Spacecraft Motion Planning and Control under Probabilistic Uncertainty for Coordinated Inspection and Safe Learning

Thesis by
Yashwanth Kumar Nakka

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

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2021
Defended April 30, 2021

Yashwanth Kumar Nakka
ORCID: 0000-0001-7897-3644

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Dr. Soon-Jo Chung for the continuous support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in my research and writing of this thesis. I am grateful to him for directing me at critical junctures of my Ph.D. journey.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Joel Burdick, Dr. Richard Murray, Dr. Fred Hadaegh, and Dr. Yisong Yue, for their insightful comments and encouragement, but also for the critical questions which incented me to widen my research.

My sincere thanks also go to Dr. Yisong Yue, who provided me an opportunity to join his team in January 2021. Without his precious support, it would not be possible to conduct this research. I thank my fellow labmates for the stimulating discussions and for all the fun we have had in the last five years.

I would like to thank my family: Mom and Dad for supporting me in the last 5 years and in general, which helped me push through the toughest times. Last but not least, I'd like to thank my wife, Robin, who has been a constant source of strength and support throughout the Ph.D.

ABSTRACT

During a spacecraft mission design process, engineers often balance the following three criteria: science return, optimality in performance, and safety. Given a science criterion, engineers design the orbit parameters with predefined performance and safety. Often in this approach, the spacecraft has no understanding of the expected outcome or the knowledge of the mission safety criteria. Autonomous science-driven orbit (or goal) selection and planning for safety under uncertainty enable efficient and adaptable missions. To this end, we propose an architecture for information-based guidance and control for coordinated inspection, motion planning and control algorithms for safe and optimal guidance under uncertainty, and architecture for safe exploration.

In the first part of this thesis, we present an architecture for inspection or mapping of a target spacecraft in a low Earth orbit using multiple observer spacecraft. We use an information gain approach to directly consider the trade-off between gathered data and fuel/energy cost. The estimated information gain is a crucial input to the motion planner, which computes orbits and reconfiguration strategies for each of the observers to maximize the information gain from distributed observations of the target spacecraft. The resulting motion trajectories jointly consider observational coverage of the target spacecraft and fuel/energy cost. We validate our architecture in a mission simulation to visually inspect the target spacecraft and on the three degree-of-freedom robotic spacecraft dynamics simulator testbed.

In the second part of the thesis, we present gPC-SCP, *G*eneralized *P*olynomial *C*haos-based *S*equential *C*onvex *P*rogramming method, to compute a sub-optimal solution for a continuous-time chance-constrained stochastic nonlinear optimal control (SNOC) problem. The approach enables motion planning and control of robotic systems under uncertainty. The proposed method involves two steps. The first step is to derive a deterministic nonlinear optimal control problem (DNOC) with convex constraints that are surrogate to the SNOC by using gPC expansion and the distributionally-robust convex subset of the chance constraints. The second step is to solve the DNOC problem using sequential convex programming (SCP) for trajectory generation and control. We prove that in the unconstrained case, the optimal value of the DNOC converges to that of SNOC asymptotically and that any feasible solution of the constrained DNOC is a feasible solution of the chance-constrained SNOC. We derive a stable stochastic model predictive controller using the gPC-SCP

for tracking a potentially unsafe trajectory in the presence of uncertainty. We empirically demonstrate the efficacy of the gPC-SCP method for the following three test cases: 1) collision checking under uncertainty in actuation, 2) collision checking with stochastic obstacles, and 3) safe trajectory tracking under uncertainty in the dynamics and obstacle location by using a receding horizon control approach. We validate the effectiveness of the gPC-SCP method on the robotic spacecraft testbed.

In the third part of this thesis, we present a new approach for optimal motion planning for safe exploration that integrates the chance-constrained stochastic optimal control with dynamics learning and feedback control. We derive an iterative convex optimization algorithm that solves an *Info*rmation-cost *S*tochastic *N*onlinear *O*ptimal *C*ontrol problem (Info-SNOC). The optimization objective encodes control cost for performance and exploration cost for learning, and the safety is incorporated as distributionally robust chance constraints. The dynamics are predicted from a robust regression model that is learned from data. The Info-SNOC algorithm is used to compute a sub-optimal pool of safe motion plans that aid in exploration for learning unknown residual dynamics under safety constraints. A stable feedback controller is used to execute the motion plan and collect data for model learning. We prove the safety of rollout from our exploration method and reduction in uncertainty over epochs, thereby guaranteeing the consistency of our learning method. We validate the effectiveness of Info-SNOC by designing and implementing a pool of safe trajectories for a planar robot. We demonstrate that our approach has a higher success rate in ensuring safety when compared to a deterministic trajectory optimization approach.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] Rebecca C. Foust, Yashwanth Kumar Nakka, Ayush Saxena, Soon-Jo Chung, and Fred Y. Hadaegh. Automated rendezvous and docking using tethered formation flight. In *9th International Workshop on Satellite Constellations and Formation Flying*, 2017. URL `https://authors.library.caltech.edu/78723/`.
Y.K.N developed the dynamical model and the simulation setup used for deriving the algorithm, participated in conducting analysis, and in writing the manuscript.

[2] Rebecca C. Foust, E. Sorina Lupu, Yashwanth Kumar Nakka, Soon-Jo Chung, and Fred Y. Hadaegh. Ultra-soft electromagnetic docking with applications to in-orbit assembly. In *International Astronautical Congress, Bremen, Germany*, 2018. URL `https://authors.library.caltech.edu/90231/`.
Y.K.N participated in conducting the experiments, analyzed the results and participated in writing the manuscript.

[3] Rebecca C Foust, E Sorina Lupu, Yashwanth Kumar Nakka, Soon-Jo Chung, and Fred Y Hadaegh. Autonomous in-orbit satellite assembly from a modular heterogeneous swarm. *Acta Astronautica*, 169:191–205, 2020. ISSN 0094-5765. doi: https://doi.org/10.1016/j.actaastro.2020.01.006.
Y.K.N developed the controller used for testing the algorithm, participated in conducting the experiments, and in writing the manuscript.

[4] Kai Matsuka, Elena Sorina Lupu, Yashwanth Kumar Nakka, Rebecca Foust, Soon-Jo Chung, and Fred Hadægh. Distributed multi-target relative pose estimation for cooperative spacecraft swarm. In *11th International Workshop on Satellite Constellations and Formation Flying*, 2019. URL `https://authors.library.caltech.edu/97314/`.
Y.K.N wrote the code to identify multiple targets in the vision data for experiments and participated in writing the manuscript.

[5] Yashwanth Kumar Nakka and Soon-Jo Chung. Trajectory optimization for chance-constrained nonlinear stochastic systems. In *IEEE Conference on Decision and Control*, pages 3811–3818, 2019. doi: 10.1109/CDC40024.2019.9028893.
Y.K.N proposed the algorithm, formulated the theoretical proofs, implemented and analyzed the method, and wrote the manuscript.

[6] Yashwanth Kumar Nakka, Rebecca C Foust, Elena Sorina Lupu, David B Elliott, Irene S Crowell, Soon-Jo Chung, and Fred Y Hadaegh. Six degree-of-freedom spacecraft dynamics simulator for formation control research. In *Advances in the Astronautical Sciences, Astrodynamics Specialist Conference*, volume 167, pages 3367–3387, 2018.

Y.K.N participated in platform design, formulated the control architecture, implemented the real-time controller, and wrote the manuscript.

[7] Yashwanth Kumar Nakka, Soon-Jo Chung, James T. Allison, Jack B. Aldrich, and Oscar S. Alvarez-Salazar. Nonlinear attitude control of a spacecraft with distributed actuation of solar arrays. *Journal of Guidance, Control, and Dynamics*, 42(3):458–475, 2019. doi: 10.2514/1.G003478.
Y.K.N derived the theoretical proofs, implemented and tested the method, and wrote the draft.

[8] Yashwanth Kumar Nakka, Wolfgang Hönig, Changrak Choi, Alexei Harvard, Amir Rahmani, and Soon-Jo Chung. Information-based guidance and control architecture for multi-spacecraft on-orbit inspection. In *AIAA Scitech Forum, Guidance Navigation and Control Conference*, 2021. doi: 10.2514/6.2021-1103.
Y.K.N participated in formulating the architecture, implemented and analyzed planning and control blocks, developed the mission design tool, conducted experiments, and participated in writing the manuscript.

[9] Yashwanth Kumar Nakka, Anqi Liu, Guanya Shi, Anima Anandkumar, Yisong Yue, and Soon-Jo Chung. Chance-constrained trajectory optimization for safe exploration and learning of nonlinear systems. *IEEE Robotics and Automation Letters*, 6(2):389–396, 2021. doi: 10.1109/LRA.2020.3044033.
Y.K.N proposed the architecture along with A.L, formulated the theoretical proofs for safety, implemented and analyzed the method, and participated in writing the manuscript.

TABLE OF CONTENTS

Acknowledgements . . . iii
Abstract . . . iv
Published Content and Contributions . . . vi
Table of Contents . . . viii
List of Illustrations . . . x
List of Tables . . . xviii
Chapter I: Introduction . . . 1
1.1 Motivation . . . 1
1.2 Contributions . . . 3
1.3 Related Work . . . 9
1.4 Organization . . . 13
Chapter II: Preliminaries . . . 15
2.1 Spacecraft Formation Flying . . . 15
2.2 Optimal Relative Orbit Initialization and Reconfiguration . . . 17
2.3 Attitude Dynamics . . . 18
2.4 Optimal Attitude Trajectory Planning Problem . . . 19
2.5 Generalized Polynomial Chaos . . . 21
2.6 Chapter Summary . . . 23
Chapter III: Information-Based Guidance and Control for On-Orbit Inspection 25
3.1 Multi-Agent Motion Planning for Inspection . . . 25
3.2 Main Distributed Inspection Algorithm . . . 27
3.3 Inspection of a Target Spacecraft . . . 38
3.4 Experiments on the Robotic Spacecraft Simulator Testbed . . . 43
3.5 Chapter Summary . . . 45
Chapter IV: Trajectory Optimization for Chance-Constrained Nonlinear Stochastic Systems . . . 49
4.1 Problem Formulation . . . 50
4.2 Deterministic surrogate of the SNOC Problem . . . 56
4.3 Chapter Summary . . . 67
Chapter V: Guidance and Control Under Uncertainty . . . 69
5.1 Motion Planning Under Uncertainty . . . 69
5.2 Tracking Control Using Stochastic Model Predictive Control . . . 75
5.3 Simulations and Experiments on Spacecraft Dynamics Simulator . . . 81
5.4 Chapter Summary . . . 89
Chapter VI: Safe Exploration and Learning . . . 90
6.1 Problem Formulation . . . 90
6.2 Info-SNOC Main Algorithm . . . 94
6.3 Analysis . . . 96
6.4 Simulation and Discussion . . . 103

6.5 Chapter Summary 106
Chapter VII: Multi-Spacecraft Testbed For Autonomy Research 108
7.1 The Spacecraft Dynamics Simulator Facility at Caltech 108
7.2 Dynamics and Control 114
7.3 Experiments 126
7.4 Chapter Summary 129
Chapter VIII: Conclusion 130
8.1 Future Work 132
Appendix A: Nonlinear Attitude Control of a Spacecraft with Distributed Actuation of Solar Arrays 151
A.1 Introduction 151
A.2 Dynamics of a 1-DOF Satellite with SASA 154
A.3 Nonlinear ODE-PDE Control Design of SASA 161
A.4 Control Implementation and Simulation Results 168
A.5 Experiments 176
A.6 Conclusion 184

LIST OF ILLUSTRATIONS

Number *Page*

1.1 The multi-spacecraft GNC system architecture for autonomous inspection and mapping proposed in Chapter 3 4

1.2 Caltech's M-STAR (Multi-Spacecraft Testbed for Autonomy Research), described in Chapter 7, planning a safe trajectory to ensure safety under uncertainty in actuation during a proximity maneuver. The motion planning problem is formulated as a chance-constrained stochastic optimal control problem solved in two steps. Step 1: Project the stochastic problem to a deterministic problem by using generalized polynomial chaos approach and distributional robustness; and Step 2: Use deterministic solvers to compute an optimal solution to the deterministic problem. 5

1.3 An episodic framework for safe exploration using chance-constrained trajectory optimization. An initial estimate of the dynamics is computed using a known safe set [53] and control policy. A probabilistic safe trajectory and policy that satisfies safety chance-constraints is computed using Info-SNOC for the estimated dynamics. This policy is used for rollout with a stable feedback controller to collect data. . . 7

2.1 Left : A visualization of the ECI $(\hat{X}, \hat{Y}, \hat{Z})$ and LVLH $(\hat{x}, \hat{y}, \hat{z})$ frames, taken from [16]. Right : Passive relative orbits for four deputy spacecraft, propagated using the initial conditions proposed in [16]. 15

2.2 An example of the optimal PRO initialization maneuver and PRO reconfiguration maneuver computed using Problem 1 and 2. We designed the PRO's using the stable subspace proposed in [16]. 18

2.3 An example of the optimal attitude trajectory computed using Problem 4 for the initial and terminal states $\mathbf{q}[0] = [1, 0, 0, 0]$, $\boldsymbol{\omega}[0] = 0$, $\mathbf{q}[T_a] = [0.5, 0.167, 0.167, 0.833]$, and $\boldsymbol{\omega}[T_a] = 0$, respectively. 21

2.4 Example gPC approximation of some standard probability distribution functions (PDF) using gPC expansion. For the beta and exponential distributions, gPC expansion represents the PDF well with just second order approximation. For a Gaussian distribution, the gPC representation is exact. 22
3.1 Concept of operations of an inspection mission for observing a target spacecraft with multiple observer spacecraft deployed. 26
3.2 The multi-level and multi-timescale hierarchical architecture for combined orbit and attitude planning and control for an inspection task. 28
3.3 Nine generated PRO candidates to cover the target spacecraft (blue in the center). 32
3.4 An example of the visibility check using ray casting database for a given camera with fixed location, pointed in $-y$ direction with a field of view of 30 degrees. 33
3.5 PRO candidate selection for $N = 2$ from the set of PROs shown in Fig. 3.3. For each of the nine candidates the expected information cost is computed (bar chart top left). PRO 5 achieves the lowest H and is selected in the first iteration. Assuming one observer is orbiting PRO 5, we estimate H for the remaining PROs in the second iteration. The lowest cost is achieved for PRO 7 (bar chart bottom left). Observer spacecraft on the selected PROs can observe the target from many perspectives (plot on the right). 34
3.6 Example solutions for the PRO reconfiguration planning problem. Each of the two observers plan to move from its current state to one of the selected PRO candidates (see Fig. 3.5) and simultaneously estimates the delta-V cost for such maneuver (see captions on each plot). 35

3.7 Centralized PRO candidate selection and information database implementation of the framework in Fig. 3.2 for inspection. The sampled PRO candidates and the information database are stored in the memory of the target spacecraft. The PRO candidates are then assigned to the observers based on coverage (information) and fuel optimality. The assigned candidates are communicated to the observers. The observers have a decentralized controller for initialization or reconfiguration to the assigned PROs. From the PRO, the observer spacecraft communicates the information update to the target spacecraft at a fixed time period for computing a new set of PRO candidates to inspect areas that have not been inspected earlier. 38

3.8 Distributed PRO candidate selection and information database implementation of the framework in Fig. 3.2 for inspection. The information database is stored on all the observers and evaluated using the updates communicated at fixed frequency from the neighbours. The PRO candidates are sampled in a distributed fashion using synchronous messages with the updated information cost. Optimal PRO assignment is done using distributed auction method discussed in [14]. 39

3.9 Inspection of Cygnus using two observers. Left: Trajectories of the two observers. The bold part shows the planned PRO reconfiguration. Right: The information cost converges to zero over time, because the entire target spacecraft is inspected. 41

