9868.
- [43] S. Jafarpour, M. Abate, A. Davydov, F. Bullo, and S. Coogan, “Robustness certificates for implicit neural networks: A mixed monotone contractive approach,” in *Proc. Conf. Learn. Dyn. Control, To Appear*, Nov. 2022.
- [44] M. Revay, R. Wang, and I. R. Manchester, “Recurrent equilibrium networks: Unconstrained learning of stable and robust dynamical models,” in *IEEE Conf. Decis. Control*, 2021, pp. 2282–2287.
- [45] P. Pauli, A. Koch, J. Berberich, P. Kohler, and F. Allgöwer, “Training robust neural networks using Lipschitz bounds,” *IEEE Control Syst. Lett.*, vol. 6, pp. 121–126, 2022.

- [46] A. Araujo, A. J. Havens, B. Delattre, A. Allauzen, and B. Hu, “A unified algebraic perspective on Lipschitz neural networks,” in *Int. Conf. Learn. Representations*, 2023.
- [47] S. Bose, “Lipschitz bound analysis of neural networks,” in *Int. Conf. Comput. Commun. Netw. Technol.*, 2022, pp. 1–5.
- [48] 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 *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2019, pp. 6878–6884.
- [49] R. Cheng, A. Verma, G. Orosz, S. Chaudhuri, Y. Yue, and J. Burdick, “Control regularization for reduced variance reinforcement learning,” in *36th Int. Conf. Mach. Learn.*, K. Chaudhuri and R. Salakhutdinov, Eds., ser. Proc. Mach. Learn. Res. Vol. 97, PMLR, Jun. 2019, pp. 1141–1150.
- [50] C. D. McKinnon and A. P. Schoellig, “Learn fast, forget slow: Safe predictive learning control for systems with unknown and changing dynamics performing repetitive tasks,” *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 2180–2187, 2019.
- [51] H. Tsukamoto and S.-J. Chung, “Learning-based robust motion planning with guaranteed stability: A contraction theory approach,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6164–6171, 2021.
- [52] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Learning-based Adaptive Control using Contraction Theory,” in *IEEE Conf. Decis. Control*, 2021, pp. 2533–2538.
- [53] H. Tsukamoto, S.-J. Chung, J.-J. Slotine, and C. Fan, “A theoretical overview of neural contraction metrics for learning-based control with guaranteed stability,” in *IEEE Conf. Decis. Control*, 2021, pp. 2949–2954.
- [54] G. Shi, W. Hönig, Y. Yue, and S.-J. Chung, “Neural-swarm: Decentralized close-proximity multirotor control using learned interactions,” in *IEEE Int. Conf. Robot. Automat.*, 2020, pp. 3241–3247.
- [55] G. Shi, W. Hönig, X. Shi, Y. Yue, and S.-J. Chung, “Neural-swarm2: Planning and control of heterogeneous multirotor swarms using learned interactions,” *IEEE Trans. Robot.*, pp. 1–17, 2021.
- [56] H. K. Khalil, *Nonlinear Systems*, 3rd. Upper Saddle River, NJ: Prentice-Hall, 2002.
- [57] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, “Learning-based model predictive control: Toward safe learning in control,” *Annu. Rev. Control Robot. Auton. Syst.*, vol. 3, no. 1, pp. 269–296, 2020.
- [58] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Neural stochastic contraction metrics for learning-based control and estimation,” *IEEE Control Syst. Lett.*, vol. 5, no. 5, pp. 1825–1830, 2021.

*Chapter 6*

## LEARNING-BASED SAFE AND ROBUST CONTROL AND ESTIMATION

- [1] H. Tsukamoto and S.-J. Chung, “Neural contraction metrics for robust estimation and control: A convex optimization approach,” *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 211–216, 2021.
- [2] H. Tsukamoto, S.-J. Chung, J.-J. Slotine, and C. Fan, “A theoretical overview of neural contraction metrics for learning-based control with guaranteed stability,” in *IEEE Conf. Decis. Control*, 2021, pp. 2949–2954.
- [3] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Neural stochastic contraction metrics for learning-based control and estimation,” *IEEE Control Syst. Lett.*, vol. 5, no. 5, pp. 1825–1830, 2021.

The CV-STEM schemes in Theorems 4.2, 4.5, and 4.6 permit the construction of optimal contraction metrics via convex optimization for synthesizing optimal, provably stable, and robust feedback control and state estimation. It is also shown in Theorems 5.2 and 5.3 that these contraction metrics are useful for obtaining stability guarantees for machine learning-based and data-driven control techniques. They also retain the CV-STEM-type optimality due to the tracking error bounds (5.11) and (5.14), which are comparable to those in Theorems 4.1 and 4.4. However, the CV-STEM requires that a nonlinear system of equations or an optimization problem be solved at each time instant, which is not suitable for systems with limited online computational capacity.

The Neural Contraction Metric (NCM) and its extensions [1]–[4], to be discussed in this chapter, have been developed to address such a computational issue by modeling the CV-STEM optimization scheme using a DNN [5]–[7], making it implementable in real-time. We will see in the following that Theorems 5.2 and 5.3 of Chapter 5 can still be used to formally prove the robustness and stability properties of the NCM methodologies [8].

Let us again consider the systems (3.1) – (3.3) for control, i.e.,

$$\dot{x} = f(x, t) + B(x, t)u + d_c(x, t) \quad (6.1)$$

$$dx = (f(x, t) + B(x, t)u)dt + G_c(x, t)d\mathcal{W}(t) \quad (6.2)$$

$$\dot{x}_d = f(x_d, t) + B(x_d, t)u_d \quad (6.3)$$

with  $\sup_{x,t} \|d_c\| = \bar{d}_c \in \mathbb{R}_{\geq 0}$  and  $\sup_{x,t} \|G_c\|_F = \bar{g}_c \in \mathbb{R}_{\geq 0}$ , and (4.3) – (4.4) for estimation, i.e.,

$$\dot{x} = f(x, t) + d_{e0}(x, t), \quad y = h(x, t) + d_{e1}(x, t) \quad (6.4)$$

$$dx = f(x, t)dt + G_{e0}d\mathcal{W}_0, \quad ydt = h(x, t)dt + G_{e1}d\mathcal{W}_1 \quad (6.5)$$

with  $\sup_{x,t} \|d_{e0}\| = \bar{d}_{e0} \in \mathbb{R}_{\geq 0}$ ,  $\sup_{x,t} \|d_{e1}\| = \bar{d}_{e1} \in \mathbb{R}_{\geq 0}$ ,  $\sup_{x,t} \|G_{e0}\|_F = \bar{g}_{e0} \in \mathbb{R}_{\geq 0}$ , and  $\sup_{x,t} \|G_{e1}\|_F = \bar{g}_{e1} \in \mathbb{R}_{\geq 0}$ . The following definitions of a DNN [5]–[7] and NCM [1], [2], [8] will be utilized extensively hereinafter.

**Definition 6.1.** *A neural network is a nonlinear mathematical model to approximately represent training samples  $\{(x_i, y_i)\}_{i=1}^N$  of  $y = \phi(x)$  by optimally tuning its hyperparameters  $W_\ell$ , and is given as follows:*

$$y_i = \varphi(x_i; W_\ell) = T_{L+1} \circ \sigma \circ T_L \circ \cdots \circ \sigma \circ T_1(x_i) \quad (6.6)$$

where  $T_\ell(x) = W_\ell x$ ,  $\circ$  denotes composition of functions, and  $\sigma$  is an activation function (e.g.  $\sigma(x) = \tanh(x)$ , note that  $\varphi(x)$  is smooth in this case). We call a neural network that has more than two layers (i.e.,  $L \geq 2$ ) a Deep Neural Network (DNN).

**Definition 6.2.** *A Neural Contraction Metric (NCM) is a DNN model of a contraction metric given in Theorem 2.1. Its training data could be sampled by solving the CV-STEM presented in Theorem 4.2 for control (or Theorem 4.6 for differential feedback control) and Theorem 4.5 for estimation. The NCM framework is summarized in Fig. 6.1.*

**Remark 6.1.** *Although a stochastic version of the NCM is called a Neural Stochastic Contraction Metric (NSCM) in [2], we also denote it as an NCM in this thesis for simplicity of presentation.*

Since the NCM only requires one function evaluation at each time instant to get  $M$  that defines a contraction metric, without solving any optimization problems unlike the CV-STEM approaches, it enables real-time optimal feedback control and state estimation in most engineering and scientific applications.

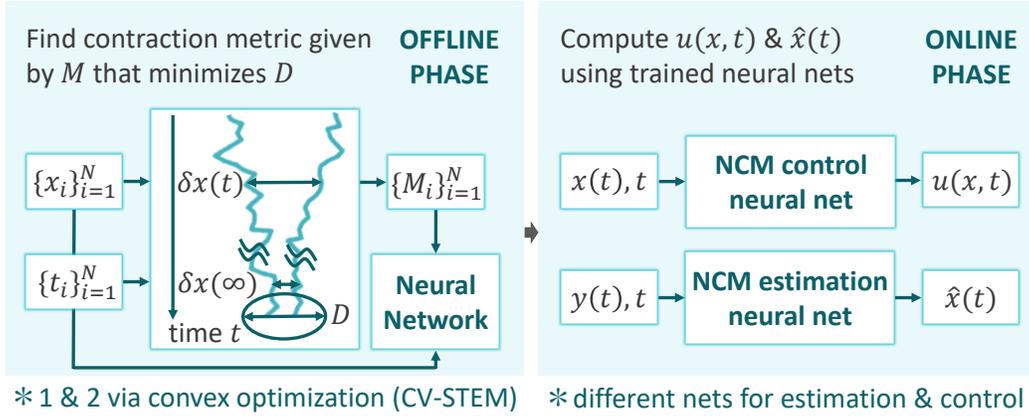


Figure 6.1: Illustration of NCM ( $x$ : system state;  $M$ : positive definite matrix that defines optimal contraction metric;  $x_i$  and  $M_i$ : sampled  $x$  and  $M$ ;  $\hat{x}$ : estimated system state;  $y$ : measurement; and  $u$ : system control input). Note that the target trajectory  $(x_d, u_d)$  is omitted in the figure for simplicity.

### 6.1 Stability of NCM-based Control and Estimation

The NCM exhibits superior incremental robustness and stability due to its internal contracting property.

**Theorem 6.1.** *Let  $\mathcal{M}$  define the NCM in Definition 6.2, and let  $u^* = u_d - R^{-1}B^\top M(x - x_d)$  for  $M$  and  $R$  given in Theorem 4.2. Suppose that the systems (6.1) and (6.2) are controlled by  $u_L$ , which computes the CV-STEM controller  $u^*$  of Theorem 4.2 replacing  $M$  by  $\mathcal{M}$ , i.e.,*

$$u_L = u_d - R(x, x_d, u_d, t)^{-1} B(x, t)^\top \mathcal{M}(x, x_d, u_d, t) \mathbf{e} \quad (6.7)$$

for  $\mathbf{e} = x - x_d$ . Note that we could use the differential feedback control  $u = u_d + \int_0^1 k d\mu$  of Theorem 4.6 for  $u^*$  and  $u_L$ . Define  $\Delta_L$  of Theorems 5.2 and 5.3 as follows:

$$\Delta_L(\varpi, t) = B(x, t)(u_L(x, x_d, u_d, t) - u^*(x, x_d, u_d, t)) \quad (6.8)$$

where we use  $q(0, t) = \xi_0(t) = x_d(t)$ ,  $q(1, t) = \xi_1(t) = x(t)$ , and  $\varpi = [x^\top, x_d^\top, u_d^\top]^\top$  in the learning-based control formulation (5.1) and (5.2). If the NCM is learned

to satisfy the learning error assumptions of Theorems 5.2 and 5.3 for  $\Delta_L$  given by (6.8), then (5.11), (5.14), and (5.15) of Theorems 5.2 and 5.3 hold as follows:

$$\|\mathbf{e}(t)\| \leq \frac{V_{\ell}(0)}{\sqrt{\underline{m}}} e^{-\alpha_{\ell} t} + \frac{\epsilon_{\ell 0} + \bar{d}_c}{\alpha_{\ell}} \sqrt{\frac{\bar{m}}{\underline{m}}} (1 - e^{-\alpha_{\ell} t}) \quad (6.9)$$

$$\mathbb{E} [\|\mathbf{e}(t)\|^2] \leq \frac{\mathbb{E}[V_{s\ell}(0)]}{\underline{m}} e^{-2\alpha_{\ell} t} + \frac{C_C \bar{m}}{2\alpha_{\ell} \underline{m}} \quad (6.10)$$

$$\mathbb{P} [\|\mathbf{e}(t)\| \geq \varepsilon] \leq \frac{1}{\varepsilon^2} \left( \frac{\mathbb{E}[V_{s\ell}(0)]}{\underline{m}} e^{-2\alpha_{\ell} t} + \frac{C_C \bar{m}}{2\alpha_{\ell} \underline{m}} \right) \quad (6.11)$$

where  $\mathbf{e} = x - x_d$ ,  $C_C = \bar{g}_c^2(2\alpha_G^{-1} + 1) + \epsilon_{\ell 0}^2 \alpha_d^{-1}$ , the disturbance upper bounds  $\bar{d}_c$  and  $\bar{g}_c$  are given in (6.1) and (6.2), respectively, and the other variables are defined in Theorems 4.2, 5.2 and 5.3.

*Proof.* Let  $g$  of the virtual systems given in Theorems 5.2 and 5.3 be  $g = \zeta$ , where  $\zeta$  is given by (3.13) (i.e.,  $\zeta = (A - BR^{-1}B^T M)(q - x_d) + \dot{x}_d$ ). By definition of  $\Delta_L$ , this has  $q = x$  and  $q = x_d$  as its particular solutions (see Example 5.1). As defined in the proof of Theorems 3.1, we have  $d = \mu d_c$  and  $G = \mu G_c$  for these virtual systems, resulting in the upper bounds of disturbances in the learning-based control formulation (5.11) and (5.14) given as

$$\bar{d} = \bar{d}_c \text{ and } \bar{g} = \bar{g}_c. \quad (6.12)$$

Since  $M$  and  $u^*$  are constructed to render  $\dot{q} = \zeta(q, \varpi, t)$  contracting as presented in Theorem 4.2, and we have  $\|\Delta_L\| \leq \epsilon_{\ell 0} + \epsilon_{\ell 1} \|\xi_1 - \xi_0\|$  due to the learning assumption (5.3), Theorem 5.2 implies the deterministic bound (6.9) and Theorem 5.3 implies the stochastic bounds (6.10) and (6.11). Note that using the differential feedback control of Theorem 4.6 as  $u^*$  also gives the bounds (6.9) – (6.11) following the same argument [8].  $\square$

Due to the estimation and control duality in differential dynamics observed in Chapter 4, we have a similar result to Theorem 6.1 for the NCM-based state estimator.

**Theorem 6.2.** *Let  $\mathcal{M}$  define the NCM in Definition 6.2, and let  $L(\hat{x}, t) = M\bar{C}^T R^{-1}$  for  $M$ ,  $\bar{C}$  and  $R$  given in Theorem 4.5. Suppose that the systems (6.4) and (6.5) are estimated with an estimator gain  $L_L$ , which computes  $L$  of the CV-STEM estimator  $\dot{\hat{x}} = f(\hat{x}, t) + L(\hat{x}, t)(y - h(\hat{x}, t))$  in Theorem 4.5 replacing  $M$  by  $\mathcal{M}$ , i.e.,*

$$\begin{aligned} \dot{\hat{x}} &= f(\hat{x}, t) + L_L(\hat{x}, t)(y - h(\hat{x}, t)) \\ &= f(\hat{x}, t) + \mathcal{M}(\hat{x}, t)\bar{C}(\varrho_c, \hat{x}, t)^T R(\hat{x}, t)^{-1}(y - h(\hat{x}, t)) \end{aligned} \quad (6.13)$$

Define  $\Delta_L$  of Theorems 5.2 and 5.3 as follows:

$$\Delta_L(\varpi, t) = (L_L(\hat{x}, t) - L(\hat{x}, t))(h(x, t) - h(\hat{x}, t)) \quad (6.14)$$

where we use  $q(0, t) = \xi_0(t) = x(t)$ ,  $q(1, t) = \xi_1(t) = \hat{x}(t)$ , and  $\varpi = [x^\top, \hat{x}^\top]^\top$  in the learning-based control formulation (5.1) and (5.2). If the NCM is learned to satisfy the learning error assumptions of Theorems 5.2 and 5.3 for  $\Delta_L$  given by (6.14), then (5.11), (5.14), and (5.15) of Theorems 5.2 and 5.3 hold as follows:

$$\|\mathbf{e}(t)\| \leq \sqrt{\bar{m}}V_{\ell}(0)e^{-\alpha_{\ell}t} + \frac{\bar{d}_a\sqrt{\frac{\bar{m}}{m}} + \bar{d}_b\bar{m}}{\alpha_{\ell}}(1 - e^{-\alpha_{\ell}t}) \quad (6.15)$$

$$\mathbb{E} [\|\mathbf{e}(t)\|^2] \leq \bar{m}\mathbb{E}[V_{s\ell}(0)]e^{-2\alpha_{\ell}t} + \frac{C_E}{2\alpha_{\ell}}\frac{\bar{m}}{m} \quad (6.16)$$

$$\mathbb{P} [\|\mathbf{e}(t)\| \geq \varepsilon] \leq \frac{1}{\varepsilon^2} \left( \bar{m}\mathbb{E}[V_{s\ell}(0)]e^{-2\alpha_{\ell}t} + \frac{C_E}{2\alpha_{\ell}}\frac{\bar{m}}{m} \right) \quad (6.17)$$

where  $\mathbf{e} = \hat{x} - x$ ,  $\bar{d}_a = \varepsilon_{\ell 0} + \bar{d}_{e0}$ ,  $\bar{d}_b = \bar{\rho}\bar{c}\bar{d}_{e1}$ ,  $C_E = (\bar{g}_{e0}^2 + \bar{\rho}^2\bar{c}^2\bar{g}_{e1}^2\bar{m}^2)(2\alpha_G^{-1} + 1) + \varepsilon_{\ell 0}^2\alpha_d^{-1}$ , the disturbance upper bounds  $\bar{d}_{e0}$ ,  $\bar{d}_{e1}$ ,  $\bar{g}_{e0}$ , and  $\bar{g}_{e1}$  are given in (6.4) and (6.5), and the other variables are defined in Theorems 4.5, 5.2 and 5.3.

*Proof.* Let  $g$  of the virtual systems given in Theorems 5.2 and 5.3 be  $g = \zeta$ , where  $\zeta$  is given by (4.10) (i.e.,  $\zeta = (A - LC)(q - x) + f(x, t)$ ). By definition of  $\Delta_L$ , this has  $q = \hat{x}$  and  $q = x$  as its particular solutions (see Example 5.2). As defined in the proof Theorem 4.3, we have

$$d = (1 - \mu)d_{e0} + \mu L_L d_{e1} \text{ and } G = [(1 - \mu)G_{e0}, \mu L_L G_{e1}]$$

for these virtual systems, resulting in the upper bounds of external disturbances in the learning-based control formulation (5.11) and (5.14) given as

$$\bar{d} = \bar{d}_{e0} + \bar{\rho}\bar{c}\bar{d}_{e1}\sqrt{\bar{m}\underline{m}} \text{ and } \bar{g} = \sqrt{\bar{g}_{e0}^2 + \bar{\rho}^2\bar{c}^2\bar{g}_{e1}^2\bar{m}^2}.$$

Since  $W = M^{-1}$  is constructed to render  $\dot{q} = \zeta(q, \varpi, t)$  contracting as presented in Theorem 4.5, and we have  $\|\Delta_L\| \leq \varepsilon_{\ell 0} + \varepsilon_{\ell 1}\|\xi_1 - \xi_0\|$  due to the learning assumption (5.3), Theorem 5.2 implies the bound (6.15), and Theorem 5.3 implies the bounds (6.16) and (6.17).  $\square$

**Remark 6.2.** As discussed in Examples 5.1 and 5.2 and in Remark 5.1, we have  $\varepsilon_{\ell 1} = 0$  for the learning error  $\varepsilon_{\ell 1}$  of (5.3) as long as  $B$  and  $h$  are bounded a compact set by the definition of  $\Delta_L$  in (6.8) and (6.14). If  $h$  and  $u^*$  are Lipschitz with respect to  $x$  in the compact set, we have non-zero  $\varepsilon_{\ell 1}$  with  $\varepsilon_{\ell 0} = 0$  in (5.3) (see Theorem 6.3).

Using Theorems 6.1 and 6.2, we can relate the learning error of the matrix that defines the NCM,  $\|\mathcal{M} - M\|$ , to the error bounds (6.9) – (6.11) and (6.15) – (6.17), if  $M$  is given by the CV-STEM of Theorems 4.2 and 4.5.

**Theorem 6.3.** *Let  $\mathcal{M}$  define the NCM in Definition 6.2, and let  $\mathcal{S}_s \subseteq \mathbb{R}^n$ ,  $\mathcal{S}_u \subseteq \mathbb{R}^m$ , and  $t \in \mathcal{S}_t \subseteq \mathbb{R}_{\geq 0}$  be some compact sets. Suppose that the systems (6.1) – (6.5) are controlled by (6.7) and estimated by (6.13), respectively, as in Theorems 6.1 and 6.2. Suppose also that  $\exists \bar{b}, \bar{c}, \bar{\rho} \in \mathbb{R}_{\geq 0}$  s.t.  $\|B(x, t)\| \leq \bar{b}$ ,  $\|C(\varrho_c, x, \hat{x}, t)\| \leq \bar{c}$ , and  $\|R^{-1}\| \leq \bar{\rho}$ ,  $\forall x, \hat{x} \in \mathcal{S}_s$  and  $t \in \mathcal{S}_t$ , for  $B$  in (6.1),  $C$  in Theorem 4.5, and  $R$  in (6.7) and (6.13). If we have for all  $x, x_d, \hat{x} \in \mathcal{S}_s$ ,  $u_d \in \mathcal{S}_u$ , and  $t \in \mathcal{S}_t$  that*

$$\|\mathcal{M}(x, x_d, u_d, t) - M(x, x_d, u_d, t)\| \leq \epsilon_\ell \text{ for control} \quad (6.18)$$

$$\|\mathcal{M}(\hat{x}, t) - M(\hat{x}, t)\| \leq \epsilon_\ell \text{ for estimation}$$

where  $M(x, x_d, u_d, t)$  and  $M(\hat{x}, t)$  (or  $M(x, t)$  and  $M(\hat{x}, t)$ , see Theorem 3.2) are the CV-STEM solutions of Theorems 4.2 and 4.5 with  $\exists \epsilon_\ell \in \mathbb{R}_{\geq 0}$  being the learning error, then the bounds (6.9) – (6.11) of Theorem 6.1 and (6.15) – (6.17) of Theorem 6.2 hold with  $\epsilon_{\ell 0} = 0$  and  $\epsilon_{\ell 1} = \bar{\rho} \bar{b}^2 \epsilon_\ell$  for control, and  $\epsilon_{\ell 0} = 0$  and  $\epsilon_{\ell 1} = \bar{\rho} \bar{c}^2 \epsilon_\ell$  for estimation, as long as  $\epsilon_\ell$  is sufficiently small to satisfy the conditions (5.10) and (5.13) of Theorems 5.2 and 5.3 for deterministic and stochastic systems, respectively.

*Proof.* For  $\Delta_L$  defined in (6.8) and (6.14) with the controllers and estimators given as  $u^* = u_d - R^{-1}B^\top M(x - x_d)$ ,  $u_L = u_d - R^{-1}B^\top \mathcal{M}(x - x_d)$ ,  $L = MC^\top R^{-1}$ , and  $L_L = \mathcal{M}C^\top R^{-1}$ , we have the following upper bounds:

$$\|\Delta_L\| \leq \begin{cases} \bar{\rho} \bar{b}^2 \epsilon_\ell \|x - x_d\| & \text{(controller)} \\ \bar{\rho} \bar{c}^2 \epsilon_\ell \|\hat{x} - x\| & \text{(estimator)} \end{cases}$$

where the relation  $\|h(x, t) - h(\hat{x}, t)\| \leq \bar{c} \|\hat{x} - x\|$ , which follows from the equality  $h(x, t) - h(\hat{x}, t) = C(\varrho_c, x, \hat{x}, t)(x - \hat{x})$  (see Lemma 3.1), is used to obtain the second inequality. This implies that we have  $\epsilon_{\ell 0} = 0$  and  $\epsilon_{\ell 1} = \bar{\rho} \bar{b}^2 \epsilon_\ell$  for control, and  $\epsilon_{\ell 0} = 0$  and  $\epsilon_{\ell 1} = \bar{\rho} \bar{c}^2 \epsilon_\ell$  for estimation in the learning error of (5.3). The rest follows from Theorems 6.1 and 6.2.  $\square$

**Example 6.1.** *The left-hand side of Fig. 6.2 shows the state estimation errors of the following Lorenz oscillator perturbed by process noise  $d_0$  and measurement noise  $d_1$  [1]:*

$$\dot{x} = [\sigma(x_2 - x_1), x_1(\rho - x_3) - x_2, x_1x_2 - \beta x_3]^\top + d_0(x, t)$$

$$y = [1, 0, 0]x + d_2(x, t)$$

where  $x = [x_1, x_2, x_3]^\top$ ,  $\sigma = 10$ ,  $\beta = 8/3$ , and  $\rho = 28$ . It can be seen that the steady-state estimation errors of the NCM and CV-STEM are below the optimal steady-state upper bound of Theorem 4.5 (dotted line in the left-hand side of Fig. 6.2) while the EKF has a larger error compared to the other two. In this example, training data is sampled along 100 trajectories with uniformly distributed initial conditions ( $-10 \leq x_i \leq 10$ ,  $i = 1, 2, 3$ ) using Theorem 4.5, and then modeled by a DNN as in Definition 6.2. Additional implementation and network training details can be found in [1].

**Example 6.2.** Consider a robust feedback control problem of the planar spacecraft perturbed by deterministic disturbances, the unperturbed dynamics of which is given as follows [9]:

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \ddot{x} = \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 1 & 1 \\ -\ell & \ell & -b & b & -\ell & \ell & -b & b \end{bmatrix} u$$

where  $x = [p_x, p_y, \phi, \dot{p}_x, \dot{p}_y, \dot{\phi}]^T$  with  $p_x$ ,  $p_y$ , and  $\phi$  being the horizontal coordinate, vertical coordinate, and yaw angle of the spacecraft, respectively, with the other variables defined as  $m =$  mass of spacecraft,  $I =$  yaw moment of inertia,  $u =$  thruster force vector,  $\ell =$  half-depth of spacecraft, and  $b =$  half-width of spacecraft (see Fig. 8 of [9] for details).

As shown in the right-hand side of Fig. 6.2, the NCM keeps their trajectories within the bounded error tube of Theorem 4.2 (shaded region) around the target trajectory (dotted line) avoiding collision with the circular obstacles, even under the presence of disturbances, whilst requiring much smaller computational cost than the CV-STEM. Data sampling and the NCM training are performed as in Example 6.1 [1].

**Remark 6.3.** We could also simultaneously synthesize a controller  $u$  and the NCM, and Theorem 6.1 can provide formal robustness and stability guarantees even for such cases [8], [10]. In Chapter 7, we generalize the NCM to learn contraction theory-based robust control using a DNN only taking  $x$ ,  $t$ , and a vector containing local environment information as its inputs, to avoid the online computation of  $x_d$  for the sake of automatic guidance and control implementable in real-time.

## 6.2 NCMs as Lyapunov Functions

We could also utilize the NCM constructed by the CV-STEM contraction metric for designing a Control Lyapunov function (CLF) [11]–[13], resulting in a stability

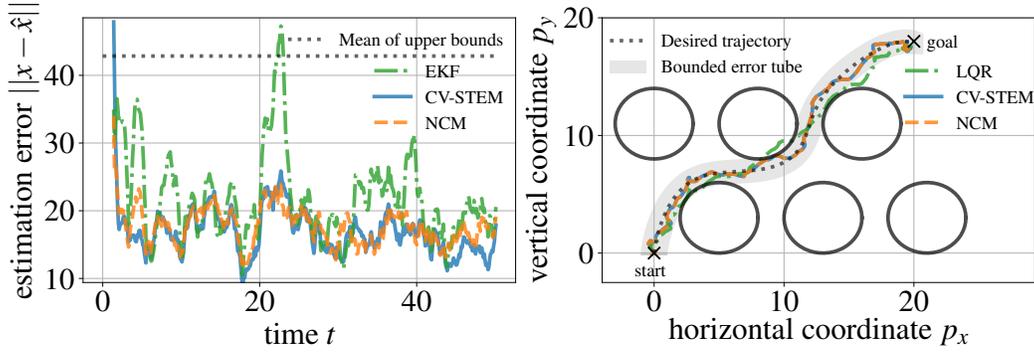


Figure 6.2: Lorenz oscillator state estimation error smoothed using a 15-point moving average filter in Example 6.1 (left), and spacecraft positions  $(p_x, p_y)$  on a planar field in Example 6.2 (right).

result different from Theorems 6.1 and 6.3. To this end, let us recall that designing optimal  $k$  of  $u = k(x, x_d, u_d, t)$  reduces to designing optimal  $K(x, x_d, u_d, t)$  of  $u = u_d - K(x, x_d, u_d, t)(x - x_d)$  due to Lemma 3.2 of Chapter 3. For such  $u$ , the virtual system of (6.1)/(6.2) and (6.3) without any perturbation, which has  $q = x$  and  $q = x_d$  as its particular solutions, is given as follows:

$$\dot{q} = (A(\varrho, x, x_d, u_d, t) - B(x, t)K(x, x_d, u_d, t))(q - x_d) + f(x_d, t) + B(x_d, t)u_d \quad (6.19)$$

where  $A$  is the SDC matrix of (6.1) in Lemma 3.1, i.e.,  $A(\varrho, x, x_d, u_d, t)(x - x_d) = f(x, t) + B(x, t)u_d - f(x_d, t) - B(x_d, t)u_d$ . Using the result of Theorem 2.3, one of the Lyapunov functions for the virtual system (6.19) with unknown  $K$  can be given as the following transformed squared length integrated over  $x$  of (6.1)/(6.2) and  $x_d$  of (6.3):

$$V_{\text{NCM}} = \int_{x_d}^x \delta q^\top \mathcal{M}(x, x_d, u_d, t) \delta q \quad (6.20)$$

where  $\mathcal{M}$  defines the NCM of Definition 6.2 modeling  $M$  of the CV-STEM contraction metric in Theorem 4.2.

**Theorem 6.4.** *Suppose (6.1) and (6.2) are controlled by  $u = u_d - K^*(x, x_d, u_d, t)\mathbf{e}$ , where  $\mathbf{e} = x - x_d$ , and that  $K^*$  is designed by the following convex optimization for given  $(x, x_d, u_d, t)$ :*

$$(K^*, p^*) = \arg \min_{K \in \mathcal{C}_K, p \in \mathbb{R}} \|K\|_F^2 + p^2 \quad (6.21)$$

$$\text{s.t. } \int_{x_d}^x \delta q^\top (\dot{\mathcal{M}} + 2 \text{sym}(\mathcal{M}A - \mathcal{M}BK)) \delta q \leq -2\alpha \int_{x_d}^x \delta q^\top (\mathcal{M} + \beta \mathbf{I}) \delta q + p \quad (6.22)$$

where  $\mathcal{M}$  defines the NCM of Definition 6.2 that models  $M$  of the CV-STEM contraction metric in Theorem 4.2 as in (6.18),  $C_K$  is a convex set containing admissible  $K$ ,  $\alpha$  is the contraction rate, and  $\beta$  is as defined in theorem 3.1 (i.e.,  $\beta = 0$  for deterministic systems and  $\beta = \alpha_s$  for stochastic systems). Then Theorems 2.4 and 2.5 still hold for the Lyapunov function defined in (6.20), yielding the following bounds:

$$\|\mathbf{e}(t)\|^2 \leq \frac{V_{\text{NCM}}(0)}{\underline{m}} e^{-2\alpha_p t} + \frac{\bar{p} + \frac{\bar{d}_c^2 \bar{m}}{\alpha_d \underline{m}}}{2\alpha_p} (1 - e^{-2\alpha_p t}) \quad (6.23)$$

$$\mathbb{E} [\|\mathbf{e}(t)\|^2] \leq \frac{\mathbb{E}[V_{\text{NCM}}(0)]}{\underline{m}} e^{-2\alpha t} + \frac{C_C \bar{m} + \bar{p}}{2\alpha \underline{m}} \quad (6.24)$$

$$\mathbb{P} [\|\mathbf{e}(t)\| \geq \varepsilon] \leq \frac{1}{\varepsilon^2} \left( \frac{\mathbb{E}[V_{\text{NCM}}(0)]}{\underline{m}} e^{-2\alpha t} + \frac{C_C \bar{m} + \bar{p}}{2\alpha \underline{m}} \right) \quad (6.25)$$

where  $\bar{p} = \sup_{x, x_d, u_d, t} p$ ,  $\underline{m}\mathbf{I} \leq \mathcal{M} \leq \bar{m}\mathbf{I}$  as in (2.26),  $\alpha_d \in \mathbb{R}_{>0}$  is an arbitrary positive constant (see (6.26)) selected to satisfy  $\alpha_p = \alpha - \alpha_d/2 > 0$ ,  $C_C = \bar{g}_c^2(2\alpha_G^{-1} + 1)$ ,  $\alpha_G \in \mathbb{R}_{>0}$  is an arbitrary constant as in Theorem 2.5,  $\varepsilon \in \mathbb{R}_{\geq 0}$ , and the disturbance terms  $\bar{d}_c$  and  $\bar{g}_c$  are given in (6.1) and (6.2), respectively. Note that the problem (6.21) is feasible due to the constraint relaxation variable  $p$ .

Furthermore, if  $\epsilon_\ell = 0$  in (6.18), (6.21) with  $p = 0$  is always feasible, and the optimal feedback gain  $K^*$  minimizes its Frobenius norm under the contraction constraint (6.22).

*Proof.* Computing  $\dot{V}_{\text{NCM}}$  for the deterministic system (6.1) along with the virtual dynamics (6.19) and the stability condition (6.22) yields

$$\dot{V}_{\text{NCM}} \leq -2\alpha V_{\text{NCM}} + \bar{d}_c \sqrt{\bar{m}} \sqrt{V_{\text{NCM}}} + \bar{p} \leq -2(\alpha - \alpha_d/2) V_{\text{NCM}} + \bar{d}_c^2 \bar{m} / \alpha_d + \bar{p} \quad (6.26)$$

where the relation  $2ab \leq \alpha_d^{-1} a^2 + \alpha_d b^2$ , which holds for any  $a, b \in \mathbb{R}$  and  $\alpha_d \in \mathbb{R}_{>0}$  ( $a = \bar{d}_c \sqrt{\bar{m}}$  and  $b = \sqrt{V_{\text{NCM}}}$  in this case), is used to obtain the second inequality. Since we can arbitrarily select  $\alpha_d$  to have  $\alpha_p = \alpha - \alpha_d/2 > 0$ , (6.26) gives the exponential bound (6.23) due to the comparison lemma of Lemma 2.1. Similarly, computing  $\mathcal{L}V_{\text{NCM}}$  for the stochastic system (6.2) yields

$$\mathcal{L}V_{\text{NCM}} \leq -2\alpha V_{\text{NCM}} + C_c \bar{m} + \bar{p}$$

which gives (6.24) and (6.25) due to Lemma 2.2. If  $\epsilon_\ell = 0$  in (6.18),  $K = R^{-1} B^\top M$  satisfies (6.22) with  $p = 0$  as we have  $\mathcal{M} = M$  and  $M$  defines the contraction metric of Theorem 4.2, which implies that (6.21) with  $p = 0$  is always feasible.  $\square$

We could also use the CCM-based differential feedback formulation of Theorem 4.6 in Theorem 6.4 (see [14], [15]). To this end, define  $E_{\text{NCM}}$  (Riemannian energy) as follows:

$$E = \int_0^1 \gamma_\mu(\mu, t)^\top \mathcal{M}(\gamma(\mu, t), t) \gamma_\mu(\mu, t) d\mu \quad (6.27)$$

where  $\mathcal{M}$  defines the NCM of Definition 6.2 that models  $M$  of the CCM in Theorem 4.6,  $\gamma$  is the minimizing geodesic connecting  $x(t) = \gamma(1, t)$  of (6.1) and  $x_d(t) = \gamma(0, t)$  of (6.3) (see Theorem 2.3), and  $\gamma_\mu = \partial\gamma/\partial\mu$ .

**Theorem 6.5.** *Suppose  $u = u^*$  of (6.1) is designed by the following convex optimization for given  $(x, x_d, u_d, t)$ :*

$$\begin{aligned} (u^*, p^*) = \arg \min_{u \in C_u, p \in \mathbb{R}} & \|u - u_d\|^2 + p^2 & (6.28) \\ \text{s.t. } & \frac{\partial E}{\partial t} + 2\gamma_\mu(1, t)^\top \mathcal{M}(x, t)(f(x, t) + B(x, t)u) \\ & - 2\gamma_\mu(0, t)^\top \mathcal{M}(x_d, t)(f(x_d, t) + B(x_d, t)u_d) \leq -2\alpha E + p & (6.29) \end{aligned}$$

where  $\mathcal{M}$  defines the NCM of Definition 6.2 that models the CCM of Theorem 4.6, and  $C_u$  is a convex set containing admissible  $u$ . Then the bound (6.23) of Theorem 6.4 holds.

Furthermore, if  $\epsilon_\ell = 0$  for the NCM learning error  $\epsilon_\ell$  in (6.18), then the problem (6.28) with  $p = 0$  is always feasible, and  $u^*$  minimizes the deviation of  $u$  from  $u_d$  under the contraction constraint (6.29).

*Proof.* Since the right-hand side of (6.29) represents  $\dot{E}$  of (6.27) if  $d_c = 0$  in (6.1), the comparison lemma of Lemma 2.1 gives (6.23) as in Theorem 6.4. The rest follows from the fact that  $u$  of Theorem 4.6 is a feasible solution of (6.28) if  $p = 0$  [14], [15].  $\square$

**Remark 6.4.** *As discussed in Remark 4.2, Theorem 6.5 can also be formulated in a stochastic setting as in Theorem 6.4 using the technique discussed in [8].*

Theorems 6.4 and 6.5 provide a perspective on the NCM stability different from Theorems 6.1 and 6.3 written in terms of the constraint relaxation variable  $p$ , by directly using the NCM as a contraction metric instead of the CV-STEM contraction metric or the CCM. Since the CLF problems are feasible with  $p = 0$  when we have zero NCM modeling error, it is implied that the upper bound of  $p$  (i.e.,  $\bar{p}$  in (6.23))

and (6.24)) decreases as we achieve a smaller learning error  $\epsilon_\ell$  using more training data for verifying (6.18) (see Remark 5.1). This intuition is also formalized later in [16]. See [14], [15] for how we compute the minimizing geodesic  $\gamma$  and its associated Riemannian energy  $E$  in Theorem 6.5.

**Example 6.3.** *If  $C_K = \mathbb{R}^{m \times n}$  and  $C_u = \mathbb{R}^m$  in (6.21) and (6.28), respectively, they can both be expressed as the following quadratic optimization problem:*

$$\begin{aligned} v^* &= \arg \min_{v \in \mathbb{R}^v} \frac{1}{2} v^\top v + c(x, x_d, u_d, t)^\top v \\ \text{s.t. } &\varphi_0(x, x_d, u_d, t) + \varphi_1(x, x_d, u_d, t)^\top v \leq 0 \end{aligned}$$

by defining  $c \in \mathbb{R}^v$ ,  $\varphi_0 \in \mathbb{R}$ ,  $\varphi_1 \in \mathbb{R}^v$ , and  $v \in \mathbb{R}^v$  appropriately. Applying the KKT condition [17, pp. 243-244] to this problem, we can show that

$$v^* = \begin{cases} -c - \frac{\varphi_0 - \varphi_1^\top c}{\varphi_1^\top \varphi_1} \varphi_1 & \text{if } \varphi_0 - \varphi_1^\top c > 0 \\ -c & \text{otherwise.} \end{cases}$$

This implies that the controller uses  $u_d$  with the feedback  $((\varphi_0 - \varphi_1^\top c)/\varphi_1^\top \varphi_1)\varphi_1$  only if  $\dot{x} = f(x, t) + B(x, t)u$  with  $u = u_d$  is not contracting, i.e.,  $\varphi_0 - \varphi_1^\top c > 0$ .

**Example 6.4.** *When  $q$  is parameterized linearly by  $\mu$  in Theorem 6.4, i.e.,  $q = \mu x + (1 - \mu)x_d$ , we have  $V_{\text{NCM}} = (x - x_d)^\top \mathcal{M}(x - x_d)$  as in the standard Lyapunov function formulation [11]–[13]. Also, a linear input constraint,  $u_{\min} \leq u \leq u_{\max}$  can be implemented by using*

$$\begin{aligned} C_K &= \{K \in \mathbb{R}^{m \times n} \mid u_d - u_{\max} \leq K\mathbf{e} \leq u_d - u_{\min}\} \\ C_u &= \{u \in \mathbb{R}^m \mid u_{\min} \leq u \leq u_{\max}\} \end{aligned}$$

in (6.21) and (6.28), respectively.

**Remark 6.5.** *For the CV-STEM and NCM state estimation in Theorems 4.5, 6.2, and 6.3, we could design a similar Lyapunov function-based estimation scheme if its contraction constraints of Theorem 4.5 do not explicitly depend on the actual state  $x$ . This can be achieved by using  $A(\varrho_A, \hat{x}, t)$  and  $C(\varrho_C, \hat{x}, t)$  instead of  $A(\varrho_A, x, \hat{x}, t)$  and  $C(\varrho_C, x, \hat{x}, t)$  in Theorem 4.5 as in Theorem 3.2 [18], leading to exponential boundedness of system trajectories as derived in [18]–[20].*

### 6.3 Remarks in NCM Implementation

This section briefly summarizes several practical techniques in constructing NCMs using the CV-STEM of Theorems 4.2, 4.5, and 4.6.

### 6.3.I Neural Network Training

We model  $M$  of the CV-STEM contraction metric sampled offline by the DNN of Definition 6.1, optimizing its hyperparameters by stochastic gradient descent [21], [22] to satisfy the learning error condition (5.3) (see Remark 5.1). The positive definiteness of  $M$  could be exploited to reduce the dimension of the DNN target output due to the following lemma [1].

**Lemma 6.1.** *A matrix  $A > 0$  has a unique Cholesky decomposition, i.e., there exists a unique upper triangular matrix  $U \in \mathbb{R}^{n \times n}$  with strictly positive diagonal entries s.t.  $A = U^T U$ .*

*Proof.* See [23, pp. 441]. □

Selecting  $\Theta$  of  $M = \Theta^T \Theta$  as the unique Cholesky decomposition of  $M$  and training the DNN using only the non-zero entries of  $\Theta$ , we can reduce the dimension of the target output by  $n(n-1)/2$  without losing any information on  $M$ . The pseudocode to obtain NCMs, using this approach depicted in Fig. 6.1, can be found in [1].

Instead of solving the convex optimization in Theorems 4.2, 4.5, and 4.6 to sample training data, we could also use them directly for training and optimizing the DNN hyperparameters, treating the constraints and the objective functions as the DNN loss functions as demonstrated in [10]. Although this approach no longer preserves the convexity and can lead only to a sub-optimal solution, this still gives the exponential tracking error bounds as long as we can get sufficiently small  $\epsilon_{\ell 0}$  and  $\epsilon_{\ell 1}$  in the learning error assumption (5.3), as discussed in Theorems 5.2, 5.3, and 6.1 – 6.3. See [8] for more details on the robustness and stability properties of this approach.

### 6.3.II Lipschitz Condition and Spectral Normalization

Let us first define a spectrally-normalized DNN, a useful mathematical tool designed to overcome the instability of network training by constraining (6.6) of Definition 6.1 to be Lipschitz, i.e.,  $\exists L_{nn} \in \mathbb{R}_{\geq 0}$  s.t.  $\|\varphi(x) - \varphi(x')\| \leq L_{nn}\|x - x'\|$ ,  $\forall x, x'$  [24], [25].

**Definition 6.3.** *A spectrally-normalized DNN is a DNN of Definition 6.1 with its weights  $W_\ell$  normalized as  $W_\ell = (C_{nn}\Omega_\ell)/\|\Omega_\ell\|$ , where  $C_{nn} \in \mathbb{R}_{\geq 0}$  is a given constant.*

**Lemma 6.2.** *A spectrally-normalized DNN given in Definition 6.3 is Lipschitz continuous with its 2-norm Lipschitz constant  $C_{nn}^{L+1}L_\sigma^L$ , where  $L_\sigma$  is the Lipschitz*

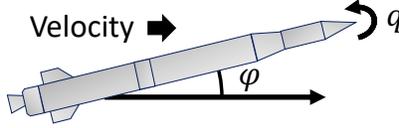


Figure 6.3: Rocket model (angle of attack  $\varphi$ , pitch rate  $q$ ).

constant of the activation function  $\sigma$  in (6.6). Also, it is robust to perturbation in its input.

*Proof.* Let  $\|f\|_{\text{Lip}}$  represent the Lipschitz constant of a function  $f$  and let  $W_\ell = (C_{nn}\Omega_\ell)/\|\Omega_\ell\|$ . Using the property  $\|f_1 \circ f_2\|_{\text{Lip}} \leq \|f_1\|_{\text{Lip}}\|f_2\|_{\text{Lip}}$ , we have for a spectrally-normalized DNN in Definition 6.3 that

$$\|\varphi(x; W_\ell)\| \leq \|\sigma\|_{\text{Lip}}^L \prod_{\ell=1}^{L+1} \|T_\ell\|_{\text{Lip}} = L_\sigma^L \prod_{\ell=1}^{L+1} \|W_\ell\|$$

where we used the fact that  $\|f\|_{\text{Lip}}$  of a differentiable function  $f$  is equal to the maximum spectral norm (induced 2-norm) of its gradient over its domain to obtain the last equality. Since  $\|W_\ell\| = C_{nn}$  holds, the Lipschitz constant of  $\varphi(x_i; W_\ell)$  is  $C_{nn}^{L+1} L_\sigma^L$  as desired. Since  $\varphi$  is Lipschitz, we have for small perturbation  $x_\epsilon$  that

$$\|\varphi(x + x_\epsilon; W_\ell) - \varphi(x; W_\ell)\| \leq C_{nn}^{L+1} L_\sigma^L \|(x + x_\epsilon) - x\| = C_{nn}^{L+1} L_\sigma^L \|x_\epsilon\|$$

implying robustness to input perturbation.  $\square$

In general, a spectrally-normalized DNN is useful for improving the robustness and generalization properties of DNNs, and for obtaining the learning error bound as in (5.3), as delineated in Remark 5.1 [24], [26], [27]. Using it for system identification yields a DNN-based nonlinear feedback controller with a formal stability guarantee as shall be seen in Sec. 8.2.II.

For the NCM framework with stochastic perturbation, we can utilize the spectrally-normalized DNN of Definition 6.3 to guarantee the Lipschitz condition on  $\partial M/\partial x_i$ , appeared in the stochastic contraction condition of Theorem 2.5 (see Proposition 1 of [2]). We could also utilize the technique proposed in [28]–[30], which designs Lipschitz-bounded equilibrium neural networks using contraction theory, for obtaining a result analogous to Lemma 6.2. The pseudocode for the NCM construction in a stochastic setting can be found in [2].

**Example 6.5.** *Let us demonstrate how having the Lipschitz condition on  $\partial M/\partial x_i$  in Theorem 2.5 would affect the NCM-based control and estimation performance.*

Table 6.1: Notations in Example 6.5.