3.10 Information cost and POI coverage over time with varying number of observer spacecraft. The results are averaged over 5 trials with standard deviation shown as shaded area. Left: Using multiple observers allows to significantly reduce the time until a target information cost value is reached. For example, $H = 2$ is achieved in less than half the time when using 5 observers compared to when using one observer. Right: A lower information cost leads to a better coverage of POIs. 42

3.11 A plot showing comparison of minimum distance between observers when 3, 4, and 5 CubeSats are used for the inspection task as function of time. As expected, 3 observer configuration has larger separation compared to 5 observer configuration. 42

3.12 Attitude trajectory generated using the slerp interpolation and sequential convex programming. The control required to track the slerp trajectory and the smoothed trajectory are compared. The smoothed trajectory satisfies the torque limits of $\pm 2\,\mathrm{mNm}$, while tracking the slerp trajectory easily saturates the reaction wheels. 44
3.13 A picture of the information-based planning (Fig. 3.2) experiment on the multi-spacecraft testbed [30] for autonomy research. 45
3.14 Block diagram showing the hierarchical planner as applied to the three degree-of-freedom spacecraft dynamics simulator. 46
3.15 Output of Algorithm 1. Observer 1 is actively inspecting the target spacecraft using the hierarchical information-based planning algorithm. 47
3.16 Information-based planning to inspect all the four surfaces of the target spacecraft using a single agent. 48
4.1 Effect of uncertainty in the state estimate and actuation on the PRO maintenance, shown in LVLH frame. We show the PRO trajectories generated over 100 Monte-Carlo trials, by sampling the uncertainty in state and actuation. 49
4.2 Example gPC propagation for a pendulum. The figure compares the mean and 2σ confidence computed using gPC Projection ($P_{\mathrm{gPC}} = 1$), linear covariance propagation and Monte Carlo (MC) propagation of the simple pendulum dynamics $\ddot{\theta} = -\sin\theta - 0.8\dot{\theta} + \sqrt{0.001}\xi(t)$. It is observed that the gPC approximation overestimates the variance compared to MC and the linear covariance propagation underestimates the variance. The $P_{\mathrm{gPC}} = 1$ projection corresponds to a Gaussian approximation that includes the cross correlation between the state and uncertainty. 57
5.1 An illustration of the convex linear constraint used for collision checking in deterministic SCP at a particular instant in time is shown on the left. The linear chance constraint under stochastic dynamics is projected to gPC space forming a second-order cone constraint. The cone constraint is visualized as a robustness bound on the robot's state as shown in the figure on right. 70

5.2 An illustration of the second-order cone constraint used for collision checking with uncertainty in dynamics and the obstacle position at an instant in time is shown on the right. For a given risk of collision probability ϵ, the uncertainty in obstacle position is visualized as an additional uncertainty in the robots state. 72
5.3 The top and side view of the Caltech's robotic spacecraft dynamics simulator. 83
5.4 The figure demonstrates convergence of the mean and the variance (σ_x, σ_y) of the states (x, y) with increasing P_{gPC} for $\sigma = \{0.01, 0.1\}$. 83
5.5 Left: We compare the probabilistic safe trajectories computed using distributionally robust and Gaussian collision chance constraint. Center: We compare the trajectories for various risks $(\epsilon = 0.05, 0.25, 0.5)$ of collision constraint violation. Right: We demonstrate collision checking under uncertainty in both robot dynamics and obstacle location. 84
5.6 We compare the trajectories generated for different sizes $c = \{0.316, 0.224, 0.071\}$ (shown as the green circle) of the terminal set. We show the terminal state of the robot (blue), when a nominal trajectory (sampled from the probabilistic trajectory) is executed using an exponentially stable controller. 85
5.7 We show trajectory tracking and safety under uncertainty in dynamics and obstacle location in real-time over 5 trials by using the proposed stochastic model predictive controller. We compare the distributionally robust (DRLCC) collision constraint with the Gaussian collision constraint. Left: In the case with small uncertainty ($\Sigma_p = 1e-4$), both constraints perform safe tracking. Right: With large uncertainty ($\Sigma_p = 1e-2$), the DRLCC provides safety in all the trials, while Gaussian collision constraint fails in 2 trials. 86
5.8 The guidance, navigation, and control loop used for planning a distributionally-robust safe trajectory using gPC-SCP and controlling the 3 DOF spacecraft simulators. 87

5.9 We show the output of the gPC-SCP method at each stage of Algorithm 2 and 10 trials of closed-loop trajectory tracking by using an exponentially stable feedback controller designed in [30]. Top: We show the output of AO-RRT for 5000 nodes and the SCP for the nominal dynamics. Middle: We show the probabilistic safe trajectory generated using the gPC-SCP method with a risk measure of $\epsilon = 0.05$ for collision checking. Bottom: We observe one failure in the 10 trials of the closed-loop trajectory execution. 88
6.1 Info-SNOC applied to Scenario 1. In Fig. (a), we show the motion plan along with the 2σ confidence in position of the performance trajectory ($\rho = 0$) and the information trajectory ($\rho = 1$) computed using Info-SNOC, and the nominal trajectory computed using SCP under nominal dynamics. In Fig. (b), we show the trace of $\Sigma_{\mathbf{x}}$ w.r.t time. The information trajectory ($\rho = 1$) has higher $\Sigma_{\mathbf{x}}$ compared to the performance trajectory ($\rho = 0$). We compare total open-loop fuel computed at each time step in Fig. (c), and in Fig. (d) we demonstrate collision avoidance during exploration for 20 trials of rollout using a safety augmented stable controller. 104
6.2 Info-SNOC applied to Scenario 2. In Fig. (a), we show a comparison of the performance trajectory ($\rho = 0$), the information trajectory ($\rho = 1$), and an intermediate safe trajectory (green) computed using Info-SNOC and the nominal trajectory computed using deterministic SCP under nominal dynamics. In Fig. (b), we compare a sampled trajectory with the trajectories generated by feedback tracking and rollout with a safety filter. 105
6.3 Performance over epochs for Scenario 1. Left: we show decrease in the terminal position variance over epochs demonstrating improved goal reaching with epoch. Center: the differential entropy of the prediction variance Σ_g for information trajectory ($\rho = 1$) is larger compared to the performance trajectory ($\rho = 0$). Right: the number of collisions during rollout for 1000 trials decrease as the learning converges, validating Theorem 6. 105
7.1 Multiple 6-DOF M-STAR spacecraft at Caltech's Aerospace Robotics and Control Lab. 109
7.2 M-STAR spacecraft dynamics simulator. 109
7.3 Flowchart of pneumatic system on translation and attitude stage. . . . 110

7.4 Attitude stage architecture. 112
7.5 Section view of Caltech's custom-made reaction wheels. 113
7.6 Software architecture design. 113
7.7 Coordinate Systems used for the derivation of the dynamic model. . . 116
7.8 Attitude Stage with actuator configuration and nomenclature in the body frame. 116
7.9 Thruster characterization setup and sample results recorded by the load cell, showing rise time and fall time. 124
7.10 Experimental data and linear fit of average force and impulse vs. thruster firing time at 40 psi, 50 psi, and 60 psi operating pressure. . . 124
7.11 Closed-loop control implementation for the 6DOF simulator. 126
7.12 Closed-loop waypoint reaching experimental result- test case 1. . . . 128
7.13 Closed-loop waypoint reaching experimental result- test case 2. . . . 128
A.1 1-DOF cylinder and flexible solar array model. 155
A.2 Distributed Piezo and and strain distribution across the composite beam. 158
A.3 Front and top view of the experimental setup. 171
A.4 Closed-loop ODE system as implemented in Simulink for simulations. 172
A.5 Trajectory tracking (simulation) for $\delta = 1$ with gains $\lambda_\theta = 3$, and $K_\theta = 0.5$, $K_\xi = 0.5$. 172
A.6 Trajectory tracking (simulation) for $\delta = 0.5$ with gains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$. 173
A.7 Trajectory tracking (simulation) for $\delta = 0$ with gains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$. 173
A.8 Slewing (simulation) for $\delta = 1$ with gains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$. 173
A.9 Slewing (simulation) for $\delta = 0.5$ with gains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$. 174
A.10 Slewing (simulation) for $\delta = 0$ with gains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$. 174
A.11 Trajectory tracking (simulation) for $\delta = 0.5$ in configuration 2 with gains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$. 174
A.12 Trajectory tracking (simulation) for $\delta = 0$ in configuration 2 with gains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$. 175
A.13 Slewing (simulation) for $\delta = 0.5$ in configuration 2 with gains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$. 175
A.14 Slewing (simulation) for $\delta = 0$ in configuration 2 with hains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$. 175

A.15 Open-loop beam experimental setup. 176
A.16 Open-loop experiment vs simulation, beam tip deflection. 176
A.17 SASA setup 178
A.18 SASA setup views 178
A.19 Top view of the setup showing coordinate systems used in experiments. 179
A.20 Closed-loop ODE system as implemented in Simulink for experiments. 180
A.21 Open-loop experiment versus simulation, bus rotation. 180
A.22 Trajectory tracking (experiment) for $\delta = 1$ with gains $\lambda_\theta = 1.5$, $K_\theta = 0.5$, $K_\xi = 0.5$. 181
A.23 Trajectory tracking (experiment) for $\delta = 0.5$ with gains $\lambda_\theta = 1.5$, $K_\theta = 0.5$, $K_\xi = 0.5$. 181
A.24 Trajectory tracking (experiment) for $\delta = 0$ with gains $\lambda_\theta = 2$, $K_\theta = 0.5$, $K_\xi = 0.5$. 182
A.25 Slewing (experiment) for $\delta = 1$ with gains $\lambda_\theta = 1$, $K_\theta = 0.25$, $K_\xi = 0.5$. 182
A.26 Slewing (experiment) for $\delta = 0.5$ with gains $\lambda_\theta = 1$, $K_\theta = 0.25$, $K_\xi = 0.5$. 182
A.27 Slewing (experiment) for $\delta = 0$ with gains $\lambda_\theta = 1$, $K_\theta = 0.25$, $K_\xi = 0.5$. 183
A.28 Trajectory tracking (experiment) for $\delta = 0.5$ in configuration 2 with gains $\lambda_\theta = 2$, $K_\theta = 0.5$, $K_\xi = 0.5$. 183
A.29 Trajectory tracking (experiment) for $\delta = 0$ in configuration 2 with gains $\lambda_\theta = 2$, $K_\theta = 0.5$, $K_\xi = 0.5$. 183
A.30 Slewing (experiment) for $\delta = 0.5$ in configuration 2 with gains $\lambda_\theta = 2$, $K_\theta = 0.5$, $K_\xi = 0.5$. 184
A.31 Slewing (experiment) for $\delta = 0$ in configuration 2 with gains $\lambda_\theta = 2$, $K_\theta = 0.5$, $K_\xi = 0.5$. 184

LIST OF TABLES

Number *Page*

1.1 Spacecraft simulators from other institutions. [107–123] 13
5.1 Number of collisions over 1000 trails. 85
7.1 List of components on the translation stage. 111
7.2 List of components on the attitude stage. 112
7.3 Nominal torque and angular momentum of the spacecraft. 113
7.4 Constraints on the angular motion of the attitude stage. 115
7.5 Linear fit equations for firing time computation for a given control signal F_r at time t and control frequency f_{cl}. 123
A.1 Physical, geometrical and structural parameters of bus, array, and PZT. 171

Chapter 1

INTRODUCTION

1.1 Motivation

Autonomous on-orbit servicing [1] and inspection of a spacecraft is a key enabling technology for increasing the mission lifetime, improving the safety of the human spaceflight, and gaining an insight into the health of the spacecraft. Inspection and mapping of known or unknown space objects in Earth's orbit is the first essential step towards on-orbit servicing capabilities that include refueling, repairing, assembling, and upgrading of space assets [2, 3]. State-of-the-art approaches for on-orbit inspection include robotic manipulators with rails for extended workspace, single servicing spacecraft [4, 5], and astronaut intervention. These approaches have severe limitations: the capability of manipulators is limited due to mechanical constraints [4]; the single service orbiter has payload constraints, and astronaut intervention is expensive and potentially dangerous. A team of collaborating spacecraft has the potential to provide augmented capabilities with enhanced robustness and versatility at a reduced cost and time [6, 7]. Thus, we propose to use a distributed spacecraft formation for on-orbit inspection. The multi-spacecraft approach has additional challenges [8, 9] of coordination, synchronization, and communication. The technologies for small spacecraft such as CubeSats are maturing quickly, making multi-spacecraft servicing closer to a reality [10–13]. In this thesis, we present information-based guidance, navigation, and control (GNC) architecture for inspection or mapping of a target spacecraft, referred to as chief, in an orbit around Earth using multiple spacecraft, referred to as deputies, in stable passive relative orbits (PROs). This architecture integrates higher-level information-based orbit selection with model-based guidance [14] and control [15] algorithms to demonstrate coordinated inspection.

Model-based design strategies for guidance [16, 17] and control [18] of spacecraft and robotic systems often take a deterministic approach with robustness guarantees [19] to quantify performance under worst-case uncertainties. These approaches assume a bounded value of uncertainty leading to conservative trajectories and control laws. Confidence-based motion planning [20–23] and control algorithms [24, 25], which incorporate uncertainties in the dynamic model and en-

vironment to guarantee safety and performance with high probability, enable safe operation of robots and autonomous systems in partially known and dynamic environments. A probabilistic approach can allow for integration with a higher-level discrete decision-making algorithm for information gathering [26, 27], and for safe exploration [28, 29] to learn the interaction with an unknown environment. Examples of autonomous systems that require safety guarantees under uncertainty include spacecraft with thrusters as actuators during proximity operations [27, 30], powered descent on Mars [31], and quadrotors flying in turbulent winds [32, 33]. We propose a new trajectory optimization algorithm that systematically accounts for the uncertainty in dynamics and environment to compute safe and optimal motion plans. We extend this algorithm for tracking a trajectory under uncertainty by using a stochastic model predictive formulation.

Modeling the complex dynamic interactions with the environment requires high-fidelity techniques that are often computationally expensive. Machine-learning models can remedy this difficulty by approximating the dynamics from data [34–37]. The learned models typically require off-line training with labeled data that are often not available or hard to collect in many applications. Safe exploration is an efficient approach to collect ground truth data by safely interacting with the environment. In this thesis, we integrate the proposed motion planning method with a learning algorithm and an exponentially stable controller to derive a safe exploration algorithm for active and safe data collection.

Spacecraft formation flying technologies using smallsats, such as microsatellites and CubeSats, as individual agents offer a robust, adaptable, and cost-effective way to establish space telescopes [38], communication systems [39] for observation, and various other applications [13, 40, 41]. The ability of these systems to perform equivalent to a monolithic system depends on achieving high-precision relative navigation, guidance, control, and synchronization of the individual agents in the formation. A ground-based robotic dynamics simulator that can mimic the frictionless motion in a disturbance torque-free environment with flight-like subsystems provides a platform to test and validate the GNC algorithms required to design and build such a multi-agent spacecraft system. To this end, we design and build a novel robotic six-degree-of-freedom spacecraft simulator for autonomy research.

1.1.1 Problem Statement

In this thesis, we study the problem of autonomous coordinated inspection and safe exploration. Both the problems can be formulated as an information-driven chance-constrained optimal control problem. Depending on the time scales of the dynamics, we propose and study the following three simplifications of this problem.

Coordinated Inspection

We formulate the multi-agent coordinated inspection problem as an information-cost optimal control problem. In this framework, we use the information cost for computing informative orbits and attitude plans. We use the orbits and attitude plan as an input to the optimal guidance and control algorithms. We derive the information cost by using the confidence on the output of the inspection sensor model.