Notation	Description
$\varphi$	angle of attack (state variable)
$q$	pitch rate (state variable)
$u$	tail fin deflection (control input)
$y$	measurement
$M$	Mach number
$p_0$	static pressure at 20,000 ft
$S$	reference area
$d$	reference diameter
$I_y$	pitch moment of inertia
$g$	acceleration of gravity
$m$	rocket mass
$V$	rocket speed

We consider a rocket autopilot problem perturbed by stochastic disturbances as in [2], the unperturbed dynamics of which is depicted in Fig. 6.3 and given as follows, assuming that the pitch rate  $q$  and specific normal force are available as a measurement via rate gyros and accelerometers [31], [32]:

$$\frac{d}{dt} \begin{bmatrix} \varphi \\ q \end{bmatrix} = \begin{bmatrix} \frac{gC(M) \cos(\varphi) \phi_z(\varphi, M)}{\frac{mV}{C(M)d\phi_m(\varphi)}} + q \\ \frac{C(M)\bar{d}_m}{I_y} \end{bmatrix} + \begin{bmatrix} \frac{gC(M)\bar{d}_n \cos(\varphi)}{\frac{mV}{C(M)\bar{d}_m}} \\ \frac{C(M)\bar{d}_n}{I_y} \end{bmatrix} u$$

$$y = \begin{bmatrix} q \\ \frac{C(M)\phi_z(\varphi, M)}{m} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{C(M)\bar{d}_n}{m} \end{bmatrix} u$$

where  $C(M) = p_0 M^2 S$ ,  $\phi_z(\varphi, M) = 0.7(a_n \varphi^3 + b_n \varphi |\varphi| + c_n(2 + M/3)\varphi)$ ,  $\phi_m(\varphi, M) = 0.7(a_m \varphi^3 + b_m \varphi |\varphi| - c_m(7 - 8M/3)\varphi)$ ,  $\bar{d}_n = 0.7d_n$ ,  $\bar{d}_m = 0.7d_m$ ,  $(a_n, b_n, c_n, d_n)$  and  $(a_m, b_m, c_m, d_m)$  are given in [32], and the notations are defined in Table 6.1. Since this example explicitly takes into account stochastic perturbation, the spectrally-normalized DNN of Definition 6.3 is used to guarantee the Lipschitz condition of Theorem 2.5 by Lemma 6.2.

Figure 6.4 shows the state estimation and tracking error performance of each estimator and controller, where the NSCM is the stochastic counterpart of the NCM aforementioned in Remark 6.1. It is demonstrated that the steady-state errors of the NSCM and CV-STEM are indeed smaller than its steady-state upper bounds of Theorems 4.2 and 4.5 (solid black line), while the other controllers violate this condition. In particular, the optimal contraction rate of the deterministic NCM for state

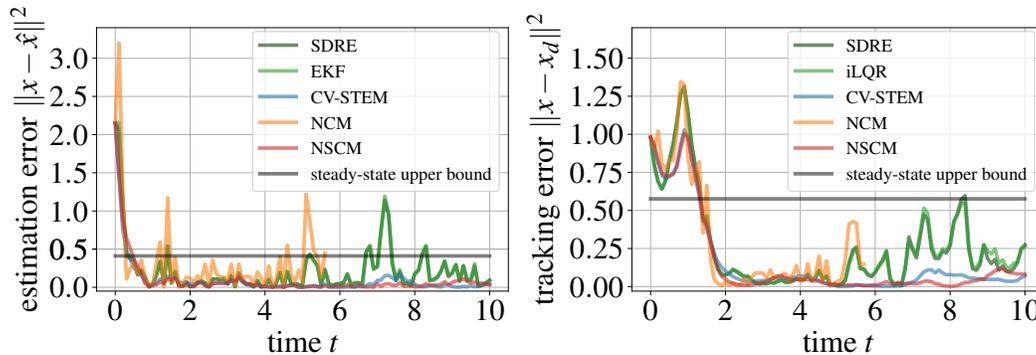


Figure 6.4: State estimation and tracking errors in Example 6.5 ( $x = [\varphi, q]^T$ ).

estimation is turned out to be much larger than the NSCM as it does not account for stochastic perturbation, which makes the NCM trajectory diverge around  $t = 5.8$  in Fig. 6.4. The NSCM circumvents this difficulty by imposing the Lipschitz condition on  $\partial M / \partial x_i$  as in Theorem 2.5 using spectral normalization of Lemma 6.2. See [2] for additional details.

## References

- [1] H. Tsukamoto and S.-J. Chung, “Neural contraction metrics for robust estimation and control: A convex optimization approach,” *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 211–216, 2021.
- [2] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Neural stochastic contraction metrics for learning-based control and estimation,” *IEEE Control Syst. Lett.*, vol. 5, no. 5, pp. 1825–1830, 2021.
- [3] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Learning-based Adaptive Control using Contraction Theory,” in *IEEE Conf. Decis. Control*, 2021, pp. 2533–2538.
- [4] H. Tsukamoto and S.-J. Chung, “Learning-based robust motion planning with guaranteed stability: A contraction theory approach,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6164–6171, 2021.
- [5] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989, ISSN: 0893-6080.
- [6] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Math. Control Signals Syst.*, vol. 2, no. 4, pp. 303–314, Dec. 1989, ISSN: 1435-568X.

- [7] K.-I. Funahashi, “On the approximate realization of continuous mappings by neural networks,” *Neural Netw.*, vol. 2, no. 3, pp. 183–192, 1989, ISSN: 0893-6080.
- [8] H. Tsukamoto, S.-J. Chung, J.-J. Slotine, and C. Fan, “A theoretical overview of neural contraction metrics for learning-based control with guaranteed stability,” in *IEEE Conf. Decis. Control*, 2021, pp. 2949–2954.
- [9] Y. Nakka, R. Foust, E. Lupu, *et al.*, “A six degree-of-freedom spacecraft dynamics simulator for formation control research,” in *Astrodyn. Special Conf.*, 2018.
- [10] D. Sun, S. Jha, and C. Fan, *Learning certified control using contraction metric*, arXiv:2011.12569, Nov. 2020.
- [11] E. Sontag, “A Lyapunov-like characterization of asymptotic controllability,” *SIAM J. Control Optim.*, vol. 21, no. 3, pp. 462–471, 1983.
- [12] E. D. Sontag, “A ‘universal’ construction of Artstein’s theorem on nonlinear stabilization,” *Syst. Control Lett.*, vol. 13, no. 2, pp. 117–123, 1989.
- [13] H. K. Khalil, *Nonlinear Systems*, 3rd. Upper Saddle River, NJ: Prentice-Hall, 2002.
- [14] I. R. Manchester and J.-J. E. Slotine, “Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design,” *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 3046–3053, Jun. 2017.
- [15] S. Singh, A. Majumdar, J.-J. E. Slotine, and M. Pavone, “Robust online motion planning via contraction theory and convex optimization,” in *IEEE Int. Conf. Robot. Automat.*, May 2017, pp. 5883–5890.
- [16] N. M. Boffi, S. Tu, N. Matni, J.-J. E. Slotine, and V. Sindhvani, *Learning stability certificates from data*, arXiv:2008.05952, 2020.
- [17] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Mar. 2004.
- [18] A. P. Dani, S.-J. Chung, and S. Hutchinson, “Observer design for stochastic nonlinear systems via contraction-based incremental stability,” *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 700–714, Mar. 2015.
- [19] H. Tsukamoto and S.-J. Chung, “Robust controller design for stochastic nonlinear systems via convex optimization,” *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4731–4746, 2021.
- [20] H. Tsukamoto and S.-J. Chung, “Convex optimization-based controller design for stochastic nonlinear systems using contraction analysis,” in *IEEE Conf. Decis. Control*, Dec. 2019, pp. 8196–8203.
- [21] H. Robbins and S. Monro, “A stochastic approximation method,” *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 1951.

- [22] P. M. Wensing and J.-J. E. Slotine, “Beyond convexity – Contraction and global convergence of gradient descent,” *PLOS ONE*, vol. 15, pp. 1–29, Aug. 2020.
- [23] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd. Cambridge University Press, 2012, ISBN: 0521548233.
- [24] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” in *Int. Conf. Learn. Representations*, 2018.
- [25] G. Shi, X. Shi, M. O’Connell, *et al.*, “Neural lander: Stable drone landing control using learned dynamics,” in *IEEE Int. Conf. Robot. Automat.*, May 2019, pp. 9784–9790.
- [26] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, “Spectrally-normalized margin bounds for neural networks,” in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
- [27] B. Neyshabur, S. Bhojanapalli, and N. Srebro, “A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks,” in *Int. Conf. Learn. Representations*, 2018.
- [28] M. Revay, R. Wang, and I. R. Manchester, *Lipschitz bounded equilibrium networks*, arXiv:2010.01732, Oct. 2020.
- [29] M. Revay, R. Wang, and I. R. Manchester, *Recurrent equilibrium networks: Flexible dynamic models with guaranteed stability and robustness*, arXiv:2104.05942, Apr. 2021.
- [30] I. R. Manchester, M. Revay, and R. Wang, “Contraction-based methods for stable identification and robust machine learning: A tutorial,” in *IEEE Conf. Decis. Control*, 2021, pp. 2955–2962.
- [31] J. S. Shamma and J. R. Cloutier, “Gain-scheduled missile autopilot design using linear parameter varying transformations,” *AIAA J. Guid. Control Dyn.*, vol. 16, no. 2, pp. 256–263, 1993.
- [32] S. Bennani, D. Willemsen, C. Scherer, C. Scherer, S. Bennani, and D. Willemsen, “Robust LPV control with bounded parameter rates,” in *Guid. Navigation Control Conf.*, 1997.

## Chapter 7

## LEARNING-BASED SAFE AND ROBUST MOTION PLANNING

- [1] H. Tsukamoto and S.-J. Chung, “Learning-based robust motion planning with guaranteed stability: A contraction theory approach,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6164–6171, 2021.
- [2] S. Singh, H. Tsukamoto, B. T. Lopez, S.-J. Chung, and J.-J. E. Slotine, “Safe motion planning with tubes and contraction metrics,” in *IEEE Conf. Decis. Control*, 2021, pp. 2943–2948.
- [3] H. Tsukamoto, B. Rivière, C. Choi, A. Rahmani, and S.-J. Chung, “CART: Collision avoidance and robust tracking augmentation in learning-based motion planning for multi-agent systems,” in *IEEE Conf. Decis. Control*, *under review*, 2023.

One of the limitations of the CV-STEM and NCM control methods is that they require the computation of target trajectories, which could be difficult to perform optimally in real time, e.g., when we consider nonlinear systems possibly with multiple robotic agents and obstacles. The purpose of this chapter is to show that the CV-STEM and NCM are also useful for designing a real-time robust motion planning algorithm, especially for systems perturbed by deterministic and stochastic disturbances as in (6.1) and (6.2). This is not for proposing a new motion planner to compute  $x_d$  and  $u_d$  of (6.3), but for augmenting existing learning-based motion planners with real-time robustness, stability, and safety guarantees via contraction theory.

Table 7.1: Comparison of LAG-ROS [1] with the learning-based and robust tube-based motion planners.

Motion planning scheme	State tracking error	Computational load
(a) $(o_\ell, t) \mapsto u_d$	Increases exponentially (Thm. 5.1)	One DNN evaluation
(b) $(x, x_d, u_d, t) \mapsto u^*$	Exponentially bounded (Thm. 4.2, 4.6)	Computation of $(x_d, u_d)$
(c) $(x, o_\ell, t) \mapsto u^*$	Exponentially bounded (Thm. 7.1)	One DNN evaluation

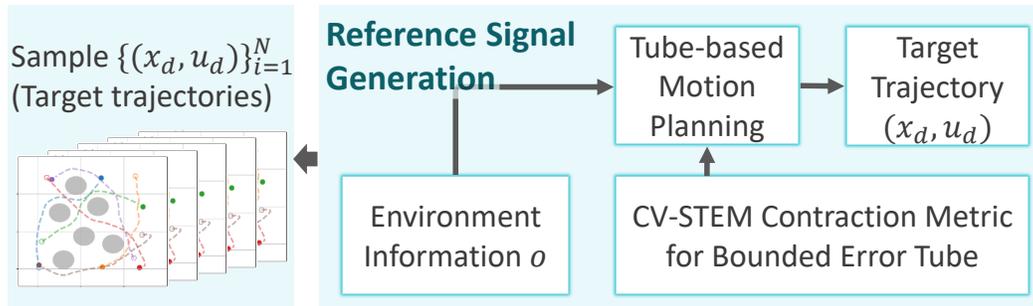


Figure 7.1: Block diagram of tube-based planning using contraction theory.

## 7.1 Overview of Existing Motion Planners

Let us briefly review the following standard motion planning techniques and their limitations:

- (a) Learning-based motion planner (see, e.g., [2]–[8]):  $(o_\ell(x, o_g), t) \mapsto u_d(o_g, t)$ , modeled by a DNN, where  $o_g \in \mathbb{R}^g$  is a vector containing global environment information, and  $o_\ell : \mathbb{R}^n \times \mathbb{R}^g \mapsto \mathbb{R}^\ell$  with  $\ell \leq g$  is local environment information extracted from  $o_g \in \mathbb{R}^g$  [2].
- (b) Robust tube-based motion planner (see, e.g., [9]–[25]):  $(x, x_d, u_d, t) \mapsto u^*$ , where  $u^*$  is robust feedback tracking control of Theorems 4.2, 4.6, 6.4, or 6.5.

The robust tube-based motion planner (b) ensures that the perturbed trajectories  $x$  of (6.1) and (6.2) stay in an exponentially bounded error tube around the target trajectory  $x_d$  of (6.3) [9]–[17], [23]–[25] given as in Fig. 7.1, using Theorems 2.4 and 2.5. However, it requires the online computation of  $(x_d, u_d)$  as an input to their control policy given in Theorems 4.2, 4.6, 6.4, and 6.5, which is not realistic for systems with limited computational resources. The learning-based motion planner (a) circumvents this issue by modeling the target policy  $(o_\ell, t) \mapsto u_d$  by a DNN. In essence, our approach, to be proposed in Theorem 7.1, is for providing (a) with the contraction theory-based stability guarantees of (b), thereby significantly enhancing the performance of (a) that only assures the tracking error  $\|x - x_d\|$  to be bounded by a function which increases exponentially with time, as derived previously in Theorem 5.1 of Chapter 5.

## 7.2 Stability Guarantees of LAG-ROS

The method of LAG-ROS, Learning-based Autonomous Guidance with ROBustness and Stability guarantees [1], bridges the gap between (a) and (b) by ensuring that the distance between the target and controlled trajectories to be exponentially bounded.

- (c) LAG-ROS (see Fig. 7.6 and [1]):  $(x, o_\ell(x, o_g), t) \mapsto u^*(x, x_d(o_g, t), u_d(o_g, t), t)$ , modeled by  $u_L$  to be given in Theorem 7.1, where  $u^*(x, x_d, u_d, t)$  of (b) is viewed as a function of  $(x, o_\ell, t)$ .

Table 7.1 summarizes the differences of the existing motion planners (a) and (b) from (c), which is defined as follows.

**Definition 7.1.** *Learning-based Autonomous Guidance with RObustness and Stability guarantees (LAG-ROS) is a DNN model for approximating a function which maps state  $x$ , local environment information  $o_\ell$ , and time  $t$  to an optimal robust feedback control input  $u^*$  given by (3.10) of Theorem 4.2, i.e.,  $u^* = u_d - K(x - x_d)$ , or (4.22) of Theorem 4.6, i.e.,  $u = u_d + \int_0^1 k d\mu$ , where its training data is sampled as explained in Sec. 7.3 (see Figures 7.3, 7.4, and 7.6).*

The LAG-ROS in Definition 7.1 achieves online computation of  $u$  without solving a motion planning problem, and it still possesses superior robustness and stability properties due to its internal contracting architecture [1].

**Theorem 7.1.** *Let  $u_L = u_L(x, o_\ell(x, o_g), t)$  be the LAG-ROS in Definition 7.1, and let  $u^*$  be given by (3.10) of Theorem 4.2, i.e.,  $u^* = u_d - K(x - x_d)$ , or (4.22) of Theorem 4.6, i.e.,  $u^* = u_d + \int_0^1 k d\mu$ . Define  $\Delta_L$  of Theorems 5.2 and 5.3 as follows:*

$$\Delta_L = B(u_L(x, o_\ell(x, o_g), t) - u^*(x, x_d(o_g, t), u_d(o_g, t), t))$$

where  $B = B(x, t)$  is the actuation matrix given in (6.1) and (6.2). Note that we use  $q(0, t) = \xi_0(t) = x_d(o_g, t)$  and  $q(1, t) = \xi_1(t) = x(t)$  in the learning-based control formulation of Theorems 5.2 and 5.3.

If the LAG-ROS  $u_L$  is learned to satisfy (5.3) with  $\epsilon_{\ell 1} = 0$ , i.e.,  $\|\Delta_L\| \leq \epsilon_{\ell 0}$  for all  $x \in \mathcal{S}_s$ ,  $o_g \in \mathcal{S}_o$ , and  $t \in \mathcal{S}_t$ , where  $\mathcal{S}_s \subseteq \mathbb{R}^n$ ,  $\mathcal{S}_o \subseteq \mathbb{R}^s$ , and  $\mathcal{S}_t \subseteq \mathbb{R}_{\geq 0}$  are some compact sets, and if the control-affine nonlinear systems (6.1) and (6.2) are controlled by  $u = u_L$ , then (5.11) of Theorem 5.2 and (5.14) of Theorem 5.3 hold with  $\bar{d}$  and  $\bar{g}$  replaced as in (6.12), i.e., we have the following for  $\mathbf{e}(t) = x(t) - x_d(o_g, t)$ :

$$\|\mathbf{e}(t)\| \leq \frac{V_\ell(0)}{\sqrt{\underline{m}}} e^{-\alpha t} + \frac{\bar{d}_{\epsilon_\ell}}{\alpha} \sqrt{\frac{\bar{m}}{\underline{m}}} (1 - e^{-\alpha t}) = r_\ell(t) \quad (7.1)$$

$$\mathbb{E} [\|\mathbf{e}(t)\|^2] \leq \frac{\mathbb{E}[V_{s\ell}(0)]}{\underline{m}} e^{-2\alpha t} + \frac{C_{\epsilon_\ell} \bar{m}}{2\alpha \underline{m}} = r_{s\ell}(t) \quad (7.2)$$

$$\mathbb{P} [\|\mathbf{e}(t)\| \geq \varepsilon] \leq \frac{1}{\varepsilon^2} \left( \frac{\mathbb{E}[V_{s\ell}(0)]}{\underline{m}} e^{-2\alpha t} + \frac{C_{\epsilon_\ell} \bar{m}}{2\alpha \underline{m}} \right) \quad (7.3)$$

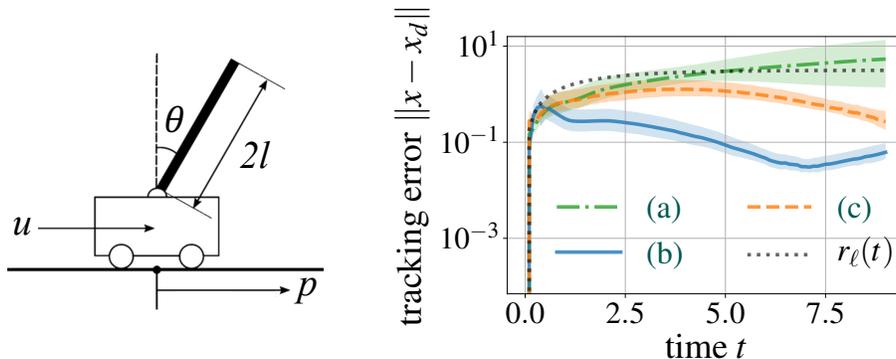


Figure 7.2: Cart-pole balancing:  $x = [p, \theta, \dot{p}, \dot{\theta}]^\top$ ,  $x$  and  $x_d$  are given in (6.1) and (6.3), (a) – (c) are given in Table 7.1, and  $r_\ell$  is given in (7.4). The shaded area denotes the standard deviation ( $+1\sigma$  and  $-0.5\sigma$ ).

where  $x_d = x_d(o_g, t)$  and  $u_d = u_d(o_g, t)$  are as given in (6.3),  $\bar{d}_{\epsilon_\ell} = \epsilon_{\ell 0} + \bar{d}_c$ ,  $C_{\epsilon_\ell} = \bar{g}_c^2(2\alpha_G^{-1} + 1) + \epsilon_{\ell 0}^2 \alpha_d^{-1}$ , and the other variables are as defined in Theorems 5.2 and 5.3 with the disturbance terms in (6.12) of Theorem 6.1.

*Proof.* We have  $f(x, t) + B(x, t)u_L = f(x, t) + B(x, t)u^* + B(x, t)(u_L - u^*)$  and  $\|\Delta_L\| = \|B(x, t)(u_L - u^*)\| \leq \epsilon_{\ell 0}$  by (5.3) with  $\epsilon_{\ell 1} = 0$  for  $x_d = x_d(o_g, t)$  and  $u_d = u_d(o_g, t)$ . Since the virtual system which has  $x$  of  $\dot{x} = f(x, t) + B(x, t)u^*$  and  $x_d$  of (6.3) as its particular solutions is contracting due to Theorems 4.2 or 4.6, the application of Theorems 5.2 and 5.3 yields the desired relations (7.1) – (7.3).  $\square$

Due to its internal feedback structure, the LAG-ROS achieves exponential boundedness of the trajectory tracking error as in Theorem 7.1, thereby improving the performance of existing learning-based feedforward control laws to model  $(o_\ell, t) \mapsto u_d$  as in (a), whose tracking error bound increases exponentially with time as in Theorem 5.1 (see [1]). This property enables its use in safety-critical guidance and control tasks which we often encounter in modern machine learning applications.

**Example 7.1.** Let us consider the cart-pole balancing task in Fig. 7.2, perturbed externally by deterministic perturbation, to demonstrate the differences of (a) – (c) summarized in Table 7.1. Its dynamical system is given in [26], [27] with its values defined in [27].

The target trajectories  $(x_d, u_d)$  are sampled with the objective function  $\int_0^T \|u\|^2 dt$ , and the LAG-ROS of Theorem 7.1 is modeled by a DNN independently of the target trajectory as detailed in [1]. The environment information  $o_g$  is selected as random

initial and target terminal states, and we let  $V_\ell(0) = 0$  and  $\epsilon_{\ell 0} + \bar{d}_c = 0.75$  in (7.1) of Theorem 7.1, yielding the following exponential bound:

$$r_\ell(t) = \frac{V_\ell(0)}{\sqrt{\bar{m}}} + \frac{\bar{d}_{\epsilon_\ell}}{\alpha} \sqrt{\frac{\bar{m}}{m}} (1 - e^{-\alpha t}) = 3.15(1 - e^{-0.60t}) \quad (7.4)$$

where  $\bar{d}_{\epsilon_\ell} = \epsilon_{\ell 0} + \bar{d}_c$ .

Figure 7.2 shows the tracking errors of each motion planner averaged over 50 simulations at each time instant  $t$ . The LAG-ROS (c) and robust tube-based planner (b) indeed satisfy the bound (7.4) (dotted black curve) for all  $t$  with a small standard deviation  $\sigma$ , unlike learning-based motion planner (a) with a diverging bound in Theorem 5.1 and increasing deviation  $\sigma$ . This example demonstrates one of the major advantages of contraction theory, which enables such quantitative analysis on robustness and stability of learning-based planners.

As implied in Example 7.1, the LAG-ROS indeed possesses the robustness and stability guarantees of  $u^*$  as proven in Theorem 7.1, unlike (a), while retaining significantly lower computational cost than that of (b). See [1] for a more detailed discussion of this simulation result.

### 7.3 Tube-based State Constraint Satisfaction

We exploit the result of Theorem 7.1 in the tube-based motion planning [9] for generating LAG-ROS training data, which satisfies given state constraints even under the existence of the learning error and disturbance of Theorem 5.2. In particular, we sample target trajectories  $(x_d, u_d)$  of (6.3) in  $\mathcal{S}_x \times \mathcal{S}_u$  of Theorem 7.1 by solving the following, assuming the learning error  $\epsilon_{\ell 0}$  and disturbance upper bound  $\bar{d}_c$  of Theorem 7.1 are selected *a priori* [28]:

$$\min_{\substack{\bar{x}=\bar{x}(o_g,t) \\ \bar{u}=\bar{u}(o_g,t)}} \int_0^T c_0 \|\bar{u}\|^2 + c_1 P(\bar{x}, \bar{u}, t) dt \quad (7.5)$$

$$\text{s.t. } \dot{\bar{x}} = (\bar{x}, t) + B(\bar{x}, t)\bar{u}, \quad \bar{x} \in \bar{\mathcal{X}}(o_g, t), \quad \text{and } \bar{u} \in \bar{\mathcal{U}}(o_g, t)$$

where  $c_0 > 0$ ,  $c_1 \geq 0$ ,  $P(\bar{x}, \bar{u}, t)$  is some performance-based cost function (e.g., information-based cost [29]),  $\bar{\mathcal{X}}$  is robust admissible state space defined as  $\bar{\mathcal{X}}(o_g, t) = \{v(t) \in \mathbb{R}^n \mid \forall \xi(t) \in \{\xi(t) \mid \|v(t) - \xi(t)\| \leq r_\ell(t)\}, \xi(t) \in \mathcal{X}(o_g, t)\}$ ,  $\mathcal{X}(o_g, t)$  is given admissible state space,  $r_\ell(t)$  is given by (7.1) of Theorem 7.1, and  $\bar{u} \in \bar{\mathcal{U}}(o_g, t)$  is an input constraint. The following lemma shows that the LAG-ROS ensures the perturbed state  $x$  of (6.1) to satisfy  $x \in \mathcal{X}$ , due to the contracting property of  $u^*$ .

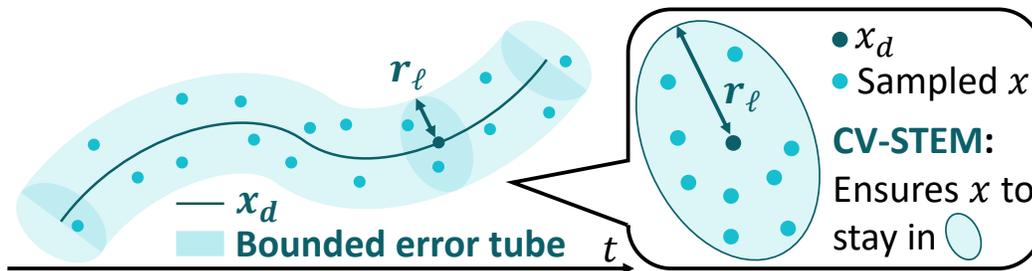


Figure 7.3: Illustration of state sampling in a robust bounded error tube.

**Lemma 7.1.** *If the solution of (7.5) gives  $(x_d, u_d)$ , the LAG-ROS  $u_L$  of Theorem 7.1 ensures the perturbed solution  $x(t)$  of  $\dot{x} = f(x, t) + B(x, t)u_L + d_c(x, t)$  in (6.1) to stay in the admissible state space  $\mathcal{X}(o_g, t)$ , i.e.,  $x(t) \in \mathcal{X}(o_g, t)$ , even with the learning error  $\epsilon_{\ell 0}$  of  $u_L$  in Theorem 7.1.*

*Proof.* See [1]. □

Lemma 7.1 implies that the perturbed trajectory (6.1) controlled by LAG-ROS will not violate the given state constraints as long as  $x_d$  is sampled by (7.5), which helps greatly reduce the need for safety control schemes such as [30]. The localization method in [2] allows extracting  $o_\ell$  of (a) by  $o_g$  of (7.5), to render LAG-ROS applicable to different environments in a distributed way with a single policy.

We therefore sample artificially perturbed states  $x$  in the tube  $S(x_d(o_g, t)) = \{\xi(t) \in \mathbb{R}^n \mid \|\xi(t) - x_d(o_g, t)\| \leq r_\ell(t)\}$  as depicted in Figures 7.3 and 7.4 [1]. The robust control inputs  $u^*$  for training LAG-ROS of Theorem 7.1 is then sampled by computing the CV-STEM of Theorem 4.2 or Theorem 4.6. The LAG-ROS framework is summarized in Fig. 7.6, and a pseudocode for its offline construction can be found in [1].

**Remark 7.1.** *For stochastic systems, we could sample  $x$  around  $x_d$  using a given probability distribution of the disturbance using (7.2) and (7.3) of Theorem 7.1.*

**Example 7.2.** *Let us again consider the spacecraft dynamics of Example 6.2, but now for designing a real-time robust motion planner using the LAG-ROS in a multi-agent setting. In this example, each agent is assumed to have access only to the states of the other agents/obstacles located in a circle with a limited radius, 2.0 m, centered around it. The performance of LAG-ROS is compared with (a), (b), and the following centralized motion planner (d) which is not computable in real-time:*

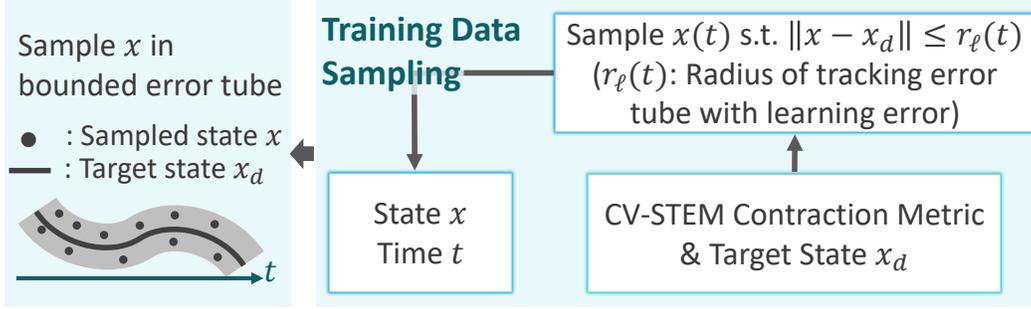


Figure 7.4: Block diagram of state sampling in a robust bounded error tube.

(d) *Centralized robust motion planner:*

$(x, x_d, u_d, t) \mapsto u^*$ , *offline centralized solution of (b).*

The objective function to sample target trajectories of (6.3) is selected as  $\int_0^T \|u\|^2 dt$  with their target terminal positions  $x_f = [p_{xf}, p_{yf}, 0, 0]^\top$ , where  $(p_{xf}, p_{yf})$  is a random position in  $(0, 0) \leq (p_{xf}, p_{yf}) \leq (5, 5)$ . For the state constraint satisfaction in Lemma 7.1, the following error tube (7.1):

$$r_\ell(t) = (\bar{d}_{\epsilon_\ell} / \alpha) \sqrt{\chi} (1 - e^{-\alpha t}) = 0.125(1 - e^{-0.30t})$$

is used with an input constraint  $u_i \geq 0, \forall i$ , to avoid collisions with a random number of multiple circular obstacles and of other agents, even under the learning error and disturbances. See [1] for the LAG-ROS training details [1].

Figure 7.5 shows one example of the trajectories of the motion planners (a) – (d) under deterministic external disturbances. It implies the following:

- For (a), the tracking error accumulates exponentially with time due to the lack of robustness as proven in Theorem 5.1.
- (b) only yields locally optimal  $(x_d, u_d)$  of (6.3), as its time horizon has to be small enough to make the problem solvable online with limited computational capacity, only with local environment information. This renders some agents stuck in local minima as depicted in Fig. 7.5.
- LAG-ROS (c) tackles these two problems by providing formal robustness and stability guarantees of Theorem 7.1, whilst implicitly knowing the global solution (only from the local information  $o_\ell$  as in [2]) without computing it online. It indeed satisfies the state constraints due to Lemma 7.1 as can be seen from Fig. 7.5.

See [1] for an in-depth discussion on this simulation result.

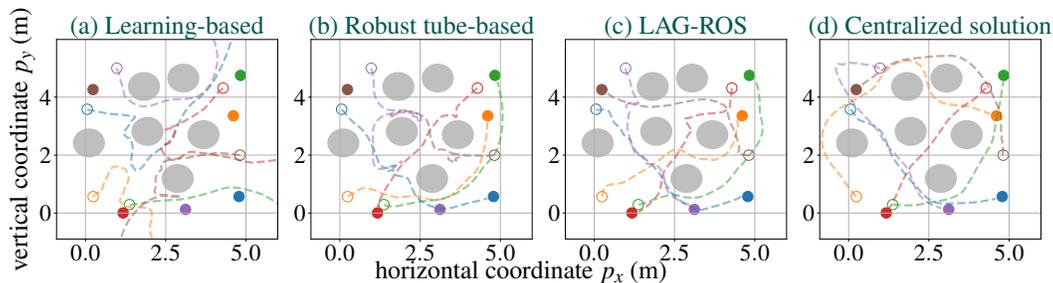


Figure 7.5: Trajectories for the learning-based planner (a), robust tube-based planner (b), LAG-ROS (c), and offline centralized solution (d) (○: start, ●: goal).

#### 7.4 Safe Exploration and Learning of Disturbances

All the theorems presented so far assume that the sizes of unknown external disturbances are fixed (i.e.,  $\bar{d}$  in Theorem 2.4 and  $(\bar{g}_0, \bar{g}_1)$  in Theorem 2.5 are known and fixed). Since such an assumption could yield conservative state constraints, e.g., in utilizing motion planning of Lemma 7.1 with the tube of Theorem 7.1, we could consider better estimating the unknown parts,  $d(x, t)$  in Theorem 2.4 or  $G_0(x, t)$  and  $G_1(x, t)$  in Theorem 2.5, also using contraction theory and machine learning. For example, it is demonstrated in [22] that the stochastic bounds of Theorem 2.5 can be used to ensure safe exploration and reduction of uncertainty over epochs in learning unknown, control non-affine residual dynamics. Here, its optimal motion plans are designed by solving chance-constrained trajectory optimization and executed using a feedback controller such as Theorem 4.2 with a control barrier function-based safety filter [30], [31]. It is also shown in [13]–[15] that the state-dependent disturbance  $d(x)$  in Theorem 2.5 can be learned adaptively with the Gaussian process while ensuring safety.

These techniques are based on the robustness and stability properties of contraction theory-based control and estimation schemes, which guarantee their state trajectories to stay in a tube around the target trajectory as in Theorems 2.4, 2.5, 5.2, and 5.3, and at the same time, enable safely sampling training data for learning the unknown part of the dynamics. Chapter 8 is for giving an overview of these concepts in the context of contraction theory-based adaptive control [26], [32], [33] and robust control synthesis for learned models [28], [34]–[37].

#### 7.5 Collision Avoidance and Robust Tracking Augmentation

One limitation of the LAG-ROS mentioned earlier in this chapter is that the size of the disturbance depends on the learning error, although we can prove robustness

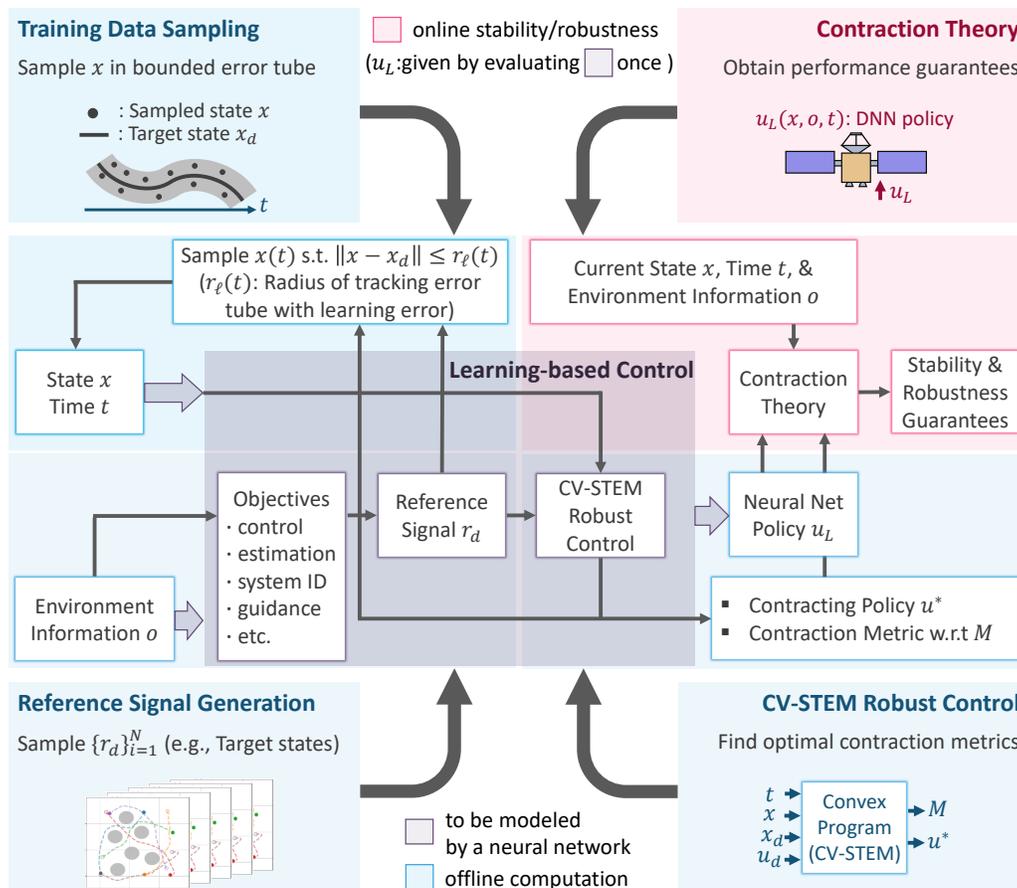


Figure 7.6: Detailed block diagram of machine learning-based control using contraction theory, including LAG-ROS, where Fig. 4.1, Fig. 7.1, and Fig. 7.4 are utilized as building blocks.

as in Theorem 7.1 due to its internal contracting structure. This section presents an analytical framework for providing control theoretical collision avoidance and robust tracking guarantees to given learned motion planning policies for nonlinear multi-agent systems, independently of the performance of the learning approaches used in designing the learned policy. Although our approach can be extended further to handle general notions of safety, we focus on collision-free operation as the objective of safety. We call our approach CART (Collision Avoidance and Robust Tracking) [38] and its concept is to be summarized in the following.

### 7.5.I Augmenting Learned Policy with Safety and Robustness

Let us first recall that, as implied throughout this chapter, directly using the learned motion planning policy has the following two issues in practice: (i) even in nominal settings without external disturbance in (7.6) and (7.7), the system solution

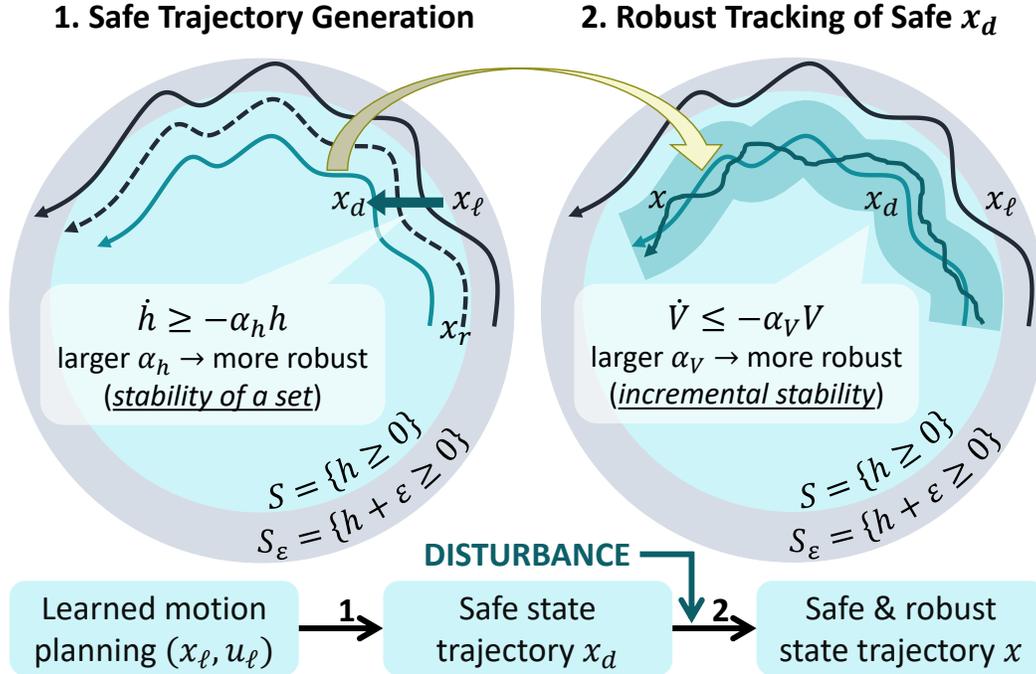


Figure 7.7: Conceptual illustration of CART, showing the hierarchical combination of our safety filter and robust filter, where  $\alpha_h, \alpha_V > 0$ ,  $h \geq 0$  is a safety function given,  $S$  is a safe set given,  $S_\epsilon$  is some fictitious set containing learned trajectory  $x_\ell$  with learning error  $\epsilon > 0$ ,  $x_d$  is a safe trajectory,  $x_r$  is a reference trajectory given by global motion planner (7.8),  $x$  is the actual state trajectory subject to disturbance, and  $V$  is an incremental Lyapunov function for safe and robust trajectory tracking. Note that we use a log-barrier formulation in CART for its distributed implementation and analytical solution, but this figure uses  $h$  for the simplicity of our concept description.

trajectories computed with the learned motion planning policy could violate safety requirements due to learning errors, and (ii) the learned policy lacks a formal mathematical guarantee of safety and robustness under the presence of external disturbance. Before going into details, let us see how we address these two problems analytically in real-time for the general systems (7.6) and (7.7), optimally and independently of the performance of the learning method used in the learned motion planning policy.

### 7.5.I-A Safety Filter and Built-in Robustness

Let  $S = \{x \in \mathbb{R}^{2n} | h(x) \geq 0\}$  be a set defining safety. Given the learned policy  $u_\ell^i$ , we can slightly modify it using a safety filter (Control Barrier Function, CBF) to ensure the agents' safe operation using a constraint  $\dot{h} \geq -\alpha_h h$ ,  $\alpha_h > 0$  [39] even

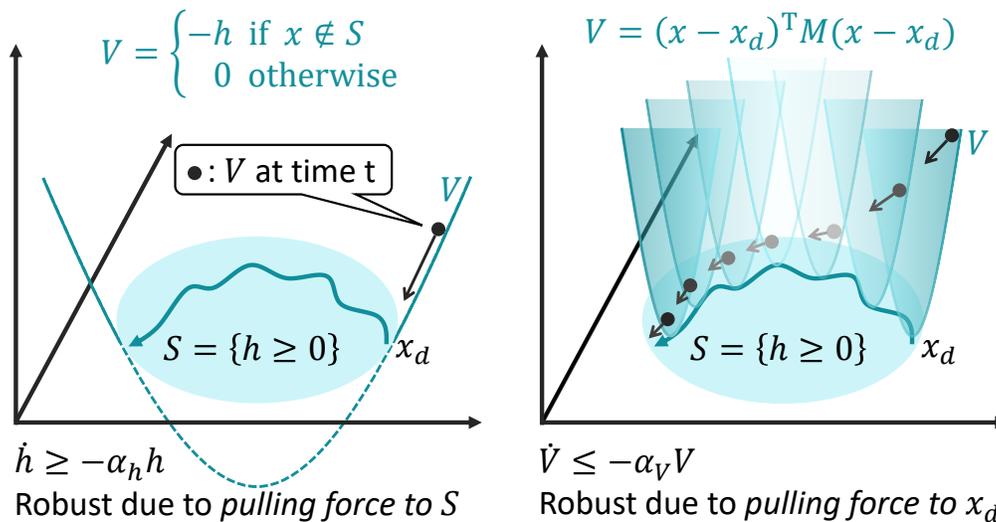


Figure 7.8: Different sources of robustness against external disturbance, where our safety filter is robust due to stability of a safe set and our robust filter is robust due to incremental stability of the closed-loop system with respect to a target trajectory.

with the presence of learning error  $\varepsilon > 0$  ( $-\alpha_h h$  can be  $-\alpha(h)$  for a class  $\mathcal{K}$  function  $\alpha$ ). Intuitively, since the learning error  $\varepsilon$  is expected to be small empirically, the contribution required for the safety filter is also expected to be small in practice, especially in the absence of external disturbance as depicted in the left-hand side of Fig. 7.7.

Also, such a safety filter is robust as can be shown using a Lyapunov function  $V = -h$  for  $x \notin S$  and  $V = 0$  for  $x \in S$  [40]. This implies that the safe set  $S$  is constructed to be exponentially (asymptotically for class  $\mathcal{K}$  functions) stable, where the robustness results from the pulling force to the safe set  $S$  that could be undesirably large leading to a large tracking error, e.g., in the real world scenario involving the discretization of the control and dynamics (see the left-hand side of Fig. 7.8).

### 7.5.I-B Robust Filter and Tracking-based Robustness

Instead of handling both safety and robustness just by the safety filter, we can further utilize a robust filter hierarchically to mitigate the burden of the safety filter in dealing with the disturbance, i.e., to robustly track the safe trajectory  $x_d$  slightly modified from the reference trajectory  $x_r$  using the safety filter, as depicted in the right-hand side of Fig. 7.7. We still use the Lyapunov formulation as in the safety filter for robustness, but now the Lyapunov function is defined incrementally as  $V = (x - x_d)^T M (x - x_d)$ , where  $M > 0$  is to be defined in the proceeding sections.

Improving the robustness performance here will simply result in closer tracking of the safe trajectory  $x_d$  (see the right-hand side of Fig. 7.8) without losing too much information of the learned motion planning policy  $u_\ell^i$ . This is mainly because the safety is handled indirectly with  $\dot{h} \geq -\alpha_h h$  in the safety filter and directly with  $h \geq 0$  in the robust filter.

These observations imply that

- (a) when the learning error is much larger than the size of the external disturbance, then we can use the safety filter and its build-in robustness, and
- (b) when the learning error is much smaller than the size of the external disturbance, which is often the case, then we can modify the learned policy slightly with the safety filter in a nominal setting, handling disturbance using the tracking-based robust filter on behalf of the safety filter.

Note that we use the log-barrier formulation from now on instead of  $\dot{h} \geq -\alpha_h h$ , which allows for the distributed implementation of our safety filter in a multi-agent setting.

### 7.5.I-C Relationship with Existing Methods

The nonlinear robustness and stability can also be analyzed using a Lyapunov function, which gives a finite tracking error with respect to a given target trajectory under the presence of external disturbances, including approximation errors of given learned motion planning policies. This property can be used further to establish a safety guarantee, by utilizing a conservative constraint that a tube around the computed target trajectory will not violate a given safety requirement [17], [18], [20], [41], [42]. This framework thus provides a one way to ensure safety and robustness as in the LAG-ROS framework with Lemma 7.1 (see also [43]–[45] and references therein), which depends on the knowledge and the size of the approximation error of a given learned motion planning policy. Such information could be conservative for previously unseen data or available only empirically (e.g., by using a Lipschitz bound [46]).

The CLF-CBF control [39], [40], [47] also considers safety and robustness in real-time without any knowledge of the learned motion planning errors at all [39], [40], [47] (see also [30], [48]–[50] for stochastic and higher-order systems and [51]–[54] for learning-based CBF methods), without solving nonlinear optimization as

in MPC-based methods for robustness and safety [41]. In our context, it constructs an optimal control input by solving a QP to minimize its deviation from the learned motion planning policy, subject to the safety constraint  $\dot{h} \geq -\alpha_h h$  and the relaxed incremental stability constraint  $\dot{V} \leq -\alpha_V V + \rho$ , where  $\rho$  is for QP feasibility and  $V$  is now defined as  $V = (x - x_\ell)^\top M(x - x_\ell)$  for the learned trajectory  $x_\ell$  of Fig. 7.7. We list key differences between the CLF-CBF controller and our approach, CART:

- (a) The primary distinction between CART and CLF-CBF is the direction in which the respective tracking component steers the system. The CLF-CBF stability component steers towards the learned trajectory, which, because of learning error, might not be safe. In contrast, CART's robust filter steers the system toward a certified safe trajectory, generated by the previous layer in the hierarchy, the safety filter. Because of this distinction, whereas CART can safely reject large disturbances with a large safety gain  $\alpha_V$ , this strategy is not practical for the CLF-CBF controller, which is forced to reject disturbances with large safety gain  $\alpha_h$ , which pulls the system overconservatively towards the interior of the safety set.
- (b) The secondary distinction between CART and CLF-CBF is that CLF-CBF requires solving a QP with a given Lyapunov function, while CART provides an explicit way to construct the incremental Lyapunov function using contraction theory, and gives an analytical solution for the optimal control input. This makes CART end-to-end trainable with a faster computation evaluation time.