Chance-Constrained Stochastic Optimal Control

We formulate the optimal guidance and control under uncertainty in dynamics and environment as a chance-constrained stochastic optimal control problem. For this problem, given an uncertainty model, we derive a novel motion planning algorithm for computing a safe and optimal motion plan under uncertainty.

Safe Exploration

We formulate the safe exploration problem as an information-cost chance-constrained stochastic optimal control problem. We integrate this problem with learning and an exponentially stable feedback controller to demonstrate safe and active data collection for model learning. In this case, we derive the information cost using confidence in the model quality. Essentially, using this architecture, we can improve the model.

1.2 Contributions

The main contributions of this thesis can be summarized as follows.

1.2.1 Chapter 3

We present an information-based GNC architecture for inspection or mapping of a target spacecraft, referred to as chief, in an orbit around Earth using multiple spacecraft, referred to as deputies, in stable Passive Relative Orbits (PROs). We assume

that the target spacecraft is in a low Earth orbit, and the deputies are deployed from the target spacecraft. We design stable relative parking orbits for initial formation deployment. From the onset of formation deployment, our guidance algorithm focuses on designing PROs and reconfiguration strategies for each of the deputies to maximize the information gain from distributed observation of the target. The resulting motion trajectories jointly consider (a) observational coverage of the target spacecraft, (b) fuel/energy cost, and (c) collision checking while iteratively acting upon updates from estimation. The planner strives towards computational efficiency to make real-time on-board computation possible, while making trade-off between optimality as needed. The capabilities of the proposed planner framework are demonstrated on a design reference mission involving up to 5 spacecraft performing coordinated inspection of a target spacecraft using simulations.

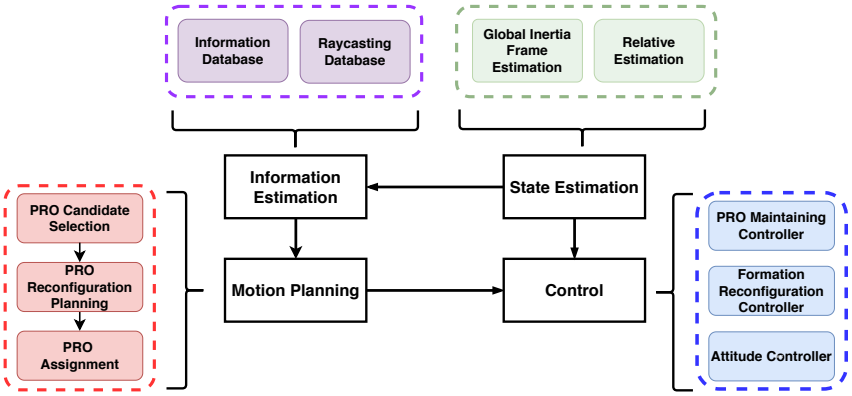


Figure 1.1: The multi-spacecraft GNC system architecture for autonomous inspection and mapping proposed in Chapter 3

The four components of the proposed architecture, shown in Fig. 1.1, are 1) information estimation, 2) state and inertial frame estimation, 3) guidance (or) motion planning, and 4) control. In this work, we assume that the state and inertial frame estimation algorithms are available and focus on motion planning for information maximization. The information estimation quantifies the information gain of the target spacecraft, given past and potential future poses of all the observer spacecraft. The estimated information gain is a crucial input to the motion planner, which considers a set of alternative PROs and decides when and how to reconfigure the observer spacecraft, while approximately maximizing the information gain and minimizing delta-V. For computational efficiency, both information estimation and motion planning modules rely on simplified attitude dynamics. Our controls component ensures PRO initialization, safe reconfiguration, and instrument pointing.

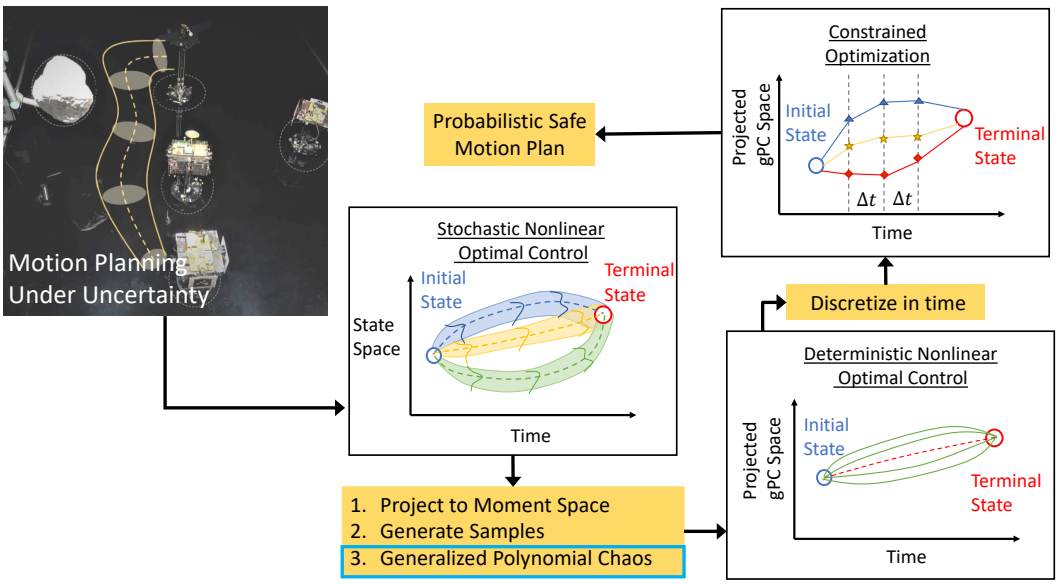


Figure 1.2: Caltech's M-STAR (Multi-Spacecraft Testbed for Autonomy Research), described in Chapter 7, planning a safe trajectory to ensure safety under uncertainty in actuation during a proximity maneuver. The motion planning problem is formulated as a chance-constrained stochastic optimal control problem solved in two steps. Step 1: Project the stochastic problem to a deterministic problem by using generalized polynomial chaos approach and distributional robustness; and Step 2: Use deterministic solvers to compute an optimal solution to the deterministic problem.

1.2.2 Chapters 4 and 5

Motion planning problem considering safety in conjunction with optimality under uncertainty can be formulated as a continuous-time continuous-space stochastic nonlinear optimal control problem (SNOC) with chance constraints. We propose the generalized polynomial chaos-based sequential convex programming (gPC-SCP) method, as described in Fig. 1.2, to solve a chance-constrained SNOC problem. The method involves deriving a deterministic nonlinear optimal control (DNOC) problem with convex constraints that are a surrogate to the SNOC problem with linear and quadratic chance constraints. We derive the DNOC problem by accounting for nonlinear stochastic dynamics using generalized polynomial chaos expansions (gPC) [42–44] and obtaining deterministic convex approximations of linear and quadratic chance constraints using distributional robustness [45–47]. The DNOC problem is then solved using sequential convex programming (SCP) [14, 15, 48] for trajectory optimization and for nonlinear stochastic model predictive control (SMPC).

The main contributions of the Chapters 4 and 5 can be summarized as follows:

(a) We present a systematic sequence of approximations for the chance-constrained SNOC problem to compute a convex-constrained DNOC problem using gPC projection. We analyze the gPC projection of the stochastic dynamics for existence and uniqueness [49, 50] of a solution in the gPC space. Examples are provided to study the effect of projection on both the controllability of surrogate dynamics and the feasibility of the DNOC problem. We prove the convexity of the distributionally-robust linear and quadratic chance constraints in the gPC space.

(b) In order to characterize the deterministic approximation obtained using gPC projection, we present analysis on convergence of the DNOC problem to the SNOC problem for the unconstrained case. Then, we prove that any feasible solution of the constrained DNOC problem is a feasible solution of chance-constrained SNOC problem with an appropriate gPC transformation step applied.

(c) We derive provably conservative convex surrogates for collision checking with both deterministic and stochastic obstacles. We integrate this collision constraint with a sampling-based planning method [17, 51] to derive an algorithm that computes safe and optimal motion plans under uncertainty.

(d) We extend the gPC-SCP method to derive an iterative algorithm that solves the SMPC formulation. We prove that, if the terminal cost used in the SMPC problem is derived by an exponentially-stabilizing controller (e.g. [24]) for the stochastic dynamics, then the stochastic model predictive controller is stable and the cost converges to an upper bound.

(e) We validate the convergence and stability theorems, and the safety provided by the convex constraints in simulation, on a three degree-of-freedom robot dynamics. We show empirically that the gPC-SCP method, for both planning and control, has a higher success rate in comparison to the Gaussian approximation [25, 52] of the collision chance constraints. We demonstrate the efficacy of the gPC-SCP method by computing a safe trajectory for a spacecraft proximity maneuver with stochastic obstacles on the robotic spacecraft dynamics simulator [30] hardware platform and by executing the trajectory in real-time closed-loop experiments.

1.2.3 Chapter 6

We present an episodic learning and control algorithm for safe exploration, as shown in Fig. 1.3, that integrates learning, stochastic trajectory planning, and rollout for active and safe data collection. *Rollout* is defined as executing the computed safe trajectory and policy using a stable feedback controller. The planning problem is formulated as an Information-cost Stochastic Nonlinear Optimal Control (Info-SNOC) problem that maximizes exploration and minimizes the control effort. Safety constraints are formulated as chance constraints. The propagation

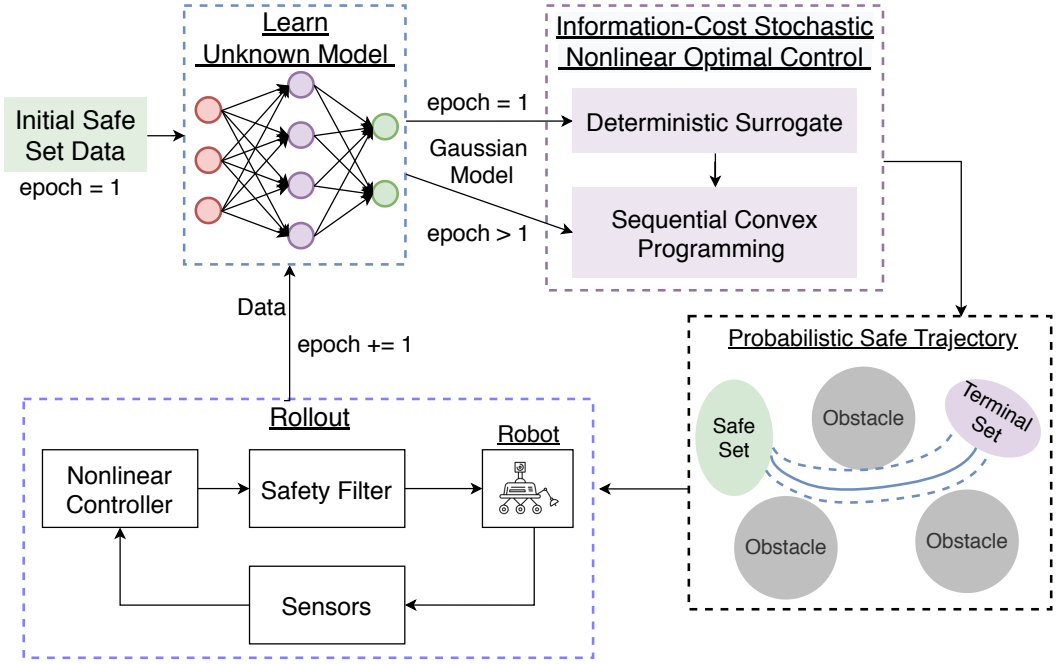


Figure 1.3: An episodic framework for safe exploration using chance-constrained trajectory optimization. An initial estimate of the dynamics is computed using a known safe set [53] and control policy. A probabilistic safe trajectory and policy that satisfies safety chance-constraints is computed using Info-SNOC for the estimated dynamics. This policy is used for rollout with a stable feedback controller to collect data.

of uncertainty in the dynamic model and chance constraints in Info-SNOC are addressed by projecting the problem to the gPC space and computing a distributionally robust [20, 46] convex approximation. By building on [20], we derive a sequential convex optimization solution to the Info-SNOC problem to plan a pool of sub-optimal safe and information-rich trajectories with the learned approximation of the dynamics. A sample of the trajectory pool is used as an input to the rollout stage to collect new data. To ensure real-time safety, the nonlinear feedback

controller with a safety filter used in the rollout stage certifies bounded stochastic stability [54]. The new data is used to learn an improved dynamic model.

The contributions of the Chapter 6 are as follows: a) we propose a new safe exploration and motion planning method by directly incorporating safety as chance constraints and ensuring stochastic nonlinear stability in the closed-loop control along with a safety filter; b) we derive a new solution method to the Info-SNOC problem for safe and optimal motion planning, while ensuring the consistency and reduced uncertainty of our dynamics learning method; and c) we use a multivariate robust regression model [55] under a covariate shift assumption to compute the multi-dimensional uncertainty estimates of the unknown dynamics used in Info-SNOC.

1.2.4 Chapter 7

We describe the development of a new 6-DOF spacecraft simulator, the Multi-Spacecraft Testbed for Autonomy Research (M-STAR). The spacecraft simulator hardware was designed to have decentralized control and information sharing capabilities with neighboring agents in view of the future goal of testing multi-agent GNC algorithms using up to five of these simulators. Each spacecraft has 16 thrusters and 4 reaction wheels to study fault-tolerant control.

In view of the model-based GNC algorithms a detailed nonlinear dynamic model for the 5-DOF system was derived by modelling it as a 3D pendulum on a gliding planar platform with a center of gravity offset in the 3D pendulum. The nonlinear dynamic model is decoupled by assuming a small center of gravity offset. A nonlinear hierarchical control law is proposed for fast attitude dynamics and slower position dynamics due to the time-scale separation. The control law computes forces and torques collocated to the dynamics. Control allocation [56] is done to map the collocated control signal to the actuator signal. Optimization formulations [57] can be used to solve the control allocation problem, typically formulated as a linear program. For the M-STAR control allocation, we implement a generalized pseudo-inverse method for control allocation with a weighted influence matrix to account for actuator saturation limits, as the optimization formulations are computationally expensive for real-time implementation.

The position control of the M-STAR is performed using on-off solenoids, which are inherently nonlinear due to mechanical delays and varying pressure in the manifold that supplies compressed air to the solenoids. The solenoids are characterized [58]

by measuring the force produced for varying on-off time, using a calibrated load cell. A linear model to compute the on time of a thruster is developed using the measured data for a given force requirement at each time step. The control law, control allocation scheme, and thruster model are tested for position tracking using a Robot Operating System (ROS)-based software framework.

1.3 Related Work

1.3.1 Autonomous On-Orbit Inspection

The prior research on autonomous on-orbit servicing can be classified into three lines of work: 1) vision-based navigation and mapping with control [59–61], 2) formation maintenance control [62] for inspection, and 3) single spacecraft flight mission [4]. The main focus of the vision-based navigation [63] research has been to develop inspection and mapping algorithms using sensors such as monocular cameras and laser range finders. Using the navigation estimates, a pointing control and formation maintenance control was designed to ensure that the target body is within the field of view of the sensors. This work is applicable to single spacecraft missions and does not take into account the overall formation maintenance and safety during the inspection task. We propose an architecture that can be integrated with any of the aforementioned navigation algorithms and extends to multi-spacecraft collaborative inspection by using an information metric to design stable relative orbits and minimum energy attitude motion plan for inspection.

A very recent work considers a centralized approach [62] to integrate formation relative orbit planning and control for the inspection task but is limited to the scenario where the target spacecraft are continuously tracked by the observer spacecraft using attitude control. The algorithm presented in this thesis is applicable to a wide range of scenarios when the full pose information about the observer spacecraft is available. Along with the work on algorithm development, the recent mission Seeker [4] demonstrated on-orbit inspection using a single observer to track and estimate the bounding box of the Cygnus spacecraft in both dark and light background with neural networks.