CART proposes a hierarchical approach to combine the best of both of these methods for safety and robustness, by performing contraction theory-based robust tracking of a provably collision-free trajectory, generated by a safety filter with the learned policy. The additional tracking-based robust filter is for reducing the burden of the safety filter in dealing with disturbances. CART can be viewed also as a generalization of the frameworks [2], [55] to multi-agent Lagrangian and general control-affine nonlinear systems with deterministic and stochastic disturbance, constructed on top of the end-to-end learned motion planning policy augmented with the optimality guarantee of the analytical solution to our safety and robust filters.

The trade-off of Sec. 7.5.I-B and the strengths implied in Sec. 7.5.I-C will be demonstrated in Chapter 10.

## 7.6 Theoretical Overview of CART

We specifically consider multi-agent settings in this section. In particular, we consider the following multi-agent Lagrangian dynamical system, perturbed by deterministic disturbance  $d^i(x, t)$  with  $\sup_{x,t} \|d^i(x, t)\| = \bar{d}^i \in [0, \infty)$  and Gaussian white noise driven by a Wiener process  $\mathcal{W}^i(t)$  with  $\sup_{x,t} \|\Gamma^i(x, t)\|_F = \bar{\gamma}^i \in [0, \infty)$ :

$$\begin{aligned} & M^i(p^i)dv^i + (C^i(p^i, v^i)v^i + G^i(p^i) + D^i(p^i, v^i))dt \\ & = (u^i + d^i(x^i, t))dt + \Gamma^i(x^i, t)d\mathcal{W}^i(t), \quad i = 1, \dots, N \end{aligned} \quad (7.6)$$

where  $t \in \mathbb{R}_+$ ,  $x^i = [p^{i\top}, v^{i\top}]^\top$ ,  $i$  is the index of the  $i$ th agent,  $p^i : \mathbb{R}_+ \mapsto \mathbb{R}^n$  and  $v^i : \mathbb{R}_+ \mapsto \mathbb{R}^n$  are the generalized position and velocity of the  $i$ th agent ( $\dot{p}^i = v^i$ ),  $u^i \in \mathbb{R}^m$  ( $m = n$  in this case) is the system control input,  $M^i$ ,  $C^i$ ,  $G^i$ , and  $D^i$  are known smooth functions that define the Lagrangian system,  $d^i$  and  $\Gamma^i$  are unknown bounded functions for external disturbances,  $\mathcal{W}^i$  is a  $w$ -dimensional Wiener process, and we consider the case where  $\bar{d}^i, \bar{\gamma}^i \in [0, \infty)$  are given. Note that we have  $M^i(p^i) > 0$  and that the matrix  $C^i(p^i, v^i)$  is selected to make  $\dot{M}^i(p^i) - 2C^i(p^i, v^i)$  skew-symmetric, so we have a useful property s.t.  $z^\top (\dot{M}^i(p^i) - 2C^i(p^i, v^i))z = 0, \forall z \in \mathbb{R}^n$  [56, p. 392]. We also consider the following general control-affine nonlinear system:

$$dv^i = (f^i(p^i, v^i, t) + B(p^i, v^i, t)u^i)dt + d^i(x^i, t)dt + \Gamma^i(x^i, t)d\mathcal{W}^i(t), \quad i = 1, \dots, N \quad (7.7)$$

where  $f^i$ , and  $B^i$  are known smooth functions,  $u^i \in \mathbb{R}^m$  is the system control input, and the other notations are consistent with the ones of (7.6). We assume the existence and uniqueness conditions of the solutions of (7.6) and (7.7) as in [57, p. 105].

The nonlinear motion planning problem of our interest is defined as follows:

$$\begin{aligned} u_r(t) &= \arg \min_{\{u^i(t) \in \mathbb{R}^m\}_{i=1}^N} \int_0^{t_f} c(x(\tau), u(\tau), \tau) d\tau \\ &\text{s.t. (7.6) or (7.7) with } d^i = 0 \text{ and } \Gamma^i = 0, \quad x^i(t_f) = x_f^i, \quad x^i(0) = x_0^i \\ &\quad \|p^i(t) - p^j(t)\| \geq r_s, \quad \forall t, \forall i, j \neq i \end{aligned} \quad (7.8)$$

where  $x(t) = \{x^i(t)\}_{i=1}^N$ ,  $u(t) = \{u^i(t)\}_{i=1}^N$ ,  $c(x(t), u(t), t)$  is a user-specified cost at time  $t$ ,  $t_f$  is the terminal time,  $x_0^i$  and  $x_f^i$  are the initial and terminal state, respectively,  $r_s$  is the minimal safe distance between  $i$ th agent and other objects, and  $j$  is the index denoting other agents and obstacles.

### 7.6.I Learned Distributed Motion Planning Policy

Let  $\mathcal{N}$  denote the set of all the  $N$  agents,  $\mathcal{M}$  denote the set of all the  $M$  static obstacles, and  $o^i$  denote the local observation of the  $i$ th agent given as follows:

$$o^i = (x^i, \{x^j\}_{j \in \mathcal{N}^i}, \{p^j\}_{j \in \mathcal{M}^i}) \quad (7.9)$$

where  $x^i$  is the state of the  $i$ th agent,  $\{x^j\}_{j \in \mathcal{N}^i}$  are the states of neighboring agents defined with  $\mathcal{N}^i = \{j \in \mathcal{N} \mid \|p^i - p^j\| \leq r_{\text{sen}}\}$ ,  $\{p^j\}_{j \in \mathcal{M}^i}$  are the positions of neighboring static obstacles defined with  $\mathcal{M}^i = \{j \in \mathcal{M} \mid \|p^i - p^j\| \leq r_{\text{sen}}\}$ , and  $r_{\text{sen}}$  is the sensing radius.

In this section, we assume that we have access to a learned distributed motion planning policy  $u_\ell^i(o^i)$  obtained by [2]. In particular, (i) we generate demonstration trajectories by solving the global nonlinear motion planning (7.8) and (ii) extract local observations from them for deep imitation learning to construct  $u_\ell^i(o^i)$ . Our differentiable safety and robust filters to be seen in Sec. 7.6.II and 7.6.III can be used also in this phase to allow for end-to-end policy training as in [2].

### 7.6.II Analytical Form of Optimal Safety Filter

We are now ready to present an analytical way to design a safety filter that guarantees safe operations of the systems (7.6) and (7.7) when  $d^i = 0$  and  $\Gamma^i = 0$ . One of the benefits of the log-barrier formulation in the following is that the global safety violation can be decomposed as the sum of the local safety violations, allowing for the distributed implementation of our analytical safety filter in a multi-agent setting:

$$\psi^i(o^i) = -\log \prod_{j \in \mathcal{S}^i} h(p^{ij}), \quad h(p^{ij}) = \frac{\|p_{ij}\|_{\Xi} - (r_s + \Delta r_s)}{r_{\text{sen}} - (r_s + \Delta r_s)} \quad (7.10)$$

where  $p^{ij} = p^j - p^i$ ,  $\mathcal{S}^i$  is the set of all the neighboring objects (i.e., agents and obstacles),  $r_s$  is the minimal safe distance between  $i$ th agent and other objects,  $r_{\text{sen}}$  is the sensing radius, and  $\Delta r_s > 0$  is a positive scalar parameter to account for the external disturbance to be formally defined in Sec. 7.6.III. We use the weighted 2-norm  $\|\cdot\|_{\Xi}$  with the weight  $\Xi > 0$  to consider a collision boundary defined by an ellipsoid. The parameters are selected to satisfy  $\sup_{\|p_{ij}\| \leq r_{\text{sen}}} \|p_{ij}\|_{\Xi} \leq r_{\text{sen}}$  and  $r_{\text{sen}} - (r_s + \Delta r_s) > 0$  to ensure  $\psi^i(o^i) \geq 0$  always when the distance to the  $j$ th object is less than  $r_{\text{sen}}$ . Having a negative value of  $h$  implies a safety violation. The idea for our safety filter is first to construct a safe target velocity given as

$$v_s^i(o^i) = -k_p^i \nabla_{p^i} \psi^i(o^i) \quad (7.11)$$

where  $k_p^i > 0$ , and then to realize it optimally using the knowledge of Lagrangian systems and contraction theory.

When dealing with general safety  $h(x^i, x^j) \geq 0$  for each  $i$  and  $j$ , we can also use the local safety function (7.10) modified as  $\psi^i(o^i) = -\log \prod_{j \in \mathcal{S}^i} h(x^i, x^j)$ . We focus on collision-free operation as the objective of safety in this paper as it is one of the most critical safety requirements in a multi-agent setting.

**Remark 7.2.** *The velocity (7.11) renders the single integrator system (i.e.,  $\dot{p}^i = v^i$ ) safe [2]. Note that instead of the condition  $\dot{\psi} \leq 0$  in [2], [55], we could use  $\dot{\psi} \leq \alpha(h)$  to increase the available set of control inputs [39], where  $\psi$  is a barrier function,  $h$  is a safety function associated with  $\psi$ , and  $\alpha$  is a class  $\mathcal{K}$  function [58, p. 144]. This requires an additional global Lipschitz assumption on  $\alpha$  as seen in [47].*

## 7.6.II-A Optimal Safety Filter for Lagrangian Systems

Given the learned motion planning policy  $u_\ell^i(o^i)$  of Sec. 7.6.I for the system (7.6), we design a control policy  $u_s^i$  processed by our proposed safety filter as follows:

$$u_s^i(o^i) = u_\ell^i(o^i) - \begin{cases} \frac{e_v^i (u_\ell^i(o^i) - \bar{u}_s^i)^\top e_v^i}{\|e_v^i\|^2} & \text{if } (u_\ell^i(o^i) - \bar{u}_s^i)^\top e_v^i > 0 \\ 0 & \text{otherwise} \end{cases} \quad (7.12)$$

where  $e_v^i = v^i - v_s^i(o^i)$  for  $v_s^i(o^i)$  of (7.11),  $\bar{u}_s^i$  is given as

$$\bar{u}_s^i = M^i \dot{v}_s^i + C^i v_s^i + G^i + D^i + v_s^i - k_v^i M^i e_v^i \quad (7.13)$$

with its arguments omitted and  $k_p^i, k_v^i > 0$  being design parameters. The controller (7.12) is well-defined even with the division by  $\|e_v^i\|$  as the relation  $(u_\ell^i(o^i) - \bar{u}_s^i)^\top e_v^i = 0 \leq 0$  holds when  $\|e_v^i\| = 0$ . We have the following for the safety guarantee.

**Theorem 7.2.** *Consider the following optimization problem:*

$$u_{\text{opt}}^i = \arg \min_{u^i \in \mathbb{R}^m} \|u^i - u_\ell^i(o^i)\|^2 \text{ s.t. } (u^i - \bar{u}_s^i)^\top e_v^i \leq 0. \quad (7.14)$$

where  $\bar{u}_s^i$  is given by (7.13). Suppose that there exists a control input  $u^i$  that satisfies the constraint of (7.14) for each  $i$ . The safety of the system (7.6) is then guaranteed when  $d^i = 0$  and  $\Gamma^i = 0$ , i.e., all the agents will not collide with the other objects when there is no external disturbance.

Also, the problem (7.14) is always feasible and its optimal solution is given by (7.12), thereby minimizing the deviation of the safe control input from  $u_\ell^i(o^i)$ .

*Proof.* Let us consider the following Lyapunov-type function:

$$\mathcal{V} = \psi(X) + \sum_{i=1}^N \frac{\|v^i - v_s^i\|_{M^i(p^i)}^2}{2} = \psi(X) + \sum_{i=1}^N \frac{\|e_v^i\|_{M^i(p^i)}^2}{2}$$

where  $v_s^i$  is given in (7.12),  $e_v^i = v^i - v_s^i$ , and  $\psi$  is given as

$$\psi(X) = -\log \prod_{i=1}^N \prod_{j \in \mathcal{N}^i} h(p^{ij}) \prod_{i=1}^N \prod_{j \in \mathcal{M}^i} h(p^{ij}) \quad (7.15)$$

for  $X = \{x^1, \dots, x^N\}$ . By the definition of  $h$  in (7.10), the collision-free operation of the system (7.6) is guaranteed as long as  $\psi^i$  of (7.10) is bounded. Taking the time derivative of  $\mathcal{V}$ , we get

$$\dot{\mathcal{V}} = \sum_{i=1}^N k_p^i \nabla_{p^i} \psi^{i\top} v_s^i - k_v^i \|e_v^i\|_{M^i}^2 + (u^i - \bar{u}_s^i)^\top e_v^i \quad (7.16)$$

by using (7.6) for  $d^i = 0$  and  $\Gamma^i = 0$  along with the relation  $z^\top (\dot{M}^i - 2C^i)z = 0$ ,  $\forall z \in \mathbb{R}^n$ , where the arguments are omitted. Having  $(u^i - \bar{u}_s^i)^\top e_v^i \leq 0$  as in the constraint of (7.14) gives  $\dot{\mathcal{V}} \leq \sum_{i=1}^N -k_p^i \|\nabla_{p^i} \psi^i\|^2 - k_v^i \|e_v^i\|_{M^i}^2$ , which guarantees the boundedness of  $V$  and then  $\psi^i$  for all  $i$ , implying no safety violation as long as the system is initially safe. Also, the constraint  $(u^i - \bar{u}_s^i)^\top e_v^i \leq 0$  is always feasible for  $u^i$  given as  $u^i = \bar{u}_s^i$ . Finally, applying the KKT condition [59, pp. 243-244] to (7.14) results in  $u_{\text{opt}}^i = u_s^i(o^i)$  for  $u_s^i(o^i)$  given in (7.12).  $\square$

## 7.6.II-B Optimal Safety Filter for General Nonlinear Systems

The results of Theorem 7.2 can be generalized to obtain a safety filter nonlinear systems given by (7.7). We use the following incremental Lyapunov function as in contraction theory [43], [60], which leads to a safety analysis analogous to that of LTV systems and Lagrangian systems (7.6):

$$\mathcal{E}^i(o^i, t) = \frac{\|v^i - v_s^i(o^i)\|_{M^i(x^i, t)}^2}{2} \quad (7.17)$$

$$\text{s.t. } \dot{M}^i + M^i A_s^i + A_s^{i\top} M^i - 2M^i B^i R^{i-1} B^{i\top} M^i \leq -k_v^i M^i \quad (7.18)$$

where  $M^i(x^i, t) > 0$ ,  $k_v^i$  is a design parameter, and  $A_s^i(o^i, t)$  is a nonlinear state-dependent coefficient matrix for the system of (7.7) defined as follows:

$$A_s^i(o^i, t)(v^i - v_s^i(o^i)) = f^i(p^i, v^i, t) - f^i(p^i, v_s^i(o^i), t).$$

The arguments are omitted in (7.18) and the notations (7.17) and (7.18) are intentionally consistent with the ones of Theorem 7.2 to imply the analogy between the methods in Sec. 7.6.II-A and Sec. 7.6.II-B. Note that the nonlinear matrix  $A_s$  always exists when  $f^i$  is continuously differentiable as proved in Lemma 3.1 [10]. Although we use one of the simple versions of a contraction metric in this paper for simplicity of discussion, we can consider more general types of Lyapunov functions and contraction metrics for the sake of the broader applicability of our approach by using, e.g., path integrals, geodesics, virtual systems, and differential dynamics, as discussed earlier in this thesis.

Let us also introduce the following assumption for generalizing the result of Theorem 7.2.

**Assumption 7.1.** *Consider a Lyapunov-type function for (7.7) defined as follows:*

$$\mathcal{V}^i(o^i, t) = k_p^i \psi^i(o^i) + \frac{\|v^i - v_s^i(o^i)\|_{M^i(x^i, t)}^2}{2}$$

for  $k_p^i > 0$ ,  $\psi^i(o^i)$  of (7.10),  $v_s^i(o^i)$  of (7.11), and  $M^i(x^i, t)$  of (7.18). We assume that

$$e_v^i = B(p^i, v^i, t)^\top \nabla_{v^i} \mathcal{V}^i = B(p^i, v^i, t)^\top M^i(v^i - v_s^i) = 0 \quad (7.19)$$

$$\Rightarrow \dot{\mathcal{V}}^i \leq -k_p^i \|\nabla_{p^i} \psi^i(o^i)\|^2 - k_v^i \mathcal{E}^i(o^i, t) \quad (7.20)$$

where  $e_v^i = v^i - v_s^i$  for  $v_s^i$  of (7.11),  $\psi^i(o^i)$  is given in (7.10),  $\alpha$  is given in (7.11),  $\mathcal{E}^i(o^i, t)$  is given in (7.17), and the arguments of  $\mathcal{V}$  and  $M^i$  are omitted for notational simplicity.

This assumption simply says that the system naturally satisfies the safety condition (7.20) when the velocity displacements are in the directions orthogonal to the span of the actuated directions as discussed in [23].

**Remark 7.3.** *Assumption 7.1 always holds for fully and over-actuated systems s.t.  $B(p^i, v^i, t)B(p^i, v^i, t)^\dagger = I_{n \times n}$ , where  $B(p^i, v^i, t)^\dagger$  the Moore-Penrose pseudo inverse. Furthermore, even when the system is under-actuated and Assumption 7.1 does not hold, the error associated with the under-actuation can be treated robustly as to be seen in Sec. 7.6.III.*

Given the learned motion planning policy  $u_\ell^i(o^i)$  of Sec. 7.6.I for the system (7.7), we design a control policy  $u_s^i$  processed by our proposed safety filter as (7.12), where  $e_v^i$  is now given by (7.19) and  $\bar{u}_s^i$  is defined as

$$\bar{u}_s^i = \begin{cases} \frac{e_v^i (e_v^i)^\top M^i (\dot{v}_s^i - f_s^i) - k_p^i e_v^i \nabla_{p^i} \psi^i}{\|e_v^i\|^2} & \text{if } \|e_v^i\| \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (7.21)$$

with  $f_s^i = f^i(p^i, v_s^i, t)$  and  $k_p^i$  being a design parameter. The controller (7.12) is well-defined with the division by  $\|e_v^i\|$  under Assumption 7.1, because the relation  $(u_\ell^i(o^i) - \bar{u}_s^i)^\top e_v^i = 0 \leq 0$  holds when  $\|e_v^i\| = 0$ .

**Theorem 7.3.** *Consider the optimization problem (7.14), where  $u_\ell^i$  is now given by the learned motion planning policy for (7.8) with (7.7),  $e_v^i$  is by (7.19),  $u_\ell^i$  is by (7.21). Suppose that Assumption 7.1 holds and that there exists a control input  $u^i$  that satisfies the constraint of (7.14) for each  $i$ . The safety of the system (7.7) is then guaranteed when  $d^i = 0$  and  $\Gamma^i = 0$ , i.e., all the agents will not collide with the other objects when there is no external disturbance.*

*Also, the problem (7.14) is always feasible and its optimal solution is given by (7.12), thereby minimizing the deviation of the safe control input from  $u_\ell^i(o^i)$ .*

*Proof.* Let us consider a Lyapunov-type function  $\mathcal{V} = \psi(X) + \sum_{i=1}^N \mathcal{E}^i(o^i, t)$ , where  $\psi$  is given in (7.15),  $v_s^i$  is given in (7.12), and  $\mathcal{E}^i(o^i, t)$  is given in (7.17). Using the relation (7.18) and Assumption 7.1, we have  $\dot{\mathcal{V}} \leq \sum_{i=1}^N -k_p^i \|\nabla_{p^i} \psi^i\|^2 - k_v^i \mathcal{E}^i(o^i, t) + (u^i - \bar{u}_s^i)^\top e_v^i$  when  $d^i = 0$  and  $\Gamma^i = 0$  in (7.7) as in the proof of Theorem 7.2, where  $\psi^i(o^i)$  is given in (7.10),  $\alpha$  is given in (7.11), and  $\mathcal{E}^i(o^i, t)$  is given in (7.17). The rest follows from the proof of Theorem 7.2 below (7.16).  $\square$

**Remark 7.4.** *The safety filter of Theorem 7.3 minimizes the deviation of the safe control input from the learned motion planning input of the general system (7.7), which implies that it instantaneously minimizes the contribution of the under-actuation error when Assumption 7.1 does not hold. This error can be then treated robustly as discussed in Remark 7.3.*

### 7.6.III Augmenting Robustness

The results in Sec. 7.6.II depend on the assumption that  $d^i = 0$  and  $\Gamma^i = 0$  in (7.6) and (7.7). This section discusses the safety of these systems in the presence of external disturbance.

### 7.6.III-A Revisiting Built-in Robustness of Safety Filter

Due to its Lyapunov-type formulation in Theorems 7.2 and 7.3, our safety filter inherits the robustness properties discussed in, e.g., [58]. It also inherits the robustness of the barrier function of [39], [40], [47] ([50], [61] for stochastic disturbance) as seen in Sec. 7.6.I with Fig. 7.7 and 7.8.

The robust filter of CART is for hierarchically augmenting such built-in robustness with the tracking-based robustness to lighten the burden of the safety filter in dealing with the disturbance. Given a safety condition  $h \geq 0$ , these two ways of augmenting the learned motion planning with robustness as in Fig. 7.7 are achieved by

- (a) changing our safety filter parameters (e.g., making  $k_p$  and  $k_v$  larger in Theorems 7.2 and 7.3),
- (b) tracking a safe trajectory that satisfies  $h \geq 0$ , ensuring the deviation from the perturbed trajectory is finite.

The first approach (a) could lead to a large repelling force due to the stability of the safe set originating from the use of  $\dot{h}$ , especially when we use the log-barrier formulation with the dynamics discretization (i.e., we get a larger control input as the agents get closer to the safety boundary, implying a large discretization error). In contrast, (b) does not involve such behavior as it simply attempts to track the safe trajectory satisfying  $h \geq 0$ . As illustrated in Fig. 7.8, there are the following two sources of robustness in our approach:

- (a) asymptotic/exponential stability of the safe set,
- (b) incremental asymptotic/exponential stability of the system trajectory with respect to a safe target trajectory,

and the robust filter of CART is about (b), which significantly reduces the responsibility of the safety filter (a) in meeting the robustness requirement, allowing the safe set to be less stable (i.e., the unsafe set to be less repelling, meaning more freedom in choosing the safety filter parameters). Note that the robust filter here is essentially the tracking control policy introduced in Chapter 3, Chapter 4, and Chapter 6, where the target trajectory is obtained hierarchically by the safety filter of Theorems 7.2 and 7.3 as in fig. 7.7 (see [38] for more details). These observations will be more appreciable in the numerical simulations in Chapter 10.

## References

- [1] H. Tsukamoto and S.-J. Chung, “Learning-based robust motion planning with guaranteed stability: A contraction theory approach,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6164–6171, 2021.
- [2] B. Rivière, W. Hönig, Y. Yue, and S.-J. Chung, “GLAS: Global-to-local safe autonomy synthesis for multi-robot motion planning with end-to-end learning,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4249–4256, 2020.
- [3] M. Everett, Y. F. Chen, and J. P. How, “Motion planning among dynamic, decision-making agents with deep reinforcement learning,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2018, pp. 3052–3059.
- [4] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, “Safe model-based reinforcement learning with stability guarantees,” in *Adv. Neural Inf. Process. Syst.*, vol. 30, Curran Associates, Inc., 2017, pp. 908–918.
- [5] J. Carius, F. Farshidian, and M. Hutter, “MPC-Net: A First Principles Guided Policy Search,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2897–2904, 2020.
- [6] J. Ho and S. Ermon, “Generative adversarial imitation learning,” in *Adv. Neural Inf. Process. Syst.*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, Curran Associates, Inc., 2016, pp. 4565–4573.
- [7] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, “Social GAN: Socially acceptable trajectories with generative adversarial networks,” in *IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 2255–2264.
- [8] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, “Imitating driver behavior with generative adversarial networks,” in *IEEE Intell. Vehicles Symp.*, 2017, pp. 204–211.
- [9] S. Singh, A. Majumdar, J.-J. E. Slotine, and M. Pavone, “Robust online motion planning via contraction theory and convex optimization,” in *IEEE Int. Conf. Robot. Automat.*, May 2017, pp. 5883–5890.
- [10] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Neural stochastic contraction metrics for learning-based control and estimation,” *IEEE Control Syst. Lett.*, vol. 5, no. 5, pp. 1825–1830, 2021.
- [11] D. Sun, S. Jha, and C. Fan, *Learning certified control using contraction metric*, arXiv:2011.12569, Nov. 2020.
- [12] B. T. Lopez, J.-J. E. Slotine, and J. P. How, “Dynamic tube MPC for nonlinear systems,” in *Proc. Amer. Control Conf.*, 2019, pp. 1655–1662.
- [13] A. Lakshmanan, A. Gahlawat, and N. Hovakimyan, “Safe feedback motion planning: A contraction theory and  $\mathcal{L}_1$ -adaptive control based approach,” in *IEEE Conf. Decis. Control*, 2020, pp. 1578–1583.

- [14] A. Gahlawat, A. Lakshmanan, L. Song, *et al.*, “Contraction  $\mathcal{L}_1$ -adaptive control using Gaussian processes,” in *Proc. Conf. Learn. Dyn. Control*, ser. Proc. Mach. Learn. Res. Vol. 144, Jun. 2021, pp. 1027–1040.
- [15] D. Sun, M. J. Khojasteh, S. Shekhar, and C. Fan, *Uncertainty-aware safe exploratory planning using Gaussian process and neural control contraction metric*, arXiv:2105.06567, 2021.
- [16] H. Tsukamoto and S.-J. Chung, “Convex optimization-based controller design for stochastic nonlinear systems using contraction analysis,” in *IEEE Conf. Decis. Control*, Dec. 2019, pp. 8196–8203.
- [17] P. Zhao, A. Lakshmanan, K. Ackerman, A. Gahlawat, M. Pavone, and N. Hovakimyan, *Tube-certified trajectory tracking for nonlinear systems with robust control contraction metrics*, arXiv:2109.04453, Sep. 2021.
- [18] A. Bemporad and M. Morari, “Robust model predictive control: A survey,” in *Robustness in identification and control*, London: Springer London, 1999, pp. 207–226.
- [19] D. Mayne, M. Seron, and S. Raković, “Robust model predictive control of constrained linear systems with bounded disturbances,” *Automatica*, vol. 41, no. 2, pp. 219–224, 2005, ISSN: 0005-1098.
- [20] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi, “Tube-based robust nonlinear model predictive control,” *Int. J. Robust Nonlinear Control*, vol. 21, no. 11, pp. 1341–1353, 2011.
- [21] J. Schulman, Y. Duan, J. Ho, *et al.*, “Motion planning with sequential convex optimization and convex collision checking,” *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1251–1270, 2014.
- [22] Y. K. Nakka, A. Liu, G. Shi, A. Anandkumar, Y. Yue, and S.-J. Chung, “Chance-constrained trajectory optimization for safe exploration and learning of nonlinear systems,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 389–396, 2021.
- [23] I. R. Manchester and J.-J. E. Slotine, “Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design,” *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 3046–3053, Jun. 2017.
- [24] H. Tsukamoto and S.-J. Chung, “Robust controller design for stochastic nonlinear systems via convex optimization,” *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4731–4746, 2021.
- [25] H. Tsukamoto and S.-J. Chung, “Neural contraction metrics for robust estimation and control: A convex optimization approach,” *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 211–216, 2021.
- [26] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Learning-based Adaptive Control using Contraction Theory,” in *IEEE Conf. Decis. Control*, 2021, pp. 2533–2538.

- [27] A. G. Barto, R. S. Sutton, and C. W. Anderson, “Neuronlike adaptive elements that can solve difficult learning control problems,” *IEEE Trans. Syst. Man Cybern.*, vol. SMC-13, no. 5, pp. 834–846, 1983.
- [28] G. Shi, X. Shi, M. O’Connell, *et al.*, “Neural lander: Stable drone landing control using learned dynamics,” in *IEEE Int. Conf. Robot. Automat.*, May 2019, pp. 9784–9790.
- [29] Y. K. Nakka, W. Hönig, C. Choi, A. Harvard, A. Rahmani, and S.-J. Chung, “Information-based guidance and control architecture for multi-spacecraft on-orbit inspection,” in *AIAA Scitech 2021 Forum*, 2021.
- [30] S. Kolathaya and A. D. Ames, “Input-to-state safety with control barrier functions,” *IEEE Control Syst. Lett.*, vol. 3, no. 1, pp. 108–113, 2019.
- [31] A. D. Ames, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs with application to adaptive cruise control,” in *IEEE Conf. Decis. Control*, 2014, pp. 6271–6278.
- [32] B. T. Lopez and J.-J. E. Slotine, “Universal adaptive control of nonlinear systems,” *IEEE Control Syst. Lett.*, vol. 6, pp. 1826–1830, 2022.
- [33] B. T. Lopez and J.-J. E. Slotine, “Adaptive nonlinear control with contraction metrics,” *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 205–210, 2021.
- [34] S. Singh, S. M. Richards, V. Sindhvani, J.-J. E. Slotine, and M. Pavone, “Learning stabilizable nonlinear dynamics with contraction-based regularization,” *Int. J. Robot. Res.*, Aug. 2020.
- [35] N. M. Boffi, S. Tu, N. Matni, J.-J. E. Slotine, and V. Sindhvani, *Learning stability certificates from data*, arXiv:2008.05952, 2020.
- [36] I. R. Manchester, M. Revay, and R. Wang, “Contraction-based methods for stable identification and robust machine learning: A tutorial,” in *IEEE Conf. Decis. Control*, 2021, pp. 2955–2962.
- [37] C. Blocher, M. Saveriano, and D. Lee, “Learning stable dynamical systems using contraction theory,” in *IEEE URAI*, 2017, pp. 124–129.
- [38] H. Tsukamoto, B. Rivière, C. Choi, A. Rahmani, and S.-J. Chung, “CART: Collision avoidance and robust tracking augmentation in learning-based motion planning for multi-agent systems,” in *IEEE Conf. Decis. Control*, *under review*, 2023.
- [39] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, 2017.
- [40] 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, ISSN: 2405-8963.

- [41] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, “Learning-based model predictive control: Toward safe learning in control,” *Annu. Rev. Control Robot. Auton. Syst.*, vol. 3, no. 1, pp. 269–296, 2020.
- [42] S. Singh, H. Tsukamoto, B. T. Lopez, S.-J. Chung, and J.-J. E. Slotine, “Safe motion planning with tubes and contraction metrics,” in *IEEE Conf. Decis. Control*, 2021, pp. 2943–2948.
- [43] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview,” *Annu. Rev. Control*, vol. 52, pp. 135–169, 2021, ISSN: 1367-5788.
- [44] 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 *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2019, pp. 6878–6884.
- [45] 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 Trans. Robot.*, pp. 1–19, 2023.
- [46] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” in *Int. Conf. Learn. Representations*, 2018.
- [47] A. J. Taylor, P. Ong, T. G. Molnar, and A. D. Ames, “Safe backstepping with control barrier functions,” in *IEEE Conf. Decis. Control*, 2022, pp. 5775–5782.
- [48] X. Tan, W. S. Cortez, and D. V. Dimarogonas, “High-order barrier functions: Robustness, safety, and performance-critical control,” *IEEE Trans. Autom. Control*, vol. 67, no. 6, pp. 3021–3028, 2022.
- [49] S. Prajna, A. Jadbabaie, and G. Pappas, “Stochastic safety verification using barrier certificates,” in *IEEE Conf. Decis. Control*, vol. 1, 2004, pp. 929–934.
- [50] A. Clark, “Control barrier functions for stochastic systems,” *Automatica*, vol. 130, p. 109 688, 2021, ISSN: 0005-1098.
- [51] A. Taylor, A. Singletary, Y. Yue, and A. Ames, “Learning for safety-critical control with control barrier functions,” in *LADC*, 2020.
- [52] R. K. Cosner, Y. Yue, and A. D. Ames, “End-to-end imitation learning with safety guarantees using control barrier functions,” in *IEEE Conf. Decis. Control*, 2022, pp. 5316–5322.
- [53] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, “Learning safe multi-agent control with decentralized neural barrier certificates,” in *Int. Conf. Learn. Representations*, 2021.

- [54] C. Dawson, Z. Qin, S. Gao, and C. Fan, “Safe nonlinear control using robust neural Lyapunov-barrier functions,” in *Proc. Conf. Robot Learn.*, ser. Proc. Mach. Learn. Res. Vol. 164, Nov. 2022, pp. 1724–1735.
- [55] P. F. Hokayem, D. M. Stipanović, and M. W. Spong, “Coordination and collision avoidance for Lagrangian systems with disturbances,” *Appl. Math. Comput.*, vol. 217, no. 3, pp. 1085–1094, 2010, ISSN: 0096-3003.
- [56] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Upper Saddle River, NJ: Pearson, 1991.
- [57] L. Arnold, *Stochastic Differential Equations: Theory and Applications*. Wiley, 1974.
- [58] H. K. Khalil, *Nonlinear Systems*, 3rd. Upper Saddle River, NJ: Prentice-Hall, 2002.
- [59] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Mar. 2004.
- [60] W. Lohmiller and J.-J. E. Slotine, “On contraction analysis for nonlinear systems,” *Automatica*, vol. 34, no. 6, pp. 683–696, 1998, ISSN: 0005-1098.
- [61] H. J. Kushner, *Stochastic Stability and Control*, English. Academic Press New York, 1967, xiv, 161 p.

## LEARNING-BASED ADAPTIVE CONTROL AND CONTRACTION THEORY FOR LEARNED MODELS

- [1] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Learning-based Adaptive Control using Contraction Theory,” in *IEEE Conf. Decis. Control*, 2021, pp. 2533–2538.
- [2] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview,” *Annu. Rev. Control*, vol. 52, pp. 135–169, 2021, ISSN: 1367-5788.

All the results presented so far assume that we have sufficient information for a nominal problem setting and thus our nominal model is sufficiently good for designing control, estimation, or motion planning policies. However, future aerospace and robotic exploration missions require that autonomous agents perform complex control tasks in challenging unknown environments while ensuring stability, optimality, and safety even for poorly-modeled dynamical systems. In this chapter, we consider the situations where the model and environmental uncertainties are too large to be treated robustly as external disturbances, and present real-time implementable control methods that extensively utilize online and offline data for their enhanced performance certificates and autonomous capabilities. Note that these methods can be used on top of the robust control approaches in the previous chapters.

### 8.1 Learning-based Adaptive Control

In this section, we consider the following smooth nonlinear system with an uncertain parameter  $\theta \in \mathbb{R}^c$ :

$$\dot{x} = f(x, \theta) + B(x, \theta)u, \quad \dot{x}_d = f(x_d, \theta) + B(x_d, \theta)u_d \quad (8.1)$$

where  $x$ ,  $u$ ,  $x_d$ , and  $u_d$  are as defined in (6.1) and (6.3), and  $f : \mathbb{R}^n \times \mathbb{R}^c \mapsto \mathbb{R}^n$  and  $B : \mathbb{R}^n \times \mathbb{R}^c \mapsto \mathbb{R}^{n \times m}$  are known smooth functions with the uncertain parameter  $\theta \in \mathbb{R}^c$ . Due to Theorems 2.4 and 5.2, we can see that the robust control techniques presented earlier in Theorems 4.2, 4.6, 6.4, and 6.5 are still useful in this case if the modeling errors  $\|f(x, \theta) - f(x, \theta_n)\|$  and  $\|B(x, \theta) - B(x, \theta_n)\|$  are bounded, where  $\theta_n$  is a nominal guess of the true parameter  $\theta$ . However, there are situations where such assumptions are not necessarily true.

We present a method of deep learning-based adaptive control for nonlinear systems with parametric uncertainty, thereby further improving the real-time performance of robust control in Theorems 5.2 and 5.3 for model-based systems, and Theorems 8.3 and 8.4 for model-free systems. Although we consider continuous-time dynamical systems in this chapter, discrete-time changes could be incorporated in this framework using [1], [2]. Also, note that the techniques in this chapter [3] can be used with differential state feedback frameworks [4]–[9] as described in Theorem 4.6, trading off added computational cost for generality (see Table 3.1 and [10], [11]).

### 8.1.I Adaptive Control with CV-STEM and NCM

Let us start with the following simple case.

**Assumption 8.1.** *The matrix  $B$  in (8.1) does not depend on  $\theta$ , and  $\exists \Pi(x) \in \mathbb{R}^{c \times n}$  s.t.  $\Pi(x)^\top \vartheta = f(x, \theta_n) - f(x, \theta)$ , where  $\vartheta = \theta_n - \theta$  and  $\theta_n$  is a nominal guess of the uncertain parameter  $\theta$ .*

Under Assumption 8.1, we can write (8.1) as

$$\dot{x} = f(x, \theta_n) + B(x)u - \Pi(x)^\top \vartheta$$

leading to the following theorem [3] for the NCM-based adaptive control. Note that we could also use the SDC formulation with respect to a fixed point as delineated in Theorem 3.2 [12]–[14].

**Theorem 8.1.** *Suppose that Assumption 8.1 holds and let  $\mathcal{M}$  defines the NCM of Definition 6.2, which models  $M$  of the CV-STEM contraction metric in Theorem 4.2 for the nominal system (8.1) with  $\theta = \theta_n$ , constructed with an additional convex constraint given as  $\partial_{b_i(x)} \bar{W} + \partial_{b_i(x_d)} \bar{W} = 0$ , where  $\partial_{b_i(q)} \bar{W} = \sum_i (\partial \bar{W} / \partial q_i) b_i(q)$  for  $B = [b_1, \dots, b_m]$  (see [4], [10]). Suppose also that the matched uncertainty condition [10] holds, i.e.,  $(\Pi(x) - \Pi(x_d))^\top \vartheta \in \text{span}(B(x))$  for  $\Pi(x)$ , and that (8.1) is controlled by the following adaptive control law:*

$$u = u_L + \varphi(x, x_d)^\top \hat{\vartheta} \tag{8.2}$$

$$\dot{\hat{\vartheta}} = -\Gamma(\varphi(x, x_d)B(x)^\top \mathcal{M}(x, x_d, u_d)(x - x_d) + \sigma \hat{\vartheta}) \tag{8.3}$$

where  $u_L$  is given by (6.7) of Theorem 6.3, i.e.,  $u_L = u_d - R^{-1}B^\top \mathcal{M}(x - x_d)$ ,  $\Gamma \in \mathbb{R}^{c \times c}$  is a diagonal matrix with positive elements that governs the rate of adaptation,  $\sigma \in \mathbb{R}_{\geq 0}$ , and  $(\Pi(x) - \Pi(x_d))^\top \vartheta = B(x)\varphi(x, x_d)^\top \vartheta$ .

If  $\exists \underline{\gamma}, \bar{\gamma}, \bar{b}, \bar{\rho}, \bar{\phi}, \bar{\vartheta} \in \mathbb{R}_{>0}$  s.t.  $\underline{\gamma}\mathbf{I} \leq \Gamma \leq \bar{\gamma}\mathbf{I}$ ,  $\|B(x)\| \leq \bar{b}$ ,  $\|R^{-1}(x, x_d, u_d)\| \leq \bar{\rho}$ ,  $\|\varphi(x, x_d)\| \leq \bar{\phi}$ , and  $\|\vartheta\| \leq \bar{\vartheta}$ , and if  $\Gamma$  and  $\sigma$  of (8.3) are selected to satisfy the following relation for the learning error  $\|\mathcal{M} - M\| \leq \epsilon_\ell$  in some compact set  $\mathcal{S}$  as in (6.18) of Theorem 6.3:

$$\begin{bmatrix} -2\alpha_\ell \underline{m} & \bar{\phi}\bar{b}\epsilon_\ell \\ \bar{\phi}\bar{b}\epsilon_\ell & -2\sigma \end{bmatrix} \leq -2\alpha_a \begin{bmatrix} \bar{m} & 0 \\ 0 & 1/\underline{\gamma} \end{bmatrix} \quad (8.4)$$

for  $\exists \alpha_a \in \mathbb{R}_{>0}$ ,  $\alpha_\ell$  given in Theorem 6.3, and  $\underline{m}$  and  $\bar{m}$  given in  $\underline{m}\mathbf{I} \leq M \leq \bar{m}\mathbf{I}$  of (2.26), then the system (8.1) is robust against bounded deterministic and stochastic disturbances with  $\sigma \neq 0$ , and we have the following bound in the compact set  $\mathcal{S}$ :

$$\|\mathbf{e}(t)\| \leq (V_\ell(0)e^{-\alpha_a t} + \alpha_a^{-1}\sigma\sqrt{\bar{\gamma}\bar{\vartheta}}(1 - e^{-\alpha_a t}))/\sqrt{\underline{m}} \quad (8.5)$$

where  $\mathbf{e} = x - x_d$ , and  $V_\ell = \int_{\xi_0}^{\xi_1} \|\Theta\delta q\|$  is defined in Theorem 2.3 with  $M = \Theta^\top\Theta$  replaced by  $\text{diag}(M, \Gamma^{-1})$  for  $\xi_0 = [x_d^\top, \vartheta^\top]^\top$  and  $\xi_1 = [x^\top, \hat{\vartheta}^\top]^\top$ . Furthermore, if the learning error  $\epsilon_\ell = 0$  (CV-STEM control), (8.3) with  $\sigma = 0$  guarantees asymptotic stability of  $x$  to  $x_d$  in (8.1).

*Proof.* The proof can be found in [3], but here we emphasize the use of contraction theory. For  $u$  given by (8.2), the virtual system of a smooth path  $q(\mu, t) = [q_x^\top, q_\vartheta^\top]^\top$  parameterized by  $\mu \in [0, 1]$ , which has  $q(\mu = 0, t) = [x_d^\top, \vartheta^\top]^\top$  and  $q(\mu = 1, t) = [x^\top, \hat{\vartheta}^\top]^\top$  as its particular solutions, is given as follows:

$$\dot{q} = \begin{bmatrix} \zeta(q_x, x, x_d, u_d) + B\varphi^\top q_\vartheta - \Pi(x)^\top \vartheta \\ -\Gamma(\varphi B^\top \mathcal{M}(q_x - x_d) + \sigma(q_\vartheta - \vartheta)) + d_{q_\vartheta}(\mu, \vartheta) \end{bmatrix}$$

where  $d_{q_\vartheta}(\mu, \vartheta) = -\mu\sigma\vartheta$ . Note that  $\zeta$  is as given in (3.13), i.e.,  $\zeta = (A - BR^{-1}B^\top \mathcal{M})(q_x - x_d) + \dot{x}_d$ , where the SDC matrix  $A$  is defined as (see Lemma 3.1)

$$A(x - x_d) = f(x, \theta_n) + B(x)u_d - f(x_d, \theta_n) - B(x_d)u_d.$$

The arguments of  $A(\varrho, x, x_d, u_d)$ ,  $B(x)$ ,  $\mathcal{M}(x, x_d, u_d)$ , and  $\varphi(x, x_d)$  are omitted for notational simplicity. Since  $\dot{q}_x = \zeta(q_x, x, x_d, u_d, t)$  is contracting due to Theorems 4.2 and 6.3 with a contraction rate given by  $\alpha_\ell$  in Theorem 6.3, we have for a Lyapunov function  $V = \delta q_x^\top M \delta q_x + \delta q_\vartheta^\top \Gamma^{-1} \delta q_\vartheta$  that

$$\dot{V}/2 \leq -\alpha_\ell \delta q_x^\top M \delta q_x + \delta q_x^\top (M - \mathcal{M}) B \varphi^\top \delta q_\vartheta - \sigma \|\delta q_\vartheta\|^2 + \delta q_\vartheta^\top \delta d_{q_\vartheta}. \quad (8.6)$$

Applying (8.4) with  $\|\mathcal{M} - M\| \leq \epsilon_\ell$  of (6.18), we get

$$\begin{aligned} \dot{V}/2 - \delta q_\vartheta^\top \delta d_{q_\vartheta} &\leq -(\alpha_\ell \underline{m}) \|\delta q_x\|^2 + \bar{\phi}\bar{b}\epsilon_\ell \|\delta q_x\| \|\delta q_\vartheta\| - \sigma \|\delta q_\vartheta\|^2 \\ &\leq -\alpha_a (\bar{m} \|\delta q_x\|^2 + \|\delta q_\vartheta\|^2 / \underline{\gamma}) \leq -\alpha_a V. \end{aligned}$$

Since we have  $\|\partial d_{q\vartheta}/\partial\mu\| = \sigma\bar{\vartheta}$ , this implies  $\dot{V}_\ell \leq -\alpha_a V_\ell + \sigma\sqrt{\bar{\gamma}}\bar{\vartheta}$ , yielding the bound (8.5) due to Lemma 2.1 [3]. Robustness against deterministic and stochastic disturbances follows from Theorem 3.1 if  $\sigma \neq 0$ . If  $\epsilon_\ell = 0$  and  $\sigma = 0$ , the relation (8.6) reduces to  $\dot{V}/2 \leq -\alpha\delta q_x^\top M\delta q_x$ , which results in asymptotic stability of  $x$  to  $x_d$  in (8.1) due to Barbalat's lemma [15, p. 323] as in the proof of Theorem 2 in [10].  $\square$

**Remark 8.1.** *Although Theorem 8.1 is for the case where  $f(x)$  is affine in its parameter, it is also useful for the following types of systems with an uncertain parameter  $\theta \in \mathbb{R}^c$  and a control input  $u$  (see [3]):*

$$H(x)p^{(n)} + h(x) + \Pi(x)\theta = u$$

where  $p \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^n$ ,  $h : \mathbb{R}^n \mapsto \mathbb{R}^n$ ,  $H : \mathbb{R}^n \mapsto \mathbb{R}^{n \times n}$ ,  $\Pi : \mathbb{R}^n \mapsto \mathbb{R}^{n \times c}$ ,  $x = [(p^{(n-2)})^\top, \dots, (p)^\top]^\top$ , and  $p^{(k)}$  denotes the  $k$ th time derivative of  $p$ . In particular, adaptive sliding control [16] designs  $u$  to render the system of the composite variable  $s$  given as  $s = p^{(n-1)} - p_r^{(n-1)}$  to be contracting, where  $p_r^{(n-1)} = p_d^{(n-1)} - \sum_{i=0}^{n-2} \lambda_i e^{(i)}$ ,  $e = p - p_d$ ,  $p_d$  is a target trajectory, and  $\kappa^{n-1} + \lambda_{n-2}\kappa^{n-2} + \dots + \lambda_0$  is a stable (Hurwitz) polynomial in the Laplace variable  $\kappa$  (see Example 2.6). Since we have  $e^{(n-1)} = s - \sum_{i=0}^{n-2} \lambda_i e^{(i)}$  and the system for  $[e^{(0)}, \dots, e^{(n-2)}]$  is also contracting if  $s = 0$  due to the Hurwitz property, the hierarchical combination property [17], [18] of contraction in Theorem 2.7 guarantees  $\lim_{t \rightarrow \infty} \|p - p_d\| = 0$  [19, p. 352] (see Example 2.7).

**Example 8.1.** *Using the technique of Remark 8.1, we can construct adaptive control for the Lagrangian system in Example 2.6 as follows:*

$$\mathcal{H}(q)\ddot{q} + C(q, \dot{q})\dot{q} + \mathcal{G}(q) = u(q, \dot{q}, t) \quad (8.7)$$

$$u(q, \dot{q}, t) = -\mathcal{K}(t)(\dot{q} - \dot{q}_r) + \hat{\mathcal{H}}(q)\ddot{q}_r + \hat{C}(q, \dot{q})\dot{q}_r + \hat{\mathcal{G}}(q) \quad (8.8)$$

where  $\hat{\mathcal{H}}$ ,  $\hat{C}$ , and  $\hat{\mathcal{G}}$  are the estimates of  $\mathcal{H}$ ,  $C$ , and  $\mathcal{G}$ , respectively, and the other variables are as defined in Example 2.6. Suppose that the terms  $\mathcal{H}$ ,  $C$ , and  $\mathcal{G}$  depend linearly on the unknown parameter vector  $\theta$  as follows [19, p. 405]:

$$\mathcal{H}(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + \mathcal{G}(q) = Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\theta.$$

Updating the parameter estimate  $\hat{\theta}$  using the adaptation law,  $\dot{\hat{\theta}} = -\Gamma Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)^\top (\dot{q} - \dot{q}_r)$ , as in (8.3) where  $\Gamma > 0$ , we can define the following virtual system which has  $q = \xi_0 = [\dot{q}_r^\top, \theta^\top]^\top$  and  $q = \xi_1 = [\dot{q}^\top, \hat{\theta}^\top]^\top$  as its particular solutions:

$$\begin{bmatrix} \mathcal{H} & 0 \\ 0 & \Gamma^{-1} \end{bmatrix} (\dot{q} - \dot{\xi}_0) + \begin{bmatrix} C + \mathcal{K} & -Y \\ Y^\top & 0 \end{bmatrix} (q - \xi_0) = 0$$

where the relation  $u = -\mathcal{K}(t)(\dot{q} - \dot{q}_r) + Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\hat{\theta}$  is used. Thus, for a Lyapunov function  $V = \delta q^\top \begin{bmatrix} \mathcal{H} & 0 \\ 0 & \Gamma^{-1} \end{bmatrix} \delta q$ , we have that

$$\dot{V} = \delta q^\top \begin{bmatrix} \mathcal{K} & Y \\ -Y^\top & 0 \end{bmatrix} \delta q = \delta q^\top \begin{bmatrix} \mathcal{K} & 0 \\ 0 & 0 \end{bmatrix} \delta q$$

which results in asymptotic stability of  $\delta q$  (i.e., semi-contraction [20], see also Barbalat's lemma [19, p. 405-406]).

### 8.1.II Parameter-Dependent Contraction Metric (aNCM)

Although Theorem 8.1 utilizes the NCM designed for the nominal system (8.1) with  $\theta = \theta_n$ , we could improve its representational power by explicitly taking the parameter estimate  $\hat{\theta}$  as one of the NCM arguments [3], leading to the concept of an adaptive NCM (aNCM).

In this section, we consider multiplicatively-separable nonlinear systems given in Assumption 8.2, which holds for many types of systems including robotics systems [19], spacecraft high-fidelity dynamics [21], [22], and systems modeled by basis function approximation and DNNs [23], [24].

**Assumption 8.2.** *The dynamical system (8.1) is multiplicatively separable in terms of  $x$  and  $\theta$ , i.e.,  $\exists Y_f : \mathbb{R}^n \mapsto \mathbb{R}^{n \times c_z}$ ,  $Y_{b_i} : \mathbb{R}^n \mapsto \mathbb{R}^{n \times c_z}$ , and  $Z : \mathbb{R}^c \mapsto \mathbb{R}^{c_z}$  s.t.*

$$Y_f(x)Z(\theta) = f(x, \theta), \quad Y_{b_i}(x)Z(\theta) = b_i(x, \theta)$$

where  $B(x, \theta) = [b_1(x, \theta), \dots, b_m(x, \theta)]$ . We could define  $\theta$  as  $[\theta^\top, Z(\theta)^\top]^\top$  so we have  $Y_f(q)\theta = f(q, \theta)$  and  $Y_{b_i}(q)\theta = b_i(q; \theta)$ . Such an over-parameterized system could be regularized using the Bregman divergence as in [25] (see Example 8.4), and thus we denote  $[\theta^\top, Z(\theta)^\top]^\top$  as  $\theta$  in the subsequent discussion.

Under Assumption 8.2, the dynamics for  $e = x - x_d$  of (8.1) can be expressed as follows:

$$\dot{e} = A(\varrho, x, x_d, u_d, \hat{\theta})e + B(x, \hat{\theta})(u - u_d) - \tilde{Y}(\hat{\theta} - \theta) \quad (8.9)$$

where  $A$  is the SDC matrix of Lemma 3.1,  $\hat{\theta}$  is the current estimate of  $\theta$ , and  $\tilde{Y}$  is defined as

$$\tilde{Y} = Y - Y_d = (Y_f(x) + Y_b(x, u)) - (Y_f(x_d) + Y_b(x_d, u_d)) \quad (8.10)$$

where  $Y_b(x, u) = \sum_{i=1}^m Y_{b_i}(q)u_i$ .

**Definition 8.1.** *The adaptive Neural Contraction Metric (aNCM) in Fig. 8.1 is a DNN model for the optimal parameter-dependent contraction metric, given by solving the adaptive CV-STEM, i.e., (4.2) of Theorem 4.2 (or Theorem 4.6 for differential feedback) with its contraction constraint replaced by the following convex constraint:*

$$-\Xi + 2 \operatorname{sym}(A\bar{W}) - 2\nu BR^{-1}B^\top \leq -2\alpha\bar{W} \quad (8.11)$$

for deterministic systems, and

$$\begin{bmatrix} -\Xi + 2 \operatorname{sym}(A\bar{W}) - 2\nu BR^{-1}B^\top + 2\alpha\bar{W} & \bar{W} \\ \bar{W} & -\frac{\nu}{\alpha_s} \mathbf{I} \end{bmatrix} \leq 0 \quad (8.12)$$

for stochastic systems, where  $W = M(x, x_d, u_d, \hat{\theta})^{-1} > 0$  (or  $W = M(x, \hat{\theta})^{-1} > 0$ , see Theorem 3.2),  $\bar{W} = \nu W$ ,  $\nu = \bar{m}$ ,  $R = R(x, x_d, u_d) > 0$  is a weight matrix on  $u$ ,  $\Xi = (d/dt)|_{\hat{\theta}} \bar{W}$  is the time derivative of  $\bar{W}$  computed along (8.1) with  $\theta = \hat{\theta}$ ,  $A$  and  $B$  are given in (8.9),  $\alpha$ ,  $\underline{m}$ ,  $\bar{m}$ , and  $\alpha_s$  are as given in Theorem 4.2, and the arguments  $(x, x_d, u_d, \hat{\theta})$  are omitted for notational simplicity.

The aNCM given in Definition 8.1 has the following stability property along with its optimality due to the CV-STEM of Theorem 4.2 [3].

**Theorem 8.2.** *Suppose that Assumption 8.2 holds and let  $\mathcal{M}$  define the aNCM of Definition 8.1. Suppose also that the true dynamics (8.1) is controlled by the following adaptive control law:*

$$u = u_d - R(x, x_d, u_d)^{-1} B(x, \hat{\theta})^\top \mathcal{M}(x, x_d, u_d, \hat{\theta}) \mathbf{e} \quad (8.13)$$

$$\hat{\theta} = \Gamma((Y^\top d\mathcal{M}_x + Y_d^\top d\mathcal{M}_{x_d} + \tilde{Y}^\top \mathcal{M}) \mathbf{e} - \sigma \hat{\theta}) \quad (8.14)$$

where  $d\mathcal{M}_q = [(\partial\mathcal{M}/\partial q_1)\mathbf{e}, \dots, (\partial\mathcal{M}/\partial q_n)\mathbf{e}]^\top/2$ ,  $\mathbf{e} = x - x_d$ ,  $\Gamma > 0$ ,  $\sigma \in \mathbb{R}_{\geq 0}$ , and  $Y$ ,  $Y_d$ ,  $\tilde{Y}$  are given in (8.10). Suppose further that the learning error in  $\|\mathcal{M} - M\| \leq \epsilon_\ell$  of Theorem 6.3 additionally satisfies  $\|d\mathcal{M}_{x_d} - dM_{x_d}\| \leq \epsilon_\ell$  and  $\|d\mathcal{M}_x - dM_x\| \leq \epsilon_\ell$  in some compact set  $\mathcal{S}$ .

If  $\exists \bar{b}, \bar{\rho}, \bar{y} \in \mathbb{R}_{>0}$  s.t.  $\|B(x, \hat{\theta})\| \leq \bar{b}$ ,  $\|R^{-1}(x, x_d, u_d)\| \leq \bar{\rho}$ ,  $\|Y\| \leq \bar{y}$ ,  $\|Y_d\| \leq \bar{y}$ , and  $\|\tilde{Y}\| \leq \bar{y}$  in (8.13) and (8.14), and if  $\Gamma$  and  $\sigma$  of (8.14) are selected to satisfy the following as in Theorem 8.1:

$$\begin{bmatrix} -2\alpha_\ell \underline{m} & \bar{y}\epsilon_\ell \\ \bar{y}\epsilon_\ell & -2\sigma \end{bmatrix} \leq -2\alpha_a \begin{bmatrix} \bar{m} & 0 \\ 0 & 1/\underline{\gamma} \end{bmatrix} \quad (8.15)$$

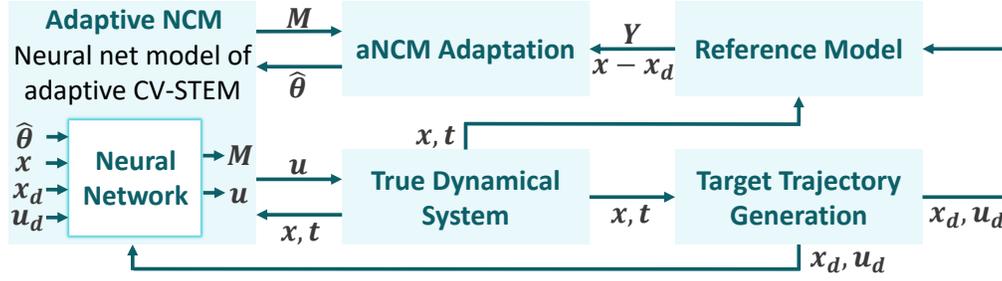


Figure 8.1: Illustration of aNCM ( $M$ : positive definite matrix that defines aNCM;  $\hat{\theta}$ : estimated parameter;  $Y$ : error signal, see (8.10);  $x$ : system state;  $u$ : system control input; and  $(x_d, u_d)$ : target state and control input trajectory).

for  $\exists \alpha_a \in \mathbb{R}_{>0}$ ,  $\alpha_\ell$  given in Theorem 6.3, and  $\underline{m}$  and  $\bar{m}$  given in  $\underline{m}\mathbf{I} \leq M \leq \bar{m}\mathbf{I}$  of (2.26), then the system (8.1) is robust against bounded deterministic and stochastic disturbances, and we have the exponential bound (8.5) in the compact set  $\mathcal{S}$ . Furthermore, if  $\epsilon_\ell = 0$  (adaptive CV-STEM control), (8.14) with  $\sigma = 0$  guarantees asymptotic stability of  $x$  to  $x_d$  in (8.1).

*Proof.* Replacing the contraction constraints of the CV-STEM in Theorem 4.2 by (8.11) and (8.12), the bound (8.5) and the asymptotic stability result can be derived as in the proof of Theorem 8.1 (see Theorem 4 and Corollary 2 of [3] for details). Theorems 2.4 and 2.5 guarantee robustness of (8.1) against bounded deterministic and stochastic disturbances for  $\sigma \neq 0$ .  $\square$

Let us again emphasize that, by using Theorem 4.6, the results of Theorems 8.1 and 8.2 can be extended to adaptive control with CCM-based differential feedback [10], [11] (see Table 3.1 for the trade-offs).

Since the adaptation laws (8.3) and (8.14) in Theorems 8.1 and 8.2 yield an explicit bound on the steady-state error as in (8.5), it could be used as the objective function of the CV-STEM in Theorem 4.2, regarding  $\Gamma$  and  $\sigma$  as extra decision variables to get  $M$  optimal in a sense different from Theorem 4.2. Smaller  $\epsilon_\ell$  would lead to a weaker condition on them in (8.4) and (8.15). Also, the size of  $\|\vartheta\| \leq \bar{\vartheta}$  in (8.5) can be adjusted simply by rescaling it (e.g.,  $\vartheta \rightarrow \theta/\bar{\vartheta}$ ). However, such a robustness guarantee comes with a drawback of having  $\lim_{t \rightarrow \infty} \|\hat{\theta}(t)\| = 0$  for  $\sigma \neq 0$  in (8.3), leading to the trade-offs in different types of adaptation laws, some of which are given in Examples 8.2 – 8.4 (see also Remark 8.2).

**Example 8.2.** Let us briefly describe the following projection operator-based adaptation law, again for the Lagrangian system (8.7) of Example 8.1 with the unknown parameter vector  $\theta$ :

$$\dot{\hat{\theta}} = \text{Proj}(\hat{\theta}, -\Gamma Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)^\top (\dot{q} - \dot{q}_r), p) \quad (8.16)$$

where  $\text{Proj}$  is the projection operator and  $p$  is a convex boundary function (e.g.,  $p(\hat{\theta}) = (\hat{\theta}^\top \hat{\theta} - \theta_{\max}^2) / (\epsilon_\theta \theta_{\max}^2)$ ) for given positive constants  $\theta_{\max}$  and  $\epsilon_\theta$ . If  $p(\hat{\theta}) > 0$  and  $\nabla p(\hat{\theta})^\top \xi > 0$ ,

$$\text{Proj}(\hat{\theta}, \xi, p) = \xi - (\nabla p(\hat{\theta}) \nabla p(\hat{\theta})^\top / \|\nabla p(\hat{\theta})\|^2) \xi p(\hat{\theta}),$$

otherwise,  $\text{Proj}(\hat{\theta}, \xi, p) = \xi$ . The projection operator has the following useful property which allows bounding the parameter estimate  $\hat{\theta}$ .

**Lemma 8.1.** If  $\hat{\theta}$  is given by (8.16) with  $\hat{\theta}(0) \in \Omega_\theta = \{\theta \in \mathbb{R}^c | p(\theta) \leq 1\}$  for a convex function  $p(\theta)$ , then  $\hat{\theta}(t) \in \Omega_\theta, \forall t \geq 0$  (e.g.,  $\|\hat{\theta}\| \leq \theta_{\max} \sqrt{1 + \epsilon_\theta}$  if  $p(\hat{\theta}) = (\hat{\theta}^\top \hat{\theta} - \theta_{\max}^2) / (\epsilon_\theta \theta_{\max}^2)$ ).

*Proof.* See [26]. □

Since Lemma 8.1 guarantees the boundedness of  $\|\hat{\theta}\|$ , the adaptive controller (8.8),  $u = -\mathcal{K}(\dot{q} - \dot{q}_r) + Y\theta + Y\tilde{\theta}$ , can be viewed as the exponentially stabilizing controller  $-\mathcal{K}(\dot{q} - \dot{q}_r) + Y\theta$  (see Example 2.6) plus a bounded external disturbance  $Y\tilde{\theta}$ , implying robustness due to Theorem 2.4 [11], [26]–[28].

Let us also remark that, as in Example 8.1, the projection operator-based adaptation law (8.16) still achieves asymptotic stabilization. Applying the control law (8.8) with the adaptation (8.16) yields the following virtual system which has  $q = \xi_0 = [\dot{q}_r^\top, \theta^\top]^\top$  and  $q = \xi_1 = [\dot{q}^\top, \hat{\theta}^\top]^\top$  as its particular solutions:

$$\begin{bmatrix} \mathcal{H} & 0 \\ 0 & \mathbf{I} \end{bmatrix} (\dot{q} - \dot{\xi}_0) + \begin{bmatrix} (C + \mathcal{K})(q_s - \dot{q}_r) - Y(q_\theta - \theta) \\ \text{Proj}(\hat{\theta}, \Gamma Y^\top (q_s - \dot{q}_r), p) \end{bmatrix} = 0$$

where  $q = [q_s^\top, q_\theta^\top]^\top$ . Thus, for a Lyapunov function  $V_{s\ell} = \frac{1}{2} \int_{\xi_0}^{\xi_1} \delta q^\top \begin{bmatrix} \mathcal{H}(q) & 0 \\ 0 & \Gamma^{-1} \end{bmatrix} \delta q$ , we have that

$$\dot{V}_{s\ell} = \int_{\xi_0}^{\xi_1} -\delta q_s^\top \mathcal{K} \delta q_s + \delta q_\theta^\top (\text{Proj}(\hat{\theta}, Y^\top \delta q_s, p) - Y^\top \delta q_s).$$

Using the convex property of the projection operator, i.e.,  $\tilde{\theta}^\top (\text{Proj}(\hat{\theta}, \xi, p) - \xi) \leq 0$  [26], [27], this gives  $\dot{V}_{s\ell} \leq -\int_{\xi_0}^{\xi_1} \delta q_s^\top \mathcal{K} \delta q_s$ , which results in asymptotic stability of  $\delta q$  due to Barbalat's lemma [19, p. 405-406] as in Example 8.1.

**Example 8.3.** The dependence on  $u$  and  $\hat{\theta}$  in  $(d/dt)|_{\hat{\theta}}M$  can be removed by using  $\partial_{b_i(x)}M + \partial_{b_i(x_d)}M = 0$  as in Theorem 8.1, and by using adaptation rate scaling introduced in [11]. In essence, the latter multiplies the adaptation law (8.14) by any strictly-increasing and strictly-positive scalar function  $v(2\rho)$ , and update  $\rho$  as

$$\dot{\rho} = \frac{1}{2} \frac{v(2\rho)}{v_\rho(2\rho)} \sum_{i=1}^c \frac{1}{V_e + \eta} \frac{\partial V_e}{\partial \hat{\theta}_i} \dot{\theta}_i \quad (8.17)$$

where  $v_\rho = \partial v / \partial \rho$ ,  $\eta \in \mathbb{R}_{>0}$ , and  $V_e = \mathbf{e}^\top M(x, x_d, u_d, \hat{\theta}) \mathbf{e}$  for  $\mathbf{e} = x - x_d$  and  $M$  given in Definition 8.1, so the additional term due to (8.17) cancels out the term involving  $\hat{\theta}$  in  $(d/dt)|_{\hat{\theta}}\bar{W}$  of (8.11) (see [11] for details). Its robustness property follows from Theorem 8.2 also in this case.

**Example 8.4.** Using the Bregman divergence-based adaptation in [25], we could implicitly regularize the parameter estimate  $\hat{\theta}$  as follows:

$$\lim_{t \rightarrow \infty} \hat{\theta} = \arg \min_{\vartheta \in \mathcal{A}} d_\psi(\vartheta \| \theta^*) = \arg \min_{\vartheta \in \mathcal{A}} d_\psi(\vartheta \| \hat{\theta}(0))$$

where  $d_\psi$  is the Bregman divergence defined as  $d_\psi(x \| y) = \psi(x) - \psi(y) - (x - y)^\top \nabla \psi(y)$  for a convex function  $\psi$ , and  $\mathcal{A}$  is a set containing only parameters that interpolate the dynamics along the entire trajectory. If  $\hat{\theta}(0) = \arg \min_{b \in \mathbb{R}^p} \psi(b)$ , we have  $\lim_{t \rightarrow \infty} \hat{\theta} = \arg \min_{\vartheta \in \mathcal{A}} \psi(\vartheta)$ , which regularizes  $\hat{\theta}$  to converge to a parameter that minimizes  $\psi$ . For example, using 1-norm for  $\psi$  would impose sparsity on the steady-state parameter estimate  $\hat{\theta}$  [25].

These extensions of adaptive control techniques described in Examples 8.2 – 8.4 could be utilized with contraction theory and learning-based control as in Theorems 8.1 and 8.2.

**Remark 8.2.** Note that the results presented earlier in this chapter do not necessarily mean parameter convergence,  $\lim_{t \rightarrow \infty} \|\tilde{\theta}\| = 0$ , as the adaptation objective is to drive the tracking error  $\|x - x_d\|$  to zero [19, p. 331], not to find out the true parameter  $\theta$  out of the many that achieve perfect tracking.

Asymptotic parameter convergence,  $\lim_{t \rightarrow \infty} \|\tilde{\theta}\| = 0$  for  $\theta$  of (8.1) under Assumption 8.2, could be guaranteed if there is no disturbance and learning error with  $\sigma = 0$ , and we have  $\exists T, \alpha_{PE} \in \mathbb{R}_{>0}$  s.t.  $\int_t^{t+T} \tilde{Y}^\top \tilde{Y} d\tau \geq \alpha_{PE} \mathbf{I}$ ,  $\forall t$  for  $\tilde{Y}$  given in (8.10) (the persistent excitation condition [19, p. 366]). We could also utilize the Bregman divergence-based adaptation to regularize the behavior of  $\lim_{t \rightarrow \infty} \|\hat{\theta}\|$  as in Example 8.4.

## 8.2 Contraction Theory for Learned Models

Let us next consider the case where the dynamical system is partially or completely unknown and we only have access to a large amount of system trajectory data, where the assumptions in Sec. 8.1 for using learning-based adaptive control techniques are no longer valid.

### 8.2.I Robust System Identification with Contraction Theory

A typical approach is to perform system identification [6], [29]–[31] using trajectory data generated by (8.19):

$$\dot{x} = f_L(x, u(x, t), t) \quad (8.18)$$

$$\dot{x}^* = f_{\text{true}}(x^*, u(x^*, t), t) \quad (8.19)$$

where  $x : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  is the system state,  $u : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^m$  is the system control input,  $f_{\text{true}} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  is a smooth function of the true dynamical system (8.19), which is unknown and thus modeled by a learned smooth function  $f_L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  of (8.18). If we can learn  $f_L$  to render the system (8.18) contracting, contraction theory still allows us to ensure the robustness and stability of these systems.

**Theorem 8.3.** *Let  $q(0, t) = \xi_0(t) = x(t)$ ,  $q(1, t) = \xi_1(t) = x^*(t)$ , and  $g = f_L$  in Theorems 5.2 and 5.3 as in Example 5.3, and define  $\Delta_L$  as*

$$\Delta_L = f_{\text{true}}(x^*, u(x^*, t), t) - f_L(x^*, u(x^*, t), t) \quad (8.20)$$

for the learning error condition  $\|\Delta_L\| \leq \epsilon_{\ell 0} + \epsilon_{\ell 1} \|\xi_1 - \xi_0\|$  in (5.3). If the function  $f_L$  is learned to satisfy (5.3) with  $\epsilon_{\ell 1} = 0$ , i.e.,  $\|\Delta_L\| \leq \epsilon_{\ell 0}$  in some compact set  $\mathcal{S}$ , and if there exists a contraction metric defined by  $M$  bounded as  $\underline{m}\mathbf{I} \leq M \leq \overline{m}\mathbf{I}$  as in (2.26), which renders (8.18) contracting as in (2.4) of Theorem 2.1 for deterministic systems, i.e.,

$$\dot{M} + M(\partial f_L / \partial x) + (\partial f_L / \partial x)^\top M \leq -2\alpha M, \quad (8.21)$$

and (2.38) of Theorem 2.5 for stochastic systems, i.e.,

$$\dot{M} + M(\partial f_L / \partial x) + (\partial f_L / \partial x)^\top M \leq -2\alpha M - \alpha_s \mathbf{I}, \quad (8.22)$$

then we have the following in the compact set  $\mathcal{S}$ :

$$\|x(t) - x^*(t)\| \leq \frac{V_\ell(0)}{\sqrt{\underline{m}}} e^{-\alpha t} + \frac{\epsilon_{\ell 0}}{\alpha} \sqrt{\frac{\overline{m}}{\underline{m}}} (1 - e^{-\alpha t}) \quad (8.23)$$

for  $x$  and  $x^*$  in (8.18) and (8.19), where  $V_\ell = \int_{x^*}^x \|\Theta \delta x\|$  as in Theorem 2.3 with  $M = \Theta^\top \Theta$ . Furthermore, the systems (8.18) and (8.19) are robust against bounded deterministic and stochastic disturbances.

*Proof.* Let  $p_t^* = (x^*, u(x^*(t), t), t)$  for notational simplicity. Since (8.19) can be written as  $\dot{x}^* = f_{\text{true}}(p_t^*) = f_L(p_t^*) + (f_{\text{true}}(p_t^*) - f_L(p_t^*))$  (see Example 5.3) and  $\|\Delta_L\| = \|f_{\text{true}}(p_t^*) - f_L(p_t^*)\| \leq \epsilon_{\ell 0}$ , Theorem 2.4 holds with  $\bar{d} = \epsilon_{\ell 0}$  as (8.18) is contracting, resulting in (8.23). Also, defining  $\Delta_L$  as (8.20) in Theorems 5.2 and 5.3 results in robustness of (8.18) and (8.19) against bounded deterministic and stochastic disturbances due to (8.21) and (8.22), respectively.  $\square$

Theorem 8.3 is the theoretical foundation for stability analysis of model-free nonlinear dynamical systems. The bound (8.23) becomes tighter as we achieve smaller  $\epsilon_{\ell 0}$  using more training data for verifying  $\|\Delta_L\| \leq \epsilon_{\ell 0}$  (see Remark 5.1). From here onwards, we utilize contraction theory to provide stability guarantees to such model-free systems, partially enabling the use of the aforementioned model-based techniques.

## 8.2.II Robust Control of Systems Modeled by DNNs

One challenge in applying Theorem 8.3 in practice is to find contraction metrics for the control non-affine nonlinear systems (8.18). This section delineates one way to construct a contraction metric in Theorem 8.3 for provably-stable feedback control, using the CV-STEM and NCM of Theorems 4.2, 4.6, 6.1, 6.3 – 6.5, and 7.1, along with the spectrally-normalized DNN of Definition 6.3.

To this end, let us assume that  $f_{\text{true}}$  of the dynamics (8.19) can be decomposed into a known control-affine part  $f(x^*, t) + B(x^*, t)u$  and an unknown control non-affine residual part  $r(x^*, u, t)$  as follows:

$$\dot{x}^* = f_{\text{true}} = f(x^*, t) + B(x^*, t)(u + r(x^*, u, t)). \quad (8.24)$$

Ideally, we would like to design  $u$  as

$$u = u^*(x, t) - r_L(x, u, t) \quad (8.25)$$

to cancel out the unknown term  $r(x, u, t)$  of the dynamical system (8.24) by the model  $r_L(x, u, t)$  learned using trajectory data, where  $u^*$  is a nominal stabilizing control input for  $\dot{x} = f(x, t) + B(x, t)u$  given by, e.g., Theorems 4.2 and 4.6. However, the equation (8.25) depends implicitly on  $u$ , which brings extra computational burden

especially if the learned model  $r_L(x, u, t)$  is highly nonlinear as in DNNs. In [29], a discrete-time nonlinear controller is proposed to iteratively solve (8.25).

**Lemma 8.2.** *Define a mapping  $\mathcal{F}$  as  $\mathcal{F}(u) = u^*(x, t) - r_L(x, u, t)$ , where  $u^*$  and  $r_L$  are given in (8.25). If  $r_L$  is Lipschitz in  $u$  with a 2-norm Lipschitz constant  $L_u < 1$ , i.e.,  $\|r_L(x, u, t) - r_L(x, u', t)\| \leq L_u \|u - u'\|$ ,  $\forall u, u'$ , then  $\mathcal{F}$  is a contraction mapping for fixed  $x, t$ . Therefore, if  $x, t$  are fixed, discrete-time nonlinear control  $u_k$  defined as*

$$u_k = \mathcal{F}(u_{k-1}) = u^*(x, t) - r_L(x, u_{k-1}, t) \quad (8.26)$$

*converges to a unique solution  $u$  given by  $u = \mathcal{F}(u)$ .*

*Proof.* Since  $r_L$  is Lipschitz, we have that

$$\|\mathcal{F}(u) - \mathcal{F}(u')\| \leq \|r_L(x, u, t) - r_L(x, u', t)\| \leq L_u \|\Delta u\|$$

where  $\Delta u = u - u'$ . Thus, the assumption  $L_u < 1$  ensures that  $\mathcal{F}$  is a contraction mapping for fixed  $x, t$  [29].  $\square$

By applying contraction theory to the discrete-time controller (8.26) of Lemma 8.2, we can guarantee the stability of (8.24) if  $r_L$  is modeled by a spectrally-normalized DNN of Definition 6.3.

**Theorem 8.4.** *Let  $x$  be the trajectory of the following ideal system without the unknown part  $r$  of the dynamics (8.24):*

$$\dot{x} = f(x, t) + B(x, t)u^*(x, t) \quad (8.27)$$

*where  $u^*$  is a stabilizing controller that renders (8.27) contracting as in Theorem 8.3 for  $M$  which satisfies  $\underline{m}\mathbf{I} \leq M \leq \bar{m}\mathbf{I}$  of (2.26). Note that such  $u^*$  can be designed by using, e.g., Theorems 4.2 and 4.6. Suppose that the true dynamics (8.24) is controlled by (8.26) and*

$$\exists \rho \in \mathbb{R}_{\geq 0} \text{ s.t. } \|u_k - u_{k-1}\| \leq \rho \|x - x^*\| \quad (8.28)$$

*for  $x^*$  in (8.24) [29]. If  $\exists \bar{b} \in \mathbb{R}_{\geq 0}$  s.t.  $\|B(x, t)\| \leq \bar{b}$ , and if  $r_L$  is modeled by a spectrally-normalized DNN of Definition 6.3 to have*

$$\|r_L(x, u, t) - r(x, u, t)\| \leq \epsilon_\ell \quad (8.29)$$

for all  $x \in \mathcal{S}_s$ ,  $u \in \mathcal{S}_u$ , and  $t \in \mathcal{S}_t$ , where  $\mathcal{S}_s \subseteq \mathbb{R}^n$ ,  $\mathcal{S}_u \subseteq \mathbb{R}^m$ , and  $\mathcal{S}_t \subseteq \mathbb{R}_{\geq 0}$  are some compact sets, then  $r_L$  is Lipschitz continuous, and the controller (8.26) applied to (8.24) gives the following bound in the compact set:

$$\|x(t) - x^*(t)\| \leq \frac{V_\ell(0)}{\sqrt{m}} e^{-\alpha_\ell t} + \frac{\bar{b}\epsilon_\ell}{\alpha_\ell} \sqrt{\frac{m}{m}} (1 - e^{-\alpha_\ell t}) \quad (8.30)$$

as long as the Lipschitz constant of  $r_L$  is selected to have

$$\exists \alpha_\ell \in \mathbb{R}_{>0} \text{ s.t. } \alpha_\ell = \alpha - \bar{b}L_u\rho\sqrt{m/m} > 0 \quad (8.31)$$

where  $V_\ell = \int_{x^*}^x \|\Theta\delta x\|$  as in Theorem 2.3 with  $M = \Theta^\top\Theta$ . Furthermore, the system (8.24) with the controller (8.26) is robust against deterministic and stochastic disturbances.

*Proof.* If  $r_L$  is modeled by a spectrally-normalized DNN, we can arbitrarily choose its Lipschitz constant  $L_u$  by Lemma 6.2. Applying (8.26) to (8.24) yields

$$\dot{x}^* = f_{cl}(x^*, t) + B(x^*, t)(r(x^*, u_k, t) - r_L(x^*, u_{k-1}, t))$$

where  $f_{cl}(x^*, t) = f(x^*, t) + B(x^*, t)u^*$ . Using the Lipschitz condition on  $r_L$  and the learning error assumption (8.29), we have that

$$\|r(x^*, u_k, t) - r_L(x^*, u_{k-1}, t)\| \leq \epsilon_\ell + L_u\|u_k - u_{k-1}\| \leq \epsilon_\ell + L_u\rho\|x - x^*\|$$

where (8.28) is used to obtain the second inequality. Since we have  $\|B(x, t)\| \leq \bar{b}$  and the closed-loop system  $\dot{x} = f_{cl}(x, t)$  is contracting, Theorem 5.2 holds with  $\epsilon_{\ell 0} = \bar{b}\epsilon_\ell$  and  $\epsilon_{\ell 1} = \bar{b}L_u\rho$  as long as we select  $L_u$  to satisfy (8.31), resulting in the bound (8.30). The final robustness statement follows from Theorems 5.2 and 5.3.  $\square$

Theorem 8.4 implies that the control synthesis algorithms via contraction theory, including robust control of Theorems 4.2 and 4.6 (CV-STEM), learning-based robust control of Theorems 6.1, 6.3 – 6.5, and 7.1 (NCM, LAG-ROS), can be enhanced to provide explicit robustness and stability guarantees even for systems partially modeled by DNNs that depend nonlinearly on  $u$ .

**Example 8.5.** Let us consider the following Lagrangian system of Example 2.6 perturbed externally by unknown control non-affine residual disturbance:

$$\mathcal{H}(q)\ddot{q} + C(q, \dot{q})\dot{q} + \mathcal{G}(q) = u + r(x, u) \quad (8.32)$$

where  $x = [\mathbf{q}^\top, \dot{\mathbf{q}}^\top]^\top$  and the other variables are as given in Example 2.6. Using the result of Theorem 8.4, we can design a discrete-time nonlinear controller by augmenting the exponentially stabilizing controller of Example 2.6 with a learned residual part  $r_L(x, u)$  as follows:

$$u_k = -\mathcal{K}(t)(\dot{\mathbf{q}} - \dot{\mathbf{q}}_r) + \mathcal{H}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \mathcal{G}(\mathbf{q}) - r_L(x, u_{k-1})$$

where  $\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d(t) - \Lambda(t)(\mathbf{q} - \mathbf{q}_d(t))$ ,  $\mathcal{K} : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{n \times n}$ ,  $\Lambda : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{n \times n}$ , and  $(\mathbf{q}_d, \dot{\mathbf{q}}_d)$  is the target trajectory of the state  $(\mathbf{q}, \dot{\mathbf{q}})$ , and  $\mathcal{K}, \Lambda > 0$  are control gain matrices (design parameters). Again, note that  $\dot{\mathcal{H}} - 2\mathbf{C}$  is skew-symmetric with  $\mathcal{H} > 0$  by construction. This gives us the following closed-loop virtual system of a smooth path  $q(\mu, t)$  parameterized by  $\mu \in [0, 1]$ , which has  $q(\mu = 0, t) = \dot{\mathbf{q}}_r$  and  $q(\mu = 1, t) = \dot{\mathbf{q}}$  as its particular solutions as in Example 2.6, but now with non-zero perturbation due to  $r(x, u)$ :

$$\mathcal{H}(\dot{\mathbf{q}} - \ddot{\mathbf{q}}_r) + (\mathbf{C} + \mathcal{K})(\mathbf{q} - \dot{\mathbf{q}}_r) = \mu(r(x, u_k) - r_L(x, u_{k-1})).$$

After some algebra as in the proof of Theorem 8.4, we can show that

$$\frac{d}{dt} \int_0^1 \|\Theta \partial_\mu q\| \leq -\left(\frac{k_\ell}{h_u} - \frac{L_u \rho}{h_\ell}\right) \int_0^1 \|\Theta \partial_\mu q\| + \frac{\epsilon_\ell}{\sqrt{h_\ell}}$$

where  $\mathcal{H} = \Theta^\top \Theta$ ,  $h_\ell \mathbf{I} \leq \mathcal{H} \leq h_u \mathbf{I}$ ,  $k_\ell \mathbf{I} \leq \mathcal{K}$ ,  $\|u_k - u_{k-1}\| \leq \rho \|\dot{\mathbf{q}} - \dot{\mathbf{q}}_r\|$ ,  $\epsilon_\ell$  is the learning error of  $r_L$ , and  $L_u$  is the Lipschitz constant of  $r_L$  (assuming  $r_L$  is modeled by a spectrally-normalized DNN of Definition 6.3). This indeed indicates that the tracking error of the Lagrangian system (8.32) is exponentially bounded as proven in Theorem 8.4.

In [29], the technique in Theorem 8.4 and in Example 8.5 is used to perform precise near-ground trajectory control of multi-rotor drones, by learning complex aerodynamic effects caused by high-order interactions between multi-rotor airflow and the environment. It is demonstrated that it significantly outperforms a baseline nonlinear tracking controller in both landing and cross-table trajectory tracking tasks. Theorem 8.4 enables applying it to general nonlinear systems with state and control dependent uncertainty, as long as we have a nominal exponentially stabilizing controller (which can be designed by Theorems 4.2 and 4.6, or approximately by Theorems 6.1, 6.3 – 6.5, and 7.1).

## References

- [1] N. M. Boffi, S. Tu, and J.-J. E. Slotine, *Regret bounds for adaptive nonlinear control*, arXiv:2011.13101, 2020.

- [2] K. El Rifai, O. El Rifai, and K. Youcef-Toumi, “On robust adaptive switched control,” in *Proc. Amer. Control Conf.*, vol. 1, 2005, pp. 18–23.
- [3] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Learning-based Adaptive Control using Contraction Theory,” in *IEEE Conf. Decis. Control*, 2021, pp. 2533–2538.
- [4] I. R. Manchester and J.-J. E. Slotine, “Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design,” *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 3046–3053, Jun. 2017.
- [5] S. Singh, A. Majumdar, J.-J. E. Slotine, and M. Pavone, “Robust online motion planning via contraction theory and convex optimization,” in *IEEE Int. Conf. Robot. Automat.*, May 2017, pp. 5883–5890.
- [6] S. Singh, S. M. Richards, V. Sindhvani, J.-J. E. Slotine, and M. Pavone, “Learning stabilizable nonlinear dynamics with contraction-based regularization,” *Int. J. Robot. Res.*, Aug. 2020.
- [7] R. Wang, R. Tóth, and I. R. Manchester, “A comparison of LPV gain scheduling and control contraction metrics for nonlinear control,” in *3rd IFAC Workshop on LPVS*, vol. 52, 2019, pp. 44–49.
- [8] R. Wang, R. Tóth, and I. R. Manchester, *Virtual CCMs: Convex nonlinear feedback design via behavioral embedding*, arXiv:2003.08513, Mar. 2020.
- [9] I. R. Manchester and J.-J. E. Slotine, “Robust control contraction metrics: A convex approach to nonlinear state-feedback  $\mathcal{H}^\infty$  control,” *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 333–338, 2018.
- [10] B. T. Lopez and J.-J. E. Slotine, “Adaptive nonlinear control with contraction metrics,” *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 205–210, 2021.
- [11] B. T. Lopez and J.-J. E. Slotine, “Universal adaptive control of nonlinear systems,” *IEEE Control Syst. Lett.*, vol. 6, pp. 1826–1830, 2022.
- [12] H. Tsukamoto and S.-J. Chung, “Robust controller design for stochastic nonlinear systems via convex optimization,” *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4731–4746, 2021.
- [13] A. P. Dani, S.-J. Chung, and S. Hutchinson, “Observer design for stochastic nonlinear systems via contraction-based incremental stability,” *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 700–714, Mar. 2015.
- [14] H. Tsukamoto and S.-J. Chung, “Convex optimization-based controller design for stochastic nonlinear systems using contraction analysis,” in *IEEE Conf. Decis. Control*, Dec. 2019, pp. 8196–8203.
- [15] H. K. Khalil, *Nonlinear Systems*, 3rd. Upper Saddle River, NJ: Prentice-Hall, 2002.
- [16] J.-J. E. Slotine and J. A. Coetsee, “Adaptive sliding controller synthesis for non-linear systems,” *Int. J. Control*, vol. 43, no. 6, pp. 1631–1651, 1986.

- [17] J.-J. E. Slotine, “Modular stability tools for distributed computation and control,” *Int. J. Adapt. Control Signal Process.*, vol. 17, no. 6, pp. 397–416, 2003.
- [18] J.-J. E. Slotine and W. Lohmiller, “Modularity, evolution, and the binding problem: A view from stability theory,” *Neural Netw.*, vol. 14, no. 2, pp. 137–145, 2001.
- [19] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Upper Saddle River, NJ: Pearson, 1991.
- [20] W. Wang and J.-J. E. Slotine, “On partial contraction analysis for coupled nonlinear oscillators,” *Biol. Cybern.*, vol. 92, no. 1, pp. 38–53, Jan. 2005, ISSN: 0340-1200.
- [21] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics*. AIAA Education Series, 1999.
- [22] D. Morgan, S.-J. Chung, L. Blackmore, B. Acikmese, D. Bayard, and F. Y. Hadaegh, “Swarm-keeping strategies for spacecraft under  $J_2$  and atmospheric drag perturbations,” *AIAA J. Guid. Control Dyn.*, vol. 35, no. 5, pp. 1492–1506, 2012.
- [23] O. Nelles, *Nonlinear Dynamic System Identification*. Springer Berlin Heidelberg, 2001, pp. 547–577.
- [24] R. M. Sanner and J.-J. E. Slotine, “Gaussian networks for direct adaptive control,” *IEEE Trans. Neur. Netw.*, vol. 3, no. 6, pp. 837–863, Nov. 1992, ISSN: 1045-9227.
- [25] N. M. Boffi and J.-J. E. Slotine, *Implicit regularization and momentum algorithms in nonlinear adaptive control and prediction*, arXiv:1912.13154, 2020.
- [26] E. Lavretsky and T. E. Gibson, *Projection operator in adaptive systems*, arXiv:1112.4232, Oct. 2012.
- [27] Z. Cai, M. de Queiroz, and D. Dawson, “A sufficiently smooth projection operator,” *IEEE Trans. Autom. Control*, vol. 51, no. 1, pp. 135–139, 2006.
- [28] S.-J. Chung, S. Bandyopadhyay, I. Chang, and F. Y. Hadaegh, “Phase synchronization control of complex networks of Lagrangian systems on adaptive digraphs,” *Automatica*, vol. 49, no. 5, pp. 1148–1161, 2013.
- [29] G. Shi, X. Shi, M. O’Connell, *et al.*, “Neural lander: Stable drone landing control using learned dynamics,” in *IEEE Int. Conf. Robot. Automat.*, May 2019, pp. 9784–9790.
- [30] I. R. Manchester, M. Revay, and R. Wang, “Contraction-based methods for stable identification and robust machine learning: A tutorial,” in *IEEE Conf. Decis. Control*, 2021, pp. 2955–2962.

- [31] C. Blocher, M. Saveriano, and D. Lee, “Learning stable dynamical systems using contraction theory,” in *IEEE URAI*, 2017, pp. 124–129.

## **Part III: Real-World Applications**

*Chapter 9*

## INTERSTELLAR OBJECT EXPLORATION

- [1] H. Tsukamoto, S.-J. Chung, B. Donitz, M. Ingham, D. Mages, and Y. K. Nakka, “Neural-Rendezvous: Learning-based robust guidance and control to encounter interstellar objects,” *AIAA J. Guid. Control Dyn.*, under review, Aug. 2022.

Interstellar objects (ISOs) represent one of the last unexplored classes of solar system objects (see fig. 9.1), considered to be physical laboratories that can enable the study of exosolar systems in-situ rather than remotely using telescopes such as the Hubble or James Webb Space Telescopes. As discussed in Chapter 1, using a dedicated spacecraft to flyby an ISO opens the doors to high-resolution imaging, mass or dust spectroscopy, and a larger number of vantage points than Earth observation. It could also resolve the target’s nucleus shape and spin, characterize the volatiles being shed from an active body, reveal fresh surface material using an impactor, and more [1].



Figure 9.1: Interstellar objects. Above: Artist’s illustration 1I/‘Oumuamua (credit: NASA, ESA, and STScI). Below: 2I/Borisov near and at perihelion (credit: NASA, ESA, and D. Jewitt (UCLA)). Click the picture to watch our YouTube summary of the ISO exploration project ([https://youtu.be/8h60B\\_p1fyQ](https://youtu.be/8h60B_p1fyQ)).

The discovery and exploration of ISOs are challenging for three main reasons: 1) they are not discovered until they are close to Earth, meaning that launches to encounter them often require high launch energy; 2) their orbital properties are poorly constrained at launch, generally leading to significant on-board resources to encounter; and 3) the encounter speeds are typically high ( $> 10$  of km/s) requiring fast response autonomous operations. As outlined in Fig. 9.2, the guidance, navigation, and control (GNC) of spacecraft for the ISO encounter are split into two segments: 1) the cruise phase, where the spacecraft utilizes state estimation obtained by ground-based telescopes and navigates via ground-in-the-loop operations, and; 2) the terminal phase, where it switches to fully autonomous operation with existing onboard navigation frameworks. This chapter presents Neural-Rendezvous, proposed in [2] for performing the second phase of the autonomous terminal guidance and control (G&C) with the on-board state estimates. The overall mission outline and motivation are visually summarized at [https://youtu.be/8h60B\\_p1fyQ](https://youtu.be/8h60B_p1fyQ) (see Fig. 9.1).

### **Outline of Neural-Rendezvous**

We utilize a dynamical system-based SN-DNN for designing a real-time guidance policy that approximates nonlinear model predictive control (MPC), which is known to be near-optimal in terms of dynamic regret (i.e., the MPC performance minus the optimal performance in hindsight [3]). This is to avoid solving nonlinear MPC optimization at each time instant and compute a spacecraft's control input autonomously, even with its limited online computational capacity. Consistent with our objective of encountering ISOs, our SN-DNN uses a loss function that directly imitates the MPC state trajectory performing dynamics integration [4], [5], as well as indirectly imitating the MPC control input as in existing methods [6]. We then introduce learning-based min-norm feedback control to be used on top of this guidance policy. This provides an optimal and robust control input that minimizes its instantaneous deviation from that of the SN-DNN guidance policy, under the incremental stability condition as in the one of contraction theory [7]. Our contributions here are summarized as follows.

### **Contribution**

If the SN-DNN guidance policy is equipped with the learning-based min-norm control, the state tracking error bound with respect to the desired state trajectory decreases exponentially in expectation with a finite probability, robustly against the

state uncertainty. This indicates that the terminal spacecraft deliver error at the ISO encounter (i.e., the shaded blue region in Fig. 9.2) is probabilistically bounded in expectation, where its size can be modified accordingly to the mission requirement by tuning its feedback control parameters. Our main contribution is to provide a rigorous proof of this fact by leveraging stochastic incremental stability analysis. It is based on constructing a non-negative function with a supermartingale property for finite-time tracking performance, explicitly accounting for the ISO state uncertainty and the local nature of nonlinear state estimation guarantees. Also, the pointwise min-norm controller design is addressed and solved in analytical form for Lagrangian dynamical systems [8, p. 392], which describe a variety of general robotic motions, not just the one used in our paper for ISO exploration. We further show that the SN-DNN guidance policy possesses a verifiable optimality gap with respect to the optimization-based G&C, under the assumption that the nonlinear MPC policy is Lipschitz.

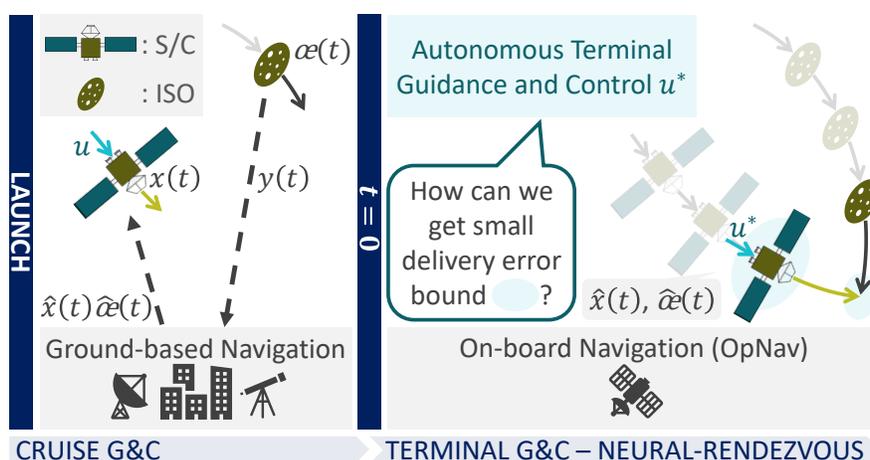


Figure 9.2: Illustration of cruise and terminal GNC ( $t$ : time;  $\alpha(t)$ : ISO state;  $x(t)$ : spacecraft state relative to  $\alpha(t)$ ;  $y(t)$ : state measurement of the ISO and spacecraft;  $u$ : control input;  $\hat{\alpha}(t)$ : estimated ISO state; and  $\hat{x}(t)$ : estimated spacecraft relative state). Neural-Rendezvous enables obtaining a verifiable delivery error bound even under the large ISO state uncertainty and high-velocity challenges.

Such verifiable and real-time optimality, stability, and robustness guarantees offer indispensable analytical insight into conventional black-box machine learning and AI-based G&C approaches. It is worth noting that Neural-Rendezvous and its guarantees are general enough to be used not only for encountering ISOs, but also for solving various nonlinear autonomous rendezvous problems accurately in real-time under external disturbances and various sources of uncertainty (e.g., state measurement noise, process noise, control execution error, unknown parts of dynamics, or

parametric/non-parametric variations of dynamics and environments). To further corroborate these arguments, the performance of Neural-Rendezvous is also empirically validated in Chapter 10 using our thruster-based spacecraft simulator called M-STAR [9] and in high-conflict and distributed UAV swarm reconfiguration with up to 20 UAVs (see (b) and (c) of Fig. 1.4).

### **Related Work**

The state of practice in realizing asteroid and comet rendezvous missions is to pre-construct an accurate spacecraft state trajectory to the target before launch, and then perform a few trajectory correction maneuvers (TCMs) along the way based on the state measurements obtained by ground-based and onboard autonomous navigation schemes as discussed in [10], [11]. Such a G&C approach is only feasible for targets with sufficient information on their orbital properties in advance, which is not realistic for ISOs visiting our solar system from interstellar space with large state uncertainty.