From robotics perspective, active information acquisition [64] was studied for both single-agent [28, 65, 66] and multi-agent systems [67, 68] to planning [67] and control [69]. The prior research was applied to ground robots and quadrotors. We extend this work, to incorporate spacecraft formation flying and pointing dynamics for active inspection in low Earth orbit. While the earlier research focuses on indi-

vidual aspects of autonomy stack, we propose an architecture that works for both spacecraft and ground robots.

1.3.2 Trajectory Optimization Under Uncertainty

Existing methods to solve a chance-constrained stochastic nonlinear optimal control (SNOC) problem use moment space propagation [25, 70–72], unscented transformation-based propagation [73], Monte Carlo sample propagation [21, 22, 74], and scenario-based [73, 75] approaches to construct a deterministic surrogate problem. Although these methods alleviate the curse of dimensionality, they do not provide asymptotic convergence guarantees for a DNOC problem. Monte Carlo methods provide asymptotic convergence guarantees, but often require large samples to estimate the constraint satisfaction for nonlinear systems and use mixed-integer programming [21] solvers for computing a solution. We use generalized polynomial chaos (gPC) propagation [44] to construct a deterministic nonlinear optimal control (DNOC) problem that converges to the SNOC problem, asymptotically. The gPC projection transforms the chance constraints from being a non-convex constraint in moment space to a convex constraint in the gPC space. This enables the use of sequential convex programming (SCP) [14, 15] method for computing a solution. Additionally, we study the existence and uniqueness [76] of a solution and the controllability of the deterministic surrogate dynamics of the stochastic dynamics.

Earlier work [22, 25, 77] uses a Gaussian approximation of the linear and the quadratic chance constraint for collision checking and for terminal constraint satisfaction. While this avoids multi-dimensional integration of chance constraints for feasibility checking, Gaussian approximation might not be an equivalent representation (or) even a subset of the feasible set in the presence of stochastic process noise in dynamics. We use distributional robustness [46, 47] property to propose a new deterministic second-order cone constraint and a quadratic constraint approximation of the linear and quadratic chance constraints. We prove that the deterministic approximations are a subset of the respective chance constraints.

In [21, 22], linear chance constraints were considered for probabilistic optimal planning for linear systems. The literature on chance-constrained programming focuses on problems with deterministic decision variable and uncertain system parameters for both linear [46] and nonlinear [47] cases. The results [45, 46] on distributional robust subset and convex approximations of the chance constraints can be readily transformed to the case with a random decision variable for an unknown measure.

The quadratic chance constraint would lead to an inner semi-definite program [78] that adds complexity to the SNOC problem considered in this thesis. The linear chance constraint for collision checking was first presented in [20]. In [79], authors show that linearized chance constraint is a subset of the original nonlinear chance constraint for a Gaussian confidence-based constraint. Since the local Gaussian assumption might not be valid for a nonlinear systems, we present proof for the distributionally robust convex constraint formulation that extends to include stochastic obstacles for a nonlinear stochastic differential equation.

From a stochastic model predictive control (SMPC) perspective, recent work [80–83] on control of discrete-time linear stochastic dynamical systems provides conditions for recursive feasibility, constraint satisfaction, convergence and stability by using a probabilistic invariant set as the domain of operation and a control Lyapunov function as the terminal cost function. Research on control of nonlinear stochastic dynamics [25, 84, 85] is focused on implementation by using nonlinear programming methods. In [86], authors formulate a bounding semi-definite optimization problem on moments using global polynomial optimization method [87] for controlling a nonlinear stochastic system, but do not incorporate state constraints or prove the stability of the system. We propose a SMPC method to control nonlinear stochastic differential equation that uses a stochastic control contraction metric [24, 88] as the terminal cost function. Assuming recursive feasibility and constraint satisfaction, we prove the convergence and stability of the SMPC method. We solve the SMPC problem using the gPC-SCP method to track a potentially unsafe trajectory in the presence of uncertainty in dynamics and environment.

The gPC expansion approach was used for stability analysis and control design of uncertain systems [84, 85, 89–92]. For trajectory optimization, recent work focused on nonlinear systems with parametric uncertainty [93, 94] with no constraints on the state, or linear systems with linear chance-constraints that do not extend to the SNOC problem considered here and lack analysis on the deterministic approximation of the uncertain system. The gPC approach was used to compute a moment-space receding horizon approximation [95], which was solved using nonlinear programming methods. We extend prior work to incorporate nonlinear dynamics and include analysis on the deterministic approximation. We formulate convex constraints for linear and quadratic constraints in gPC space and use this formulation to design algorithms for motion planning and control of a nonlinear stochastic dynamic system.

1.3.3 Safe Exploration

Safe exploration for continuous dynamical systems has been studied using the following three frameworks: learning-based model-predictive control (MPC), dual-control, and active dynamics learning. Learning-based MPC [53, 96–98] has been studied extensively for controlling the learned system. These deterministic techniques are also applied for planning an information trajectory[1] to learn online. The approach has limited analysis on safety of the motion plans that use recursive feasibility argument and by appending a known safe control policy. In contrast, stochastic trajectory planning [20, 22, 99, 100] naturally extends to incorporate probabilistic learned dynamic model. The safety constraints formulated as joint chance constraints [22] facilitate a new approach to analyze the safety of the motion plans computed using Info-SNOC. The effect of uncertainty in the learned model on the propagation of dynamics is estimated using the method gPC expansion for propagation, which has asymptotic convergence to the original distribution, and provides guarantees on the constraint satisfaction.

Estimating unknown parameters while simultaneously optimizing for performance has been studied as a dual control problem [101]. Dual control is an optimal control problem formulation to compute a control policy that is optimized for performance and guaranteed parameter convergence. In some recent work [99, 102], the convergence of the estimate is achieved by using the condition of persistency of excitation in the optimal control problem. Our method uses SCP [14, 15, 103] to compute the persistent excitation trajectory. Recent work [104] uses nonlinear programming tools to solve optimal control problems with an upper-confidence bound [105] cost for exploration without safety constraints. We follow a similar approach but formulate the planning problem as an SNOC with distributionally robust linear and quadratic chance constraints for safety. The distributionally robust chance constraints are convexified via projection to the gPC space. The algorithm proposed in this thesis can be used in the MPC framework with appropriate terminal conditions for feasibility and to solve dual control problems with high efficiency using the interior point methods.

1.3.4 Robotic Spacecraft Dynamics Testbed

Historically, air bearing [106] platforms have been a popular choice to build spacecraft dynamics simulators. Air bearing spacecraft simulation platforms were de-

[1] An information trajectory is defined as a result of optimal motion planning that has more information about the unknown model compared to a fuel-optimal trajectory.

Organization	Name	DOF
Naval Postgraduate School	POSEIDYN	3
Georgia Institute of Technology	ASTROS	5
Florida Institute of Technology	ORION	6
University of Florida	ADAMUS	6
Yonsei University	ASTERIX	5
NASA Jet Propulsion Laboratory (JPL)	FCT SSDT	5 3
German Aerospace Center (DLR)	TEAMS	3 and 5
Massachusetts Institute of Technology	SPHERES ARGOS	3 3 (attitude)

Table 1.1: Spacecraft simulators from other institutions. [107–123]

veloped by several research laboratories [107–123]; a selection of these simulation platforms is shown in Table 1.1. Existing air bearing platforms can be classified into four types based on the mode of operation: 3 degrees-of-freedom (DOF) planar [109, 116, 120], 3-DOF attitude [108, 110, 120], 5-DOF planar and attitude [107, 114, 117, 122], and 6-DOF planar and attitude with gravity-axis motion [113, 119, 121]. The air bearing system acts as a ground-based simulator platform for flight-like actuators and sensors, which provides an opportunity to test flight algorithms and emulate space dynamics [124]. In this thesis, we describe the development of a new 6-DOF spacecraft simulator, the Multi-Spacecraft Testbed for Autonomy Research (M-STAR), that is designed to be modular and accommodates 3-DOF, 4-DOF, 5-DOF, and 6-DOF operation with minimal mechanical modifications.

1.4 Organization

The organization of the dissertation is as follows. In Chapter 2, we discuss spacecraft formation flying dynamics, optimal control problem for orbit initialization and reconfiguration, optimal attitude trajectory design, and generalized polynomial chaos method for propagation of a stochastic differential equation. We build on this material and present an information-based guidance and control architecture in Chapter 3. We demonstrate inspection of a target spacecraft in simulation and on the robotic spacecraft simulator testbed. In Chapter 4, we derive a deterministic surrogate trajectory optimization problem for a stochastic nonlinear optimal control problem. We study the asymptotic performance and controllability of the deterministic surrogate problem. In Chapter 5, we design a motion planning and

control algorithm to compute safe and optimal motion plans under uncertainty in dynamics and environment. In Chapter 5, we extend the motion planning algorithm to incorporate an information cost to design safe and informative trajectories for exploration. We integrate this planning algorithm with robust regression and an exponentially stable feedback controller to demonstrate safe exploration. In Chapter 7, we describe the design and development of a six degree of freedom spacecraft dynamics simulator. We conclude in Chapter 8 with a summary of the contributions made in the thesis and a note on future work.

Chapter 2

PRELIMINARIES

To understand the information-based guidance and control architecture proposed in Chapter 3, we introduce the spacecraft formation flying dynamics, optimal control problem used for optimal orbit initialization and reconfiguration, attitude planning problem. We show example planning outputs by solving the optimal control problem using the sequential convex programming method. Following this, we describe the generalized polynomial chaos (gPC) method used for uncertainty propagation in Chapters 4, 5, and 6.

2.1 Spacecraft Formation Flying

The relative orbit dynamics [125, 126] of the observer spacecraft with respect to the target spacecraft in chief orbit are described in a Local-Vertical Local-Horizontal (LVLH) frame $(\hat{x}, \hat{y}, \hat{z})$ as shown in Fig. 2.1 that is fixed to the target spacecraft. The $\hat{x}$ coordinate is in the chief orbit's radial direction, the $\hat{z}$ coordinate is defined in the angular momentum direction, and the $\hat{y}$ coordinate completes the right handed coordinate system.

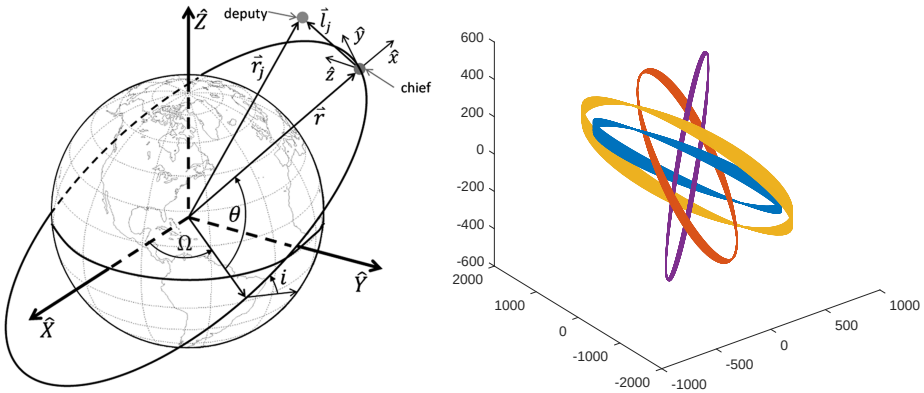


Figure 2.1: Left : A visualization of the ECI $(\hat{X}, \hat{Y}, \hat{Z})$ and LVLH $(\hat{x}, \hat{y}, \hat{z})$ frames, taken from [16]. Right : Passive relative orbits for four deputy spacecraft, propagated using the initial conditions proposed in [16].

We assume that the target spacecraft's dynamics is defined by the Gauss's variational equation [16],

$$\dot{\mathbf{æ}} = \mathbf{f}_c(\mathbf{æ}) + \mathbf{G}(\mathbf{æ})\mathbf{u}_c, \tag{2.1}$$

where $\mathbf{æ} \in \mathbb{R}^6$ is the orbital element vector in the ECI frame, the bias term $\mathbf{f}_\mathrm{c}$ determines the rate of change of $\mathbf{æ}$, and $\mathbf{G}$ is the control allocation matrix for the control input $\mathbf{u}_\mathrm{c} \in \mathbb{R}^3$. In the LVLH frame, the functional form of the relative orbital dynamics of the i^{th} observer is defined as follows:

$$\dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}_i, \mathbf{æ}, \dot{\mathbf{æ}}, \mathbf{u}_i), \tag{2.2}$$

where the bias term $\mathbf{f}_i$ depends on the target's dynamics $\mathbf{æ}$ and the force input $\mathbf{u}_i$. For the definitions of $\mathbf{f}_\mathrm{c}$, $\mathbf{f}_i$, and $\mathbf{G}$ see [16]. We assume that the target spacecraft is in a circular orbit, the linearized relative dynamics $\mathbf{f}_i$ of the i^{th} observer spacecraft for continuous control input u_i in the LVLH frame are as follows:

$$\ddot{x} - 2n\dot{y} - 3n^2x = \frac{f_\mathrm{thr}}{m}u_x, \quad \ddot{y} + 2n\dot{x} = \frac{f_\mathrm{thr}}{m}u_y, \quad \ddot{z} + n^2z = \frac{f_\mathrm{thr}}{m}u_z, \tag{2.3}$$

where $n = \sqrt{\mu/r_0^3}$ is the mean motion of the target spacecraft, μ is the gravitational constant, r_0 is the radius of the chief orbit with respect to the ECI frame, and m is the mass of the observer spacecraft. The control input $\mathbf{u} = [u_x, u_y, u_z]^\top$ scales the thruster force f_thr similar to a pulse-width modulated signal. These dynamics in Linear Time Invariant (LTI) state-space form are given by:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\frac{\mathbf{u}}{m}, \tag{2.4}$$

where $\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^\top$. The matrices $\mathbf{A}$ and $\mathbf{B}$ are given in (2.38) (appendix).

The discrete-time dynamics for is given by:

$$\mathbf{x}[k+1] = \mathbf{A}_\mathrm{d}\mathbf{x}[k] + \mathbf{B}_\mathrm{d}\mathbf{u}[k], \text{ with } \quad \mathbf{A}_\mathrm{d} = e^{\mathbf{A}\Delta t}, \quad \mathbf{B}_\mathrm{d} = \int_{t=0}^{\Delta t} e^{\mathbf{A}\tau}\frac{\mathbf{B}}{m}d\tau, \tag{2.5}$$

where k is the current time-step, and Δt is the discretization time step. The matrix $\mathbf{A}_\mathrm{d}$ is given in (2.39). For continuous control $\mathbf{u}(t)$ the input matrix $\mathbf{B}_\mathrm{d}$ is given in (2.40).

The LTI dynamics with control input $\mathbf{u}$ is used in designing trajectories for reaching a stable relative orbit and for reconfiguration maneuvers to a new set of stable relative orbits. The relative orbits for parking and inspection are designed using the energy matching condition discussed below in (2.6), which results in passively stable orbits that require minimum fuel for maintenance.

The global criterion for stable relative orbits is given by the energy matching

$$\frac{1}{2}\left((\dot{x} - ny)^2 + (\dot{y} + n(x + r_0))^2 + \dot{z}^2\right) - \frac{\mu}{\sqrt{(r_0 + x)^2 + y^2 + z^2}} = -\frac{\mu}{2r_0}. \tag{2.6}$$

In (2.6), the energy of the observer is matched with that of the target spacecraft at all time t. Using this criteria to design the orbits of the deputies results in a bounded relative motion with respect to the LVLH frame. For simplified analysis, we use a subset of the energy matched orbits defined by the following initial condition constraint for the HCW dynamics when $\mathbf{u} = 0$:

$$\dot{y}_0 = -2nx_0, \tag{2.7}$$

where $\dot{y}_0$ is the initial velocity in y direction and x_0 is the initial position in the LVLH frame. The initial ΔV in y_0 is computed using the initial position information. We use this condition to generate a pool of stable relative trajectories and search for an optimal trajectory in the pool for conducting the inspection mission.