The ISO rendezvous problem can be cast as a robust motion planning and control problem that has been investigated in numerous studies in the field of robotics. The most well-developed and commercialized of these G&C methods is the robust nonlinear MPC [12], which extensively utilizes knowledge about the underlying nonlinear dynamical system to design an optimal control input at each time instant, thereby allowing a spacecraft to use the most updated ISO state information. When the MPC is augmented with feedback control, robustness against the state uncertainty and various external disturbances can be shown using, e.g., Lyapunov and contraction theory (see [7] and references therein). However, as mentioned earlier, the spacecraft's onboard computational power is not necessarily sufficient to solve the MPC optimization at each time instant of its TCM, which could lead to failure in accounting for the ISO state that changes dramatically in a few seconds due to the body's high velocity. Even when we solve the MPC in a discrete manner, the computational load of solving it online is not negligible, especially if the computational resource of the agent is limited and not sufficient to perform nonlinear optimization in real time.

Learning-based control designs have been considered a promising solution to this problem, as they allow replacing these optimization-based G&C algorithms with computationally-cheap mathematical models, e.g., neural networks [6]. Neural-Rendezvous can be viewed as a novel variant of a learning-based control design

with formal optimality, stability, and robustness guarantees, which are obtained by extending the results of [7] for general nonlinear systems to the case of the ISO rendezvous problem. Our proposed approach is also for providing an approximate online solution even for complex and large-scale nonlinear motion planning problems in general, (e.g., distributed and intelligent motion planning and control of high-conflict UAV swarm reconfiguration, the problem of which is highly nonlinear and thus solving even a single optimization online could become unrealistic), considering the ISO encounter problem as one example. In Chapter 10, we will see also experimental validation on these scenarios to further support this argument.

### 9.1 Technical Challenges in ISO Exploration

In this chapter, we consider the following translational dynamical system of a spacecraft relative to an Interstellar Object (ISO), equipped with feedback control  $u \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^m$ :

$$\dot{x}(t) = f(x(t), \alpha(t), t) + B(x(t), \alpha(t), t)u(\hat{x}(t), \hat{\alpha}(t), t) \quad (9.1)$$

where  $t \in \mathbb{R}_{\geq 0}$  is time,  $\alpha \in \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  are the ISO orbital elements evolving by a separate equation of motion (see [13] for details),  $x \in \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  is the state of the spacecraft relative to  $\alpha$  in a local-vertical local-horizontal (LVLH) frame centered on the ISO [14, pp. 710-712],  $f : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  and  $B : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{n \times m}$  are known smooth functions [15] (see (9.2)), and  $\hat{\alpha} \in \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  and  $\hat{x} \in \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$  are the estimated ISO and spacecraft relative state in the LVLH frame given by an on-board navigation scheme, respectively. Particularly when we select  $x$  as  $x = [p^\top, \dot{p}^\top]^\top$  as its state, where  $p \in \mathbb{R}^3$  is the position of the spacecraft relative to the ISO, we have that

$$f(x, \alpha, t) = \begin{bmatrix} \dot{p} \\ -m_{\text{sc}}(t)^{-1} (C(\alpha)\dot{p} + G(p, \alpha)) \end{bmatrix}, \quad B(x, \alpha, t) = \begin{bmatrix} O_{3 \times 3} \\ m_{\text{sc}}(t)^{-1} I_{3 \times 3} \end{bmatrix} \quad (9.2)$$

where  $m_{\text{sc}}(t)$  is the mass of the spacecraft described by the Tsiolkovsky rocket equation, and the functions  $G$  and  $C$  are as given in [13], [15].

**Remark 9.1.** *In general, the estimation errors  $\|\hat{x}(t) - x(t)\|$  and  $\|\hat{\alpha}(t) - \alpha(t)\|$  are expected to decrease locally with the help of the state-of-the-art onboard navigation schemes as the spacecraft gets closer to the ISO, utilizing more accurate ISO state measurements obtained from an onboard sensor as detailed in [11]. For example, if the extended Kalman filter or contraction theory-based estimator [7] is used for navigation, their expected values can be shown to be locally bounded and*

*exponentially decreasing in  $t$ . Although G&C are the focus of our study and thus developing such a navigation technique is beyond our scope, those interested in this field can also refer to, e.g., [10], [11], to tighten the estimation error bound using practical knowledge specific to the ISO dynamics. Online learning and adaptive control methods [16], [17] can also be used to actively deal with such uncertainty with some formal guarantees.*

Since the ISO state and its onboard estimate in (9.1) change dramatically in time due to their poorly constrained orbits with high inclinations and high relative velocities, using a fixed desired trajectory computed at some point earlier in time could fail to utilize the radically changing ISO state information as much as possible. Even with the use of MPC with receding horizons, the online computational load of solving it is not negligible, especially if the computational resource of the agent is limited and not sufficient to perform nonlinear optimization in real-time. Also, when we consider more challenging and highly nonlinear G&C scenarios (see Sec. 10.2), solving even a single optimization online could become unrealistic. We, therefore, construct a guidance policy to optimally achieve the smallest spacecraft delivery error for given estimated states  $\hat{x}(t)$  and  $\hat{\alpha}(t)$  in (9.1) at time  $t$ , and then design a learning-based guidance algorithm that approximates it with a verifiable optimality gap, so the spacecraft can update its desired trajectory autonomously in real-time using the most recent state estimates  $\hat{x}(t)$  and  $\hat{\alpha}(t)$ , which become more accurate as the spacecraft gets closer to the ISO as discussed in Remark 9.1. Note that the size of the neural network to be used is selected to be small enough to significantly reduce the online computational load required in solving optimization.

The major design challenge here is how we provide the learning approach with a formal guarantee for realizing the ISO encounter, i.e., achieving a sufficiently small spacecraft delivery error with respect to a given desired relative position to flyby or impact the ISO, which leads to the problem formulated as follows.

### **Autonomous terminal G&C for ISO encounter with formal guarantees**

We aim to design an autonomous nonlinear G&C algorithm that robustly tracks the desired trajectory with zero spacecraft delivery error computed by utilizing the above learning-based guidance, and that guarantees a finite tracking error bound even under the presence of the state uncertainty and the learning error.

## 9.2 Autonomous Guidance via Dynamical System-Based Deep Learning

Before proceeding to solve 9.1 in Sec. 9.1, this section describes dynamical system-based deep learning to design the autonomous terminal guidance algorithm to be used in the subsequent sections. It utilizes the known dynamics of (9.1) and (9.2) in approximating an optimization-based guidance policy, thereby directly minimizing the deviation of the learned state trajectory from the optimal state trajectory with the smallest spacecraft delivery error at the ISO encounter, possessing a verifiable optimality gap with respect to the optimal guidance policy.

### 9.2.1 Model Predictive Control Problem

Let us first introduce the following definition of an ISO state flow, which maps the ISO state at any given time to the one at time  $t$ , so we can account for the rapidly changing ISO state estimate of (9.1) in our proposed framework.

**Definition 9.1.** A flow  $\varphi^t(\alpha_0)$ , where  $\alpha_0$  is some given ISO state, defines the solution trajectory of the autonomous ISO dynamical system [13] at time  $t$ , which satisfies  $\varphi^0(\alpha_0) = \alpha_0$  at  $t = 0$ .

Utilizing the ISO flow given in Definition 9.1, we consider the following optimal guidance problem for the ISO encounter, given estimated states  $\hat{x}(\tau)$  and  $\hat{\alpha}(\tau)$  in (9.1) at  $t = \tau$ :

$$u^*(\hat{x}(\tau), \hat{\alpha}(\tau), t, \rho) = \arg \min_{u(t) \in \mathcal{U}(t)} \left( c_0 \|p_\xi(t_f) - \rho\|^2 + c_1 \int_\tau^{t_f} P(u(t), \xi(t)) dt \right) \quad (9.3)$$

$$\text{s.t. } \dot{\xi}(t) = f(\xi(t), \varphi^{t-\tau}(\hat{\alpha}(\tau)), t) + B(\xi(t), \varphi^{t-\tau}(\hat{\alpha}(\tau)), t)u(t), \quad \xi(\tau) = \hat{x}(\tau) \quad (9.4)$$

where  $t \in (\tau, t_f]$  in (9.4),  $\tau \in [0, t_f)$  is the current time at which the spacecraft solves (9.3),  $\xi$  is the fictitious spacecraft relative state of the dynamics (9.4),  $\int_\tau^{t_f} P(u(t), \xi(t)) dt$  is some performance-based cost function, such as  $L^2$  control effort and  $L^2$  trajectory tracking error, and information-based cost [18],  $p_\xi(t_f)$  is the terminal relative position of the spacecraft satisfying  $\xi(t_f) = [p_\xi(t_f)^\top, \dot{p}_\xi(t_f)^\top]^\top$ ,  $\rho$  is a mission-specific predefined terminal position relative to the ISO at given terminal time  $t_f$  ( $\rho = 0$  for impacting the ISO),  $\mathcal{U}(t)$  is a set containing admissible control inputs, and  $c_0 \in \mathbb{R}_{>0}$  and  $c_1 \in \mathbb{R}_{\geq 0}$  are the weights on each objective function. We assume that the terminal time  $t_f$  is not a decision variable but a fixed constant, as varying it is demonstrated to have a small impact on the objective value of (9.3) in our simulation setup in Sec. 10.1. Note that  $\rho$  is explicitly considered

as one of the inputs to  $u^*$  to account for the fact that it could change depending on the target ISO. Note that the policy  $u^*$  only depends on the information available at the current time  $t$  without using the information of future time steps. The future estimated states are computed by integrating the nominal dynamics as mentioned in definition 9.1.)

**Remark 9.2.** *Since it is not realistic to match the spacecraft velocity with that of the ISOs due to their high inclination nature, the terminal velocity tracking error is intentionally not included in the cost function, although it could be with an appropriate choice of  $P$  in (9.3). Also, we can set  $c_0 = 0$  and augment the problem with a constraint  $\|p_\xi(t_f) - \rho\| = 0$  if the problem is feasible with this constraint.*

Since the spacecraft relative state changes dramatically due to the ISO's high relative velocity, and the actual dynamics are perturbed by the ISO and spacecraft relative state estimation uncertainty, which decreases as  $t$  gets closer to  $t_f$  (see Remark 9.1), it is expected that the delivery error at the ISO encounter (i.e.,  $\|p_\xi(t_f) - \rho\|$ ) becomes smaller as the spacecraft solves (9.3) more frequently onboard using the updated state estimates in the initial condition (9.4) as in (9.1). More specifically, it is desirable to apply the optimal guidance policy solving (9.3) at each time instant  $t$  as follows as in model predictive control (MPC) [19]:

$$u_{\text{mpc}}(\hat{x}(t), \hat{\alpha}(t), t, \rho) = u^*(\hat{x}(t), \hat{\alpha}(t), t, \rho) \quad (9.5)$$

where  $u$  is the control input of (9.1). Note that  $\tau$  of  $u^*$  in (9.5) is now changed to  $t$  unlike (9.3), implying we only utilize the solution of (9.3) at the initial time  $t = \tau$ . Due to the predictive nature of the MPC, which leverages future predictions of the states  $x(t)$  and  $\alpha(t)$  for  $t \in [\tau, t_f]$  obtained by integrating their dynamics given  $\hat{x}(\tau)$  and  $\hat{\alpha}(\tau)$  as in (9.4), the solution of its linearized and discretized version can be shown to be near-optimal [3] in terms of dynamic regret, i.e., the MPC performance minus the optimal performance in hindsight. This would imply that the nonlinear MPC also enjoys similar optimality guarantees when it is solved by the sequential convex programming approach, which is proven to converge to a point satisfying the KKT conditions (see [20]). However, solving the nonlinear optimization problem (9.3) at each time instant to obtain (9.5) is not realistic for a spacecraft with limited computational power. Again, even when we use MPC with receding horizons, the online computational load of solving it is not negligible, especially if the computational resource of the agent is limited and not sufficient to perform nonlinear optimization in real-time. Using offline learning to replace

online optimization could enable our approach applicable also to more challenging and highly nonlinear G&C scenarios (see Sec. 10.2), where solving even a single optimization online could become unrealistic.

### 9.2.II Imitation Learning of MPC State and Control Trajectories

In order to compute the MPC of (9.5) in real-time with a verifiable optimality guarantee, our proposed learning-based terminal guidance policy models it using an SN-DNN [21], which is a deep neural network constructed to be Lipschitz continuous and robust to input perturbation by design (see, e.g., Lemma 6.2 of [7]). Note that the size of the neural network to be used is selected to be small enough to significantly reduce the online computational load required in solving optimization.

Let  $u_\ell(\hat{x}(t), \hat{\alpha}(t), t, \rho; \theta_{\text{nn}})$  denote the proposed learning-based terminal guidance policy, which models the MPC policy  $u_{\text{mpc}}(\hat{x}(t), \hat{\alpha}(t), t, \rho)$  of (9.5) using the SN-DNN, where  $\theta_{\text{nn}}$  is its hyperparameter. The following definition of the process induced by the spacecraft dynamics with  $u_{\text{mpc}}$  and  $u_\ell$ , which map the ISO and spacecraft relative state at any given time to their respective spacecraft relative state at time  $t$ , is useful for simplifying notation in our framework.

**Definition 9.2.** *Mappings denoted as  $\varphi_\ell^t(x_\tau, \alpha_\tau, \tau, \rho; \theta_{\text{nn}})$  and  $\varphi_{\text{mpc}}^t(x_\tau, \alpha_\tau, \tau, \rho)$  (called processes [22, p. 24]) define the solution trajectories of the following non-autonomous dynamical systems at time  $t$ , controlled by the SN-DNN and MPC policy, respectively, both with the initial condition  $\xi(\tau) = x_\tau$ :*

$$\dot{\xi}(t) = f(\xi(t), \varphi^{t-\tau}(\alpha_\tau), t) + B(\xi(t), \varphi^{t-\tau}(\alpha_\tau), t)u_\ell(\xi(t), \varphi^{t-\tau}(\alpha_\tau), t, \rho; \theta_{\text{nn}}) \quad (9.6)$$

$$\dot{\xi}(t) = f(\xi(t), \varphi^{t-\tau}(\alpha_\tau), t) + B(\xi(t), \varphi^{t-\tau}(\alpha_\tau), t)u_{\text{mpc}}(\xi(t), \varphi^{t-\tau}(\alpha_\tau), t, \rho) \quad (9.7)$$

where  $\tau \in [0, t_f]$ ,  $t_f$  and  $\rho$  are the given terminal time and relative position at the ISO encounter as in (9.4),  $\alpha_\tau$  and  $x_\tau$  are some given ISO and spacecraft relative state at time  $t = \tau$ , respectively,  $f$  and  $B$  are given in (9.1) and (9.2), and  $\varphi^{t-\tau}(\alpha_\tau)$  is the ISO state trajectory with  $\varphi^0(\alpha_\tau) = \alpha_\tau$  at  $t = \tau$  as given in Definition 9.1.

Let  $(\bar{x}, \bar{\alpha}, \bar{t}, \bar{\rho}, \Delta\bar{t})$  denote a sampled data point for the spacecraft state, ISO state, current time, desired terminal relative position, and time of integration to be used in (9.8), respectively. Also, let  $\text{Unif}(\mathcal{S})$  be the uniform distribution over a compact set  $\mathcal{S}$ , which produces  $(\bar{x}, \bar{\alpha}, \bar{t}, \bar{\rho}, \Delta\bar{t}) \sim \text{Unif}(\mathcal{S})$ . Using Definition 9.2, we introduce

the following new loss function to be minimized by optimizing the hyperparameter  $\theta_{\text{nn}}$  of the SN-DNN guidance policy  $u_\ell(\hat{x}(t), \hat{a}(t), t, \rho; \theta_{\text{nn}})$ :

$$\begin{aligned} \mathcal{L}_{\text{nn}}(\theta_{\text{nn}}) = \mathbb{E} \left[ & \|u_\ell(\bar{x}, \bar{a}, \bar{t}, \bar{\rho}; \theta_{\text{nn}}) - u_{\text{mpc}}(\bar{x}, \bar{a}, \bar{t}, \bar{\rho})\|_{C_u}^2 \right. \\ & \left. + \|\varphi_\ell^{\bar{t}+\Delta\bar{t}}(\bar{x}, \bar{a}, \bar{t}, \bar{\rho}; \theta_{\text{nn}}) - \varphi_{\text{mpc}}^{\bar{t}+\Delta\bar{t}}(\bar{x}, \bar{a}, \bar{t}, \bar{\rho})\|_{C_x}^2 \right] \end{aligned} \quad (9.8)$$

where  $\|(\cdot)\|_{C_x}$  and  $\|(\cdot)\|_{C_u}$  are the weighted Euclidean 2-norm given as  $\|(\cdot)\|_{C_u}^2 = (\cdot)^\top C_u (\cdot)$  and  $\|(\cdot)\|_{C_x}^2 = (\cdot)^\top C_x (\cdot)$  for symmetric positive definite weight matrices  $C_u, C_x > 0$ , and  $\varphi_\ell^t(\bar{x}, \bar{a}, \bar{t}, \bar{\rho}; \theta_{\text{nn}})$  and  $\varphi_{\text{mpc}}^t(\bar{x}, \bar{a}, \bar{t}, \bar{\rho})$  are the solution trajectories with the SN-DNN and MPC guidance policy given in Definition 9.2, respectively. As illustrated in Fig. 9.3, we train the SN-DNN to also minimize the deviation of the state trajectory with the SN-DNN guidance policy from the desired MPC state trajectory [4], [5]. The learning objective is thus not just to minimize  $\|u_\ell - u_{\text{mpc}}\|$ , but to imitate the optimal trajectory with the smallest spacecraft delivery error at the ISO encounter. We will see how the selection of the weights  $C_u$  and  $C_x$  affects the performance of  $u_\ell$  in Sec. 10.1.

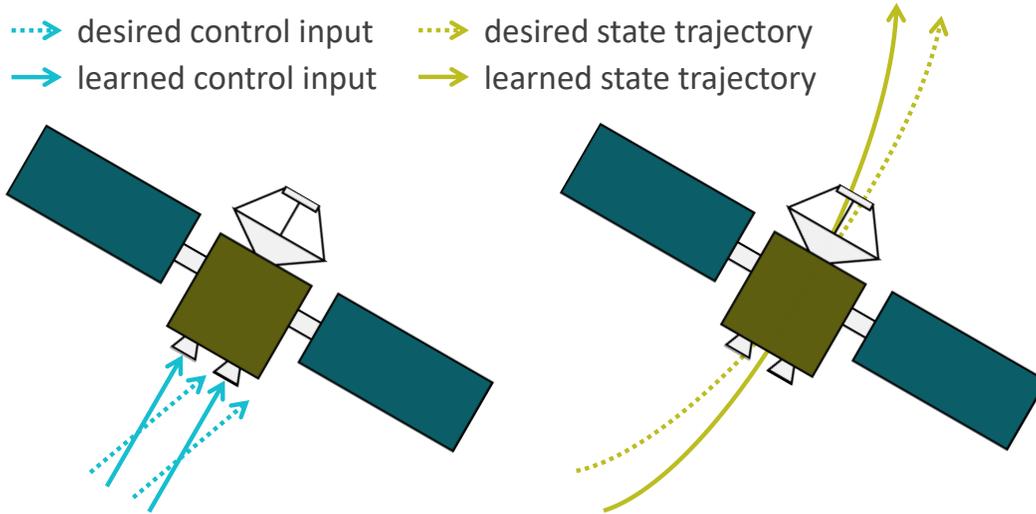


Figure 9.3: Neural net loss functions. Left: SN-DNN trained to imitate a desired control input (first term of (9.8)); right: SN-DNN trained to imitate a desired state trajectory (second term of (9.8)).

**Remark 9.3.** As in standard learning algorithms for neural networks including stochastic gradient descent (SGD), the expectation of the loss function (9.8) can be approximated using sampled data points as follows:

$$\begin{aligned} \mathcal{L}_{\text{emp}}(\theta_{\text{nn}}) = & \sum_{i=1}^{N_d} \left( \|u_{\ell}(\bar{x}_i, \bar{\alpha}_i, \bar{t}_i, \bar{\rho}_i; \theta_{\text{nn}}) - u_{\text{mpc}}(\bar{x}_i, \bar{\alpha}_i, \bar{t}_i, \bar{\rho}_i)\|_{C_u}^2 \right. \\ & \left. + \|\varphi_{\ell}^{\bar{t}_i + \Delta \bar{t}_i}(\bar{x}_i, \bar{\alpha}_i, \bar{t}_i, \bar{\rho}_i; \theta_{\text{nn}}) - \varphi_{\text{mpc}}^{\bar{t}_i + \Delta \bar{t}_i}(\bar{x}_i, \bar{\alpha}_i, \bar{t}_i, \bar{\rho}_i)\|_{C_x}^2 \right) \end{aligned} \quad (9.9)$$

where the training data points  $\{(\bar{x}_i, \bar{\alpha}_i, \bar{t}_i, \bar{\rho}_i, \Delta \bar{t}_i)\}_{i=1}^{N_d}$  are drawn independently from  $\text{Unif}(\mathcal{S})$ .

### 9.2.III Optimality Gap of Deep Learning-Based Guidance

Since an SN-DNN is Lipschitz bounded by design and robust to perturbation, the optimality gap of the guidance framework  $u_{\ell}$  introduced in Sec. 9.2.II can be bounded as in the following theorem, where the notations are summarized in Table 9.1 and the proof concept is illustrated in Fig. 9.4.

Table 9.1: Notations in Lemma 9.1.

Notation	Description
$L_{\ell}$	2-norm Lipschitz constant of $u_{\ell}$ in $\mathbb{R}_{>0}$ , guaranteed to exist by design
$N_d$	Number of training data points
$\mathcal{S}_{\text{test}}$	Any compact test set containing $(x, \alpha, t, \rho)$ , not necessarily the training set $\mathcal{S}_{\text{train}}$ itself
$u_{\ell}$	Learning-based terminal guidance policy that models $u_{\text{mpc}}$ by the SN-DNN using the loss function given in (9.8)
$u_{\text{mpc}}$	Optimal MPC terminal guidance policy given in (9.5)
$(x, \alpha, t, \rho)$	Test data point for the S/C relative state, ISO state, current time, and desired terminal relative position of (9.4), respectively
$(\bar{x}_i, \bar{\alpha}_i, \bar{t}_i, \bar{\rho}_i)$	Training data point for the S/C relative state, ISO state, current time, and desired terminal relative position of (9.4), respectively, where $i \in \mathbb{N} \cup [1, N_d]$
$\Pi_{\text{train}}$	Training dataset containing a finite number of training data points, i.e., $\Pi_{\text{train}} = \{(\bar{x}_i, \bar{\alpha}_i, \bar{t}_i, \bar{\rho}_i)\}_{i=1}^{N_d}$

**Lemma 9.1.** *Suppose that  $u_{\text{mpc}}$  is Lipschitz with its 2-norm Lipschitz constant  $L_{\text{mpc}} \in \mathbb{R}_{>0}$ . If  $u_\ell$  is trained using the empirical loss function (9.9) of Remark 9.3 to have  $\exists \epsilon_{\text{train}} \in \mathbb{R}_{\geq 0}$  s.t.*

$$\sup_{i \in \mathbb{N} \cup [1, N_d]} \|u_\ell(\bar{x}_i, \bar{\alpha}_i, \bar{t}_i, \bar{\rho}_i; \theta_{\text{nn}}) - u_{\text{mpc}}(\bar{x}_i, \bar{\alpha}_i, \bar{t}_i, \bar{\rho}_i)\| \leq \epsilon_{\text{train}} \quad (9.10)$$

then we have the following bound  $\forall (x, \alpha, t, \rho) \in \mathcal{S}_{\text{test}}$ :

$$\|u_\ell(x, \alpha, t, \rho; \theta_{\text{nn}}) - u_{\text{mpc}}(x, \alpha, t, \rho)\| \leq \epsilon_{\text{train}} + r(x, \alpha, t, \rho)(L_\ell + L_{\text{mpc}}) = \epsilon_{\ell u} \quad (9.11)$$

where  $r(x, \alpha, t, \rho) = \inf_{i \in \mathbb{N} \cup [1, N_d]} \sqrt{\|\bar{x}_i - x\|^2 + \|\bar{\alpha}_i - \alpha\|^2 + (\bar{t}_i - t)^2 + \|\bar{\rho}_i - \rho\|^2}$ .

*Proof.* Let  $\eta = (x, \alpha, t, \rho)$  be a test element in  $\mathcal{S}_{\text{test}}$  (i.e.,  $\eta \in \mathcal{S}_{\text{test}}$ ) and let  $\bar{\zeta}_j(\eta)$  be the training data point in  $\Pi_{\text{train}}$  (i.e.,  $\bar{\zeta}_j(\eta) \in \Pi_{\text{train}}$ ) that achieves the infimum of  $r(\eta)$  with  $i = j$  for a given  $\eta \in \mathcal{S}_{\text{test}}$ . Since  $u_\ell$  and  $u_{\text{mpc}}$  are Lipschitz by design and by assumption, respectively, we have for any  $\eta \in \mathcal{S}_{\text{test}}$  that

$$\begin{aligned} \|u_\ell(\eta; \theta_{\text{nn}}) - u_{\text{mpc}}(\eta)\| &\leq \|u_\ell(\bar{\zeta}_j(\eta); \theta_{\text{nn}}) - u_{\text{mpc}}(\bar{\zeta}_j(\eta))\| + r(\eta)(L_\ell + L_{\text{mpc}}) \\ &\leq \epsilon_{\text{train}} + r(\eta)(L_\ell + L_{\text{mpc}}) \end{aligned}$$

where the first inequality follows from the definition of  $\bar{\zeta}_j(\eta)$  and the second inequality follows from (9.10) and the fact that  $\bar{\zeta}_j(\eta) \in \Pi_{\text{train}}$ . This relation leads to the desired result (9.11) as it holds for any  $\eta \in \mathcal{S}_{\text{test}}$ .  $\square$

The SN-DNN provides a verifiable optimality gap even for data points not in its training set, which indicates that it still benefits from the near-optimal guarantee of the MPC in terms of dynamic regret as discussed below (9.5). As illustrated in Fig. 9.4, each term in the optimality gap (9.11) can be interpreted as follows:

1.  $\epsilon_{\text{train}}$  of (9.10) is the training error of the SN-DNN, expected to decrease as the learning proceeds using SGD.
2.  $r(\eta)$  is the closest distance from the test element  $\eta \in \mathcal{S}_{\text{test}}$  to a training data point in  $\Pi_{\text{train}}$ , expected to decrease as the number of data points  $N_d$  in  $\Pi_{\text{train}}$  increases and as the training set  $\mathcal{S}_{\text{train}}$  gets larger.
3. The Lipschitz constant  $L_\ell$  is a design parameter we could arbitrarily choose when constructing the SN-DNN (it is Lipschitz *by design* by spectrally normalizing the weights of the neural network [21], we can freely choose  $L_\ell$  independently of the choice of the neural net parameters).

4. We could also treat  $L_{\text{mpc}}$  as a design parameter by adding a Lipschitz constraint,  $\|\partial u_{\text{mpc}}/\partial \eta\| \leq L_{\text{mpc}}$ , in solving (9.3).

Note that  $\epsilon_{\text{train}}$  and  $r(\eta)$  can always be computed numerically for a given  $\eta$  as the dataset  $\Pi_{\text{train}}$  only has a finite number of data points.

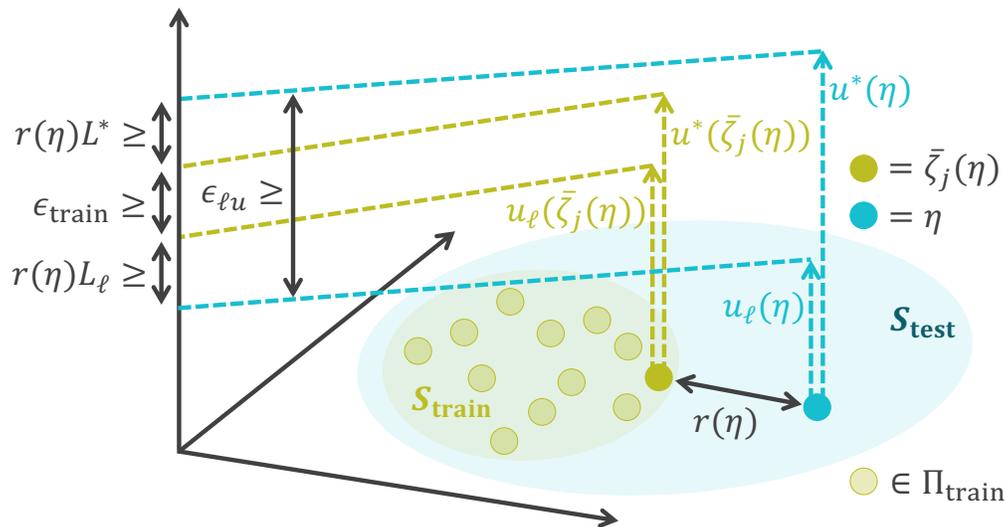


Figure 9.4: Illustration of the optimality gap (9.11) in Lemma 9.1, where  $S_{\text{train}}$  denotes the training set in which training data  $(\bar{x}_i, \bar{a}_i, \bar{t}_i, \bar{\rho}_i) \in \Pi_{\text{train}}$  is sampled.

The obvious downsides of the optimality gap (9.11) are that it still suffers the generalization issue, which is common in the field of machine learning, and that we may have to find the region that the MPC is locally Lipschitz empirically. However, at least Lemma 9.1 explicitly describes all the assumptions we need to obtain the bound on the learning error, without treating it like a black box to be validated in the experimental results implicitly. Also in practice, a smaller Lipschitz constant  $L_\ell$  could lead to a larger training error  $\epsilon_\ell$ , requiring some tuning in using it in real-world simulations (see Sec. 10.1 and Sec. 10.2).

**Remark 9.4.** *Obtaining a tighter and more general optimality gap of the learned control and state trajectories has been an active field of research. Particularly for a neural network equipped with a dynamical structure as in our proposed approach, we could refer to generalization bounds for the neural ordinary differential equations [4], or use systems and control theoretical methods to augment it with stability and robustness properties [23], [24]. We could also consider combining the SN-DNN with neural networks that have enhanced structural guarantees of robustness, including robust implicit networks [25] and robust equilibrium networks [26].*

The optimality gap discussed in Lemma 9.1 and in Remark 9.4 is useful in that they provide mathematical guarantees for learning-based frameworks only with the Lipschitz assumption (e.g., it could prove safety in the learning-based MPC framework [6]). However, when the system is perturbed by the state uncertainty as in (9.1), we could only guarantee that the distance between the state trajectories controlled by  $u_{\text{mpc}}$  of (9.5) and  $u_\ell$  of Lemma 9.1 is bounded by a function that increases exponentially with time [7]. In Sec. 9.3 and 9.4, we will see how such a conservative bound can be replaced by a decreasing counterpart.

### 9.3 Deep learning-Based Optimal Tracking Control

This section proposes a feedback control policy to be used on top of the SN-DNN terminal guidance policy  $u_\ell$  of Lemma 9.1 in Sec. 9.2. In particular, we utilize  $u_\ell$  to design the desired trajectory with zero spacecraft delivery error and then construct pointwise optimization-based tracking control with a Lyapunov stability condition, which can be shown to have an analytical solution for real-time implementation. In Sec. 9.4, we will see that this framework plays an essential part in solving our main problem 9.1 of Sec. 9.1.

**Remark 9.5.** *For notational convenience, we drop the dependence on the desired terminal relative position  $\rho$  of (9.4) in  $u_\ell$ ,  $u_{\text{mpc}}$ ,  $\varphi_\ell^t$ , and  $\varphi_{\text{mpc}}^t$  in the subsequent sections, as it is time-independent and thus does not affect the arguments to be made in the following. Note that the SN-DNN is still to be trained regarding  $\rho$  as one of its inputs, so we do not have to retrain SN-DNNs every time we change  $\rho$ .*

#### 9.3.1 Desired State Trajectory with Zero Delivery Error

Let us recall the definitions of the ISO state flow and spacecraft state processes of Definitions 9.1 and 9.2 summarized in Table 9.2. Ideally at some given time  $t = t_d \in [0, t_f)$  when the spacecraft possesses enough information on the ISO, we would like to construct a desired trajectory using the estimated states  $\hat{x}(t_d)$  and  $\hat{x}(t_d)$  s.t. it ensures zero spacecraft delivery error at the ISO encounter assuming zero state estimation errors for  $t \in [t_d, t_f]$  in (9.1). As in trajectory generation in space and aerial robotics [27], [28], this can be achieved by obtaining a desired state trajectory as  $\varphi_\ell^t(x_f, \alpha_f, t_f)$  of Table 9.2 with  $\alpha_f = \varphi^{t_f - t_d}(\hat{x}(t_d))$ , i.e., by solving the

following dynamics with the SN-DNN terminal guidance policy  $u_\ell$  (9.6) backward in time:

$$\begin{aligned} \dot{\xi}(t) &= f(\xi(t), \varphi^{t-t_d}(\hat{\alpha}(t_d)), t) + B(\xi(t), \varphi^{t-t_d}(\hat{\alpha}(t_d)), t)u_{bw}, \quad \xi(t_f) = x_f \quad (9.12) \\ u_{bw} &= u_\ell(\xi(t), \varphi^{t-t_d}(\hat{\alpha}(t_d)), t; \theta_{nn}) \end{aligned}$$

where the property of the ISO solution flow in Table 9.2 introduced in Definition 9.1,  $\varphi^{t-t_f}(\alpha_f) = \varphi^{t-t_f}(\varphi^{t_f-t_d}(\alpha(t_d))) = \varphi^{t-t_d}(\alpha(t_d))$ , is used to get (9.12),  $t_f$  is the given terminal time at the ISO encounter as in (9.4), and the ideal spacecraft terminal relative state  $x_f$  is defined as

$$x_f = \begin{bmatrix} \rho \\ C_{s2v} \varphi_\ell^{t_f}(\hat{x}(t_d), \hat{\alpha}(t_d), t_d) \end{bmatrix} \quad (9.13)$$

where  $\rho$  is the desired terminal relative position given in (9.4),  $\varphi_\ell^{t_f}(\hat{x}(t_d), \hat{\alpha}(t_d), t_d)$  is the spacecraft relative state at  $t = t_f$  obtained by integrating the dynamics forward as in Table 9.2, and  $C_{s2v} = [O_{3 \times 3} \ I_{3 \times 3}] \in \mathbb{R}^{3 \times 6}$  is a matrix that maps the spacecraft relative state to its velocity vector. Figure 9.5 illustrates the construction of such a desired trajectory.

Table 9.2: Summary of the ISO state flow and spacecraft state processes in Definitions 9.1 and 9.2, where  $u_\ell$  and  $u_{\text{mpc}}$  are the SN-DNN and MPC guidance policies, respectively. Note that the dependence on the terminal position  $\rho$  is omitted as explained in Remark 9.5.

Notation	Description
$\varphi^{t-\tau}(\alpha_\tau)$	Solution trajectory of the ISO dynamics at time $t$ which satisfies $\varphi^0(\alpha_\tau) = \alpha_\tau$
$\varphi_\ell^t(x_\tau, \alpha_\tau, \tau; \theta_{nn})$	Solution trajectory of the S/C relative dynamics at time $t$ , controlled by $u_\ell$ with no state estimation error, which satisfies $\varphi_\ell^\tau(x_\tau, \alpha_\tau, \tau; \theta_{nn}) = x_\tau$ at $t = \tau$ and $\alpha(t) = \varphi^{t-\tau}(\alpha_\tau)$
$\varphi_{\text{mpc}}^t(x_\tau, \alpha_\tau, \tau)$	Solution trajectory of the S/C relative dynamics at time $t$ , controlled by $u_{\text{mpc}}$ with no state estimation error, which satisfies $\varphi_{\text{mpc}}^\tau(x_\tau, \alpha_\tau, \tau) = x_\tau$ at $t = \tau$ and $\alpha(t) = \varphi^{t-\tau}(\alpha_\tau)$

**Remark 9.6.** Although the desired state trajectory design depicted in Fig. 9.5 involves backward and forward integration (9.12) and (9.13) as in Definition 9.2, which is common in trajectory generation in space and aerial robotics [27], [28], the SN-DNN approximation of the MPC policy allows performing it within a short

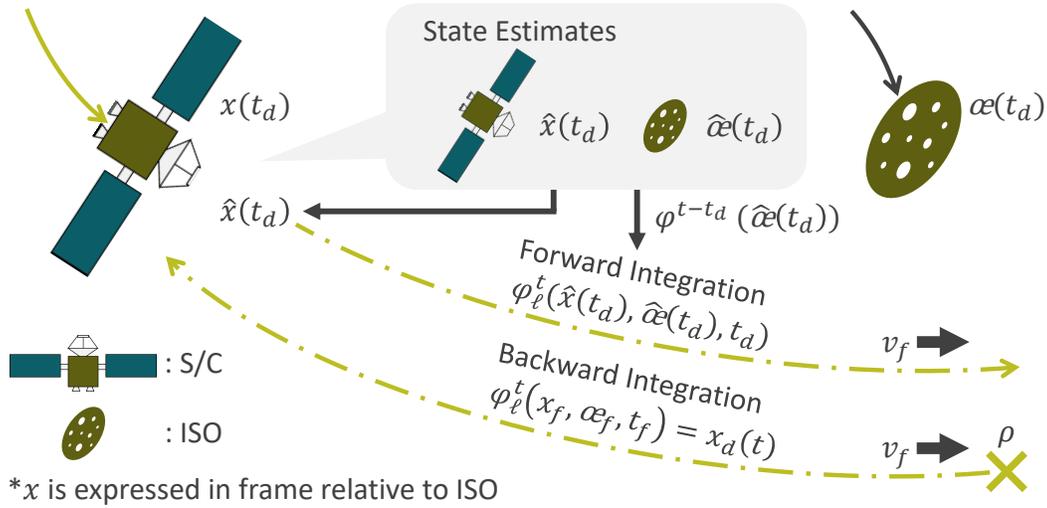


Figure 9.5: Illustration of the desired trajectory discussed in Sec. 9.3.I, where  $v_f = C_{s2v} \varphi_\ell^{t_f}(\hat{x}(t_d), \hat{\alpha}(t_d), t_d)$  as in (9.13). Spacecraft obtains the desired trajectory  $x_d$  of (9.14) as a result of backward integration (see Remark 9.6).

period. In fact, when measured using the Mid 2015 MacBook Pro laptop, it takes only about  $3.0 \times 10^{-4}$  s for numerically integrating the dynamics for one step using the fourth-order Runge–Kutta method (e.g., it takes  $\sim 3$  s to get the desired trajectory for  $t_d = 0$  (s) and  $t_f = 10000$  (s) with the discretization time step 1 s). The computational time should decrease as  $t_d$  becomes larger. Section 9.4.II delineates how we utilize such a desired trajectory in real-time with the feedback control to be introduced in Sec. 9.3.II.

### 9.3.II Deep Learning-Based Pointwise Min-Norm Control

Using the results given in Sec. 9.3.I, we design pointwise optimal tracking control resulting in verifiable spacecraft delivery error, even under the presence of state uncertainty. For notational simplicity, let us denote the desired spacecraft relative state and ISO state trajectories of (9.12), constructed using  $\hat{x}(t_d)$  and  $\hat{\alpha}(t_d)$  at  $t = t_d$  with the terminal state (9.13) as illustrated in Fig. 9.5, as follows:

$$x_d(t) = \varphi_\ell^t(x_f, \alpha_f, t_f), \quad \alpha_d(t) = \varphi^{t-t_d}(\hat{\alpha}(t_d)). \quad (9.14)$$

Also, assuming that we select  $x$  as  $x = [p^\top, \dot{p}^\top]^\top$  as the state of (9.1) as in (9.2), where  $p \in \mathbb{R}^3$  is the position of the spacecraft relative to the ISO, let  $\mathcal{f}(x, \alpha, t)$  and  $\mathcal{B}(x, \alpha, t)$  be defined as follows:

$$\mathcal{f}(x, \alpha, t) = C_{s2v} f(x, \alpha, t), \quad \mathcal{B}(x, \alpha, t) = C_{s2v} B(x, \alpha, t), \quad (9.15)$$

where  $f$  and  $B$  are given in (9.1) and  $C_{s2v} = [O_{3 \times 3} \ I_{3 \times 3}] \in \mathbb{R}^{3 \times 6}$  as in (9.13). Note that the relations (9.15) imply

$$\ddot{p} = \ell(x, \alpha, t) + \mathcal{B}(x, \alpha, t)u(\hat{x}, \hat{\alpha}, t). \quad (9.16)$$

Given the desired trajectory (9.14),  $x_d = [p_d^\top, \dot{p}_d^\top]^\top$ , which achieves zero spacecraft delivery error at the ISO encounter when the state estimation errors are zero for  $t \in [t_d, t_f]$  in (9.1), we design a controller  $u$  of (9.1) and (9.16) as follows, as considered in [7], [29] for general nonlinear systems:

$$u_\ell^*(\hat{x}, \hat{\alpha}, t; \theta_{nn}) = u_\ell(x_d(t), \alpha_d(t), t; \theta_{nn}) + k(\hat{x}, \hat{\alpha}, t) \quad (9.17)$$

$$k(\hat{x}, \hat{\alpha}, t) = \begin{cases} 0 & \text{if } \Upsilon(\hat{x}, \hat{\alpha}, t) \leq 0 \\ -\frac{\Upsilon(\hat{x}, \hat{\alpha}, t)(\dot{\hat{p}} - \varrho_d(\hat{p}, t))}{\|\dot{\hat{p}} - \varrho_d(\hat{p}, t)\|^2} & \text{otherwise} \end{cases} \quad (9.18)$$

with  $\Upsilon$  and  $\varrho_d$  defined as

$$\begin{aligned} \Upsilon(\hat{x}, \hat{\alpha}, t) &= (\dot{\hat{p}} - \varrho_d(\hat{p}, t))^\top M(t) \{ \ell(\hat{x}, \hat{\alpha}, t) - \ell(x_d(t), \alpha_d(t), t) \\ &\quad + \Lambda(\dot{\hat{p}} - \dot{p}_d(t)) + \alpha(\dot{\hat{p}} - \varrho_d(\hat{p}, t)) \} \\ \varrho_d(\hat{p}, t) &= -\Lambda(\hat{p} - p_d(t)) + \dot{p}_d(t) \end{aligned} \quad (9.19)$$

where  $\hat{p} \in \mathbb{R}^3$  is the estimated position of the spacecraft relative to the ISO s.t.  $\hat{x} = [\hat{p}^\top, \dot{\hat{p}}^\top]^\top$ ,  $u_\ell$  is the SN-DNN terminal guidance policy of Lemma 9.1,  $\alpha_d$ ,  $\ell$ , and  $\mathcal{B}$  are given in (9.14) and (9.15),  $\Lambda > 0$  is a given symmetric positive definite matrix,  $\alpha \in \mathbb{R}_{>0}$  is a given positive constant, and  $M(t) = m_{sc}(t)I_{3 \times 3}$  for the spacecraft mass  $m_{sc}(t)$  given in (9.2). This control policy can be shown to possess the following pointwise optimality property, in addition to the robustness and stability guarantees to be seen in Sec. 9.4. Note that (9.17) is well-defined even with the division by  $\|\dot{\hat{p}} - \varrho_d(\hat{p}, t)\|$  because we have  $\Upsilon(\hat{x}, \hat{\alpha}, t) = 0$  when  $\|\dot{\hat{p}} - \varrho_d(\hat{p}, t)\| = 0$  and thus  $k(\hat{x}, \hat{\alpha}, t) = 0$  in this case.

Consider the following optimization problem, which computes an optimal control input that minimizes its instantaneous deviation from that of the SN-DNN guidance policy at each time instant, under an incremental Lyapunov stability condition:

$$u_\ell^*(\hat{x}, \hat{\alpha}, t; \theta_{nn}) = \arg \min_{u \in \mathbb{R}^m} \|u - u_\ell(x_d(t), \alpha_d(t), t; \theta_{nn})\|^2 \quad (9.20)$$

$$\begin{aligned} \text{s.t. } & \frac{\partial V}{\partial \dot{\hat{p}}}(\dot{\hat{p}}, \varrho_d(\hat{p}, t), t)(\ell(\hat{x}, \hat{\alpha}, t) + \mathcal{B}(\hat{x}, \hat{\alpha}, t)u) \\ & + \frac{\partial V}{\partial \varrho_d}(\dot{\hat{p}}, \varrho_d(\hat{p}, t), t)(\dot{p}_d(t) - \Lambda(\dot{\hat{p}} - \dot{p}_d(t))) \leq -2\alpha V(\dot{\hat{p}}, \varrho_d(\hat{p}, t), t) \end{aligned} \quad (9.21)$$

where  $V$  is a non-negative function defined as

$$V(\dot{\hat{p}}, \varrho_d, t) = (\dot{\hat{p}} - \varrho_d)^\top M(t)(\dot{\hat{p}} - \varrho_d) \quad (9.22)$$

and the other notation is as given in (9.17).

**Lemma 9.2.** *The optimization problem (9.20) is always feasible, and the controller (9.17) defines its analytical optimal solution for the spacecraft relative dynamical system (9.2). Furthermore, substituting  $u = u_\ell(x_d(t), \alpha_d(t), t; \theta_{nn})$  into (9.21) yields  $\Upsilon(\hat{x}, \hat{\alpha}, t) \leq 0$ , which implies the controller (9.17) modulates the desired input,  $u_\ell(x_d(t), \alpha_d(t), t; \theta_{nn})$ , only when necessary to ensure the stability condition (9.21).*

*Proof.* Let  $u_n(\hat{x}, \hat{\alpha}, t)$  be defined as follows:

$$u_n(\hat{x}, \hat{\alpha}, t) = M(t)\dot{\varrho}_d(\hat{p}, t) + C(\hat{\alpha})\varrho_d(\hat{p}, t) + G(\hat{p}, \hat{\alpha}) - \alpha M(t)(\dot{\hat{p}} - \varrho_d(\hat{p}, t))$$

where  $\varrho_d$  is as given in (9.19). Substituting this into the left-hand side of (9.21) as  $u = u_n(\hat{x}, \hat{\alpha}, t)$  gives

$$2(\dot{\hat{p}} - \varrho_d(\hat{p}, t))^\top (-C(\hat{\alpha}) - \alpha M(t))(\dot{\hat{p}} - \varrho_d(\hat{p}, t)) = -2\alpha V(\dot{\hat{p}}, \varrho_d(\hat{p}, t), t)$$

where  $\ell$  of (9.15) is computed with (9.2), and the equality follows from the skew-symmetric property of  $C$ , i.e.,  $C(\hat{\alpha}) + C(\hat{\alpha})^\top = O_{3 \times 3}$  [15]. The relation indeed indicates that  $u = u_n(\hat{x}, \hat{\alpha}, t)$  is a feasible solution to the optimization problem (9.20). Furthermore, applying the KKT condition to (9.20) yields (9.17). Also, we get the condition  $\Upsilon(\hat{x}, \hat{\alpha}, t) \leq 0$  by substituting  $V$  of (9.22) and  $\ddot{p}$  into the stability condition (9.21) with  $u = u_\ell(x_d(t), \alpha_d(t), t; \theta_{nn})$ .  $\square$

Let us emphasize again that, as proven in Lemma 9.2, the deviation term  $k(\hat{x}, \hat{\alpha}, t)$  of the controller (9.17) is non-zero only when the stability condition (9.21) cannot be satisfied with the SN-DNN terminal guidance policy  $u_\ell$  of Lemma 9.1. This result can be viewed as an extension of the min-norm control methodology of [29] for control-affine nonlinear systems, and in Lemma 9.2, the Lagrangian system-type structure of the spacecraft relative dynamics is extensively used to obtain the analytical solution (9.17) of the quadratic optimization problem (9.20), for the sake of its real-time implementation.

### 9.3.III Assumptions for Robustness and Stability

Before proceeding to the next section on proving the robustness and stability properties of the SN-DNN min-norm control of (9.17) of Lemma 9.2, let us make a few assumptions with the notations given in Table 9.3, the first of which is that the spacecraft has access to an on-board navigation scheme that satisfies the following conditions. Due to the system nonlinearity, we can only assume the following only *locally* (see (9.23)), which is one of the reasons why we need to construct a non-negative function with a non-negative supermartingale property later in Theorem 9.1.

Table 9.3: Notations in Sec. 9.3.III.