2.2 Optimal Relative Orbit Initialization and Reconfiguration

The trajectory design for stable relative orbit initialization and orbit reconfiguration is formulated as a fixed-time optimal control problem with $\mathcal{L}_1$ fuel cost. For N observer spacecraft the optimal control problem is given in the following problem.

Problem 1. Centralized Continuous Time Relative Orbit Reconfiguration

$$\mathcal{J}_{\text{PRO}} = \min_{\mathbf{x}_i, \mathbf{u}_i, i \in \{1,\dots,N\}} \sum_{i=1}^{N} \int_{t_0}^{t_f} \|\mathbf{u}_i(t)\|_1 dt \tag{2.8}$$

$$\text{s.t} \quad \dot{\mathbf{x}}_i = \mathbf{A}\mathbf{x}_i + \mathbf{B}\mathbf{u}_i \quad \forall t \in [t_0, t_f] \tag{2.9}$$

$$\|\mathbf{C}(\mathbf{x}_i - \mathbf{x}_j)\|_2 \geq r_c \quad \forall t \in [t_0, t_f], \quad j > i, \quad j = \{2, \dots, N\} \tag{2.10}$$

$$\mathbf{u}_i \in \mathcal{U} \quad \forall t \in [t_0, t_f] \tag{2.11}$$

$$\mathbf{x}_i(t_0) = \mathbf{x}_{i0} \quad \& \quad \mathbf{x}_i(t_f) = \mathbf{x}_{if}, \tag{2.12}$$

where i denotes the observer and $i \in \{1, \dots, N\}$, $\mathbf{C} = [\mathbb{I}_{3\times3}\ 0_{3\times3}]$, r_c is the minimum safe allowable distance between two spacecraft, $\mathcal{U}$ is a convex control constraint set, and $\mathbf{x}_{i0}$ and $\mathbf{x}_{if}$ are the initial and terminal conditions of the i^{th} observer. The optimal control problem is transformed to a convex optimization problem by using the discrete dynamics in (2.5) and convexifying the collision constraint in (2.10) about a nominal trajectory $(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i)$, given the full trajectory $(\bar{\mathbf{x}}_j, \bar{\mathbf{u}}_j)$ of the neighboring observer spacecraft j, as discussed in prior work [15, 127]. For each observer spacecraft i the following decentralized convex optimization problem is solved to compute the trajectory.

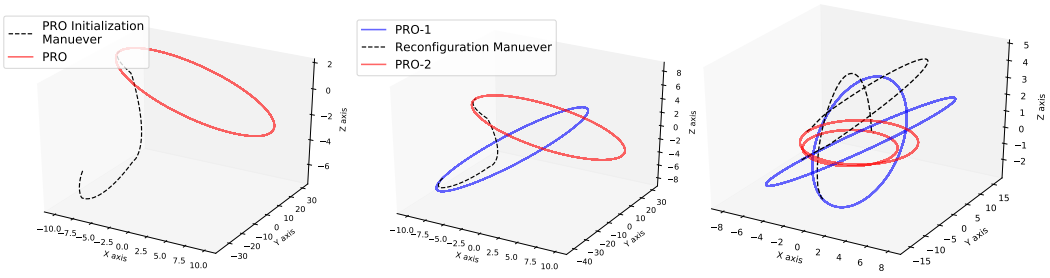


Figure 2.2: An example of the optimal PRO initialization maneuver and PRO reconfiguration maneuver computed using Problem 1 and 2. We designed the PRO's using the stable subspace proposed in [16].

Problem 2. Decentralized Discrete Time Convex Relative Orbit Reconfiguration

$$\mathcal{J}_{\mathrm{PRO}_i} = \min_{\mathbf{x}_i, \mathbf{u}_i} \sum_{k=0}^{T-1} \|\mathbf{u}_i[k]\|_1 \, \Delta t \tag{2.13}$$

$$\text{s.t} \quad \mathbf{x}_i[k+1] = \mathbf{A}_{\mathrm{d}} \mathbf{x}_i[k] + \mathbf{B}_{\mathrm{d}} \mathbf{u}_i[k] \quad \forall k \in \{1, \ldots, T-1\} \tag{2.14}$$

$$(\bar{\mathbf{x}}_i[k] - \bar{\mathbf{x}}_j[k])^\top \mathbf{C}^\top \mathbf{C} (\mathbf{x}_i[k] - \bar{\mathbf{x}}_j[k]) \geq r_c \|\mathbf{C}(\bar{\mathbf{x}}_i[k] - \bar{\mathbf{x}}_j[k])\|_2$$

$$\forall k \in \{0, 1, \ldots, T\}, \quad j > i, \quad j = \{2, \ldots, N\} \tag{2.15}$$

$$\mathbf{u}_i[k] \in \mathcal{U} \quad \forall k \in \{0, 1, \ldots, T-1\} \tag{2.16}$$

$$\mathbf{x}_i[0] = \mathbf{x}_{i0} \quad \& \quad \mathbf{x}_i[T] = \mathbf{x}_{if}, \tag{2.17}$$

where T is the number of time steps in the time interval $[t_0, t_f]$, $k = 0$ and $k = T$ correspond to the initial condition and terminal condition, respectively. Problem 2 is used to compute a safe trajectory for stable relative orbit initialization and reconfiguration. The choice of initial and terminal conditions for each phase of the mission in Fig. 3.1 is discussed in Section 3.2. The convex program is initialized with a straight line trajectory from initial to terminal conditions as the nominal trajectory. A *model predictive control* (MPC) formulation of the above problem can be used for controlling the spacecraft to track the designed trajectory [15, 127]. The orbital dynamics and attitude dynamics are assumed to be decoupled for the motion planning and control design. We discuss the attitude dynamics and trajectory design problem formulation in the following sections.

2.3 Attitude Dynamics

The attitude of the spacecraft can be represented by any attitude representations including quaternions [128], Modified Rodrigues Parameters (MRPs) [128], and SO(3) rotation matrices. For the attitude planning problem we use quaternions $\mathbf{q} \in \mathbb{R}^4$ and $\|\mathbf{q}\|_2 = 1$, $\mathbf{q} \in \mathbb{H}$, where $\mathbb{H}$ is the Hamiltonian space and have a

bijective mapping to the quaternion sphere. The attitude kinematics equation is given using the body angular rates $\boldsymbol{\omega} = [\omega_1,\ \omega_2,\ \omega_3]^\top \in \mathbb{R}^3$. The kinematics of quaternions are given as

$$\dot{\mathbf{q}} = \Omega(\boldsymbol{\omega})\mathbf{q}, \tag{2.18}$$

where $\Omega(\boldsymbol{\omega})$ is given in (2.41). The attitude dynamics is as follows:

$$\dot{\boldsymbol{\omega}} = -\mathbf{J}^{-1}(\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}) + \mathbf{J}^{-1}\boldsymbol{\tau}, \tag{2.19}$$

where $\mathbf{J}$ is inertia tensor of the observer, $\boldsymbol{\tau}$ is the input torque, and $\times$ denotes the vector cross product. The continuous dynamics is linearized around a nominal trajectory $(\bar{\mathbf{q}}, \bar{\boldsymbol{\omega}})$ and the functional form of the linear system is defined in the following equations:

$$\dot{\mathbf{q}} = \mathbf{A}_{1q}(\mathbf{q} - \bar{\mathbf{q}}) + \mathbf{A}_{1\boldsymbol{\omega}}(\boldsymbol{\omega} - \bar{\boldsymbol{\omega}}) + \Omega(\bar{\boldsymbol{\omega}})\bar{\mathbf{q}}, \tag{2.20}$$

$$\text{where } \mathbf{A}_{1q} = \frac{\partial \Omega \mathbf{q}}{\partial \mathbf{q}}\Big|_{(\mathbf{q},\boldsymbol{\omega})=(\bar{\mathbf{q}},\bar{\boldsymbol{\omega}})}, \mathbf{A}_{1\omega} = \frac{\partial \Omega \mathbf{q}}{\partial \boldsymbol{\omega}}\Big|_{(\mathbf{q},\boldsymbol{\omega})=(\bar{\mathbf{q}},\bar{\boldsymbol{\omega}})} \tag{2.21}$$

$$\dot{\boldsymbol{\omega}} = \mathbf{A}_{2\omega}(\boldsymbol{\omega} - \bar{\boldsymbol{\omega}}) + \mathbf{J}^{-1}\boldsymbol{\tau} - \mathbf{J}^{-1}(\bar{\boldsymbol{\omega}} \times \mathbf{J})\bar{\boldsymbol{\omega}}, \tag{2.22}$$

$$\text{where } \mathbf{A}_{2\omega} = -\frac{\partial(\mathbf{J}^{-1}(\bar{\boldsymbol{\omega}} \times \mathbf{J})\bar{\boldsymbol{\omega}})}{\partial \boldsymbol{\omega}}\Big|_{\boldsymbol{\omega}=\bar{\boldsymbol{\omega}}}. \tag{2.23}$$

The linearized equations are discretized for a time interval Δt_a and time steps T_a and represented as follows to use in the sequential convex programming formulation:

$$\begin{aligned}\mathbf{q}[k+1] = \mathbf{A}_{1q}(\bar{\mathbf{q}}, \bar{\boldsymbol{\omega}}, \Delta t_a)[k]\mathbf{q}[k] + \mathbf{A}_{1\omega}(\bar{\mathbf{q}}, \bar{\boldsymbol{\omega}}, \Delta t_a)[k]\mathbf{q}[k] \\ + \mathbf{C}_1(\bar{\mathbf{q}}, \bar{\boldsymbol{\omega}}, \Delta t_a)[k],\end{aligned} \tag{2.24}$$

$$\boldsymbol{\omega}[k+1] = \mathbf{A}_{2\omega}(\bar{\boldsymbol{\omega}}, \Delta t_a)[k] + \mathbf{J}^{-1}\boldsymbol{\tau}[k] + \mathbf{C}_2(\bar{\boldsymbol{\omega}}, \Delta t_a)[k], \tag{2.25}$$

where $k \in \{1, \ldots, T_a\}$ is the time step. The matrices $\mathbf{A}_{1q}$, $\mathbf{A}_{1\omega}$, $\mathbf{C}_1$, $\mathbf{A}_{2\omega}$, and $\mathbf{C}_2$ are computed using the symbolic Python package `SymPy`. The linear, discrete form of the attitude dynamics is used to construct a sequential convex programming problem for designing a smooth and optimal attitude trajectory.

2.4 Optimal Attitude Trajectory Planning Problem

The attitude planning problem is formulated as the following optimal control problem with a norm constraint on the quaternions:

Problem 3. Continuous Time Attitude Planning

$$\begin{aligned}
\mathcal{J}_{\text{attitude}} = \min_{\mathbf{q},\boldsymbol{\omega},\boldsymbol{\tau}} & \int_{t_0}^{t_f} \|\boldsymbol{\tau}(t)\|_2 dt \\
\text{s.t} \quad & \dot{\mathbf{q}} = \Omega(\boldsymbol{\omega})\mathbf{q} \quad \forall t \in [t_0, t_f] \\
& \dot{\boldsymbol{\omega}} = -\mathbf{J}^{-1}\left(\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}\right) + \mathbf{J}^{-1}\boldsymbol{\tau} \quad \forall t \in [t_0, t_f] \\
& \mathbf{q}^\top \mathbf{q} = 1 \quad \forall t \in [t_0, t_f] \\
& \|\boldsymbol{\tau}\|_\infty \leq \tau_{\max} \quad \forall\, t \in [t_0, t_f] \\
& \mathbf{q}(t_0) = \mathbf{q}_0, \boldsymbol{\omega}(t_0) = \boldsymbol{\omega}_0 \quad \& \quad \mathbf{q}(t_f) = \mathbf{q}_f, \boldsymbol{\omega}(t_f) = \boldsymbol{\omega}_f,
\end{aligned} \tag{2.26}$$

where $\tau_{\max}$ is the maximum torque that can be applied on the spacecraft using reaction wheels, and $(\mathbf{q}_0, \boldsymbol{\omega}_0)$ and $(\mathbf{q}_f, \boldsymbol{\omega}_f)$ are the initial and terminal orientation and angular velocity, respectively. The solution of Problem 3 is a minimum energy attitude maneuver from initial orientation $\mathbf{q}_0$ to a terminal orientation $\mathbf{q}_f$. We use the discrete dynamics in (2.24) and (2.25) to formulate the following convex optimization problem given a nominal attitude $\bar{\mathbf{q}}$ and angular velocity trajectory $\bar{\boldsymbol{\omega}}$.

Problem 4. Linearized Discrete Time Attitude Planning

$$\mathcal{J}_{\text{attitude}} = \min_{\mathbf{q},\boldsymbol{\omega},\boldsymbol{\tau}} \sum_{k=0}^{T_a} \|\boldsymbol{\tau}[k]\|_2 \Delta t_a \tag{2.27}$$

$$\text{s.t} \quad \text{(2.24) and (2.25)} \quad \forall k \in \{0, \ldots, T_a - 1\} \tag{2.28}$$

$$\boldsymbol{\tau}[k+1] - \boldsymbol{\tau}[k] \leq \alpha \Delta t_a \quad k \in \{0, \ldots, T_a - 2\} \tag{2.29}$$

$$\|\boldsymbol{\tau}[k]\|_\infty \leq \tau_{\max} \quad k \in \{0, \ldots, T_a\} \tag{2.30}$$

$$\mathbf{q}[0] = \mathbf{q}_0, \boldsymbol{\omega}[0] = \boldsymbol{\omega}_0 \quad \& \quad \mathbf{q}[T_a] = \mathbf{q}_f, \boldsymbol{\omega}[T_a] = \boldsymbol{\omega}_f. \tag{2.31}$$

We use a normalized nominal attitude trajectory $\frac{\bar{\mathbf{q}}}{\|\bar{\mathbf{q}}\|}$ for linearization in (2.24) and (2.25), to impose the norm constraint $\|\mathbf{q}\|_2 = 1$. We additionally introduce the constraint in (2.29) to account for the actuator dynamics, where α is a constant depending on the input signal to the actuators. For a given initial and terminal quaternion $\mathbf{q}_0$ and $\mathbf{q}_f$, we initialize Problem 4 by computing the nominal quaternion $\bar{\mathbf{q}}$ trajectory using slerp interpolation. The nominal angular velocity trajectory $\bar{\boldsymbol{\omega}}$ is computed using $\bar{\mathbf{q}}$ and the kinematics in (2.18). Problem 4 is solved using sequential convex programming. The attitude motion plan is tracked using the nonlinear control law derived in prior work [129]. In Chapter 3, we integrate Problem 2, Problem 4, and control algorithms [15, 129] with an information cost and derive an algorithm for active inspection of a spacecraft.

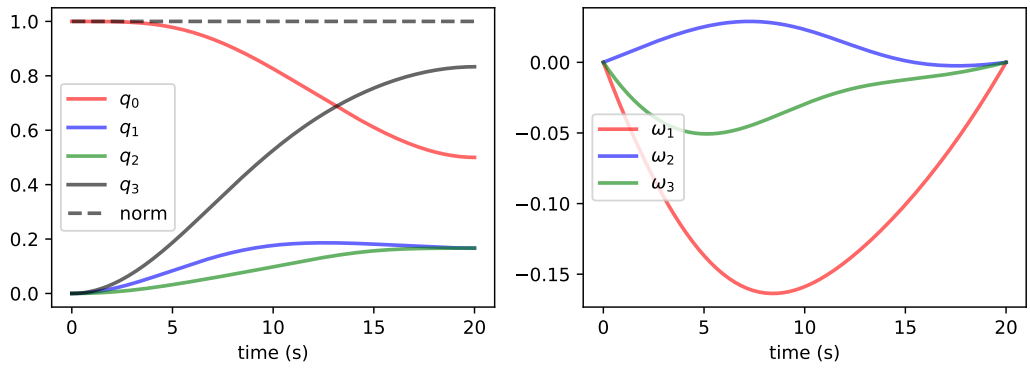


Figure 2.3: An example of the optimal attitude trajectory computed using Problem 4 for the initial and terminal states $\mathbf{q}[0] = [1, 0, 0, 0]$, $\boldsymbol{\omega}[0] = 0$, $\mathbf{q}[T_a] = [0.5, 0.167, 0.167, 0.833]$, and $\boldsymbol{\omega}[T_a] = 0$, respectively.

2.5 Generalized Polynomial Chaos

The generalized Polynomial Chaos (gPC) [42, 43, 93] expansion theory is used to model uncertainty with finite second-order moments as a series expansion of orthogonal polynomials. The polynomials are orthogonal with respect to a known density function $\boldsymbol{\zeta}(.)$. Consider the random vector ξ with independent identically distributed (i.i.d) random variables $\{\xi_i\}_{i=1}^{d_\xi}$ as elements. Each $\xi_i \sim \mathcal{N}(0, 1)$ is normally distributed with zero mean and unit variance. The random vector $\mathbf{x}(t) \in \mathcal{X} \subseteq \mathbb{R}^{d_\mathbf{x}}$ at time t, can be expressed as the following series:

$$x_i(t) = \sum_{j=0}^{\infty} x_{ij}(t)\phi_j(\xi), \tag{2.32}$$

where x_i denote the i^{th} element in the vector $\mathbf{x} \in \mathcal{X}$ and x_{ij} is the j^{th} coefficient in the series expansion. The dimension d_ξ is the sum of number of random inputs in the SDE (4.7) and the number of random initial conditions. The functions $\phi_j(\xi)$ are constructed using the Hermite polynomial [42] basis functions. The functions $\phi_j(\xi)$ are orthogonal with respect to the joint probability density function $\boldsymbol{\zeta}(\xi) = \zeta(\xi_1)\zeta(\xi_2)\cdots\zeta(\xi_{d_\xi})$, where $\zeta(\xi_k) = \frac{1}{\sqrt{2\pi}} e^{\frac{-\xi_k^2}{2}}$. The choice of the orthogonal polynomials depends on the uncertainty model effecting the dynamics. We refer to [44] for details on type and construction of the polynomials for different standard uncertainty models such as uniform, beta, and Poisson distributions.

Remark 1. The series expansion is truncated to a finite number $\ell + 1$ as $x_i \approx \sum_{j=0}^{\ell} x_{ij}(t)\phi_j(\xi)$ based on the maximum degree of the polynomials $\mathrm{P}_{\mathrm{gPC}}$ required to represent the variable $\mathbf{x}$. The minimum ℓ required to appropriately represent $\mathbf{x}$ with uncertainty parameter $\xi \in \mathbb{R}^{d_\xi}$ is given by $\ell = \binom{\mathrm{P}_{\mathrm{gPC}}+d_\xi}{d_\xi} - 1$.

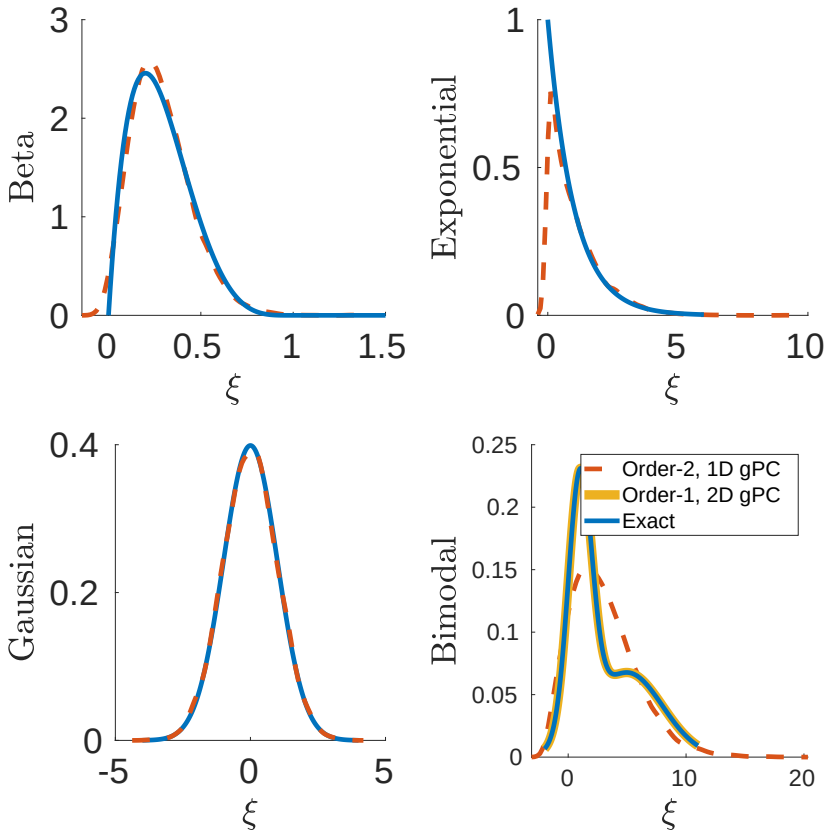


Figure 2.4: Example gPC approximation of some standard probability distribution functions (PDF) using gPC expansion. For the beta and exponential distributions, gPC expansion represents the PDF well with just second order approximation. For a Gaussian distribution, the gPC representation is exact.

The coefficients $x_{ij}(t)$ are computed using the Galerkin projection given by the following equation:

$$x_{ij}(t) = \frac{\int_{\mathbb{D}} \boldsymbol{\zeta}(\xi) x_i(t) \phi_j(\xi) d\xi}{\langle \phi_j(\xi), \phi_j(\xi) \rangle}, \tag{2.33}$$

where $\langle \phi_i(\xi), \phi_j(\xi) \rangle = \int_{\mathbb{D}} \boldsymbol{\zeta}(\xi) \phi_i(\xi) \phi_j(\xi) d\xi$. For non-polynomial functions, the Galerkin projection is computed using the Stochastic Collocation [43] method as follows:

$$\int_{\mathbb{D}} \boldsymbol{\zeta}(\xi) x_i(t) \phi_j(\xi) d\xi \approx \sum_{k=1}^{m} w_k x_i(t) \phi_j(n_k), \tag{2.34}$$

where Gauss-Hermite quadrature is used to generate the nodes n_k and the corresponding node weights w_k. Following Lemmas 1 and 2 discuss the convergence of the gPC expansion to the true distribution and the error due to truncated polynomial approximation of a distribution.

Lemma 1. *(Cameron-Martin Theorem [130]) The gPC series approximation in (2.32) converges to the true value $x_i \in \mathcal{L}_2$.*

$$\|x_i(t) - \sum_{j=0}^{\ell} x_{ij}(t)\phi_j(\xi)\|_{\mathcal{L}_2} \to 0, \text{ as } \ell \to \infty \; \forall \, t \in [t_0, t_f]. \tag{2.35}$$

Remark 2. The expectation $\mathbb{E}(x_i)$ and variance Σ_{x_i} of the random variable x_i can be expressed in terms of the coefficients of the expansion as follows:

$$\mathbb{E}(x_i) = x_{i0}, \quad \Sigma_{x_i} \approx \sum_{j=1}^{\ell} x_{ij}^2 \langle \phi_j, \phi_j \rangle \text{ as } \ell \to \infty. \tag{2.36}$$

Lemma 2. *(Truncation Error Theorem [131]) If an element x_i of the random variable $\mathbf{x}$ is represented using ℓ polynomials, then the approximation error is given as follows:*

$$\|x_i - \sum_{j=0}^{\ell} x_{ij}(t)\phi_j(\xi)\| = \|e_\ell\| \leq \sqrt{\sum_{j=\ell+1}^{\infty} x_{ij}^2 \|\phi_j\|^2}. \tag{2.37}$$

Lemmas 1 and 2, and Remark 2 will be used in studying the convergence of the gPC approximation of the cost function, the SDE and the chance constraints. Furthermore, the higher-order moments can be expressed as a polynomial function of the coefficients.

Curse of Dimensionality

The truncated polynomial expansion is a finite-dimensional approximation of the random variable. The number of polynomials ℓ grow exponentially large based on the degree of polynomial used to represent the state distribution. The large dimensionality can be reduced, inducing sparsity in the gPC expansion, by using techniques like sparse gPC [132], and data-driven gPC [133]. A cost effective approach to estimate moments up to second order is to use gPC polynomials up to degree 2, i.e., $\mathrm{P}_{\mathrm{gPC}} = 2$ [134]. The computationally complexity for $\mathrm{P}_{\mathrm{gPC}} = 2$ is equivalent to linear covariance propagation. Note that, unlike the linear covariance propagation method, the gPC method with $\mathrm{P}_{\mathrm{gPC}} = 2$ accounts for the coupling between the state x and the white-noise process dw.

2.6 Chapter Summary

We described the formation flying dynamics, optimal control problem formulation for PRO initialization and reconfiguration, optimal attitude trajectory planning problem, and the generalized polynomial chaos method for uncertainty propagation.

Appendix

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix}, \quad and \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ f_{\text{thr}} & 0 & 0 \\ 0 & f_{\text{thr}} & 0 \\ 0 & 0 & f_{\text{thr}} \end{bmatrix} \tag{2.38}$$

$$\mathbf{A}_d = \begin{bmatrix} 4-3\cos n\Delta t & 0 & 0 & \frac{1}{n}\sin n\Delta t & \frac{2}{n}(1-\cos n\Delta t) & 0 \\ 6(\sin n\Delta t - n\Delta t) & 1 & 0 & -\frac{2}{n}(1-\cos n\Delta t) & \frac{1}{n}(4\sin n\Delta t - 3n\Delta t) & 0 \\ 0 & 0 & \cos n\Delta t & 0 & 0 & \frac{1}{n}\sin n\Delta t \\ 3n\sin n\Delta t & 0 & 0 & \cos n\Delta t & 2\sin n\Delta t & 0 \\ -6n(1-\cos n\Delta t) & 0 & 0 & -2\sin n\Delta t & 4\cos n\Delta t - 3 & 0 \\ 0 & 0 & -n\sin n\Delta t & 0 & 0 & \cos n\Delta t \end{bmatrix} \tag{2.39}$$

$$\mathbf{B}_{\text{d}} = \frac{f_{\text{thr}}}{m} \begin{bmatrix} \frac{1}{n^2}(1-\cos n\Delta t) & \frac{2}{n^2}(n\Delta t - \sin n\Delta t) & 0 \\ -\frac{2}{n^2}(n\Delta t - \sin n\Delta t) & \frac{4}{n^2}(1-\cos n\Delta t) - \frac{3}{2}\Delta t^2 & 0 \\ 0 & 0 & \frac{1}{n^2}(1-\cos n\Delta t) \\ \frac{1}{n}\sin n\Delta t & \frac{2}{n}(1-\cos n\Delta t) & 0 \\ -\frac{2}{n}(1-\cos n\Delta t) & \frac{4}{n}\sin n\Delta t - 3\Delta t & 0 \\ 0 & 0 & \frac{1}{n}\sin n\Delta t \end{bmatrix} \tag{2.40}$$

$$\Omega(\boldsymbol{\omega}) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \tag{2.41}$$

Chapter 3

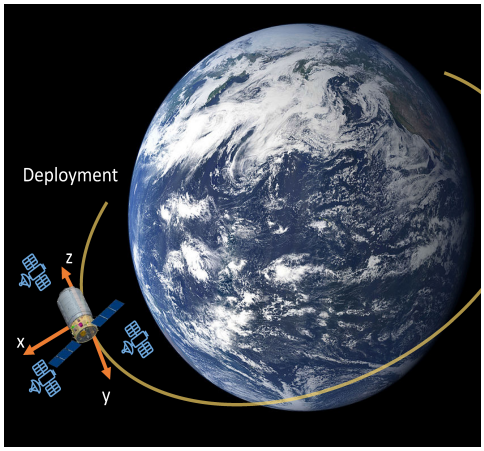
INFORMATION-BASED GUIDANCE AND CONTROL FOR ON-ORBIT INSPECTION

In this chapter, we discuss the concept of operations of a typical Earth orbit inspection mission with the target spacecraft, information-based optimal control problem formulation for inspection. We design a hierarchical algorithm to solve the information-based optimal control problem. We demonstrate inspection of a target spacecraft with multiple deputy spacecraft using the proposed algorithm. We validate the algorithm on the robotic spacecraft simulator (see Chapter 7) testbed.

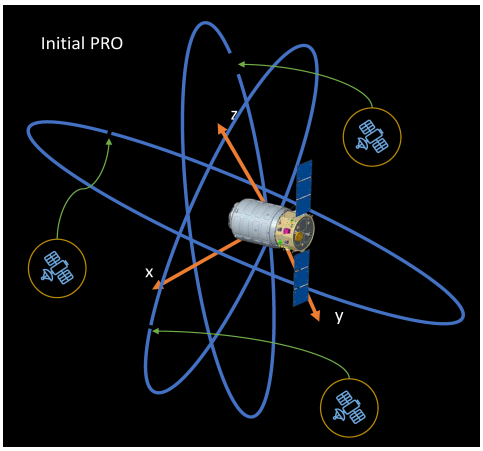
The scenario considered in this thesis has three phases (see Figure 3.1). In the first phase, the small observer spacecraft are deployed from the target spacecraft and begin a drift phase. The drifting spacecraft are then inserted into a parking PRO or an initial PRO in the second phase. In the third phase, the spacecraft in stable relative orbits are used for inspecting the target. As needed, the spacecraft reconfigure to a new set of PROs to inspect a previously unobserved surface area on the target spacecraft. In this chapter, we use the Hills-Clohessy-Wiltshire (HCW) equations to describe the relative orbital dynamics of the observer CubeSats. For the stable relative orbit initialization and reconfiguration phase, we formulate an optimal control problem with $\mathcal{L}_1$ fuel cost and safety, energy matching as constraints, and solve it using sequential convex programming (SCP), similar to prior work [14]. The planned trajectories are tracked using a model predictive control formulation of the convexified problem. During the inspection phase, we represent the attitude dynamics using quaternions. The attitude planning is done using a combination of slerp interpolation and SCP with norm constraint on the quaternions. We use an existing nonlinear feedback controller for attitude tracking [129]. In the following, we review the information-based optimal control problem as applied to the coordinated inspection task.

3.1 Multi-Agent Motion Planning for Inspection

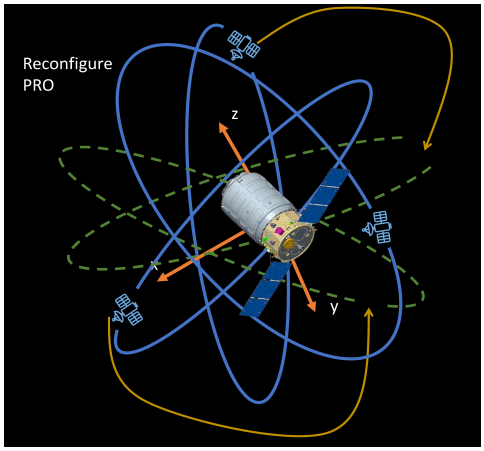
We design optimal PROs and the attitude trajectory for N observer spacecraft, inspecting M points of interest (POI) on a target spacecraft, by using the following information-based optimal control problem.