Notation	Description
$C_{\text{iso}}(r), C_{\text{sc}}(r)$	Tubes of radius $r \in \mathbb{R}_{>0}$ centered around the desired trajectories $x_d$ and $a_d$ given in (9.14), i.e., $C_{\text{iso}}(r) = \bigcup_{t \in [t_s, t_f]} \{\xi \in \mathbb{R}^n \mid \ \xi - a_d(t)\  < r\} \subset \mathbb{R}^n$ and $C_{\text{sc}}(r) = \bigcup_{t \in [t_s, t_f]} \{\xi \in \mathbb{R}^n \mid \ \xi - x_d(t)\  < r\} \subset \mathbb{R}^n$
$\mathcal{E}_{\text{iso}}, \mathcal{E}_{\text{sc}}$	Subsets of $\mathbb{R}^n$ that have the ISO and S/C state estimation error vectors $\ \hat{a}(t) - a(t)\ $ and $\ \hat{x}(t) - x(t)\ $ , respectively, where a given on-board navigation scheme is valid (e.g., region of attraction)
$\mathbb{E}_{Z_1}[\cdot]$	Conditional expected value operator s.t. $\mathbb{E}[\cdot \mid x(t_1) = x_1, \hat{x}(t_1) = \hat{x}_1, a(t_1) = a_1, \hat{a}(t_1) = \hat{a}_1]$
$t_f$	Given terminal time at the ISO encounter as in (9.4)
$t_s$	Time when the S/C activates the SN-DNN min-norm control policy (9.17) of Lemma 9.2
$Z_1$	Tuple of the true and estimated ISO and S/C state at time $t = t_1$ , i.e., $Z_1 = (a_1, \hat{a}_1, x_1, \hat{x}_1)$
$\varsigma^t(Z_1, t_1)$	Expected estimation error upper bound at time $t = t_1$ given $Z_1$ at time $t = t_1$ , which can be determined based on the choice of navigation techniques (see Remark 9.1)

**Assumption 9.1.** *Let the probability of the error vectors remaining in  $\mathcal{E}_{\text{iso}}$  and  $\mathcal{E}_{\text{sc}}$  be bounded as follows for  $\exists \varepsilon_{\text{est}} \in \mathbb{R}_{\geq 0}$ :*

$$\mathbb{P} \left[ \bigcap_{t \in [0, t_f]} (\hat{a}(t) - a(t)) \in \mathcal{E}_{\text{iso}} \cap (\hat{x}(t) - x(t)) \in \mathcal{E}_{\text{sc}} \right] \geq 1 - \varepsilon_{\text{est}}. \quad (9.23)$$

*We assume that if the event of (9.23) has occurred, then we have the following bound for any  $t_1, t_2 \in [0, t_f]$  s.t.  $t_1 \leq t_2$ :*

$$\mathbb{E}_{Z_1} \left[ \sqrt{\|\hat{a}(t) - a(t)\|^2 + \|\hat{x}(t) - x(t)\|^2} \right] \leq \varsigma^t(Z_1, t_1), \quad \forall t \in [t_1, t_2]. \quad (9.24)$$

We also assume that given the 2-norm estimation error satisfies

$$\sqrt{\|\hat{\alpha}(t) - \alpha(t)\|^2 + \|\hat{x}(t) - x(t)\|^2} < c_e$$

for  $c_e \in \mathbb{R}_{\geq 0}$  at time  $t = t_s$ , then it satisfies this bound for  $\forall t \in [t_s, t_f]$  with probability at least  $1 - \varepsilon_{\text{err}}$ , where  $\exists \varepsilon_{\text{err}} \in \mathbb{R}_{\geq 0}$ .

If the extended Kalman filter or contraction theory-based estimator [7] is used for navigation with disturbances expressed as the Gaussian white noise processes, then we have  $\zeta^t(Z_1, t_1) = e^{-\beta(t-t_1)} \sqrt{\|\hat{\alpha}_1 - \alpha_1\|^2 + \|\hat{x}_1 - x_1\|^2} + c$ , where  $Z_1 = (\alpha_1, \hat{\alpha}_1, x_1, \hat{x}_1)$  and  $\beta$  and  $c$  are some given positive constants (see Example 9.1). The last statement of the boundedness of the estimation error is expected for navigation schemes resulting in a decreasing estimation error, and can be shown formally using Ville's maximal inequality for supermartingales [30, pp. 79-83] with an appropriate Lyapunov-like function for navigation synthesis [30, pp. 79-83] (see Lemma 9.3 for similar computation in control synthesis). Note that if  $\mathcal{E}_{\text{iso}} = \mathcal{E}_{\text{sc}} = \mathbb{R}^n$ , i.e., the bound (9.24) holds globally, then we have  $\varepsilon_{\text{est}} = 0$ . Let us further make the following assumption on the SN-DNN min-norm control policy (9.17).

**Assumption 9.2.** We assume that  $k$  of (9.18) is locally Lipschitz in its first two arguments, i.e.,  $\exists r_{\text{sc}}, r_{\text{iso}}, L_k \in \mathbb{R}_{>0}$  s.t.

$$\|k(x_1, \alpha_1, t) - k(x_2, \alpha_2, t)\| \leq L_k \sqrt{\|\alpha_1 - \alpha_2\|^2 + \|x_1 - x_2\|^2} \quad (9.25)$$

for all  $\alpha_1, \alpha_2 \in \mathcal{C}_{\text{iso}}(r_{\text{iso}})$ ,  $x_1, x_2 \in \mathcal{C}_{\text{sc}}(r_{\text{sc}})$ , and  $t \in [t_s, t_f]$ . We also assume that  $r_{\text{iso}}$  and  $r_{\text{sc}}$  are large enough to have  $r_{\text{iso}} - 2c_e \geq 0$  and  $r_{\text{sc}} - 2c_e \geq 0$  for  $c_e$  of Assumption 9.1, and that the set  $\mathcal{C}_{\text{iso}}(r_{\text{iso}} - c_e)$  is forward invariant, i.e.,

$$\alpha(t_s) \in \mathcal{C}_{\text{iso}}(r_{\text{iso}} - c_e) \Rightarrow \varphi^{t-t_s}(\alpha(t_s)) \in \mathcal{C}_{\text{iso}}(r_{\text{iso}} - c_e), \forall t \in [t_s, t_f] \quad (9.26)$$

where  $\varphi^{t-t_s}(\alpha(t_s))$  is the ISO state trajectory with  $\varphi^0(\alpha(t_s)) = \alpha(t_s)$  at  $t = t_s$  as given in Table 9.2 and defined in Definition 9.1.

If  $\ell$  of (9.15) is locally Lipschitz in  $x$  and  $\alpha$ ,  $k$  of (9.18) can be expressed as a composition of locally Lipschitz functions in  $x$  and  $\alpha$ , which implies that the Lipschitz assumption (9.25) always holds for finite  $r_{\text{iso}}$  and  $r_{\text{sc}}$  (see, e.g., Theorem 2 of [31] and the references therein for further explanation). Since the estimation error is expected to decrease in general,  $c_e$  of Assumption 9.1 can be made smaller as  $t_s$  gets larger, which renders the second condition of Assumption 9.2 less strict.

## 9.4 Neural-Rendezvous: Learning-based Robust Guidance and Control to Encounter ISOs

This section finally presents Neural-Rendezvous, a deep learning-based terminal G&C approach to autonomously encounter ISOs, thereby solving the problem 9.1 of Sec. 9.1 that arise from the large state uncertainty and high-velocity challenges. It will be shown that the SN-DNN min-norm control (9.17) of Lemma 9.2 verifies a formal exponential bound on expected spacecraft delivery error, which provides valuable information in determining whether we should use the SN-DNN terminal guidance policy or enhance it with the SN-DNN min-norm control, depending on the size of the state uncertainty.

### 9.4.I Robustness and Stability Guarantee

The assumptions introduced in Sec. 9.3.III allow bounding the mean squared distance between the spacecraft relative position of (9.1) controlled by (9.17) and the desired position  $p_d(t)$  given in (9.14), even under the presence of the state uncertainty (see Fig. 9.4). We remark that the additional notations in the following theorem are summarized in Table 9.4, where the others are consistent with the ones in Table 9.3. Let us remark that the use of our incremental Lyapunov-like function, along with the supermartingale analysis, allows easier handling of the aforementioned local assumptions in estimation and control.

**Theorem 9.1.** *Suppose that Assumptions 9.1 and 9.2 hold, and that the spacecraft relative dynamics with respect to the ISO, given in (9.1), is controlled by  $u = u_c^*$ . If the estimated states at time  $t = t_s$  satisfy  $\hat{\alpha}_s \in C_{\text{iso}}(\bar{R}_{\text{iso}})$  and  $\hat{x}_s \in C_{\text{sc}}(\bar{R}_{\text{sc}})$ , then the spacecraft delivery error is explicitly bounded as follows with probability at least  $1 - \varepsilon_{\text{ctrl}}$ :*

$$\begin{aligned} & \mathbb{E}[\|p(t_f) - \rho\| | \hat{\alpha}(t_s) = \hat{\alpha}_s \cap \hat{x}(t_s) = \hat{x}_s] \\ & \leq \sup_{(\alpha_s, x_s) \in \mathcal{D}_{\text{est}}} e^{-\lambda(t_f - t_s)} \|p_s - p_d(t_s)\| + \frac{e^{-\lambda t_f} L_k}{m_{\text{sc}}(t_f)} \int_{t_s}^{t_f} \varpi^\tau(Z_s, t_s) d\tau \\ & + \begin{cases} \frac{e^{-\lambda(t_f - t_s)} - e^{-\alpha(t_f - t_s)}}{\alpha - \lambda} \frac{v(x_s, t_s)}{\sqrt{m_{\text{sc}}(t_f)}} & \text{if } \alpha \neq \underline{\lambda} \\ (t_f - t_s) e^{-\lambda(t_f - t_s)} \frac{v(x_s, t_s)}{\sqrt{m_{\text{sc}}(t_f)}} & \text{if } \alpha = \underline{\lambda} \end{cases} \end{aligned} \quad (9.27)$$

where  $\varepsilon_{\text{ctrl}} \in \mathbb{R}_{\geq 0}$  is the probability to be given in (9.42),  $\varpi^t(Z_s, t_s)$  is a time-varying function defined as  $\varpi^t(Z_s, t_s) = e^{(\lambda - \alpha)t} \int_{t_s}^t e^{\alpha\tau} \zeta^\tau(Z_s, t_s) d\tau$ . Note that (9.27) yields a bound on  $\mathbb{P}[\|p(t_f) - \rho\| \leq d | \hat{\alpha}(t_s) = \hat{\alpha}_s \cap \hat{x}(t_s) = \hat{x}_s]$  as in Theorem 2.5 of [7], where  $d \in \mathbb{R}_{\geq 0}$  is any given distance of interest, using Markov's inequality.

Table 9.4: Notations in Theorem 9.1.

Notation	Description
$\mathcal{D}_{\text{est}}$	Set defined as $\mathcal{D}_{\text{est}} = \{(\alpha_s, x_s) \in \mathbb{R}^n \times \mathbb{R}^n \mid \sqrt{\ \alpha_s - \hat{\alpha}_s\ ^2 + \ x_s - \hat{x}_s\ ^2} \leq c_e$
$L_k$	Lipschitz constant of $k$ in Assumption (9.1)
$m_{\text{sc}}(t)$	Spacecraft mass of (9.2) satisfying $m_{\text{sc}}(t) \in [m_{\text{sc}}(t_s), m_{\text{sc}}(t_f)]$ due to the Tsiolkovsky rocket equation
$\alpha_s, \hat{\alpha}_s, x_s, \hat{x}_s$	True and estimated ISO and S/C state at time $t = t_s$ , respectively
$p_d(t)$	Desired S/C relative position trajectory, i.e., $x_d(t) = [p_d(t)^\top, \dot{p}_d(t)^\top]^\top$ for $x_d$ of (9.14)
$p_s, \hat{p}_s$	True and estimated S/C relative position at time $t = t_s$ , i.e., $x_s = [p_s^\top, \dot{p}_s^\top]^\top$ and $\hat{x}_s = [\hat{p}_s^\top, \dot{\hat{p}}_s^\top]^\top$
$\bar{R}_{\text{iso}}, \bar{R}_{\text{sc}}$	Constants defined as $\bar{R}_{\text{iso}} = r_{\text{iso}} - 2c_e$ and $\bar{R}_{\text{sc}} = r_{\text{sc}} - 2c_e$ for $c_e$ of Assumption 9.1 and $r_{\text{iso}}$ and $r_{\text{sc}}$ of Assumption 9.2
$u_\ell^*$	SN-DNN min-norm control policy (9.17) of Lemma 9.2
$v(x, t)$	Non-negative function given as $v(x, t) = \sqrt{V(\dot{p}, \varrho_d(p, t), t)} = \sqrt{m_{\text{sc}}(t)} \ \Lambda(p - p_d(t)) + \dot{p} - \dot{p}_d(t)\ $ for $V$ of (9.22)
$Z_s$	Tuple of the true and estimated ISO and S/C state at time $t = t_s$ , i.e., $Z_s = (\alpha_s, \hat{\alpha}_s, x_s, \hat{x}_s)$
$\alpha$	Positive constant of (9.21)
$\underline{\lambda}$	Minimum eigenvalue of $\Lambda$ defined in (9.21)
$\rho$	Desired terminal S/C relative position of (9.4)

*Proof.* The dynamics (9.16) controlled by (9.17) can be rewritten as

$$\ddot{p} = \mathcal{f}(x, \alpha, t) + \mathcal{B}(x, \alpha, t)u_\ell^*(x, \alpha, t; \theta_{\text{nn}}) + \mathcal{B}(x, \hat{\alpha}, t)\tilde{u} \quad (9.28)$$

where  $\tilde{u} = u_\ell^*(\hat{x}, \hat{\alpha}, t; \theta_{\text{nn}}) - u_\ell^*(x, \alpha, t; \theta_{\text{nn}})$ . Since the third term of (9.28) is subject to stochastic disturbance due to the state uncertainty of Assumption 9.1, we consider the weak infinitesimal operator  $\mathcal{A}$  given in [30, p. 9], instead of taking the time derivative, for analyzing the time evolution of the non-negative function  $v$  defined as  $v(x, t) = \sqrt{V(\dot{p}, \varrho_d(p, t), t)} = \sqrt{m_{\text{sc}}(t)} \|\Lambda(p - p_d(t)) + \dot{p} - \dot{p}_d(t)\|$  for  $V$  of (9.22). To this end, let us compute the following time increment of  $v$  evaluated at the true state and time  $(x, \alpha, t)$ :

$$\begin{aligned} \Delta v &= v(x(t + \Delta t), t + \Delta t) - v(x, t) \\ &= \frac{1}{2v(x, t)} \left( \frac{\partial V}{\partial \dot{p}} \ddot{p}(t) + \frac{\partial V}{\partial \varrho_d} \ddot{\varrho}_d(p(t), t) + \frac{\partial V}{\partial t} \right) \Delta t + \mathcal{O}(\Delta t^2) \end{aligned} \quad (9.29)$$

where  $\Delta t \in \mathbb{R}_{\geq 0}$  and the arguments  $(\dot{p}(t), \varrho_d(p, t), t)$  of the partial derivatives of  $V$  are omitted for notational convenience. Dropping the argument  $t$  for the state variables for simplicity, the dynamics decomposition (9.28) gives

$$\begin{aligned} & \frac{\partial V}{\partial \dot{p}} \ddot{p} + \frac{\partial V}{\partial \varrho_d} \ddot{\varrho}_d(p, t) + \frac{\partial V}{\partial t} \\ & \leq \frac{\partial V}{\partial \dot{p}} (\mathcal{F}(x, \alpha, t) + \mathcal{B}(x, \alpha, t) u^*(\hat{x}, \hat{\alpha}, t; \theta_{\text{nn}})) + \frac{\partial V}{\partial \varrho_d} (\ddot{p}_d - \Lambda(\dot{p} - \dot{p}_d)) \\ & \leq -2\alpha V(\dot{p}, \varrho_d(p, t), t) + 2 \frac{\partial V}{\partial \dot{p}} \mathcal{B}(x, \alpha, t) \tilde{u} = -2\alpha v(x, t)^2 + 2v(x, t) \frac{\|\tilde{u}\|}{\sqrt{m_{\text{sc}}(t)}} \end{aligned} \quad (9.30)$$

where the first inequality follows from the fact that the spacecraft mass  $m_{\text{sc}}(t)$ , described by the Tsiolkovsky rocket equation, is a decreasing function and thus  $\partial V / \partial t \leq 0$ , and the second inequality follows from the stability condition (9.21) evaluated at  $(x, \alpha, t)$ , which is guaranteed to be feasible due to Lemma 9.2. Since  $u^*$  is assumed to be Lipschitz as in (9.25) of Assumption 9.2, we get the following relation for any  $x_s, \hat{x}_s \in C_{\text{sc}}(r_{\text{sc}})$ ,  $\alpha_s, \hat{\alpha}_s \in C_{\text{iso}}(r_{\text{iso}})$ , and  $t_s \in [0, t_f]$ , by substituting (9.30) into (9.29):

$$\mathcal{A}v(x_s, t_s) = \lim_{\Delta t \downarrow 0} \frac{\mathbb{E}_{Z_s} [\Delta v]}{\Delta t} \leq -\alpha v(x_s, t_s) + \frac{L_k}{\sqrt{m_{\text{sc}}(t_f)}} \sqrt{\|\hat{\alpha}_s - \alpha_s\|^2 + \|\hat{x}_s - x_s\|^2} \quad (9.31)$$

where  $\mathbb{E}_{Z_s} [\cdot] = \mathbb{E} [\cdot | x(t_s) = x_s, \hat{x}(t_s) = \hat{x}_s, \alpha(t_s) = \alpha_s, \hat{\alpha}(t_s) = \hat{\alpha}_s]$  for  $Z_s$  given as  $Z_s = (x_s, \hat{x}_s, \alpha_s, \hat{\alpha}_s)$ , and the relation  $m_{\text{sc}}(t) \geq m_{\text{sc}}(t_f)$  by the Tsiolkovsky rocket equation is also used to obtain the inequality. Applying Dynkin's formula [30, p. 10] to (9.31) gives the following with probability at least  $1 - \varepsilon_{\text{est}}$  due to (9.24) of Assumption 9.1:

$$\begin{aligned} & \mathbb{E}_{Z_s} [\|\Lambda(p(t) - p_d(t)) + \dot{p}(t) - \dot{p}_d(t)\|] \\ & \leq \frac{e^{-\alpha(t-t_s)} v(x_s, t_s)}{\sqrt{m_{\text{sc}}(t_f)}} + \frac{e^{-\alpha t} L_k}{m_{\text{sc}}(t_f)} \int_{t_s}^t e^{\alpha \tau} \varsigma^\tau(Z_s, t_s) d\tau, \quad \forall t \in [t_s, \bar{t}_e] \end{aligned} \quad (9.32)$$

where  $\varsigma^\tau(Z_s, t_s)$  is as given in (9.24) and  $\bar{t}_e$  is defined as  $\bar{t}_e = \min\{t_e, t_f\}$ , with  $t_e$  being the first exit time that any of the states leave their respective set, i.e.,

$$\begin{aligned} t_e = \inf\{t \geq t_s : \alpha(t) \notin C_{\text{iso}}(r_{\text{iso}}) \cup \hat{\alpha}(t) \notin C_{\text{iso}}(r_{\text{iso}}) \\ \cup x(t) \notin C_{\text{sc}}(r_{\text{sc}}) \cup \hat{x}(t) \notin C_{\text{sc}}(r_{\text{sc}})\} \end{aligned} \quad (9.33)$$

given that the event of (9.23) in Assumption 9.1 has occurred. Since we further have that  $\mathcal{A}\|p - p_d\| = \frac{d}{dt} \|p - p_d\| \leq \|\tilde{\varrho}\| - \underline{\lambda}(p - p_d)$  for  $\tilde{\varrho} = \Lambda(p - p_d) + \dot{p} - \dot{p}_d$

due to the hierarchical structure of  $\tilde{\varrho}$ , utilizing Dynkin's formula one more time and then substituting (9.32) result in

$$\begin{aligned} \mathbb{E}_{Z_s} [\|p(t) - p_d(t)\|] &\leq e^{-\lambda t} \int_{t_s}^t \frac{e^{(\lambda-\alpha)\tau + \alpha t_s} v(x_s, t_s)}{\sqrt{m_{sc}(t_f)}} + \frac{L_k}{m_{sc}(t_f)} \varpi^\tau(Z_s, t_s) d\tau \\ &+ e^{-\lambda(t-t_s)} \|p_s - p_d(0)\|, \quad \forall t \in [0, \bar{t}_e] \end{aligned} \quad (9.34)$$

where  $x_s = [p_s^\top, \dot{p}_s^\top]^\top$ .

What is left to show is that we can achieve  $t_e > t_f$  with a finite probability for the first exit time  $t_e$  of (9.33). For this purpose, we need the following lemma.

**Lemma 9.3.** *Suppose that the events of Assumption 9.1 has occurred, i.e., the event of (9.23) has occurred and the 2-norm estimation error satisfies*

$$\sqrt{\|\hat{\alpha}(t) - \alpha(t)\|^2 + \|\hat{x}(t) - x(t)\|^2} < c_e, \quad \forall t \in [t_s, t_f].$$

If Assumption 9.2 holds and if we have

$$\mathcal{A}v(x_s, t_s) \leq -\alpha v(x_s, t_s) + \frac{L_k}{\sqrt{m_{sc}(t_f)}} \sqrt{\|\hat{\alpha}_s - \alpha_s\|^2 + \|\hat{x}_s - x_s\|^2} \quad (9.35)$$

for any  $x_s, \hat{x}_s \in C_{sc}(r_{sc})$ ,  $\alpha_s, \hat{\alpha}_s \in C_{iso}(r_{iso})$ , and  $t_s \in [0, t_f]$  as in (9.31), then we get the following probabilistic bound for  $\exists \varepsilon_{\text{exit}} \in \mathbb{R}_{\geq 0}$ :

$$p_{\text{stay}} \geq 1 - \varepsilon_{\text{exit}} = \begin{cases} \left(1 - \frac{E_s}{\bar{v}}\right) e^{-\frac{H(t_f)}{\bar{v}}} & \text{if } \bar{v} \geq \frac{\sup_{t \in [t_s, t_f]} h(t)}{\bar{\alpha}} \\ 1 - \frac{\bar{\alpha} E_s + (e^{\bar{H}(t_f)} - 1) \sup_{t \in [t_s, t_f]} h(t)}{\bar{\alpha} \bar{v} e^{\bar{H}(t_f)}} & \text{otherwise} \end{cases} \quad (9.36)$$

$$p_{\text{stay}} = \mathbb{P}_{Z_s} \left[ \bigcap_{t \in [t_s, t_f]} x(t) \in C_{sc}(\bar{r}_{sc}) \right]$$

where  $Z_s = (\alpha_s, \hat{\alpha}_s, x_s, \hat{x}_s)$  denotes the states at time  $t = t_s$ , which satisfy  $\alpha_s \in C_{iso}(\bar{r}_{iso})$ ,  $\hat{\alpha}_s \in C_{iso}(\bar{R}_{iso})$ ,  $x_s \in C_{sc}(\bar{r}_{sc})$ , and  $\hat{x}_s \in C_{sc}(\bar{R}_{sc})$  for  $\bar{r}_{iso} = r_{iso} - c_e \geq 0$ ,  $\bar{R}_{iso} = r_{iso} - 2c_e \geq 0$ ,  $\bar{r}_{sc} = r_{sc} - c_e \geq 0$ , and  $\bar{R}_{sc} = r_{sc} - 2c_e \geq 0$ ,  $E_s = v(x_s, t_s) + \delta_p \|p_s - p_d(t_s)\|$ ,  $H(t) = \int_{t_s}^t h(\tau) d\tau$ ,  $\bar{H}(t) = 2H(t)\bar{\alpha} / \sup_{t \in [t_s, t_f]} h(t)$ , and  $h(t) = L_k \mathcal{S}^t(Z_s, t_s) / \sqrt{m_{sc}(t_f)}$ . The suitable choices of  $\bar{v}, \bar{\alpha}, \delta_p \in \mathbb{R}_{>0}$  are to be defined below.

*Proof of Lemma 9.3.* Let us define a non-negative and continuous function  $E(x, t)$  as

$$E(x, t) = v(x, t) + \delta_p \|p - p_d(t)\| \quad (9.37)$$

where  $x = [p^\top, \dot{p}^\top]^\top$  and  $\delta_p \in \mathbb{R}_{>0}$ . Note that  $E(x, t)$  of (9.37) is 0 only when  $x = x_d(t)$ . Since we have  $\mathcal{A}\|p - p_d(t)\| \leq v(x, t)/\sqrt{m_{sc}(t_f)} - \underline{\lambda}\|p - p_d(t)\|$ , designing  $\delta_p$  to have  $\alpha - \delta_p/\sqrt{m_{sc}(t_f)} > 0$ , the relation (9.35) gives

$$\mathcal{A}E(x_s, t_s) \leq -\bar{\alpha}E(x_s, t_s) + \frac{L_k}{\sqrt{m_{sc}(t_f)}} \sqrt{\|\hat{\alpha}_s - \alpha_s\|^2 + \|\hat{x}_s - x_s\|^2} \quad (9.38)$$

for any  $x_s, \hat{x}_s \in C_{sc}(r_{sc})$ ,  $\alpha_s, \hat{\alpha}_s \in C_{iso}(r_{iso})$ , and  $t_s \in [0, t_f]$ , where

$$\bar{\alpha} = \min \left\{ \alpha - \delta_p/\sqrt{m_{sc}(t_f)}, \underline{\lambda} \right\} > 0.$$

Let us define another non-negative and continuous function  $W(x, t)$  as

$$W(x, t) = E(x, t)e^{\gamma H(t)} + \frac{e^{\gamma H(t_f)} - e^{\gamma H(t)}}{\gamma}$$

where  $\gamma$  is a positive constant that satisfies  $2\bar{\alpha} \geq \gamma \sup_{t \in [0, t_f]} h(t)$ . If  $\bar{v}, \bar{w} \in \mathbb{R}_{>0}$  is selected to have  $W(x, t) < \bar{w} \Rightarrow E(x, t) < \bar{v} \Rightarrow x \in C_{sc}(\bar{r}_{sc}) = \bigcup_{t \in [0, t_f]} \{s \in \mathbb{R}^n \mid \|s - x_d(t)\| < \bar{r}_{sc}\} \subset \mathbb{R}^n$ , which is always possible since  $E(x, t)$  of (9.37) is 0 only when  $x = x_d(t)$ , then we can utilize (9.38) to compute  $\mathcal{A}W$  as follows for any  $x_s = [p_s^\top, \dot{p}_s^\top]^\top \in C_{sc}(\bar{r}_{sc})$  and  $t_s \in [0, t_f]$  that satisfies  $W(x_s, t_s) < \bar{w}$ :

$$\begin{aligned} \mathcal{A}W(x_s, t_s) &= (\gamma h(t_s)E(x_s, t_s) + \mathcal{A}E(x_s, t_s))e^{\gamma H(t_s)} - h(t_s)e^{\gamma H(t_s)} \\ &\leq (\mathcal{K}(\alpha_s, \hat{\alpha}_s, x_s, \hat{x}_s) - h(t_s))e^{\gamma H(t_s)} \end{aligned} \quad (9.39)$$

where  $\mathcal{K}(\alpha_s, \hat{\alpha}_s, x_s, \hat{x}_s) = L_k \sqrt{(\|\hat{\alpha}_s - \alpha_s\|^2 + \|\hat{x}_s - x_s\|^2)/m_{sc}(t_f)}$  and the inequality follows from the relation  $\bar{\alpha} \geq \gamma \sup_{t \in [0, t_f]} h(t)$ . Applying Dynkin's formula [30, p. 10] to (9.39), it can be verified that  $W(x(t), t)$  is a non-negative supermartingale due to the fact that  $\mathbb{E}_{Z_s} [\mathcal{K}(\alpha(t), \hat{\alpha}(t), x(t), \hat{x}(t))] \leq L_k \zeta^t(Z_s, t_s)/\sqrt{m_{sc}(t_f)} = h(t)$  by definition of  $\zeta^t(Z_s, t_s)$  in (9.24) of Assumption 9.1, which gives the following due to Ville's maximal inequality [30, p. 26]:

$$\mathbb{P}_{Z_s} \left[ \sup_{t \in [t_s, t_f]} W(x(t), t) \geq \bar{w} \right] \leq \frac{E_s + \gamma^{-1}(e^{\gamma H(t_f)} - 1)}{\bar{w}} \quad (9.40)$$

as long as we have  $\alpha(t), \hat{\alpha}(t) \in C_{iso}(r_{iso})$  and  $\hat{x}(t) \in C_{sc}(r_{sc})$  for  $\forall t \in [t_s, t_f]$ , where  $Z_s = (\alpha_s, \hat{\alpha}_s, x_s, \hat{x}_s)$  is as defined in (9.36). For  $Z_s$  satisfying  $\alpha_s \in C_{iso}(\bar{r}_{iso})$ ,  $\hat{\alpha}_s \in C_{iso}(\bar{R}_{iso})$ ,  $x_s \in C_{sc}(\bar{r}_{sc})$ , and  $\hat{x}_s \in C_{sc}(\bar{R}_{sc})$ , the occurrence of the events of Assumption 9.1 and the forward invariance condition of (9.26) of Assumption 9.2 ensure that if we have  $x(t) \in C_{sc}(\bar{r}_{sc})$  for  $\forall t \in [t_s, t_f]$ , we get  $\alpha(t) \in C_{iso}(\bar{r}_{iso})$ ,  $\hat{\alpha}(t) \in C_{iso}(r_{iso})$ , and  $\hat{x}(t) \in C_{sc}(r_{sc})$  for  $\forall t \in [t_s, t_f]$ . Therefore, selecting the constants  $\gamma$  and  $\bar{w}$  in the inequality (9.40) as in [30, pp. 82-83] yields (9.36) due to the relation  $W(x, t) < \bar{w} \Rightarrow E(x, t) < \bar{v} \Rightarrow x \in C_{iso}(\bar{r}_{iso})$ .  $\square$

Now, let us select  $c_e$  of Assumption 9.1 as  $c_e = k_e \mathbb{E} [\zeta^{t_s}(Z_0, 0)]$ , where  $Z_0$  is the tuple of the true and estimated ISO and spacecraft state at time  $t = 0$  and  $k_e \in \mathbb{R}_{>0}$ . Using Markov's inequality along with the bounds (9.23) and (9.24) of Assumption 9.1, we get the following probability bound:

$$\mathbb{P} \left[ \sqrt{\|\hat{\alpha}_s - \alpha_s\|^2 + \|\hat{x}_s - x_s\|^2} < k_e \mathbb{E} [\zeta^{t_s}(Z_0, 0)] \right] \geq (1 - \varepsilon_{\text{est}}) \left(1 - \frac{1}{k_e}\right). \quad (9.41)$$

Since we have  $\hat{\alpha}_s \in C_{\text{iso}}(\bar{R}_{\text{iso}})$  and  $\hat{x}_s \in C_{\text{sc}}(\bar{R}_{\text{sc}})$  by the theorem assumption, the occurrence of the event in (9.41) implies  $\alpha_s \in C_{\text{iso}}(\bar{r}_{\text{iso}})$  and  $x_s \in C_{\text{sc}}(\bar{r}_{\text{sc}})$  for  $\bar{r}_{\text{iso}} = r_{\text{iso}} - k_e \mathbb{E} [\zeta^{t_s}(Z_0, 0)] \geq 0$ ,  $\bar{r}_{\text{sc}} = r_{\text{sc}} - k_e \mathbb{E} [\zeta^{t_s}(Z_0, 0)] \geq 0$ , and then  $\alpha(t) \in C_{\text{iso}}(\bar{r}_{\text{iso}})$ ,  $\forall t \in [t_s, t_f]$  due to the forward invariance condition (9.26) in Assumption 9.2. Therefore, using  $\varepsilon_{\text{err}}$  of Assumption 9.1 along with the result of Lemma 9.3, we have  $\exists \varepsilon_{\text{ctrl}} \in \mathbb{R}_{\geq 0}$  s.t.

$$\mathbb{P} [\mathcal{A} | \hat{\alpha}(t_s) = \hat{\alpha}_s \cap \hat{x}(t_s) = \hat{x}_s] \geq 1 - \varepsilon_{\text{ctrl}} = (1 - \varepsilon_{\text{exit}})(1 - \varepsilon_{\text{est}})(1 - \varepsilon_{\text{err}}) \left(1 - \frac{1}{k_e}\right) \quad (9.42)$$

where  $\mathcal{A} = \bigcap_{t \in [t_s, t_f]} \alpha(t) \in C_{\text{iso}}(\bar{r}_{\text{iso}}) \cap \hat{\alpha}(t) \in C_{\text{iso}}(r_{\text{iso}}) \cap x(t) \in C_{\text{sc}}(\bar{r}_{\text{sc}}) \cap \hat{x}(t) \in C_{\text{sc}}(r_{\text{sc}}) \cap \mathcal{E}$  with  $\mathcal{E}$  being the event of (9.23) in Assumption 9.1. The desired relation (9.27) follows by evaluating the first term of the integral in (9.34), and by observing that if  $\mathcal{A}$  of (9.42) occurs, we have  $t_e > t_f$  and  $\sqrt{\|\hat{\alpha}_s - \alpha_s\|^2 + \|\hat{x}_s - x_s\|^2} < k_e \mathbb{E} [\zeta^{t_s}(Z_0, 0)]$  due to (9.41).

Note that if  $\mathcal{E}_{\text{iso}} = \mathcal{E}_{\text{sc}} = \mathbb{R}^n$  in Assumption 9.1 and  $k$  is globally Lipschitz in Assumption 9.2, the estimation bound (9.24) and the Lipschitz bound (9.25) always hold without the second condition of Assumption 9.1. This indicates that the bound (9.27) holds as long as  $(\alpha_s, x_s) \in \mathcal{D}_{\text{est}}$  for given  $\hat{\alpha}_s, \hat{x}_s \in \mathbb{R}^n$ , which occurs with probability at least  $1 - k_e^{-1}$  due to (9.41).  $\square$

As derived in Theorem 9.1, the SN-DNN min-norm control policy (9.17) of Lemma 9.2 enhances the SN-DNN terminal guidance policy of Lemma 9.1 by providing the explicit spacecraft delivery error bound (9.27), which holds even under the presence of the state uncertainty. This bound is valuable in modulating the learning and control parameters to achieve a verifiable performance guarantee consistent with a mission-specific performance requirement as to be seen in Sec. 9.4.II.

As illustrated in Fig. 9.6, the expected state tracking error bound (9.27) of Theorem 9.1 decreases exponentially in time if  $\zeta^t(Z_s, t_s)$  is a non-increasing function

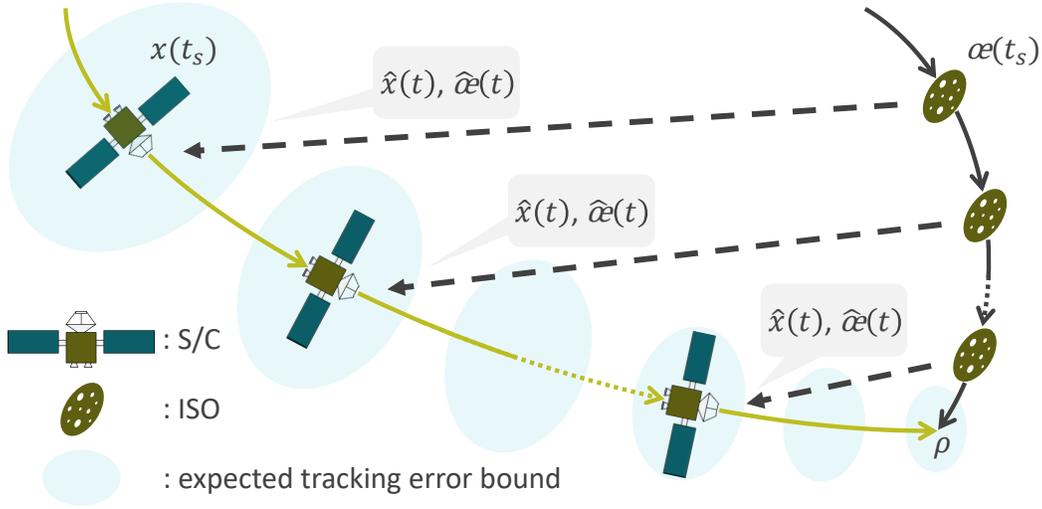


Figure 9.6: Illustration of the expected position tracking error bound (9.27) of Theorem 9.1, where  $\zeta^t(Z_s, t_s)$  is assumed to be a non-increasing function in time.

in  $t$ . The following example demonstrates how we compute the bound (9.27) in practice.

**Example 9.1.** Suppose that the estimation error is upper-bounded by a function that exponentially decreases in time, i.e., we have

$$\zeta^t(Z_s, t_s) = e^{-\beta(t-t_s)} \sqrt{\|\hat{\alpha}_s - \alpha_s\|^2 + \|\hat{x}_s - x_s\|^2} + c$$

for (9.24) in Assumption 9.1, where  $c \in \mathbb{R}_{\geq 0}$  and  $\beta \in \mathbb{R}_{>0}$ . Assuming that  $\alpha \neq \beta$ ,  $\beta \neq \underline{\lambda}$ , and  $\underline{\lambda} \neq \alpha$  for simplicity, we get

$$\begin{aligned} \text{RHS of (9.27)} &\leq e^{-\underline{\lambda}\bar{t}_f} (\|\hat{p}_s - p_d(t_s)\| + c_e) \\ &\quad + \frac{e^{-\underline{\lambda}\bar{t}_f} - e^{-\alpha\bar{t}_f}}{\alpha - \underline{\lambda}} \frac{(\bar{\lambda} + 1)(\|\hat{x}_s - x_d(t_s)\| + c_e)}{\sqrt{m_{\text{sc}}(t_f)}} \\ &\quad + \frac{L_k}{m_{\text{sc}}(t_f)} \left( \frac{c_e}{\alpha - \beta} \left( \frac{e^{-\underline{\lambda}\bar{t}_f} - e^{-\beta\bar{t}_f}}{\beta - \underline{\lambda}} - \frac{e^{-\underline{\lambda}\bar{t}_f} - e^{-\alpha\bar{t}_f}}{\alpha - \underline{\lambda}} \right) \right. \\ &\quad \left. + \frac{c}{\alpha} \left( \frac{1 - e^{-\underline{\lambda}\bar{t}_f}}{\underline{\lambda}} - \frac{e^{-\underline{\lambda}\bar{t}_f} - e^{-\alpha\bar{t}_f}}{\alpha - \underline{\lambda}} \right) \right) \end{aligned} \quad (9.43)$$

where  $\bar{t}_f = t_f - t_s$ . Note that the bound (9.43) can be computed explicitly for given  $\hat{x}_s$  and  $\hat{\alpha}_s$  at time  $t = t_s$  as long as  $\hat{\alpha}_s \in C_{\text{iso}}(\bar{R}_{\text{iso}})$  and  $\hat{x}_s \in C_{\text{sc}}(\bar{R}_{\text{sc}})$ . Its dominant term in (9.43) for large  $\bar{t}_f$  is  $L_k c / (m_{\text{sc}}(t_f) \alpha \underline{\lambda})$  due to the following relation:

$$\lim_{t_f \rightarrow \infty} \text{RHS of (9.43)} = \frac{L_k c}{m_{\text{sc}}(t_f) \alpha \underline{\lambda}}.$$

The bound 9.27 of Theorem 9.1 can be used as a tool to determine whether we should use the SN-DNN terminal guidance policy of Lemma 9.1 or the SN-DNN min-norm control policy (9.17) of Lemma 9.2, based on the trade-off between them as to be seen in the following.

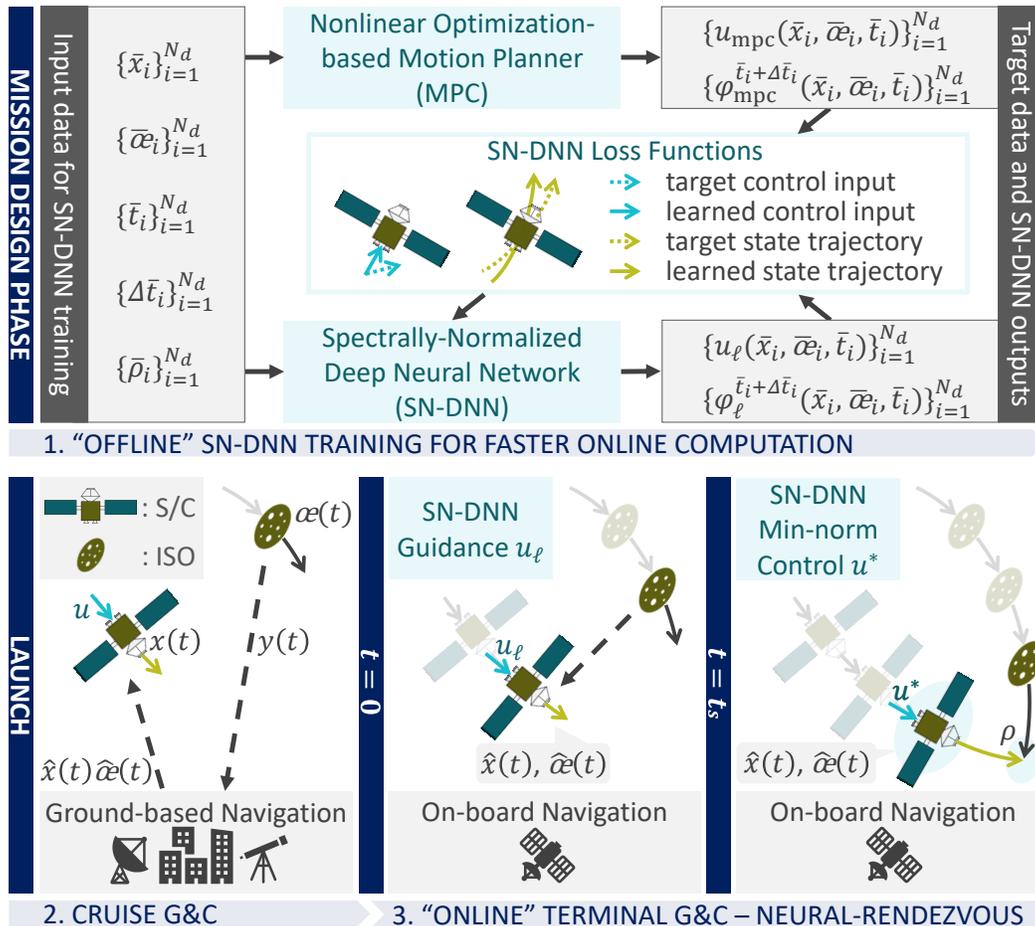


Figure 9.7: Mission timeline for encountering an ISO, where  $y(t)$  is the state measurement as in Fig. 9.2 and the other notation follows that of Lemma 9.1 and Theorem 9.1. Terminal G&C (Neural-Rendezvous) is performed using the SN-DNN guidance and min-norm control policies online as in Algorithm 1, thereby enabling fast response autonomous operations under the large ISO state uncertainty and high-velocity challenges. Note that the SN-DNN is trained offline.

## 9.4.II Neural-Rendezvous

We summarize the pros and cons of the aforementioned terminal G&C techniques.

- The SN-DNN terminal guidance policy of Lemma 9.1 can be implemented in real-time and possesses an optimality gap as in (9.11), resulting in the near-

**Algorithm 1:** Neural-Rendezvous**Inputs :**  $u_\ell$  of Lemma 9.1 and  $u_\ell^*$  of Lemma 9.2**Outputs:** Control input  $u$  of (9.1) for  $t \in [0, t_f]$ 


---

```

 $t_0 \leftarrow$  current time
 $\Delta t \leftarrow$  control time interval
 $t_k, t_d, t_{\text{int}} \leftarrow$  current time  $- t_0$ 
 $\text{flagA}, \text{flagB} \leftarrow 0$ 
while  $t_k < t_f$  do
  Obtain  $\hat{\alpha}(t_k)$  and  $\hat{x}(t_k)$  using navigation technique
  if  $\text{flagA} = 0$  then
     $u \leftarrow u_\ell(\hat{x}(t_k), \hat{\alpha}(t_k), t_k)$ 
  else
    if  $\text{flagB} = 1$  then
       $u \leftarrow u_\ell^*(\hat{x}(t_k), \hat{\alpha}(t_k), t_k)$ 
    else
      Compute RHS of (9.27) in Theorem 9.1 with  $t_s = t_k$ 
      if RHS of (9.27)  $>$  threshold then
         $u \leftarrow u_\ell(\hat{x}(t_k), \hat{\alpha}(t_k), t_k)$ 
      else
         $u \leftarrow u_\ell^*(\hat{x}(t_k), \hat{\alpha}(t_k), t_k)$ 
         $\text{flagB} \leftarrow 1$ 
    end if
  end if
  Apply  $u$  to (9.1) for  $t \in [t_k, t_k + \Delta t]$ 
  while current time  $- t_0 < t_k + \Delta t$  do
    if  $\text{flagB} = 0$  then
      Integrate dynamics to get  $x_d$  and  $\alpha_d$  of (9.14) from  $t_{\text{int}}$ 
    end if
    if integration is complete then
      Update  $x_d$  and  $\alpha_d$ 
       $t_d, t_{\text{int}} \leftarrow t_k + \Delta t$ 
       $\text{flagA} \leftarrow 1$ 
    else
       $t_{\text{int}} \leftarrow$  time when S/C stopped integration
    end if
  end while
   $t_k \leftarrow t_k + \Delta t$ 

```

---

optimal guarantee in terms of dynamic regret as discussed below (9.5) and in Lemma 9.1. However, obtaining any quantitative bound on the spacecraft delivery error with respect to the desired relative position, to either flyby or impact the ISO, is difficult in general [7].

- In contrast, the SN-DNN min-norm control policy (9.17) of Lemma 9.2, which solves the problem 9.1 of Sec. 9.1, can also be implemented in real-time and provides an explicit upper-bound on the spacecraft delivery error that decreases in time as proven in Theorem (9.1). However, the desired trajectory it tracks is subject to the large state uncertainty initially for small  $t_s$ , resulting in a large optimality gap.

Based on these observations, it is ideal to utilize the SN-DNN terminal guidance policy without any feedback control initially for large state uncertainty, to avoid having a large optimality gap, then activate the SN-DNN min-norm control once the verifiable spacecraft delivery error of (9.27) of Theorem (9.1) becomes smaller than a mission-specific threshold value for the desired trajectory (9.14), which is to be updated at  $t_d \in [0, t_f]$  along the way as discussed in Remark 9.6. The pseudo-code for the proposed learning-based approach for encountering the ISO is given in Algorithm 1, and its mission timeline is shown in Fig. 9.7.

Again, the role of offline learning in Neural-Rendezvous, with the size of the neural network selected to be small enough for real-time implementation, is to deal with the limited computational resources of the spacecraft in solving any nonlinear optimization with non-negligible online computational load, including the one of MPC with receding horizons. This also enables utilizing Neural-Rendezvous in more challenging and highly nonlinear G&C scenarios as to be seen in Sec. 10.2, where solving even a single optimization online could become unrealistic.

**Remark 9.7.** *We use  $t_{\text{int}}$  in Algorithm 1 to account for the fact that computing  $x_d$  could take more than  $\Delta t$  for small  $\Delta t$  and  $t_k$  as pointed out in Remark 9.6, and thus the spacecraft is sometimes required to compute it over multiple time steps. It can be seen that the SN-DNN terminal guidance policy  $u_\ell$  is also useful when the spacecraft does not possess  $x_d$  yet, i.e., when  $\text{flagA} = 0$ . Also, the impact of discretization introduced in Algorithm 1 will be demonstrated in Sec. 10.1.III, where the detailed discussion of its connection to continuous-time stochastic systems can be found in [32].*

## 9.5 Extensions

In this section, we present several extensions of the proposed G&C techniques for encountering ISOs, which can be used to further improve certain aspects of their performance.

### 9.5.I Optimal Lyapunov Functions and Other Types of Disturbances

The Neural-Rendezvous approach in Algorithm 1 is based on the feedback control of Lemma 9.2, constructed using the non-negative function  $V$  of (9.22). In general, we can always construct such a feedback control policy as long as there exists a

Lyapunov function  $\mathcal{V} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}_{\geq 0}$  [33, p. 154] and control policy  $u = \mu(x, \alpha, t)$  that satisfies

$$k_1 \| (x - x_d) \|^2 \leq \mathcal{V}(x, \alpha, t) \leq k_2 \| (x - x_d) \|^2 \quad (9.44)$$

$$\frac{\partial \mathcal{V}}{\partial t} + \frac{\partial \mathcal{V}}{\partial \alpha} \dot{\alpha} + \frac{\partial \mathcal{V}}{\partial x} (f(x, \alpha, t) + B(x, \alpha, t) \mu(x, \alpha, t)) \leq -2\alpha \mathcal{V}(x, \alpha, t) \quad (9.45)$$

$\forall x, \alpha \in \mathbb{R}^n$  and  $t \in \mathbb{R}_{\geq 0}$ , where  $k_1, k_2, \alpha \in \mathbb{R}_{> 0}$ , and  $u, f$ , and  $B$  are given in (9.1). The combination of such  $\mathcal{V}$  and  $\mu$  is not necessarily unique, and many studies have discussed optimal and numerically efficient ways to find them, as partially summarized in [7]. For example, contraction theory [7], [34]–[36] uses a squared differential length  $\mathcal{V} = \delta x^\top M(x, \alpha, t) \delta x$  as a Lyapunov-like function, allowing the systematic construction of  $\mathcal{V}$  and  $\mu$  of (9.44) and (9.45) via convex optimization to minimize an upper bound of the steady-state distance between the controlled and desired system trajectories [32], [35]. The computational burden of these approaches can be significantly reduced by using machine learning techniques [7], [37]–[39].

It is worth emphasizing that since our proposed feedback control of Lemma 9.2 is also categorized as a Lyapunov-based approach (which can be verified for  $\mathcal{V} = V + \epsilon(p - p_d)^\top \Lambda(p - p_d)$ , where  $V$  and  $\Lambda$  are given in (9.20) and  $\epsilon \in \mathbb{R}_{> 0}$  is as defined in [40, pp. 54-55]), we can show that it is robust not only against the state uncertainty of (9.1), but also deterministic and stochastic disturbances resulting from e.g., process noise, control execution error, parametric uncertainty, and unknown parts of dynamics as shown in [7], [41], [42].

## 9.5.II Stochastic MPC with Terminal Chance Constraints

We could utilize the expectation bound (9.27) in the guidance problem to compute a risk-constrained policy using terminal chance-constrained stochastic optimal control problem formulation in [43], with probabilistic guarantees on reaching the terminal set. This approach improves the quality of the solution that is approximated by an SN-DNN as in Theorem 9.1. Using the same notation as in (9.3) of Sec. 9.2.I, the chance-constrained stochastic optimal control problem is described as follows:

$$u^*(\hat{x}(\tau), \hat{\alpha}(\tau), t, \rho) = \arg \min_{u(t) \in \mathcal{U}(t)} \mathbb{E} \left[ \int_{\tau}^{t_f} J_S(\xi(t), u(t)) dt + J_{S_f}(\xi(t_f)) \right] \quad (9.46)$$

$$\text{s.t. } d\xi(t) = f(\xi(t), \varphi^{t-\tau}(\hat{\alpha}(\tau)), t) dt + B(\xi(t), \varphi^{t-\tau}(\hat{\alpha}(\tau)), t) dt \\ + G(\xi(t), \varphi^{t-\tau}(\hat{\alpha}(\tau)), u(t)) d\mathcal{W}(t), \forall t \in [\tau, t_f]$$

$$\mathbb{E}(\xi(\tau)) = \hat{x}(\tau), x(t_f) \in \mathcal{X}_{S_f}$$

where  $J_{S_f}$  and  $J_S$  are given transient and terminal cost functions,  $G : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n \times w}$  is a matrix-valued diffusion coefficient for stochastic disturbance,  $\mathcal{W} : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^w$  is a  $w$ -dimensional Wiener process [44, p. 100] (see also [44, p. xii] for the notations used), and the terminal constraint set  $\mathcal{X}_{S_f}$  is defined using a quadratic chance constraint as follows:

$$\mathcal{X}_{S_f} = \{x \in \mathbb{R}^n \mid \mathbb{P}[(x - \rho)^\top Q_{X_f}(x - \rho) \leq c_f] \geq 1 - \epsilon_f\},$$

where  $Q_{X_f} > 0$ ,  $c_f \in \mathbb{R}_{\geq 0}$ , and  $\epsilon_f$  is the risk measure. The terminal set defines the encounter specifications (position, velocity, and their variance, respectively) with the ISO. Unlike the formulation given in (9.3), the problem (9.46) explicitly accounts for the stochastic disturbance resulting from the ISO state uncertainty in its dynamics, leading to a more sophisticated offline solution that can be obtained using the generalized polynomial chaos-based sequential convex programming method [43].

### 9.5.III Multi-Agent Systems in Cluttered Environments

The optimization formulation with chance constraints in Sec. 9.5.II is also useful in extending our proposed approach to a multi-agent setting with obstacles, where each spacecraft is required to achieve its mission objectives in a collision-free manner. It allows expressing stochastic guidance problems as deterministic counterparts [43] so we could exploit existing methods for designing distributed, robust, and safe control policies for deterministic multi-agent systems, which can be computed in real-time [39], [45], [46].

### 9.5.IV Discrete-Time Systems

Although this chapter considers a continuous-time system and we discretize it when implementing the proposed algorithm as discussed in Remark 9.7, where the impact of discretization is to be demonstrated in Sec. 10.1.III, we could also start from a discrete-time system and analyze robustness and stability in a discrete sense, which can be performed by replacing the stability constraint (9.45) with its discrete versions introduced in [32], [34], [47]–[51].

### 9.5.V Robustness of Neural Networks

As discussed in Remark 9.4, there are several ongoing studies in the field of machine learning that view a neural network as one form of a dynamical system, so their robustness performance could be analyzed using the techniques of Lyapunov and contraction theory described in Sec. 9.5.I. These approaches typically rewrite a neural network as a discrete-time dynamical system, and utilize the stability constraint

of Sec. 9.5.IV as either a regularization loss or a structural constraint of neural networks [23]–[26], [52], [53]. This permits us to augment them with stronger robustness and stability guarantees that could further tighten the optimality gap of Theorem 9.1.

### 9.5.VI Online Learning

The proposed learning-based algorithm is based solely on offline learning and online guarantees of robustness and stability, but there could be situations where the parametric or non-parametric uncertainty of underlying dynamical systems is too large to be treated robustly. As shown in [7], robust control techniques, including our proposed approach in this chapter, can always be augmented with adaptive control techniques with formal stability [17], [54]–[57] and with static or dynamic regret bounds [58], [59] for online nonlinear control problems [60].

### References

- [1] J. Castillo-Rogez and K. Meech, “Mission design study for objects on oort cloud comet Orbits,” in *Community White Papers Planet. Sci. Decadal Surv.*, 2020.
- [2] H. Tsukamoto, S.-J. Chung, B. Donitz, M. Ingham, D. Mages, and Y. K. Nakka, “Neural-Rendezvous: Learning-based robust guidance and control to encounter interstellar objects,” *AIAA J. Guid. Control Dyn.*, under review, Aug. 2022.
- [3] C. Yu, G. Shi, S.-J. Chung, Y. Yue, and A. Wierman, “The power of predictions in online control,” in *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1994–2004.
- [4] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, “Neural ordinary differential equations,” in *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.
- [5] S. M. Richards, N. Azizan, J.-J. Slotine, and M. Pavone, “Adaptive-control-oriented meta-learning for nonlinear systems,” in *Robotics: Science and Systems*, 2021.
- [6] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, “Learning-based model predictive control: Toward safe learning in control,” *Annu. Rev. Control Robot. Auton. Syst.*, vol. 3, no. 1, pp. 269–296, 2020.
- [7] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview,” *Annu. Rev. Control*, vol. 52, pp. 135–169, 2021, ISSN: 1367-5788.

- [8] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Upper Saddle River, NJ: Pearson, 1991.
- [9] Y. Nakka, R. Foust, E. Lupu, *et al.*, “A six degree-of-freedom spacecraft dynamics simulator for formation control research,” in *Astrodyn. Special Conf.*, 2018.
- [10] B. P. S. Donitz, D. Mages, H. Tsukamoto, *et al.*, “Interstellar object accessibility and mission design,” in *IEEE Aerosp. Conf.*, 2022.
- [11] D. Mages, D. Landau, B. Donitz, and S. Bhaskaran, “Navigation evaluation for fast interstellar object flybys,” *Acta Astronaut.*, vol. 191, pp. 359–373, 2022, ISSN: 0094-5765.
- [12] A. Bemporad and M. Morari, “Robust model predictive control: A survey,” in *Robustness in identification and control*, London: Springer London, 1999, pp. 207–226.
- [13] D. Morgan, S.-J. Chung, L. Blackmore, B. Acikmese, D. Bayard, and F. Y. Hadaegh, “Swarm-keeping strategies for spacecraft under  $J_2$  and atmospheric drag perturbations,” *AIAA J. Guid. Control Dyn.*, vol. 35, no. 5, pp. 1492–1506, 2012.
- [14] J. L. Junkins and H. Schaub, *Analytical mechanics of space systems*, 2nd. AIAA, Apr. 2018.
- [15] S.-J. Chung, U. Ahsun, and J.-J. E. Slotine, “Application of synchronization to formation flying spacecraft: Lagrangian approach,” *AIAA J. Guid. Control Dyn.*, vol. 32, no. 2, pp. 512–526, 2009.
- [16] F.-Y. Wang, H. Zhang, and D. Liu, “Adaptive dynamic programming: An introduction,” *IEEE Comput. Intell. Magazine*, vol. 4, no. 2, pp. 39–47, 2009. DOI: 10.1109/MCI.2009.932261.
- [17] M. O’Connell, G. Shi, X. Shi, *et al.*, “Neural-Fly enables rapid learning for agile flight in strong winds,” *Sci. Robot.*, vol. 7, no. 66, eabm6597, 2022.
- [18] Y. K. Nakka, W. Hönig, C. Choi, A. Harvard, A. Rahmani, and S.-J. Chung, “Information-based guidance and control architecture for multi-spacecraft on-orbit inspection,” in *AIAA Scitech 2021 Forum*, 2021.
- [19] M. Morari and J. H. Lee, “Model predictive control: Past, present and future,” *Comput. Chem. Eng.*, vol. 23, no. 4, pp. 667–682, 1999, ISSN: 0098-1354.
- [20] D. Morgan, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, “Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming,” *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1261–1285, 2016.
- [21] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” in *Int. Conf. Learn. Representations*, 2018.

- [22] P. Kloeden and M. Rasmussen, *Nonautonomous Dynamical Systems*. Aug. 2011, vol. 176, ISBN: 978-0-8218-6871-3.
- [23] M. Zakwan, L. Xu, and G. Ferrari-Trecate, *On robust classification using contractive Hamiltonian neural ODEs*, arXiv:2203.11805, Mar. 2022.
- [24] I. D. J. Rodriguez, A. D. Ames, and Y. Yue, *LyaNet: A Lyapunov framework for training neural ODEs*, arXiv:2202.02526, Feb. 2022.
- [25] S. Jafarpour, A. Davydov, A. Proskurnikov, and F. Bullo, “Robust implicit networks via non-Euclidean contractions,” in *Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 9857–9868.
- [26] M. Revay, R. Wang, and I. R. Manchester, “Recurrent equilibrium networks: Unconstrained learning of stable and robust dynamical models,” in *IEEE Conf. Decis. Control*, 2021, pp. 2282–2287.
- [27] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, “A review of space robotics technologies for on-orbit servicing,” *Prog. Aerosp. Sci.*, vol. 68, pp. 1–26, 2014, ISSN: 0376-0421.
- [28] D. Mellinger, N. Michael, and V. Kumar, “Trajectory generation and control for precise aggressive maneuvers with quadrotors,” *Int. J. Robot. Res.*, vol. 31, no. 5, pp. 664–674, 2012.
- [29] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, “Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics,” *IEEE Trans. Autom. Control*, vol. 59, no. 4, pp. 876–891, 2014.
- [30] H. J. Kushner, *Stochastic Stability and Control*, English. Academic Press New York, 1967, xiv, 161 p.
- [31] A. D. Ames, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs with application to adaptive cruise control,” in *IEEE Conf. Decis. Control*, 2014, pp. 6271–6278.
- [32] H. Tsukamoto and S.-J. Chung, “Robust controller design for stochastic nonlinear systems via convex optimization,” *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4731–4746, 2021.
- [33] H. K. Khalil, *Nonlinear Systems*, 3rd. Upper Saddle River, NJ: Prentice-Hall, 2002.
- [34] W. Lohmiller and J.-J. E. Slotine, “On contraction analysis for nonlinear systems,” *Automatica*, vol. 34, no. 6, pp. 683–696, 1998, ISSN: 0005-1098.
- [35] I. R. Manchester and J.-J. E. Slotine, “Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design,” *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 3046–3053, Jun. 2017.
- [36] S. Singh, A. Majumdar, J.-J. E. Slotine, and M. Pavone, “Robust online motion planning via contraction theory and convex optimization,” in *IEEE Int. Conf. Robot. Automat.*, May 2017, pp. 5883–5890.

- [37] H. Tsukamoto and S.-J. Chung, “Neural contraction metrics for robust estimation and control: A convex optimization approach,” *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 211–216, 2021.
- [38] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Neural stochastic contraction metrics for learning-based control and estimation,” *IEEE Control Syst. Lett.*, vol. 5, no. 5, pp. 1825–1830, 2021.
- [39] H. Tsukamoto and S.-J. Chung, “Learning-based robust motion planning with guaranteed stability: A contraction theory approach,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6164–6171, 2021.
- [40] S.-J. Chung, “Nonlinear control and synchronization of multiple Lagrangian systems with application to tethered formation flight spacecraft,” Ph.D. dissertation, MIT, 2007.
- [41] H. Tsukamoto, S.-J. Chung, J.-J. Slotine, and C. Fan, “A theoretical overview of neural contraction metrics for learning-based control with guaranteed stability,” in *IEEE Conf. Decis. Control*, 2021, pp. 2949–2954.
- [42] S. Han and S.-J. Chung, *Incremental nonlinear stability analysis for stochastic systems perturbed by Lévy noise*, arXiv:2103.13338, Mar. 2021.
- [43] Y. K. Nakka and S.-J. Chung, “Trajectory optimization of chance-constrained nonlinear stochastic systems for motion planning and control,” *IEEE Trans. Robot.*, vol. to appear, 2022.
- [44] L. Arnold, *Stochastic Differential Equations: Theory and Applications*. Wiley, 1974.
- [45] B. Rivière, W. Hönig, Y. Yue, and S.-J. Chung, “GLAS: Global-to-local safe autonomy synthesis for multi-robot motion planning with end-to-end learning,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4249–4256, 2020.
- [46] H. Tsukamoto, B. Rivière, C. Choi, A. Rahmani, and S.-J. Chung, “CART: Collision avoidance and robust tracking augmentation in learning-based motion planning for multi-agent systems,” in *IEEE Conf. Decis. Control*, under review, 2023.
- [47] J.-J. E. Slotine, W. Wang, and K. El Rifai, “Contraction analysis of synchronization in networks of nonlinearly coupled oscillators,” in *Int. Symp. Math. Theory Netw. Syst.*, Jul. 2004.
- [48] W. Lohmiller and J.-J. E. Slotine, “Nonlinear process control using contraction theory,” *AIChE Journal*, vol. 46, pp. 588–596, Mar. 2000.
- [49] Q. Pham, “Analysis of discrete and hybrid stochastic systems by nonlinear contraction theory,” in *Int. Conf. Control Automat. Robot. Vision*, Dec. 2008, pp. 1054–1059.

- [50] H. Tsukamoto and S.-J. Chung, “Convex optimization-based controller design for stochastic nonlinear systems using contraction analysis,” in *IEEE Conf. Decis. Control*, Dec. 2019, pp. 8196–8203.
- [51] R. Wang, R. Tóth, and I. R. Manchester, *Discrete-time contraction-based control of nonlinear systems with parametric uncertainties using neural networks*, arXiv:2105.05432, May 2021.
- [52] S. Jafarpour, M. Abate, A. Davydov, F. Bullo, and S. Coogan, “Robustness certificates for implicit neural networks: A mixed monotone contractive approach,” in *Proc. Conf. Learn. Dyn. Control, To Appear*, Nov. 2022.
- [53] I. R. Manchester, M. Revay, and R. Wang, “Contraction-based methods for stable identification and robust machine learning: A tutorial,” in *IEEE Conf. Decis. Control*, 2021, pp. 2955–2962.
- [54] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Learning-based Adaptive Control using Contraction Theory,” in *IEEE Conf. Decis. Control*, 2021, pp. 2533–2538.
- [55] B. T. Lopez and J.-J. E. Slotine, “Universal adaptive control of nonlinear systems,” *IEEE Control Syst. Lett.*, vol. 6, pp. 1826–1830, 2022.
- [56] G. Shi, X. Shi, M. O’Connell, *et al.*, “Neural lander: Stable drone landing control using learned dynamics,” in *IEEE Int. Conf. Robot. Automat.*, May 2019, pp. 9784–9790.
- [57] Y. K. Nakka, A. Liu, G. Shi, A. Anandkumar, Y. Yue, and S.-J. Chung, “Chance-constrained trajectory optimization for safe exploration and learning of nonlinear systems,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 389–396, 2021.
- [58] N. M. Boffi, S. Tu, and J.-J. E. Slotine, “Regret bounds for adaptive nonlinear control,” in *Proc. Conf. Learn. Dyn. Control*, ser. Proc. Mach. Learn. Res. Vol. 144, Jun. 2021, pp. 471–483.
- [59] G. Shi, K. Azizzadenesheli, M. O’Connell, S.-J. Chung, and Y. Yue, “Meta-adaptive nonlinear control: Theory and algorithms,” in *Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 10 013–10 025.
- [60] S. Kakade, A. Krishnamurthy, K. Lowrey, M. Ohnishi, and W. Sun, “Information theoretic regret bounds for online nonlinear control,” in *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 15 312–15 325.

*Chapter 10*ADDITIONAL NUMERICAL SIMULATIONS AND EMPIRICAL  
VALIDATION

- [1] H. Tsukamoto, S.-J. Chung, B. Donitz, M. Ingham, D. Mages, and Y. K. Nakka, “Neural-Rendezvous: Learning-based robust guidance and control to encounter interstellar objects,” *AIAA J. Guid. Control Dyn.*, *under review*, Aug. 2022.
- [2] B. P. S. Donitz, D. Mages, H. Tsukamoto, *et al.*, “Interstellar object accessibility and mission design,” in *IEEE Aerosp. Conf.*, 2022.
- [3] H. Tsukamoto, B. Rivière, C. Choi, A. Rahmani, and S.-J. Chung, “CART: Collision avoidance and robust tracking augmentation in learning-based motion planning for multi-agent systems,” in *IEEE Conf. Decis. Control*, *under review*, 2023.

Real-world systems always involve unmodeled uncertainties not just in their state, but also in the dynamics, environment, and control actuation, which could all lead to destabilizing and unexpected behaviors different from what we learned and observed in simulations with a nominal problem setting. As discussed in the previous chapters, our learning-based approaches are shown to be provably robust against these uncertainties, thereby helping to reproduce the nominal simulation results seamlessly in real-world environments. We assess the validity of this argument using several numerical simulations with external perturbation and empirical validation with actual hardware, performed as part of the ISO exploration project and the CASTOR project in collaboration with NASA JPL.

**Remark 10.1.** *In order for readers to easily understand the motivations of this chapter, let us recall that the discovery and exploration of ISOs are challenging for three main reasons: 1) they are not discovered until they are close to Earth, meaning that launches to encounter them often require high launch energy; 2) their orbital properties are poorly constrained at launch, generally leading to significant on-board resources to encounter; and 3) the encounter speeds are typically high ( $> 10$  of km/s) requiring fast response autonomous operations. Our approach, Neural-Rendezvous, can be viewed as one way to realize such autonomous operations,*

*especially when the computational resource of the agent is limited and not sufficient to perform nonlinear optimization in real time.*

*Also, when we consider more challenging and highly nonlinear G&C scenarios, solving even a single optimization online could become unrealistic. This chapter also discusses such cases in distributed, high-conflict, and optimal guidance and control of swarms of UAVs.*

## **10.1 Numerical Simulations for ISO Exploration**

Our proposed framework, Neural-Rendezvous, is demonstrated using the data set that contains ISO candidates for possible exploration [1] to validate if it indeed solves Problem 9.1 introduced earlier in Sec. 9.1. **PyTorch** is used for designing and training neural networks and NASA's Navigation and Ancillary Information Facility (**NAIF**) is used to obtain relevant planetary data. A YouTube video that visualizes these simulation results can be found at [https://youtu.be/8h60B\\_p1fyQ](https://youtu.be/8h60B_p1fyQ).

### **10.1.I Simulation Setup**

All the G&C frameworks in this section are implemented with the control time interval 1 s unless specified, and their computational time is measured using the MacBook Pro laptop (2.2 GHz Intel Core i7, 16 GB 1600 MHz DDR3 RAM). The terminal time  $t_f$  of (9.4) for terminal guidance is selected to be  $t_f = 86400$  (s), and the wet mass of the spacecraft at the beginning of terminal guidance is assumed to be 150 kg. Also, we consider the SN-DNN min-norm control (9.17) of Theorem 9.1 designed with  $\Lambda = 1.3 \times 10^{-3}$  and  $\alpha = 8.9 \times 10^{-7}$ . The maximum control input is assumed to be  $u_{\max} = 3$  (N) in each direction with the total admissible delta-V (2-norm S/C velocity increase) being 0.6 km/s.

### **10.1.II Dynamical System-Based SN-DNN Training**

This section delineates how we train the dynamical system-based SN-DNN of Sec. 9.2 for constructing the leaning-based terminal guidance algorithm.

#### **10.1.II-A State Uncertainty Assumption**

For the sake of simplicity, we assume that the spacecraft has access to the estimated ISO and its relative state generated by the respective normal distribution  $\mathcal{N}(\mu, \Sigma)$ , with  $\mu$  being the true state and  $\Sigma$  being the navigation error covariance, where  $\text{Tr}(\Sigma)$  exponentially decaying in  $t$  as in Example 9.1. In particular, we assume that the standard deviations of the ISO and spacecraft absolute along-track position,

cross-track position, along-track velocity, and cross-track velocity, expressed in the ECLIPJ2000 frame as in JPL's [SPICE toolkit](#), are  $10^4$  km,  $10^2$  km,  $10^{-2}$  km/s, and  $10^{-2}$  km/s initially at time  $t = 0$  (s), and decays to  $\mathcal{O}(10^1)$  km,  $\mathcal{O}(10^0)$  km,  $\mathcal{O}(10^{-4})$  km/s, and  $\mathcal{O}(10^{-4})$  km/s finally at time  $t = t_f = 86400$  (s) in the along-track and cross-track direction, respectively, which can be achieved by using, e.g., the extended Kalman filter with full state measurements and estimation gains given by  $R = I_{n \times n}$  and  $Q = I_{n \times n} \times 10^{-10}$ .

**Remark 10.2.** *As discussed in Remark 9.1, G&C are the focus of our study and navigation is beyond our scope, we have simply assumed the ISO state measurement uncertainty given above partially following the discussion of [1]. The assumption can be easily modified accordingly to the state estimation schemes to be used in each aerospace and robotic problem of interest. See Sec. 10.1.III-A for more discussion.*

### 10.1.II-B Training Data Generation

We generate 499 candidate ISO and spacecraft ideal trajectories based on the ISO population analyzed in [1], and utilized the first 399 ISOs for training the SN-DNN and the other 100 for testing its performance later in this section. We then obtain 10000 time and ISO index pairs  $(t_i, I_i)$  uniformly and randomly from  $[0, t_f] \times [1, 399] \cup \mathbb{N}$  and perturbed the ISO and spacecraft ideal state with the uncertainty given in Sec. 10.1.II-A to produce the training samples  $(\bar{x}_i, \bar{\alpha}_i, \bar{t}_i)$  of (9.9) for the control input loss (i.e., the first term of (9.8)). The training data samples for the state trajectory loss (i.e., the second term of (9.8)) are obtained in the same way using the pairs generated uniformly and randomly from  $[0, 3600] \times [1, 399] \cup \mathbb{N}$ , with  $\Delta \bar{t}_i$  of (9.9) fixed to  $\Delta \bar{t}_i = 10$  (s). The desired relative positions  $\bar{\rho}_i$  in (9.9) are also sampled uniformly and randomly from the surface of a ball with radius 100 km.

The desired control input samples  $u_{\text{mpc}}(\bar{x}_i, \bar{\alpha}_i, \bar{t}_i, \bar{\rho}_i)$  and desired state trajectory samples  $\varphi_{\text{mpc}}^{\bar{t}_i + \Delta \bar{t}_i}(\bar{x}_i, \bar{\alpha}_i, \bar{t}_i, \bar{\rho}_i)$  of (9.9) are then generated by solving (9.5) using the sequential convex programming approach and by numerically integrating (9.7) using the fourth-order Runge–Kutta method, respectively, where the terminal position error is treated as a constraint  $\|p_\xi(t_f) - \rho\| = 0$  in (9.4), the cost function of (9.3) is defined with  $c_0 = 0$ ,  $c_1 = 1$ , and  $P(u(t), \xi(t)) = \|u(t)\|^2$  as in Remark 9.2, and the control input constraint  $u(t) \in \mathcal{U}(t) = \{u \in \mathbb{R}^m \mid |u_i| \leq u_{\text{max}}\}$   $u_i$  is used with  $u_i$  being the  $i$ th element of  $u$  and  $u_{\text{max}} = 3$  (N). Note that the problem is discretized with the time step 1 s, consistently with the control time interval.

### 10.1.II-C Training Data Normalization

Instead of naively training the SN-DNN with the raw data generated in Sec. 10.1.II-B, we transform the SN-DNN input data  $(\bar{x}_i, \bar{\alpha}_i, \bar{t}_i, \bar{\rho}_i)$  as follows, thereby accelerating the speed of learning process and improving neural network generalization performance:

$$\text{SN-DNN input} = \left( \bar{p}_i - \bar{\rho}_i, \frac{\bar{p}_i - \bar{\rho}_i}{t_f - \bar{t}_i} + \dot{\bar{p}}_i, \bar{\rho}_i, t_f - \bar{t}_i, \bar{\omega}_{z,i}, G(\bar{p}_i, \bar{\alpha}_i) \right) \quad (10.1)$$

where  $\bar{x}_i = [\bar{p}_i^\top, \dot{\bar{p}}_i^\top]^\top$ ,  $G(p, \alpha)$  is given in (9.2), and  $\bar{\omega}_{z,i}$  is the  $i$ th training sample of the orbital element  $\omega_z$  given in [2], which is the dominant element of the matrix function  $C(\alpha)$  of (9.2). We further normalize the input (10.1) and output  $u_\ell(x, \alpha, t, \rho; \theta_{\text{nn}})$  of the SN-DNN by dividing them by their maximum absolute values in their respective training data.

### 10.1.II-D SN-DNN Configuration and Training

We select the number of hidden layers and neurons of the SN-DNN as 6 and 64, with the spectral normalization constant ( $C_{nn}$  of Definition 6.2 in [3]) being  $C_{nn} = 25$ . The activation function is selected to be tanh, which is also used in the last layer not to violate the input constraint  $u(t) \in \mathcal{U}(t) = \{u \in \mathbb{R}^m \mid |u_i| \leq u_{\max}\} u_i$  by design. Figure 10.1 shows the terminal spacecraft position error (delivery error) and control effort (total delta-V) of the SN-DNN terminal guidance policy trained for 10000 epochs using several different weights  $c_u, c_x \in \mathbb{R}_{\geq 0}$  of the loss function (9.8) in Sec. 9.2, where its weight matrices are given as  $C_x = c_x \text{diag}(\mathbf{I}_{3 \times 3}, 10^7 \times \mathbf{I}_{3 \times 3})$  and  $C_u = c_u \mathbf{I}_{3 \times 3}$ . The results are averaged over 50 simulations for the ISOs in the test set without any state uncertainty. Consistently with the definition of (9.8), this figure indicates that

- as  $c_x/c_u$  gets smaller, the loss function (9.8) penalizes the imitation loss of the control input more heavily than that of the state trajectory, and thus the spacecraft yields smaller control effort but with larger delivery error, and
- as  $c_x/c_u$  gets larger, the loss function (9.8) penalizes the imitation loss of the state trajectory more heavily than that of the control input, and thus the spacecraft yields smaller delivery error but with larger control effort.

The weight ratio  $c_x/c_u$  of the SN-DNN to be implemented in the next section is selected as  $c_x/c_u = 10^2$ , which achieves the smallest delivery error with the smallest

standard deviation of all the weight ratios in Fig. 10.1, while having the control effort smaller than the admissible  $\Delta V$  of 0.6 km/s. Note that we could further optimize the ratio with the  $\Delta V$  constraint based on this trade-off discussed above, but this is left as future work. The SN-DNN is then trained using SGD for 10000 epochs with 10000 training data points obtained as in Sec. 10.1.II-B.

**Remark 10.3.** *The trend of Fig. 10.1 is also due to the fact that the delivery is treated as a hard constraint by MPC in this chapter as can be seen in (9.3). In general, the best choice of the ratio  $c_x/c_u$  will depend on how the motion planning is formulated and solved to sample training data (see, e.g., [4] for indirect approaches and [5] for direct approaches).*

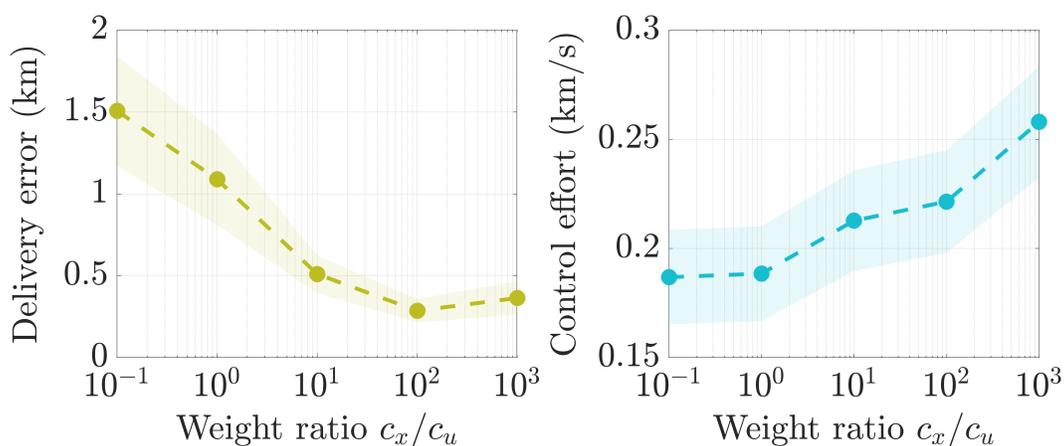


Figure 10.1: Control performances versus weight ratio of the SN-DNN loss. The shaded area denotes the standard deviations ( $\pm 2.5 \times 10^{-1}\sigma$  for the spacecraft delivery error and  $\pm 5 \times 10^{-2}\sigma$  for the control effort).

### 10.1.III Neural-Rendezvous Performance

Figure 10.2 shows the spacecraft delivery error and control effort of Neural-Rendezvous of Algorithm 1, SN-DNN terminal guidance of Sec. 9.2, PD guidance and control (i.e., PD control constructed to track a pre-computed and fixed desired trajectory, which is a JPL baseline), robust nonlinear tracking control of [6, pp. 397-402] (i.e., nonlinear sliding mode-type control for general Lagrangian systems to track a pre-computed and fixed desired trajectory), and MPC with linearized dynamics [7], where the SN-DNN min-norm control (9.17) of Theorem 9.1 is activated at time  $t = t_s = 26400$  (s). It can be seen that Neural-Rendezvous achieves  $\leq 0.2$  km delivery error for 99% of the ISOs in the test set, even under the presence of the large ISO state uncertainty given in Sec. 10.1.II-A. Also, its error is indeed less

than the dominant term of the expectation bound on the tracking error (9.27) of Theorem 9.1, which is computed assuming the estimation error is upper-bounded by a function that exponentially decreases in time as in Example 9.1. Furthermore, the SN-DNN terminal guidance can also achieve  $\leq 1$  km delivery error for 86 % of the ISOs, and as expected from the optimality gap given in (9.11) of Lemma 9.1, the control effort of Neural-Rendezvous is larger than that of the SN-DNN guidance and MPC with linearized dynamics, but it is still less than the admissible delta-V of 0.6 km/s for all the ISOs in the test set.

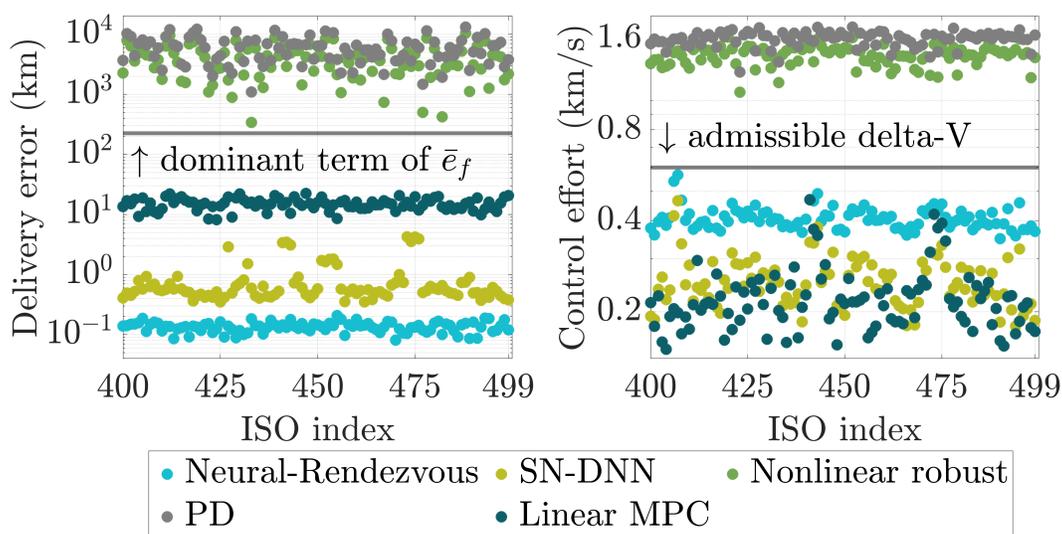


Figure 10.2: Control performances versus ISOs in the test set (400-499), where  $\bar{e}_f$  is the right-hand side of bound (9.27) in Theorem 9.1, the admissible delta-V (2-norm S/C velocity increase during the terminal guidance) is assumed to be 0.6 km/s, and each result shown above is the average of 10 simulations performed for each ISO.

Figure 10.3 then shows the spacecraft delivery error and control effort of Neural-Rendezvous of Algorithm 1 and SN-DNN terminal guidance of Sec. 9.2, averaged over 100 ISOs in the test set, versus the control time interval. Although both of these methods involve discretization when implementing them in practice as pointed out in Remark 9.7, it can be seen that Neural-Rendezvous enables having the delivery error smaller than 5 km with its standard deviation always smaller than that of the SN-DNN guidance, even for the control interval 600 s (10 min). Since the MPC problem is solved by discretizing it with the time step 1 s as explained in Sec. 10.1.II-B, and the SN-DNN min-norm control (9.17) of Theorem 9.1 is designed for continuous dynamics, Neural-Rendezvous yields less optimal control inputs for larger control time intervals, resulting in larger delivery error and control effort as expected from [8].

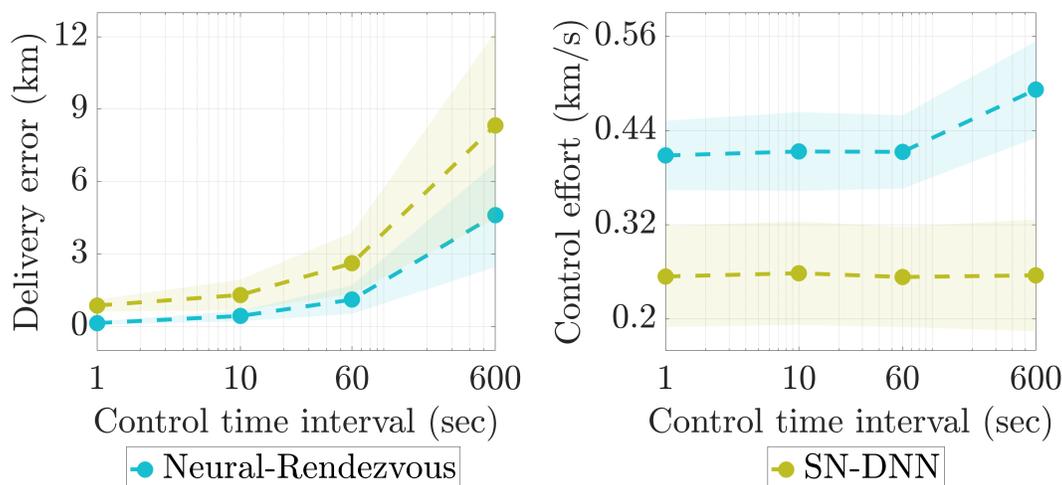


Figure 10.3: Control performances versus control time interval, where each result shown above is the average of 10 simulations performed for all 100 ISOs in the test set (400-499). The shaded area denotes the standard deviations ( $\pm 1\sigma$  both for the spacecraft delivery error and control effort).

The performance of the nonlinear MPC (the optimization-based solution we aim to reproduce with machine learning) in this mission is shown in Figure 10.4, where the robust nonlinear MPC is the nonlinear MPC with the min-norm feedback control of Theorem 9.1. The performance comparison between our learning-based control approaches and the MPC is summarized in Fig. 10.5. Comparing the SN-DNN guidance with the nonlinear MPC, we can see that that optimality gap is  $1.37 \times 10\%$  on average for the control effort, while the SN-DNN delivery error is  $1.82 \times 10^2\%$  larger than that of the MPC due to the lack of the delivery error guarantee, unlike Neural-Rendezvous. In contrast, the Neural-Rendezvous delivery error is only  $5.84 \times 10^{-3}\%$  larger than that of the robust nonlinear MPC at the slight expense of the control effort  $5.48\%$  larger than that of the robust nonlinear MPC. This is thanks to the formal probabilistic bound obtained in Theorem 9.1 equipped on top of the SN-DNN guidance, as expected.

Finally, as shown in Table 10.1, we can see that all the methods presented in this section including our proposed approaches, except for the global solution obtained with the nonlinear MPC, can be computed in  $\leq 1$  (s) and thus can be implemented in real-time. The observations so far imply that the proposed approach indeed provides one of the promising solutions to Problem 9.1 of Sec. 9.1. The simulation results summarized in this section are visualized at [https://youtu.be/8h60B\\_p1fyQ](https://youtu.be/8h60B_p1fyQ) as illustrated in Fig. 10.6.

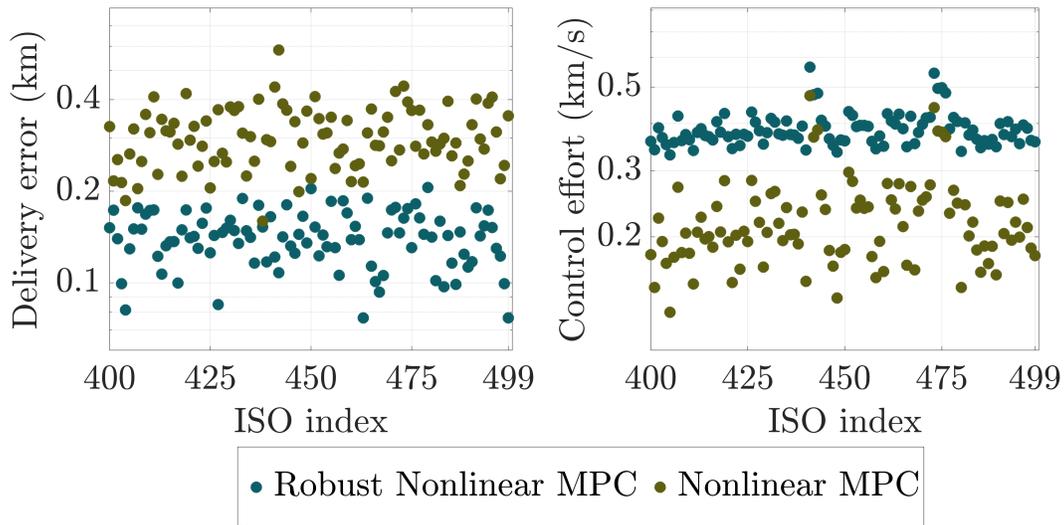


Figure 10.4: Control performances versus ISOs in the test set (400-499). Each result shown above is the average of 10 simulations performed for each ISO.

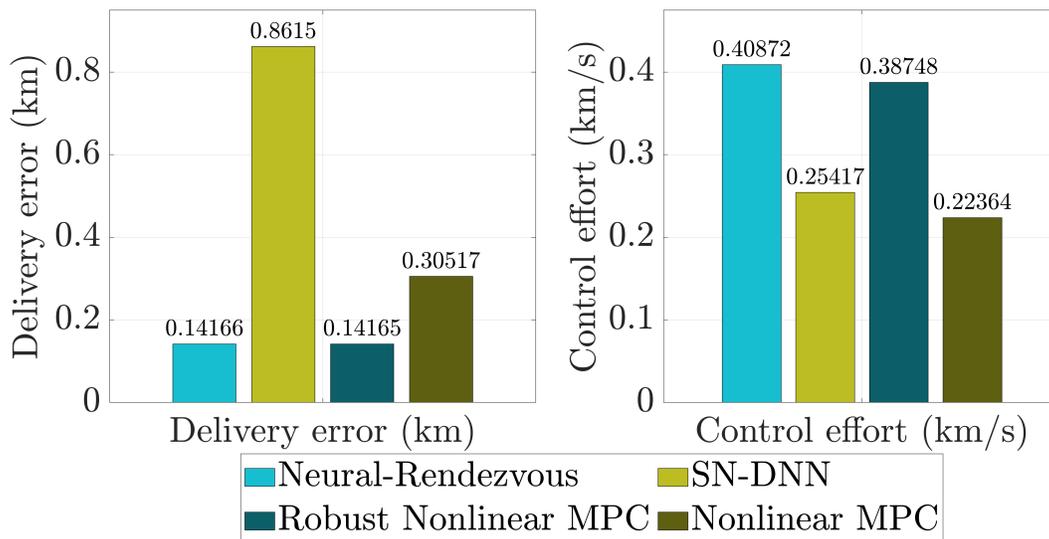


Figure 10.5: Performance comparison between our learning-based approaches and MPC. Each result shown above is the average of all ISOs in Fig. 10.2 and Fig. 10.4.

Table 10.1: Computational time of each method for the ISO encounter averaged over 100 evaluations.

	Average computational time for one step (s)
Neural-Rendezvous	$8.0 \times 10^{-4}$
SN-DNN	$5.7 \times 10^{-5}$
Nonlinear robust control	$1.2 \times 10^{-4}$
PD control	$4.3 \times 10^{-5}$
Linear approx. MPC	$1.5 \times 10^{-4}$
Global solution	$2.0 \times 10^3$

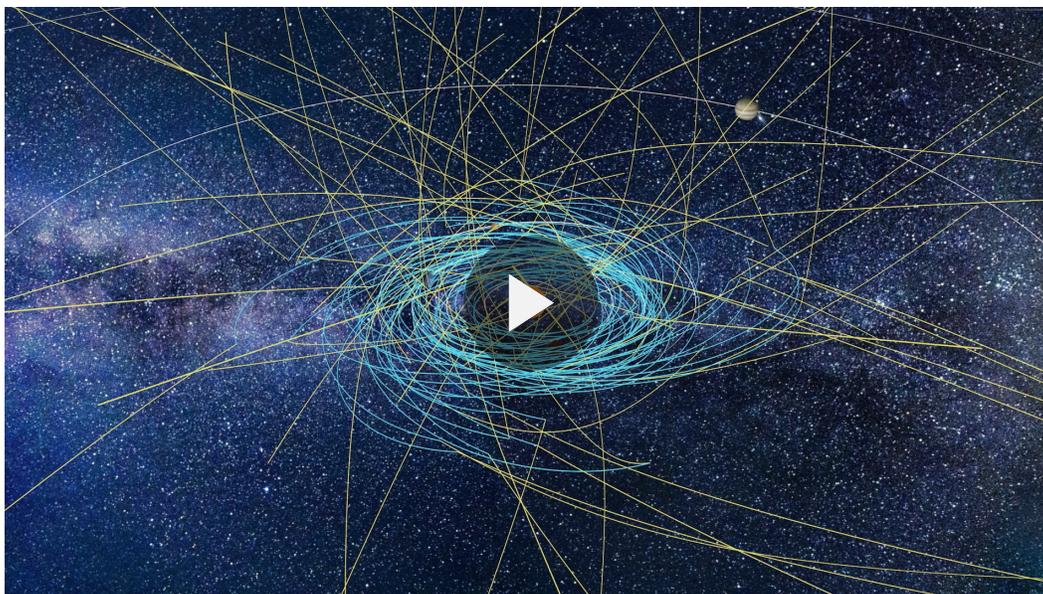


Figure 10.6: Visualized Neural-Rendezvous trajectories for ISO exploration, where yellow curves represent ISO trajectories and blue curves represent spacecraft trajectories. More details can be found [here](#).

### 10.1.III-A Numerical Simulations at NASA JPL

The comparison of the performance of Neural-Rendezvous with the JPL state-of-practice autonomous navigation systems for small bodies, including ISOs, is discussed in [9]. Neural-Rendezvous is demonstrated to outperform them under realistic ISO state uncertainty in terms of the spacecraft delivery error, under mild GNC assumptions. Note that the uncertainty history is derived from an autonomous optical navigation orbit determination filter, as is used in the AutoNav system [10], where a Monte-Carlo analysis simulating AutoNav performance is run to provide state estimation error versus time results.

## 10.2 Empirical Validation for ISO Exploration

Applying deep learning-based algorithms to real-world systems always involves unmodeled uncertainties that may not be captured in numerical simulations. Using the incremental stability-based analysis as in the proof of Theorem 9.1, we can further show that Neural-Rendezvous guarantees robustness against bounded and stochastic external disturbances and uncertainties in the dynamics and control actuation, including the SN-DNN learning error, in a partially unknown environment [3] as seen in the previous chapters. Such a strong mathematical guarantee helps us reproduce the simulation results seamlessly in real-world hardware experiments, inherently with various sources of uncertainties.

### 10.2.I Spacecraft Simulators

We first test the performance using our spacecraft simulator called M-STAR [11] and epoxy flat floors for spacecraft motion simulation with the Vicon motion capture system (<https://www.vicon.com/>). We have 14 motion capture cameras on the ceiling of this facility, with an IMU mounted on each spacecraft simulator for estimating the pose. The flatness of the epoxy floor is maintained within 0.001 inches for frictionless translation of the spacecraft dynamics using 3 flat air-bearing pads, so we can properly demonstrate its motion in deep space. Each simulator has 8 thrusters in addition to the 3 air-bearing pads mounted at the bottom, which are to be used for controlling its position and attitude as in actual spacecraft. The left-hand side of Fig. 10.7 shows the spacecraft simulators and the ISO model used for this empirical validation.

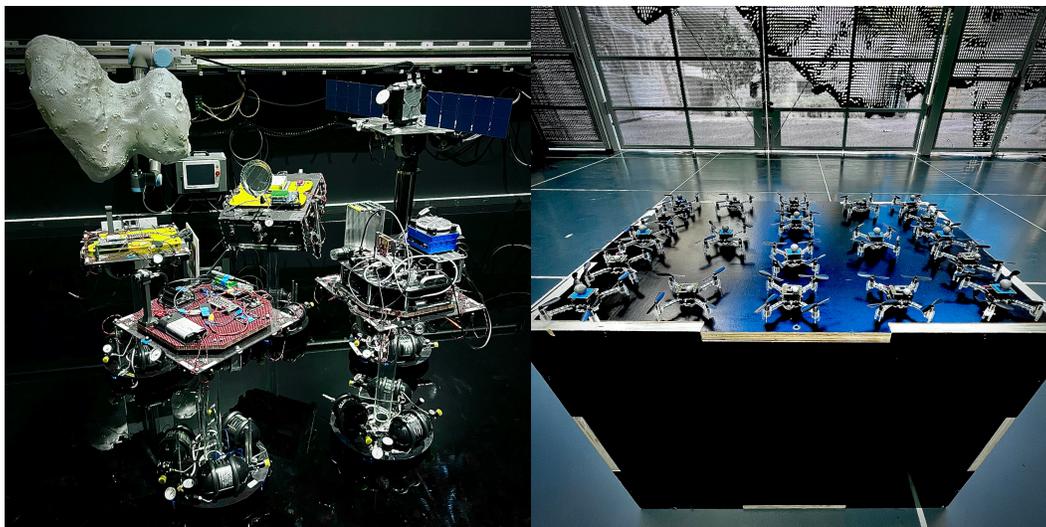


Figure 10.7: Robotic platforms. Left: Multi-Spacecraft Testbed for Autonomy Research (M-STAR). Right: Crazyflie, a versatile open-source flying development platform that only weighs 27 g.