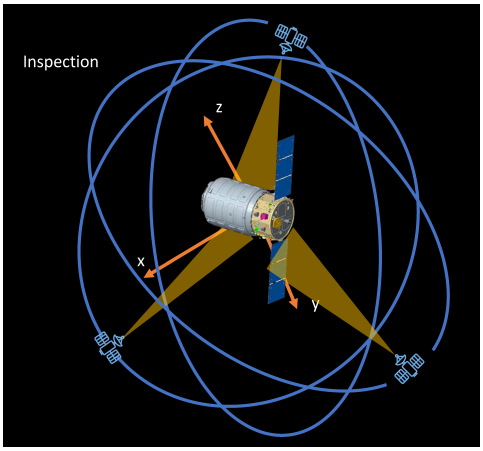
(a) Three observer spacecraft deployed from the target spacecraft.



(b) Initialization of the deployed observer spacecraft into stable relative orbits.



(c) Reconfiguration of the observer spacecraft to inspect a different surface area on the target.



(d) Pointing control for inspection of the target spacecraft.

Figure 3.1: Concept of operations of an inspection mission for observing a target spacecraft with multiple observer spacecraft deployed.

Problem 5. Information-Based Optimal Control Problem

$$\min_{\mathbf{p},\mathbf{u}_i} \quad \int_0^{t_f} \left(\sum_{\mathbf{s}_j \in \text{POI}} H(\mathbf{p}, \mathbf{s}_j) + \sum_i \|\mathbf{u}_i\| \right) dt \tag{3.1}$$

$$\textbf{s.t.} \begin{cases} \text{Dynamics Model :} & \dot{\mathbf{p}}_i = \mathbf{f}(\mathbf{p}_i, \mathbf{u}_i) \\ \text{Safe Set :} & \mathbf{p}_i \in \mathcal{P},\ \forall i \in \{1, \ldots, N\} \\ \text{Points of Interest :} & \mathbf{s}_j\ \forall j \in \{1, \ldots, M\} \\ \text{Inspection Sensor Model :} & \mathbf{z}_{i,j} = h(\mathbf{p}_i, \mathbf{s}_j) + \xi,\ \xi \sim \mathcal{N}\left(0, \Sigma_h(\mathbf{p}_i, \mathbf{s}_j)\right), \end{cases} \tag{3.2}$$

$$\text{Points of Interest :} \quad \mathbf{s}_j\ \forall j \in \{1, \ldots, M\} \tag{3.3}$$

where $\sum_{\mathbf{s}_j \in \mathrm{POI}} H(\mathbf{p}, \mathbf{s}_j)$ is the information cost, $\sum_i \|\mathbf{u}_i\|$ is the fuel cost, $\mathbf{p}_i$ is the full-pose of the observer spacecraft, $\mathbf{s}_j$ is the full-pose of the j^{th} POI on the target spacecraft. The inspection sensor model in (3.3) outputs the value of interest $\mathbf{z}_{i,j}$, when the i^{th} observer with pose $\mathbf{p}_i$ is inspecting a POI at $\mathbf{s}_j$. Minimizing the information cost $\sum_{\mathbf{s}_j \in \mathrm{POI}} H(\mathbf{p}, \mathbf{s}_j)$ ensures that the inspection task is complete. We discuss the choice of the information cost and the sensor model in Section 3.2.

We decompose the Problem 5 to derive the proposed hierarchical algorithm in Section 3.2. In the hierarchical algorithm, we use the information-cost and the sensor model to select the informative PROs and attitude for each agent. We optimize the informative PROs and attitude plan for optimal orbit insertion, reconfiguration, and attitude tracking using the optimal control problems discussed Sections 2.2 and 2.4.

3.2 Main Distributed Inspection Algorithm

Our approach has four major components: information estimation, state estimation, motion planning, and control, see Fig. 1.1. The information estimation quantifies the information gain of the target spacecraft, given past and potential future poses of all the observer spacecraft. The estimated information gain is a crucial input to the motion planner, which considers a set of alternative *passive relative orbits* (PROs) and decides when and how to reconfigure the spacecraft, while approximately maximizing the information gain and minimizing the control effort. For computational efficiency, both information estimation and motion planning modules rely on simplified attitude dynamics. Our controls component tracks the planned motions at runtime. We assume that an accurate estimate of the LVLH frame and the full state of the observer spacecraft in the LVLH frame and the target spacecraft in the ECI frame are available at all times during the guidance and control.

We first discuss the centralized implementation of the algorithm and then discuss on the extension to distributed architecture with communication between observers. The algorithm is implemented using a hierarchical framework, where different components are executed at different frequencies (see Figure 3.2). In an offline preprocessing stage, we use a rough prior model of the target spacecraft to generate PRO candidates as well as a ray casting database, which is a data structure that enables efficient computation of the information gain at runtime. Our motion planning module runs at a low rate ω_{low}, e.g., once per orbit. It uses the PRO candidates, ray casting database, as well as information database to decide whether and how the spacecraft should reconfigure to different PROs. These PROs are maintained us-

ing our PRO controller at a mid-rate ω_{mid}, e.g., every few seconds. At the same timescale, we plan for new desired attitudes, take pictures of the target spacecraft, and update our relative state estimate and information cost. At the highest rate ω_{high}, e.g., several times per second, we use attitude trajectory optimization and control. The pseudo code of our approach is given in Algorithm 1 for centralized implementation.

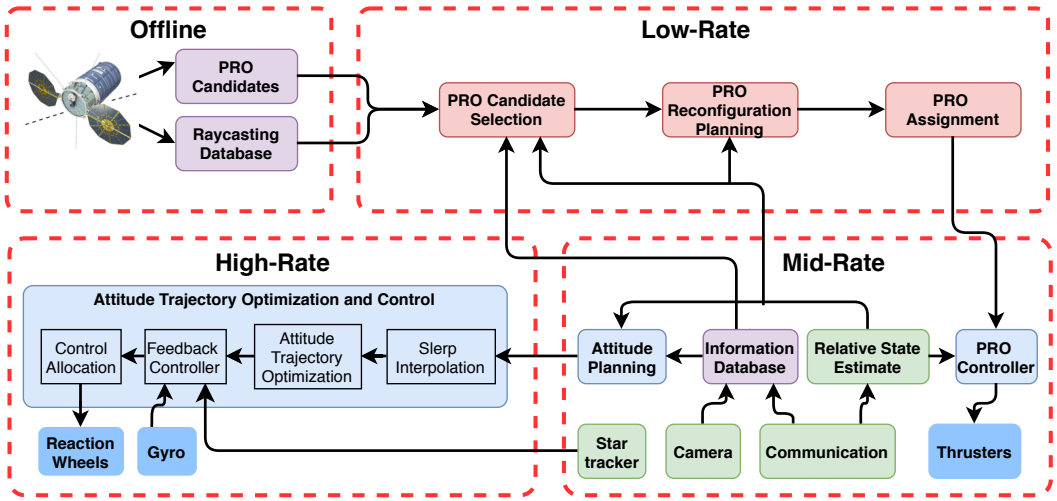


Figure 3.2: The multi-level and multi-timescale hierarchical architecture for combined orbit and attitude planning and control for an inspection task.

3.2.1 Prerequisites: Information Metric

Information Gain

To quantify the information, we assume that we have a (rough) prior model of the target spacecraft and sample *points of interest* (POIs) on its surface. The overall goal is then to minimize the variance of the estimation of all POIs. Specifically, we introduce the *information cost*, H, based on prior work [69], as follows:

$$H_{\mathrm{POI}}(\mathbf{s}) = \left(w^{-1} + \sum_{\mathbf{p} \in \mathcal{P}} \sigma(\mathbf{p}, \mathbf{s})^{-1} \right)^{-1}$$
$$H = \sum_{\mathbf{s} \in \mathrm{POIs}} H_{\mathrm{POI}}(\mathbf{s}) \phi(\mathbf{s}), \tag{3.4}$$

where $\mathbf{s} \in \mathbb{R}^3$ is a sampled POI on the target spacecraft's surface, $w \in \mathbb{R}$ is the basic variance based on the prior model of the target spacecraft, $\mathbf{p} \in SE(3)$ is the pose of a sensor mounted on a spacecraft such as a camera, $\mathcal{P}$ is the set of all sensor poses, $\sigma(\mathbf{p}, \mathbf{s})$ estimates the variance of estimating POI at $\mathbf{s}$ with the sensor

Algorithm 1: Multi-level Hierarchical Guidance and Control Architecture.

Input: Prior rough model of the target, points of interest (POI) on its surface, and the number of observers N

Input: Initialized instance of the *information cost* (Section 3.2.1) and *information database* (Section 3.2.1)

Input: Minimum safe distance $d_{\min}$ and desired sensing distance d_s from the observer to the target, sensor field of view

Input: Pose estimates of all the observers and the LVLH frame available to all the observers at all time

Output: Autonomous inspection

▷*Pre-processing [Offline]*

PreProcessing (Input)
- Generate *PRO candidate* pool as discussed in Section 3.2.2,
- Generate *ray casting database*,
- Initialize *information database* using prior model,
- **return** PRO candidates, information database, ray casting database

▷*Low-Rate: [loop frequency: ω_{low}]*

LowRate (information database, PRO candidates)
- Generate an optimal set of N PROs that minimizes *information cost*, see Section 3.2.3,
- **for** observer *in* $\{1, \ldots, N\}$ **do**
  - Compute N *reconfiguration plans* for the *observer* to each selected PRO, see Section 3.2.3,
- Compute optimal *PRO assignment* to each observer that minimizes total reconfiguration cost, see Section 3.2.3,
- **return** *reconfiguration plan*, *PRO assignment*

▷*Mid-Rate: [loop frequency: ω_{mid}]*

MidRate (reconfiguration plan, PRO assignment)
- **for** observer *in* $\{1, \ldots, N\}$ **do**
  - **if** reconfiguration plan *is available* **then**
    - Execute *reconfiguration plan* using MPC controller discussed in [15],
    - Plan attitude using visibility check with the *ray casting database*, *information update*, see Section 3.2.4,
  - **else**
    - Propagate the assigned PRO, HCW dynamics using (2.7),
    - Maintain Stable HCW using *PRO controller*, see Section 3.2.4,
    - Plan attitude using visibility check with the *ray casting database*, *information update*, see Section 3.2.4,
- **return** *attitude plan*, *information update*

▷*High-Rate: [loop frequency: ω_{high}]*

HighRate (attitude plan)
- **for** observer *in* $\{1, \ldots, N\}$ **do**
  - Attitude trajectory optimization using Problem 4,
  - Nonlinear attitude control using the reference input [129]

at $\mathbf{p}$, and $\phi(\mathbf{s}) \in \mathbb{R}$ is the relative importance of POI $\mathbf{s}$. The cost H is similar to Bayesian sensor fusion, where the estimates are fused using the confidence in the noisy sensor outputs. This formulation can handle important cases such as multiple spacecraft observing the same POI well. Additionally, minimizing H also minimizes the expected variance of the estimation of the POIs [69].

For a camera, the function $\sigma(\cdot, \cdot)$ corresponds to information per pixel. Minimizing σ entails computing the location and orientation of the camera to optimally capture a POI. The choice of σ can directly incorporate other sensor or spacecraft characteristics, such as the current uncertainty of the spacecraft's pose estimate, the accuracy of the sensor based on the distance between $\mathbf{p}$ and $\mathbf{s}$, or the lighting conditions. Here, we use a simple formulation of σ assuming an RGB camera sensor and no environmental noise [135]:

$$\sigma(\mathbf{p}, \mathbf{s}) \propto \begin{cases} \mathrm{dist}^2(\mathbf{p}, \mathbf{s}) & \mathbf{s} \text{ visible from } \mathbf{p} \\ \infty & \text{otherwise} \end{cases}, \tag{3.5}$$

where $\mathrm{dist}(\mathbf{p}, \mathbf{s})$ is the Euclidean distance between POI $\mathbf{s}$ and pose $\mathbf{p}$. In practice, computing σ requires an efficient visibility check. Since we have a rough prior model of the target spacecraft, we propose to use a *ray casting database*, which uses pre-processing to speed up visibility checks significantly, see Section 3.2.2.

Information Database

Let $\mathcal{P}$ be the set of all sensor poses. For each POI $\mathbf{s}$, we store the set of sensor poses $\mathbf{p} \in \mathcal{P}$ that can observe $\mathbf{s}$ as $\mathcal{P}_{\mathrm{POI}}(\mathbf{s})$, i.e.,

$$\mathcal{P}_{\mathrm{POI}}(\mathbf{s}) = \{\mathbf{p} \in \mathcal{P} | \sigma(\mathbf{p}, \mathbf{s}) \text{ finite}\}. \tag{3.6}$$

We call the data structure that stores $\mathcal{P}_{\mathrm{POI}}(\mathbf{s})$ for all $\mathbf{s} \in$ POIs *information database*. We use the fact that our state estimator uses the common LVLH frame for all the observers, making direct pose exchanges feasible. The information database is compact and can be stored locally on each observer. Furthermore, if the covariance of the pose estimate is available, it can be communicated and used as well for an improved uncertainty sensor model σ.

3.2.2 Pre-processing: PRO Candidates and Ray Casting

PRO Candidate Computation

Passive Relative Orbits (PROs) play an important role in the proposed GNC architecture as it provides thrust-free orbits around the target spacecraft that can be

utilized for on-orbit inspection and mapping. With prior knowledge of the target spacecraft model and the desired sensing distance of a given sensor, we can pre-compute a finite set of PRO candidates that will be utilized throughout the on-orbit inspection and mapping in real-time.

To generate a set of PRO candidates to geometrically cover the target spacecraft from varying perspectives in a systematic manner, we first express the solution to HCW (2.3) from Cartesian to Phase-Magnitude [125] form:

$$x(t) = \rho_x \sin(nt + \alpha_x), \quad y(t) = \rho_y + 2\rho_x \cos(nt + \alpha_x), \quad z(t) = \rho_z \sin(nt + \alpha_z) \tag{3.7}$$

$$\rho_x = \frac{\sqrt{\dot{x}_0^2 + x_0^2 n^2}}{n}, \quad \rho_y = y_0 - 2\dot{x}_0/n, \quad \rho_z = \frac{\sqrt{\dot{z}_0^2 + z_0^2 n^2}}{n} \tag{3.8}$$

$$\alpha_x = \tan^{-1}\frac{nx_0}{\dot{x}_0}, \qquad \alpha_z = \tan^{-1}\frac{nz_0}{\dot{z}_0}. \tag{3.9}$$

Then we set $\rho_y = 0$ to restrict the PRO candidates to be concentric around the target spacecraft. The concentric PROs ensure collision avoidance [16] between spacecraft as multiple orbits around the target spacecraft are used for inspection. We further confine the PRO candidates to have orbits whose distance to the target spacecraft varies around the desired sensing distance of the sensor d_s by setting $\rho_x = 0.5d_s$. Note that these concentric PROs will have semi-major axis of d_s and semiminor axis of $0.5d_s$ in its $x - y$ projection. Then the parameter ρ_z and α_z are varied from 0 to $2d_s$ and $-\pi/2$ to $\pi/2$, respectively, to generate PRO candidates to be utilized for the on-orbit inspection. The parameter ρ_z controls the eccentricity of the orbit and determines how far stretched a PRO is along the z-axis. The parameter α_z rotates a PRO about the z-axis and provides varying cross-sectional view of the target spacecraft. The PROs generated in this manner are essentially 3D ellipses that are different cross-sections of a 3D cylinder encompassing the target spacecraft. PROs generated using this technique cover only the radial surface of a cylinder around the target spacecraft. In order to generate the PROs that cover top and bottom surfaces of the target spacecraft, we set the parameter $\rho_z = d_s$, $\rho_y = 0$, $\alpha_x = 0$, and vary ρ_x and α_z in the range $[d_{\min}, 0.5d_s]$ and $[-\pi/2, \pi/2]$, respectively. The parameter $d_{\min}$ is the minimum safe distance to the target spacecraft. An example of generated PROs is shown in Fig. 3.3. Further, for each sampled PRO on the 3D cylinder, we compute the delta-V required to initialize stable PRO at that location.