### 10.2.I-A Relating M-STAR Dynamics to Spacecraft Dynamics Relative to ISO

Although the M-STAR actuation works as in spacecraft in deep space due to the epoxy flat floor and thruster-based control, its dynamics given in [11] is different from the one we consider in Chapter 9 as (9.1). Also, due to the spatial limitation of the facility, we need to scale down the position, velocity, and control input of (9.1) to the ones of M-STAR (see Table 10.2).

Table 10.2: Scales of the state and control in each dynamics.

	S/C w.r.t. to ISO	M-STAR
Position (km)	$\mathcal{O}(10^6)$	$\mathcal{O}(10^{-3})$
Velocity (km/s)	$\mathcal{O}(10^1)$	$\mathcal{O}(10^{-4})$
Control (km/s <sup>2</sup> )	$\mathcal{O}(10^{-5})$	$\mathcal{O}(10^{-5})$

To this end, we consider the following state  $x_{\text{sim}}$  and time  $t_{\text{sim}}$ :

$$x_{\text{sim}} = [p_{\text{sim}}, v_{\text{sim}}]^\top = \left[ \frac{p(t) - (p(0) + \dot{p}(0)t)}{c_p}, \frac{\dot{p}(t) - \dot{p}(0)}{c_v} \right]^\top, \quad t_{\text{sim}} = \frac{t}{c_t}$$

where  $t$  is the actual time of (9.1),  $p$  is the position of the spacecraft relative to the ISO as in (9.2), and  $c_p$ ,  $c_v$ , and  $c_t$  are constant scaling parameters. Taking the time derivative of  $x_{\text{sim}}$  with respect to  $t_{\text{sim}}$ , we get

$$\frac{dx_{\text{sim}}}{dt_{\text{sim}}} = c_t [\dot{p}_{\text{sim}}, \dot{v}_{\text{sim}}]^\top = \left[ \frac{c_v c_t}{c_p} v_{\text{sim}}, \frac{c_t}{c_v m_{\text{sc}}(t)} (-C(\varrho)\dot{p} - G(p, \varrho) + u) \right]^\top$$

where  $\varrho$  is the ISO state,  $u$  is the spacecraft control input, and  $C$  and  $G$  are the matrix functions defined in (9.1) and (9.2). To be dynamically consistent, we select  $c_p$ ,  $c_v$ , and  $c_t$  to satisfy  $c_v c_t / c_p = 1$ . If we allocate the control input of the 8 thrusters of M-STAR to have its 3-dimensional acceleration vector  $u_{\text{sim}}$  as follows:

$$u_{\text{sim}} = \frac{c_t}{c_v m_{\text{sc}}(t)} (-C(\varrho)\dot{p} - G(p, \varrho) + u),$$

then we can convert the control input  $u$  obtained by Neural-Rendezvous to the spacecraft simulator control input  $u_{\text{sim}}$ , to be applied to the scaled-down M-STAR dynamics. Note that these procedures rewrite the dynamics as the double integrator dynamics, which can be easily demonstrated using the generalized pseudo-inverse control allocation scheme proposed in [11].

### 10.2.I-B Experimental Setup and Results

We select NVIDIA Jetson TX2 as the main onboard computer to run the GNC and perception algorithms, where the software architecture is built on Robotic Operating System (ROS) framework. The 8 thrusters are controlled at 2 Hz control frequency (control input every 0.5 seconds) with the motion capture camera system running at 100 Hz, which encourages the use of computationally efficient, deep learning-based control schemes in place of optimization-based control schemes. The state uncertainty in the M-STAR dynamics is added externally assuming the same uncertainty as in Sec. 10.1 scaled down by the method detailed in Sec. 10.2.I-A. We select 25 spacecraft relative state trajectories out of the ones given in Sec. 10.1, which meet the spatial constraint of our spacecraft simulator facility (6.27 m in  $x$  direction and 8.05 m in  $y$  direction, see Table 10.2) after the scaling-down procedures in Sec. 10.2.I-A. Note that our spacecraft simulator facility can demonstrate 2-dimensional motion although the spacecraft in the actual mission moves in 3-dimensional space.

The demonstrated trajectories of the M-STAR are shown in Fig. 10.8 and the trajectories observed in the actual environment are shown in Fig. 10.9. The scaled-down trajectories of Fig. 10.8 and Fig. 10.9 correspond to the ones of the ISO candidates

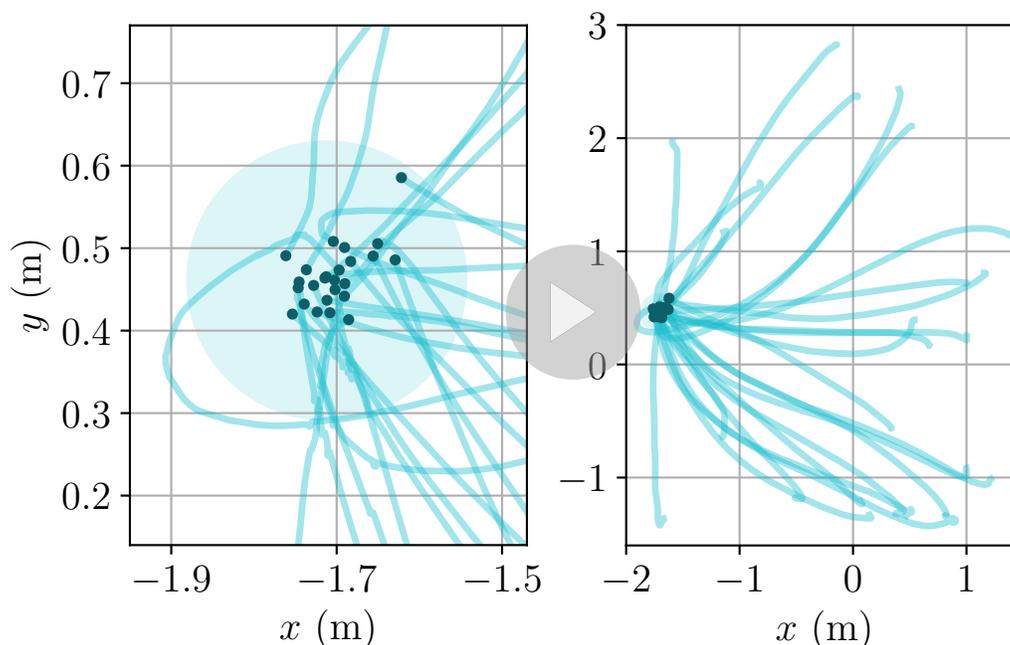


Figure 10.8: Demonstrated Neural-Rendezvous trajectories of M-STAR, where the shaded yellow circle indicates the expected delivery error bound (the left figure is a magnified view of the right figure).

400, 405, 406, 407, 408, 422, 425, 426, 430, 436, 437, 438, 439, 460, 461, 462, 463, 479, 480, 481, 490, 491, 492, 493, 494, simulated in Sec. 10.1. As can be seen from the figures and the [movie](#), Neural-Rendezvous indeed satisfies the probabilistic bound computed by Theorem 9.1, which is indicated by the yellow ball in Fig. 10.8 and scaled down to the bound of the M-STAR dynamics using the method of Sec. 10.2.I-A.

### 10.2.II High-Conflict and Distributed UAV Swarm Reconfiguration

As we briefly mentioned in the introduction, Neural-Rendezvous and its guarantees are general enough to be used for other G&C problems with completely different dynamics and objectives. To demonstrate this point, this section considers the problem of high-conflict reconfiguration of a UAV swarm, where the problem's complexity arises from the dynamics' nonlinearity, the aerodynamic nonlinear interaction between each UAV that acts as external disturbance, and highly nonlinear and distributed optimization required to efficiently avoid collisions considering the motions of multiple UAVs at each time instance, even with their limited communication radius. Another implicit focus of this validation is simply to demonstrate the performance of Neural-Rendezvous in the 3-dimensional space in addition to the

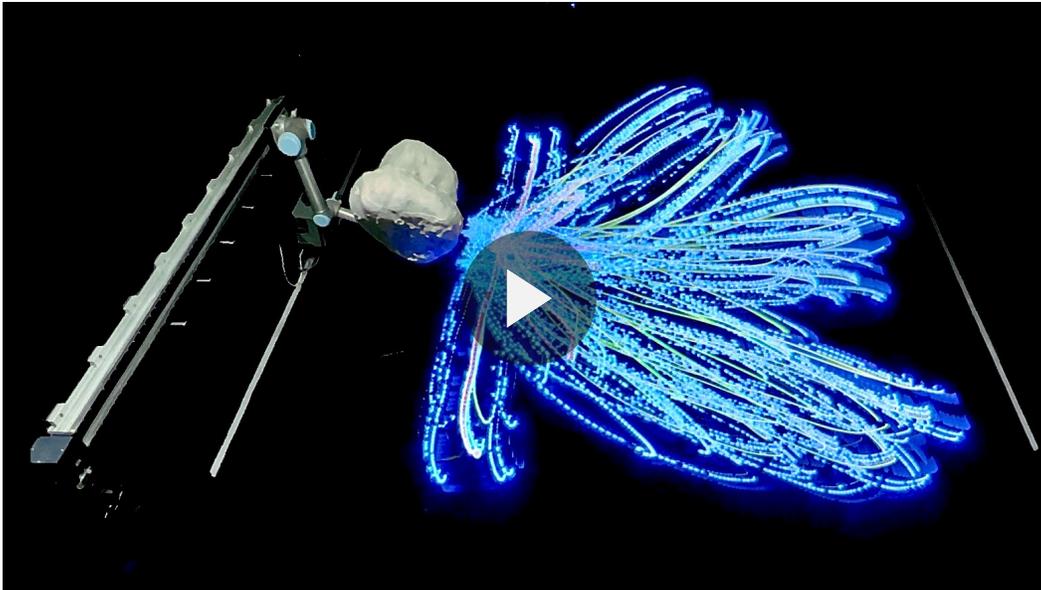


Figure 10.9: Demonstrated Neural-Rendezvous trajectories of M-STAR observed in the actual environment, illustrated using their light trails. This movie can be found [here](#).

2-dimensional setup in Sec. 10.2.I. We use 20 crazyflies shown in the right-hand side of Fig. 10.7, which are designed to be a versatile open-source flying development platform to perform various types of aerial robotics research. The product description can be found here <https://www.bitcraze.io/products/crazyflie-2-1/>.

### 10.2.II-A Problem Statement

Given multiple crazyflies, our objective is to intelligently and distributedly control each UAV to move to randomized and high-conflict target positions from randomized initial positions, optimally in a distributed manner. We solve nonlinear motion planning problems to sample target guidance and control inputs that minimize total control effort during the entire flight, using the sequential convex programming approach. The training is performed in a distributed manner as in [12], so we can use local observations to account for centralized global solutions even with the decentralized implementation of the G&C algorithm. The distributed communication radius is set to 2 m and the initial and target positions are randomly sampled from a 3-dimensional cuboid ( $x \in [-1.25, 1.25]$ ,  $y \in [0.60, 1.90]$ , and  $z \in [0.7, 2.7]$ , all in meters) under the conditions that the distances between the initial and target position of each UAV are at least 2.00 m apart and the distances between each initial/target position and the other initial/target positions are at least 0.60 m apart. The collision

avoidance constraint is implemented in the optimization using a tall ellipsoid with the lengths of the principal axes being 0.6 m, 0.6 m, and 1.2 m, in  $x$ ,  $y$ , and  $z$  directions, respectively, considering the downwash effect. The SN-DNN is trained as in Neural-Rendezvous with the robust learning approach proposed in [13], which provides machine learning-based nonlinear motion planners with formal robustness and stability guarantees, even under the presence of learning errors and external disturbances.

Some example global solution trajectories are shown in the first row of Fig. 10.10. They are obtained by solving the nonlinear motion planning problem by the sequential convex programming approach, which takes  $4.4817 \times 10^3$  s on average with the MacBook Pro laptop, 2.2 GHz Intel Core i7, 16 GB 1600 MHz DDR3 RAM). The high-conflict nature of the UAV swarm reconfiguration is implied by the naive fictitious solution trajectories shown in the second row of Fig. 10.10, which are computed just with the Buffered Voronoi Cells method [14] for collision avoidance, without solving the nonlinear motion planning.

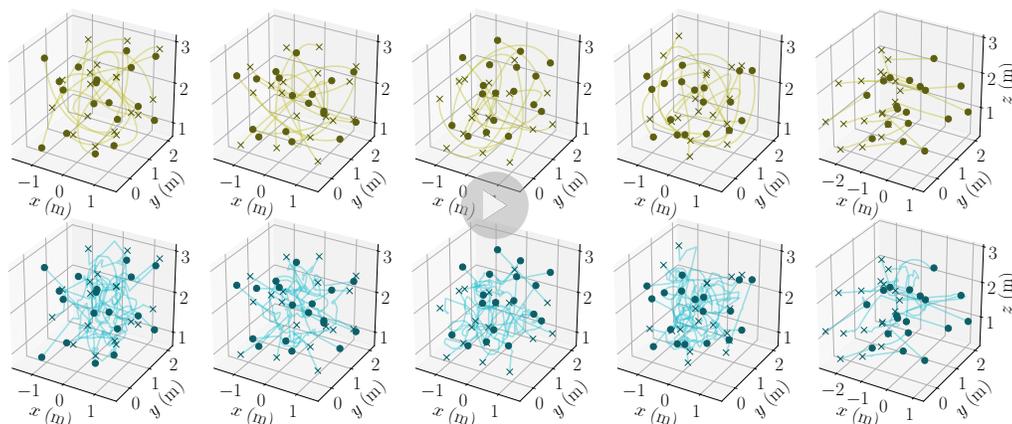


Figure 10.10: Crazyflie nominal position trajectories. First row: global solution trajectories of 18 crazyflies, found by solving the nonlinear motion planning problem by sequential convex programming. Second row: naive fictitious solution trajectories obtained with the Buffered Voronoi Cells method [14]. Initial and terminal positions are consistent with those of Fig. 10.11.

### 10.2.II-B Experimental Setup and Results

The G&C commands computed by Neural-Rendezvous are sent to crazyflies in real time over 8 low-latency/long-range radio channels, with their control frequency being 100 Hz. The software architecture is constructed using `crazyswarm`, a ROS-

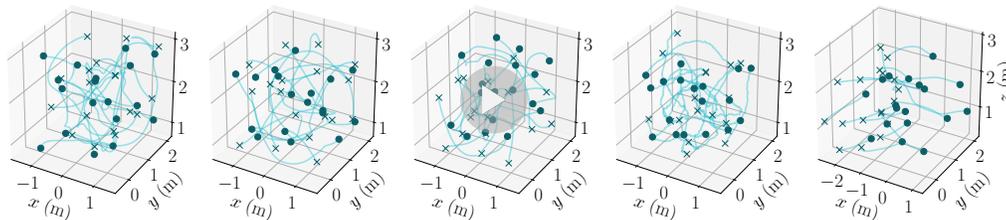


Figure 10.11: Demonstrated Neural-Rendezvous trajectories of 18 crazyflies, where the circles are their initial positions and the crosses are their target positions. The first three figures represent randomized experiments without artificial disturbance, 4th figure represents a randomized simulation with artificial disturbance, and 5th figure represents an experiment with the ISO model (see Fig. 10.12).

based open-source platform for controlling UAV swarms [15]. The pose of each UAV is estimated using the Vicon motion capture system, and we perform experiments both with and without artificial external and bounded disturbance acting on each of the UAV dynamics. When added, the disturbance  $d$  with  $\|d\| = 0.125$  is used for demonstrating the robustness property of Neural-Rendezvous.

The demonstrated trajectories of the crazyflies are shown in Fig. 10.11 and the trajectories observed in the actual environment are shown in Fig. 10.12, where the first 3 figures of Fig. 10.11 represent the randomized experiments without artificial disturbance, the 4th figure represents randomized simulation with artificial disturbance, and the 5th figure represents the experiment with the ISO model depicted in Fig. 10.12. As implied in the figures and the [movie](#), Neural-Rendezvous successfully achieves the high-conflict UAV swarm reconfiguration in a distributed manner, robustly against the real-world disturbance solving the highly-nonlinear guidance and control problem. The total 2-norm control effort for the reconfiguration is 1.7597 m/s for Neural-Rendezvous and 1.2901 m/s for the computationally-expensive global solution (which takes  $4.4817 \times 10^3$  s to be found on average when using the sequential convex programming approach on the MacBook Pro laptop, 2.2 GHz Intel Core i7, 16 GB 1600 MHz DDR3 RAM). The optimality gap arises from the SN-DNN learning error, real-world disturbances, and distributed communication of the UAVs. The performance is demonstrated up to 20 UAVs as is also visualized in this [movie](#).

### 10.3 Numerical Simulations for CASTOR

This section demonstrates one of our frameworks, CART of Chapter 7 [16], for several motion planning and control problems as part of the CASTOR project in

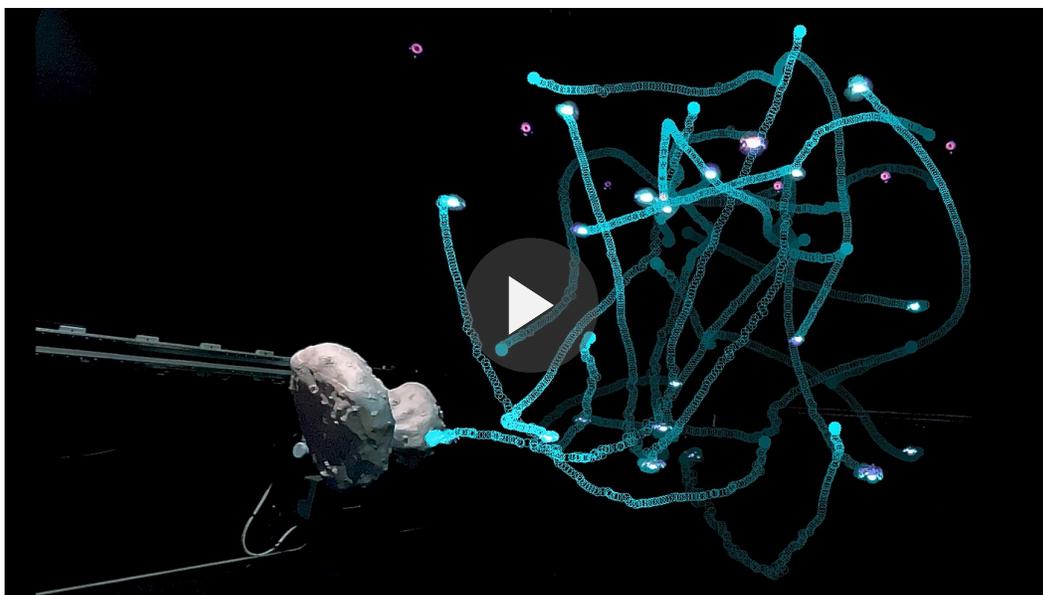


Figure 10.12: Demonstrated Neural-Rendezvous trajectories of 20 crazyflies observed in the actual environment, illustrated using their light trails. This movie can be found [here](#).

collaboration with NASA JPL. PyTorch [17] is used for designing and training neural networks and CVXPY [18] is used to solve convex optimization problems.

### 10.3.I Illustrative Examples

#### 10.3.I-A General Nonlinear System

Let us first consider a nonlinear dynamical system (7.7) with  $f^i$  given as

$$f^i(p^i, v^i, t) = \begin{bmatrix} \cos(p_1^i)p_2^i - v_1^i + v_2^i \\ -\sin(p_2^i)p_1^i v_2^i + (v_1^i)^2 - v_2^i - 2v_1^i v_2^i \end{bmatrix} \quad (10.2)$$

where  $N = 1$  (number of agents) and  $M = 5$  (number of obstacles). Our safety filter and robust filter are constructed using the methods outlined in Chapter 7, where the state-dependent coefficient matrix is used to construct the contraction metric of (7.18) as in [8]. This example does not have a clear physical interpretation, but finding the incremental Lyapunov function is non-trivial as the system is nonlinear and non-polynomial and the set of CLFs is non-convex [19].

The reference trajectory shown in the left-hand side of Fig. 10.13 is generated by a baseline contraction-based tracking control law [8] with an initial condition  $x(0) = [1.0, 1.0, 1.0, 1.0]^\top$ , violates a safety constraint at the position indicated by  $\times$ . When the disturbance is small ( $\bar{d}^i = \bar{\gamma}^i = 2.0 \times 10^{-5}$ ), our safety filter works without the robust filter thanks to its built-in robustness even with the control interval

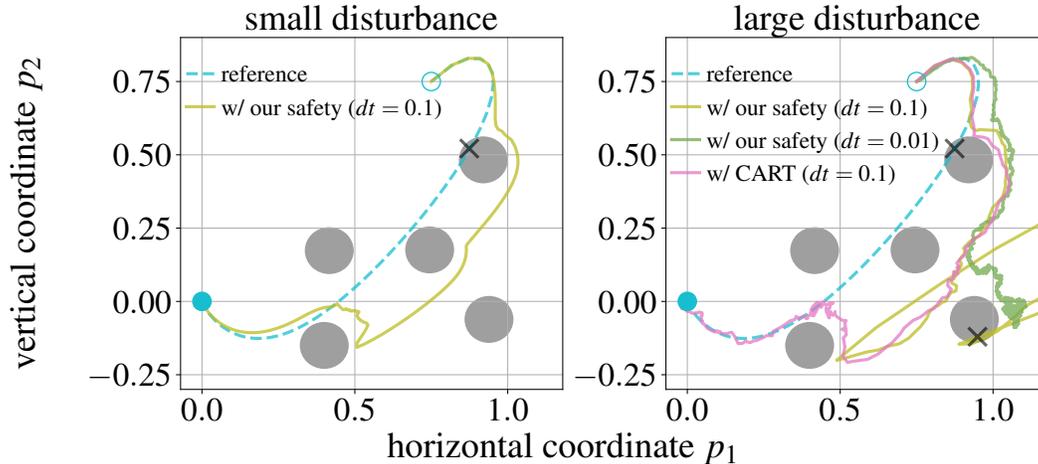


Figure 10.13: Position trajectories of the nonlinear system (7.7) with (10.2) with small and large disturbance,  $\bar{d}^i = \bar{\gamma}^i = 2.0 \times 10^{-5}$  and  $\bar{d}^i = \bar{\gamma}^i = 2.0 \times 10^{-2}$  in (7.7), respectively, which capture the trade-off discussed in (a) and (b) of Sec. 7.5.I-B.

$dt = 0.1$ , as we discussed in (a) of Sec. 7.5.I-B. Furthermore, since the reference trajectory uses an incremental Lyapunov-based robust control, this scenario can be viewed as a demonstration of the CLF-CBF controller. This result thus implies that the CLF-CBF works well in a nominal setting with a small disturbance, as expected from the argument of Sec. 7.5.I-C. When  $\bar{d}$  and  $\bar{\gamma}$  get larger ( $\bar{d}^i = \bar{\gamma}^i = 2.0 \times 10^{-2}$ ) as in the right-hand side of Fig. 10.13, however, our safety filter becomes too sensitive to the disturbance for the control interval  $dt = 0.1$ , leading to a large control input and safety violation indicated by  $\times$ . This situation can be avoided by using a smaller control interval  $dt = 0.01$ , but still, its control input for safety becomes more dominant than the ideal control input for the reference trajectory even in this case, taking longer to reach the target position as can be seen from the green trajectory of Fig. 10.13. The combination of the safety filter and robust filter, CART, indeed allows for considering robustness separately when applying the safety filter, thereby handling large disturbances without violating safety and losing too much of the reference control performance, as discussed in (b) of Sec. 7.5.I-B.

### 10.3.I-B Thruster-based Spacecraft Simulators

Such a trade-off is more evident in a practical multi-agent robotic system, where we cannot use a smaller control time interval due to hardware limitations. We next consider a nonlinear spacecraft simulator system given in [11] with  $N = 6$  (number of agents),  $M = 10$  (number of obstacles), and  $r_{\text{sen}} = 2.0$  (m) (sensing radius),

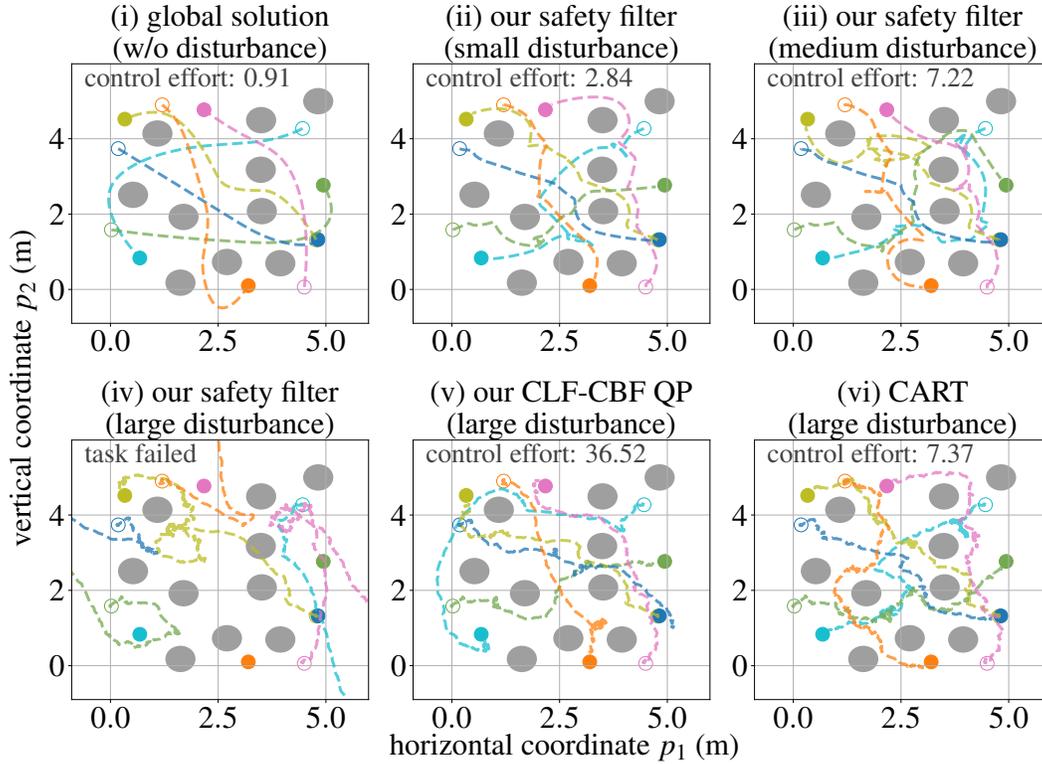


Figure 10.14: Position trajectories of the spacecraft simulator system [11] with various disturbances (small:  $\bar{d}^i = \bar{\gamma}^i = 3.0 \times 10^{-4}$ , medium:  $\bar{d}^i = \bar{\gamma}^i = 5.0 \times 10^{-3}$ , and large:  $\bar{d}^i = \bar{\gamma}^i = 5.0 \times 10^{-2}$  in (7.7)), which capture the trade-off discussed in Sec. 7.5.I.

where the control time interval  $dt$  is required to be  $dt \geq 0.1$  (s). The dynamics parameters are normalized to 1.

The learned motion planning policy detailed in Sec. 7.6.I is constructed using a neural network used in [13] with the training process outlined in [12]. It utilizes the centralized global solution data sampled by solving (7.8) using, e.g., the sequential convex programming, for the decentralized approximation by the neural network with the local observation (7.9). The initial and target states of the spacecraft and the positions of the circular static obstacles in Fig. 10.14 are randomized during the training and simulation. The cost function for the objective function is selected as  $c(x(\tau), u(\tau), \tau) = \sum_{i=1}^N \|u^i(\tau)\|^2$  in (7.8).

As can be seen from Fig. 10.14, we can indeed observe the differences between each approach as discussed in Sec. 7.5.I with Fig. 7.7 and 7.8. When the size of the disturbance is relatively smaller than the learning error ((ii) of Fig. 10.14), then the proposed safety filter works with its built-in robustness (see (a) of Sec. 7.5.I-B).

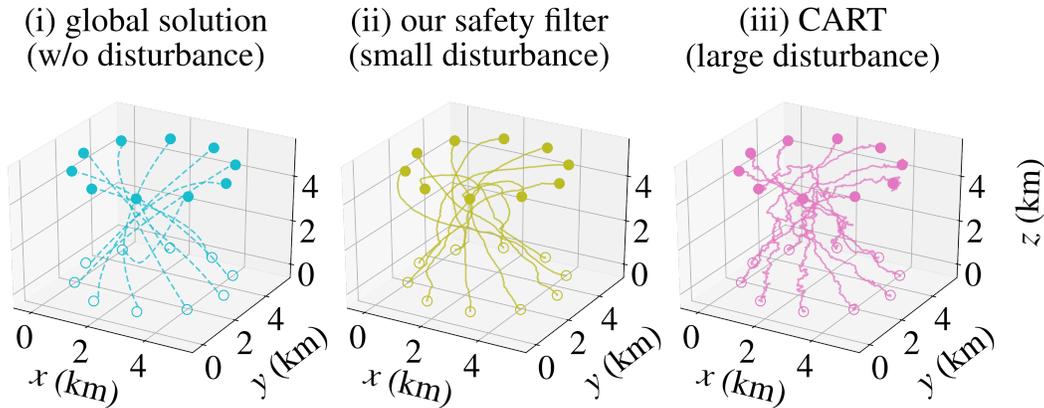


Figure 10.15: Position trajectories for the multi-spacecraft reconfiguration task in LEO [2] with various disturbances (small:  $\bar{d}^i = \bar{\gamma}^i = 1.0 \times 10^{-2}$  and large:  $\bar{d}^i = \bar{\gamma}^i = 5.0 \times 10^{-2}$  in (7.6)).

The loss of optimality in control results from the presence of disturbance and its decentralized implementation with distributed information. As the disturbance gets larger ((iii) – (v) of Fig. 10.14), our safety filter starts to fail. Also, our safety filter with the CLF-CBF approach starts to yield excessively large control input even when its QP is solved with a control input constraint ( $|(u^i)_k| \leq 1.00, k = 1, \dots, n$ ), which results in additional computational burden for each agent (see (a) – (b) of Sec. 7.5.I-C with Fig 7.8). Task failure is defined as the situation where at least one of the spacecraft does not reach the target state. Even when the size of the disturbance is relatively larger than the learning error ((vi) of Fig. 10.14), CART, the safety filter equipped with the robust filter, still works, retaining its control effort 4.96 times smaller than that of the CLF-CBF and 8.10 times greater than that of the optimal solution (see (b) of Sec. 7.5.I-B with Fig. 7.7 and 7.8).

### 10.3.II Applications to Real-World Multi-Agent Systems

Let us also discuss the potential of our approach in numerical simulations for real-world aerospace and robotic systems. These are performed as a part of the CASTOR project in collaboration with JPL.

#### 10.3.II-A Multi-Spacecraft Reconfiguration

We consider the optimal reconfiguration in Low Earth Orbit (LEO) with 10 spacecraft ( $N = 10$  and  $M = 0$ ), where the distributed learned motion planning policy of Sec. 7.6.I is trained again using [12] as discussed in Sec. 10.3.I-B with  $r_{\text{sen}} = 2.0$  (m) for the sensing radius. Its nonlinear dynamical system can be expressed as a La-

Table 10.3: Control performances for the multi-spacecraft reconfiguration in LEO averaged over 50 simulations, where  $J = \int_0^{t_f} \sum_{i=1}^N \|u^i\|^2 d\tau$ .

guidance and control methods	success rate (%)	control effort $J$
global solution (w/o disturb.)	100	1.26
our safety filter (small disturb.)	94.0	4.93
our CLF-CBF QP (large disturb.)	31.6	10.36
CART (large disturb.)	100	5.86

\*small disturb.:  $\bar{d}^i = \bar{\gamma}^i = 1.0 \times 10^{-2}$  and large disturb.:  $\bar{d}^i = \bar{\gamma}^i = 5.0 \times 10^{-2}$  in (7.6).

grangian system (7.6) as given in [2]. The task success is defined as the situation where the agent reaches, avoiding collisions, if any, a given target terminal state  $x_f$  within a given time horizon. The success rate is computed as the percentage of successful trials in the total 50 simulations. The initial and target states of the spacecraft are randomized during the training and simulation. Also, the cost function of (7.8) is again selected as  $c(x(\tau), u(\tau), \tau) = \sum_{i=1}^N \|u^i(\tau)\|^2$  in (7.8).

As implied in Fig. 10.15 and as discussed in Sec. 10.3.I-B, we can still see that our proposed safety filter works for small disturbances, and CART, the safety filter equipped with the robust filter, works for large disturbances. Such an observation can be corroborated by the results summarized in Table 10.3. In particular, CART augments the learned motion planning policy with safety and robust tracking, resulting in its success rate of 100% with its control effort 4.64 times greater than that of the optimal solution. Again, the loss of optimality in control results from the presence of disturbance and the CART's decentralized implementation with distributed information.

### 10.3.II-B Swarm Control of UAVs with Single Rotor Failures

As an example of under-actuated nonlinear systems, let us finally consider the swarm motion planning and control of UAVs each with one rotor failure [20], [21]. The dynamics parameters are selected as the ones of the crazyflie designed to be a versatile open-source flying development platform to perform various types of aerial robotics research <https://www.bitcraze.io/products/crazyflie-2-1/>. Figure 10.16 implies that our approach, CART, provides the certified safety and robust tracking also in such a challenging robotic motion planning and control problem, even under the presence of learning errors and disturbance. Note that the learned motion planning policy

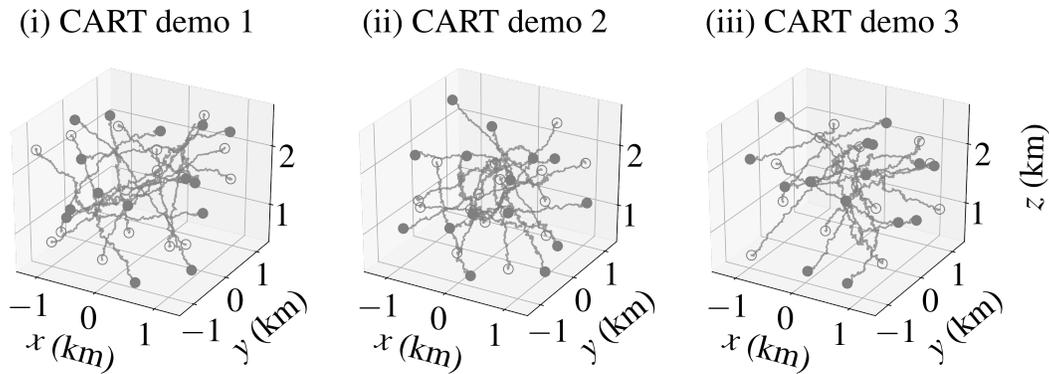


Figure 10.16: Position trajectories for the swarm motion planning and control of UAVs with one rotor failure [20], [21] with disturbance  $\bar{d}^i = \bar{\gamma}^i = 2.0 \times 10^{-2}$  in (7.7).

is obtained using [12] and the simulation is performed for randomized initial and terminal states, with  $N = 15$  (number of agents),  $M = 0$  (number of obstacles), and  $r_{\text{sen}} = 2.0$  (m) (sensing radius).

## References

- [1] D. Mages, D. Landau, B. Donitz, and S. Bhaskaran, “Navigation evaluation for fast interstellar object flybys,” *Acta Astronaut.*, vol. 191, pp. 359–373, 2022, ISSN: 0094-5765.
- [2] S.-J. Chung, U. Ahsun, and J.-J. E. Slotine, “Application of synchronization to formation flying spacecraft: Lagrangian approach,” *AIAA J. Guid. Control Dyn.*, vol. 32, no. 2, pp. 512–526, 2009.
- [3] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, “Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview,” *Annu. Rev. Control*, vol. 52, pp. 135–169, 2021, ISSN: 1367-5788.
- [4] D. Morgan, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, “Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming,” *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1261–1285, 2016.
- [5] Z. Xie, C. K. Liu, and K. Hauser, “Differential dynamic programming with nonlinear constraints,” in *IEEE Int. Conf. Robot. Automat.*, 2017, pp. 695–702.
- [6] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Upper Saddle River, NJ: Pearson, 1991.
- [7] M. Morari and J. H. Lee, “Model predictive control: Past, present and future,” *Comput. Chem. Eng.*, vol. 23, no. 4, pp. 667–682, 1999, ISSN: 0098-1354.

- [8] H. Tsukamoto and S.-J. Chung, “Robust controller design for stochastic nonlinear systems via convex optimization,” *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4731–4746, 2021.
- [9] B. P. S. Donitz, D. Mages, H. Tsukamoto, *et al.*, “Interstellar object accessibility and mission design,” in *IEEE Aerosp. Conf.*, 2022.
- [10] J. Riedel *et al.*, “Autonomous optical navigation (AutoNav) DS1 technology validation report,” in *NASA JPL, Caltech*, 2000.
- [11] Y. Nakka, R. Foust, E. Lupu, *et al.*, “A six degree-of-freedom spacecraft dynamics simulator for formation control research,” in *Astrodyn. Special. Conf.*, 2018.
- [12] B. Rivière, W. Hönig, Y. Yue, and S.-J. Chung, “GLAS: Global-to-local safe autonomy synthesis for multi-robot motion planning with end-to-end learning,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4249–4256, 2020.
- [13] H. Tsukamoto and S.-J. Chung, “Learning-based robust motion planning with guaranteed stability: A contraction theory approach,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6164–6171, 2021.
- [14] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, “Fast, on-line collision avoidance for dynamic vehicles using buffered Voronoi cells,” *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 1047–1054, 2017.
- [15] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, “Crazyswarm: A large nano-quadcopter swarm,” in *Int. Conf. Robot. Automat.*, 2017, pp. 3299–3304.
- [16] H. Tsukamoto, B. Rivière, C. Choi, A. Rahmani, and S.-J. Chung, “CART: Collision avoidance and robust tracking augmentation in learning-based motion planning for multi-agent systems,” in *IEEE Conf. Decis. Control*, *under review*, 2023.
- [17] A. Paszke, S. Gross, F. Massa, *et al.*, “PyTorch: An imperative style, high-performance deep learning library,” in *Adv. Neural Inf. Process. Syst.*, vol. 32, Curran Associates, Inc., 2019.
- [18] S. Diamond and S. Boyd, “CVXPY: A Python-embedded modeling language for convex optimization,” *J. Mach. Learn. Res.*, 2016.
- [19] I. R. Manchester and J.-J. E. Slotine, “Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design,” *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 3046–3053, Jun. 2017.
- [20] M. W. Mueller and R. D’Andrea, “Stability and control of a quadcopter despite the complete loss of one, two, or three propellers,” in *IEEE ICRA*, 2014, pp. 45–52.

- [21] S. Sun, G. Cioffi, C. de Visser, and D. Scaramuzza, “Autonomous quadrotor flight despite rotor failure with onboard vision sensors: Frames vs. events,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 580–587, 2021.

## CONCLUDING REMARKS

The theoretical contribution of this thesis is twofold. First, we present an overview of contraction theory, which provides a way to generalize and simplify Lyapunov-based stability methods for incremental exponential stability analysis of nonlinear non-autonomous (time-varying) systems (Chapter 2). The use of differential dynamics and its similarity to an LTV system allow for LMI and convex optimization formulations that are useful for systematic nonlinear control and estimation synthesis (Chapter 3 and Chapter 4). Second, various methods of machine learning-based control using contraction theory are presented to augment the existing learning frameworks with formal robustness and stability guarantees, extensively using the results of the first part of the thesis (Chapter 5). In particular, we have derived such guarantees for deep learning-based feedback control & state estimation (Chapter 6), motion planning & multi-agent collision avoidance and robust tracking augmentation (Chapter 7), adaptive control & neural net-based system identification and control (Chapter 8), for nonlinear systems perturbed externally by deterministic and stochastic disturbances.

### 11.1 Summary of Presented Methods

Let us summarize the control methods we have discussed so far, revisiting Fig. 11.1 shown also at the very beginning of this thesis in Chapter 1.

As described in Fig. 11.1,

1. if the uncertainty can be treated just robustly, then the NCM and CV-STEM methods will take care of it and provide us with a near-optimal and robust control policy, with formal guarantees even under the presence of learning errors (Chapter 3, Chapter 4, and Chapter 6) [1]–[5];
2. if the uncertainty is linearly parametric, then we can use the NCM-based adaptive control (aNCM) for online learning with the same guarantees (Chapter 8) [6];

**Overview of Learning-based Control with Formal Guarantees (in the context of nonlinear control theory)**

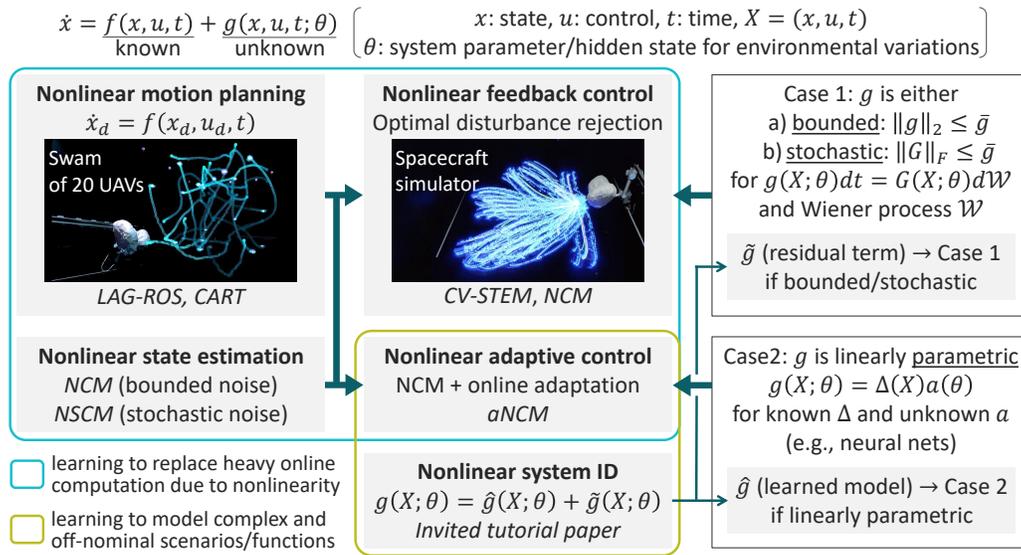


Figure 11.1: Overview of our learning-based control with formal guarantees, revisited.

3. if the uncertainty cannot be treated robustly and is non-parametric, then we can model it by the SN-DNN, and treat each term robustly and adaptively in real-time (Chapter 8) [7];
4. if the target trajectory and true state are unknown, we can use the extensions of our proposed approach to motion planning and state estimation (Chapter 3, Chapter 4, and Chapter 7) [2], [8]–[10].

We remark here that there are two different ways of using learning-based control here: 1) the methods in the blue frame of Fig. 11.1 use learning for replacing heavy computational load due to the problem complexity and nonlinearity as in meta-learning, and 2) the methods in the yellow frame of Fig. 11.1 use them for learning complex and off-nominal scenarios and functions.

The formal guarantees of the methods shown in Fig. 11.1 are essential for their real-world applications but could be difficult to obtain without accounting for a contracting property. Especially in situations where ISS and uniform asymptotic stability-based arguments render nonlinear stability analysis unnecessarily complicated, the use of exponential stability and the comparison lemma in contraction theory helps to achieve significant conceptual and methodological simplifications. A connection to the KYP and bounded real lemmas is also shown in the context of contraction-based incremental stability analysis.

Considering the promising outcomes of its utilization for model-based learning in Chapter 5 – 7 and for model-free data-driven learning in Chapter 8, the methods of contraction theory discussed in this thesis provide important mathematical tools for formally providing safety and stability guarantees of learning-based and data-driven control, estimation, and motion planning techniques for high-performance robotic and autonomous systems. Examples are elucidated to provide clear guidelines for its use in deep learning-based stability analysis and its associated control design for various nonlinear systems.

Furthermore, the ISO exploration project of Chapter 9 with the numerical simulations and experimental validation of Chapter 10 imply the real-world applicability and the superior performance of our approaches. For example, we have developed Neural-Rendezvous, a deep learning-based terminal G&C framework for achieving ISO encounter under large state uncertainty and high-velocity challenges, which uses a minimum-norm tracking controller with an optimal MPC-based guidance policy imitated by the SN-DNN. As derived in Theorem 9.1 and illustrated in Fig. 9.3 – 9.7, its major advantage is the formal optimality, stability, and robustness guarantee even with the use of machine learning, resulting in a spacecraft delivery error bound that decreases exponentially in expectation with a finite probability. The performance is validated both in numerical simulations and hardware experiments, which indicates that it works not just for ISO exploration but for general nonlinear G&C problems, solving them robustly against real-world disturbances and uncertainties. Having a verifiable performance guarantee is essential for using learning-based control in safety-critical robotic and aerospace missions, and our work provides a nonlinear control theoretical approach to formally meet this need.

Applying learning-based control algorithms to real-world systems always involves unmodeled uncertainties in the state, dynamics, environment, and control actuation, which could all lead to destabilizing behaviors different from what is learned and observed in numerical simulations. Our learning-based approaches are provably robust against these uncertainties, thereby helping to reproduce the simulation results in a nominal setting seamlessly in real-world scenarios with various sources of uncertainties.

## 11.2 Future Plans

Let us finally mention some of our future directions. Ideally speaking, if we place a robotic/aerospace explorer in a fully or partially unknown dynamics and environ-

ment, we expect our to-be-developed autonomous framework should possess the superior capability of autonomously investigating the dynamical and environmental properties performing whatever tasks with varying objectives, just like humans, but much more quickly, accurately, and intelligently with much-needed optimality, safety, robustness, and incremental stability guarantees in real-time, unlike humans.

### **11.2.I Performance-based Learning via Contraction Theory**

The performance guarantees in my previous approaches depend on GNC parameters pre-tuned to be optimal in a nominal problem setting. In highly autonomous situations with time-varying/off-nominal dynamics and environments, however, the pre-computed guarantees can become too conservative to be used as they are. We could instead utilize the real-time information of the guarantees to actively guide online learning of the optimal GNC parameters, for the sake of having the best performance even in such challenging, dynamically changing real-world scenarios. Using [6] to handle the parameter update, we could develop autonomous systems with much more radical options to explore in achieving their objectives, but with the same or increasingly better quality of the performance guarantees.

### **11.2.II Universal Learning-based Control with Robustness Guarantees**

This topic seeks to develop a learning-based nonlinear GNC method that works for broader types of systems, beyond the ones considered in [7]. One of the most important yet challenging of these is a system modeled by deep neural networks, which is worth paying special attention to considering the recent triumphs of machine learning. A promising direction is to view the network's layer-to-layer connection as a dynamical system and apply contraction theory to enjoy superior robustness guarantees by design, where we can build our learning-based GNC on top of it. Such a venture to more involved nonlinear systems enhances the nominal performance guarantees to be optimized online in 11.2.I.

### **11.2.III Task Objective-driven Meta-learning for Online GNC**

As we embark on new frontiers of autonomy, the task objectives should become more interdisciplinary and dynamic. This would imply that near-future autonomous systems are required to solve optimization problems in real-time with dynamically changing definitions of optimality. One way to address this issue is to learn the latent information shared among all the autonomy tasks offline and extract and adaptively learn the portion of the problem that dynamically changes in a real-

world environment online. Assuming we manage to express such information in a multiplicatively-separable manner, we could still utilize our previous approaches for online learning with strong incremental stability and robustness guarantees.