The total number of PRO candidates to be pre-computed will depend on the computational resources available on the spacecraft. The cardinality of the PRO candidate

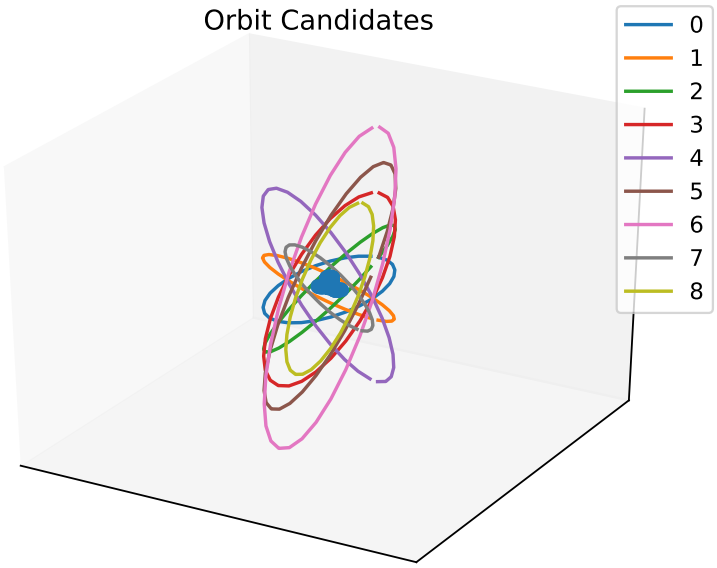


Figure 3.3: Nine generated PRO candidates to cover the target spacecraft (blue in the center).

set should be large enough to be able to densely cover the target spacecraft, but not too large to enable online computation of PRO reconfiguration and assignment planning with on-board computation in real-time.

Ray Casting Database

In practice, computing σ requires an efficient visibility check. Since we have a rough prior model of the target spacecraft, we propose to use a *ray casting database*, which uses pre-processing to speed up visibility checks significantly. First, we use the model to sample candidate points of interest (POIs). Then, we i) uniformly create potential rays, ii) for each POI we use ray casting to find the subset of the potential rays that are not occluded, and iii) we store the result in a hash map indexed by spherical coordinates of the rays for fast look-up. At runtime, we first convert the position of p to spherical coordinates and use bilinear mapping of the nearby candidate rays to determine visibility. Second, we consider the attitude of p and the field of view of the sensor. An example of the visibility check for a fixed camera is shown in Fig. 3.4.

3.2.3 Low-Rate: Orbit Motion Planning

The orbit motion planner uses the *PRO candidates*, a set of pre-computed passive relative orbits, and decides when and how each each of the observer spacecraft should switch to one of the orbits in that candidate set. Our approach uses three

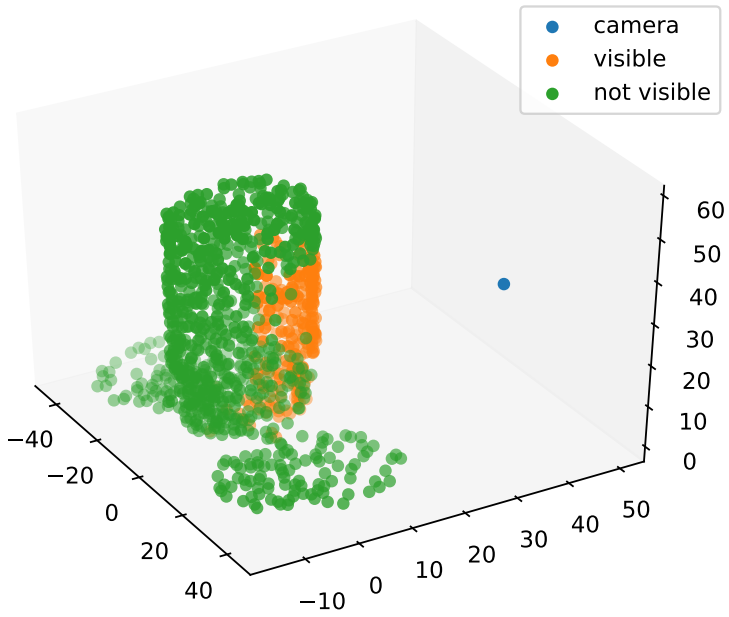


Figure 3.4: An example of the visibility check using ray casting database for a given camera with fixed location, pointed in $-y$ direction with a field of view of 30 degrees.

parts: i) *PRO candidate selection* based on minimizing the information cost H, ii) *PRO reconfiguration planning* to minimize delta-V, and iii) optimal *PRO assignment*.

PRO Candidate Selection

For the *PRO candidate selection*, we use a sequential greedy approach to generate a PRO set from the PRO candidate pool that will collectively minimize the *information cost* H when its orbits are committed by the observers. For each PRO candidate, we first compute an estimate of H that would be attained if an observer spacecraft takes measurements along the PRO. The estimation of H is computed by considering future sensor poses sampled from the candidate PRO with attitudes greedily assigned, together with the past sensor poses from all observers in the *information database*. Then, the candidate PRO with the lowest estimate of H is appended as an element of the PRO set. The procedure is repeated N times such that the cardinality of the PRO set generated is equal to the number of observers.

In our proposed multi-spacecraft GNC architecture, the sequential greedy approach is favored over combinatorial optimization for its computational efficiency. Selecting a PRO set that minimizes the *information cost* H with optimality guaran-

tee requires combinatorial computation. This approach has factorial scaling as the number of observers increase. In contrast, the sequential greedy approach scales linearly with N and is able to generate near-optimal solution whose optimality gap is sufficiently small in practice. Figure 3.5 demonstrates the *PRO candidate selection* using the sequential greedy approach and shows the resulting PRO set.

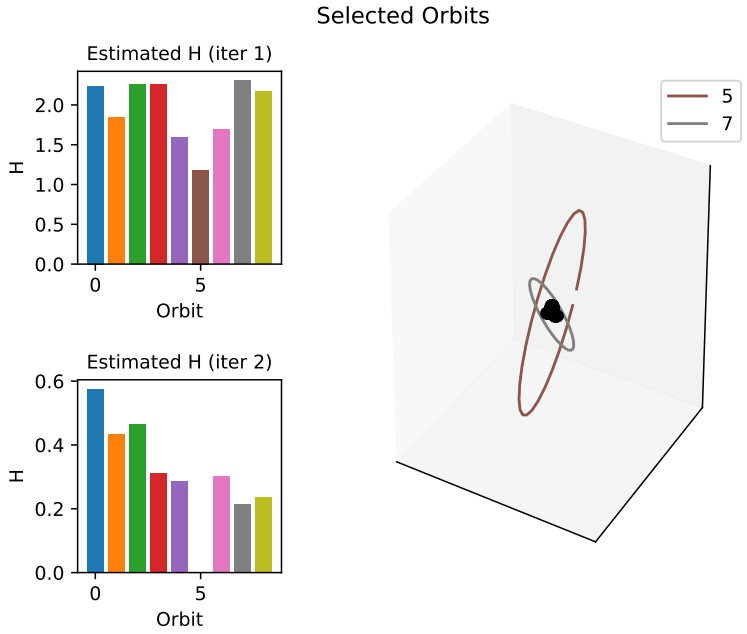


Figure 3.5: PRO candidate selection for $N = 2$ from the set of PROs shown in Fig. 3.3. For each of the nine candidates the expected information cost is computed (bar chart top left). PRO 5 achieves the lowest H and is selected in the first iteration. Assuming one observer is orbiting PRO 5, we estimate H for the remaining PROs in the second iteration. The lowest cost is achieved for PRO 7 (bar chart bottom left). Observer spacecraft on the selected PROs can observe the target from many perspectives (plot on the right).

PRO Reconfiguration Planning

For *PRO reconfiguration planning* we use sequential convex programming in Problem 1 to compute minimum fuel trajectories that transfer an observer spacecraft from their current pose to each of the N selected PROs previously generated. In computing an optimal transfer trajectory for an observer spacecraft to a target PRO, the observer solves multiple instances of Problem 1 with varying terminal conditions (2.17) sampled along the target PRO. The lowest cost solution from the multiple problem instances forms the optimal transfer trajectory, and its corresponding

estimate of the delta-V is recorded. This process is performed for each of the observers and can be easily done in a distributed manner.

A similar approach is utilized at the beginning of the mission for *PRO initialization*, which can be achieved by computing the initial thruster firing using the energy matching conditions derived in [16]. These initial conditions are shown to be robust to $J2$ perturbations.

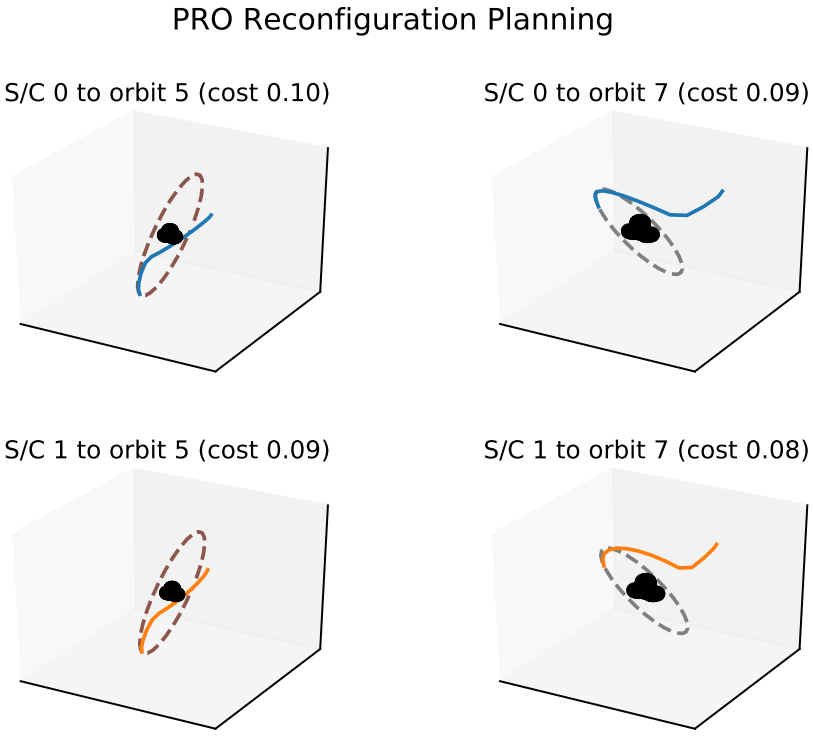


Figure 3.6: Example solutions for the PRO reconfiguration planning problem. Each of the two observers plan to move from its current state to one of the selected PRO candidates (see Fig. 3.5) and simultaneously estimates the delta-V cost for such maneuver (see captions on each plot).

PRO Assignment

For the *PRO assignment*, we use the Hungarian method [136] to assign candidate PROs to specific spacecraft, such that the estimated delta-V is minimized. For example, given 3 observers and 4 candidate PROs, the assignment is done such that $\sum_{i=1}^{3} \int_{t_0}^{t_f} \|u_i\|_1 dt$ is minimized. This PRO is initialized using the SCP formulation or by using the stable initial conditions through the reconfiguration planning. Such an approach can be easily distributed by utilizing, for example, distributed auctions [14].

In the example in Fig. 3.6 there are two possible assignments: observer 0 selects orbit 5 and observer 1 selects orbit 7, or observer 0 selects orbit 7 and observer 1 selects orbit 5. The Hungarian method computes the optimal assignment efficiently in polynomial time. Here, both costs are equal (0.18) and both assignments are optimal.

3.2.4 Mid-Rate: Attitude Planning, Information Update, and PRO Controller

Information Update

We update our current information gain H as well as the *information database* in case of two events: if a new sensor measurement was obtained or if we receive a message from another observer of a new measurement. Let $\mathbf{p}' \in SE(3)$ the new sensor pose. Then we can update the information database by i) finding the set of visible POIs from $\mathbf{p}'$ using the ray casting database, and ii) updating the database:

$$\mathcal{P}_{\mathrm{POI}}(\mathbf{s}) = \mathcal{P}_{\mathrm{POI}}(\mathbf{s}) \cup \{\mathbf{p}'\};\ \forall \mathbf{s} \in \text{POIs s.t. } \mathbf{s} \text{ visible from } \mathbf{p}'. \tag{3.10}$$

Afterwards, we can re-compute H by evaluating (3.4). If a new measurement was obtained on the spacecraft, $\mathbf{p}'$ is broadcasted to all other observers.

We note that H can also be re-evaluated incrementally more efficiently in the common case of $\mathbf{p}'$ only observing a small subset of the POIs, because only a few $H_{\mathrm{POI}}(\mathbf{s})$ values will change. Furthermore, if $\sigma(\cdot,\cdot)$ depends on the covariance of $\mathbf{p}'$, the covariance needs to be included in the information database and communicated to the neighboring observers.

Attitude Planner

While the orbit motion planner (Section 3.2.3) already considers attitudes to select the subset of suitable PROs, these attitudes might be impossible to track or are simply outdated based on the actual observations made so far. Therefore, we employ an efficient, greedy attitude planner algorithm, which takes the current value of H as well as angular velocity limits into account. A greedy approach is favored over more elaborate attitude planning, such as ones using dynamic programming [137], for computational efficiency.

Specifically, we try to point our camera towards an observable POI $\mathbf{s}^*$ with the highest variance:

$$\mathbf{s}^* = \arg\max\{H_{\mathrm{POI}}(\mathbf{s}) | \mathbf{s} \in \text{POIs and } \mathbf{s} \text{ visible from } \mathbf{p}\}, \tag{3.11}$$

where $\mathbf{p}$ is the current pose of the observer. As before, the visibility check can be done efficiently using the ray casting database. Based on $\mathbf{s}^*$, the desired attitude can be computed such that $\mathbf{s}^*$ is in the field of view.

PRO Controller

During long-term operation, the nonlinear effects like $J2$, atmospheric drag, and the uncertainty in the thruster firing leads to drift in the spacecraft position with respect to the target spacecraft. To mitigate this we can perform *regular orbit maintenance* recomputing the energy matched delta-V using (2.7) and applying it to the spacecraft (or) by using a feedback tracking controller. We assume that the target spacecraft is on a circular orbit and use HCW equations that do not consider nonlinear $(J2)$ and non-conservative effects (drag). This simplification allows us to develop an architecture that can be extend to include the nonlinear effects using initial conditions [16] that are robust to $(J2)$ effect. For reconfiguration control, we use a model predictive control (MPC) [15] formulation of SCP by computing the control at fixed time intervals.

3.2.5 High-Rate: Attitude Trajectory Optimization and Control

The attitude planner in Section 3.2.4 generates a sequence of POIs for each spacecraft to point to in a greedy manner. The sequence of POIs are interpolated to a smooth attitude trajectory that is used as reference for the attitude tracking controller by solving Problem 4 using SCP. The optimization includes control constraints and actuator dynamics to ensure feasibility during tracking. The *attitude controller* tracks an interpolated attitude profile. The tracking of the interpolated attitude profile uses an exponentially stable controller [129]. The gains of the controller are tuned for the rise time and potential noise observed in the mission scenario.

3.2.6 Properties and Remarks

Our approach scales linearly with the number of POIs, linearly with the number of PROs, and cubically with the number of spacecraft and empirically shows very promising results with respect to the solution quality.

3.2.7 Extension to Distributed Inspection

In the above formulation, the PRO candidate selection, the information database, information updates, and the PRO assignment play key role in distributed inspection.

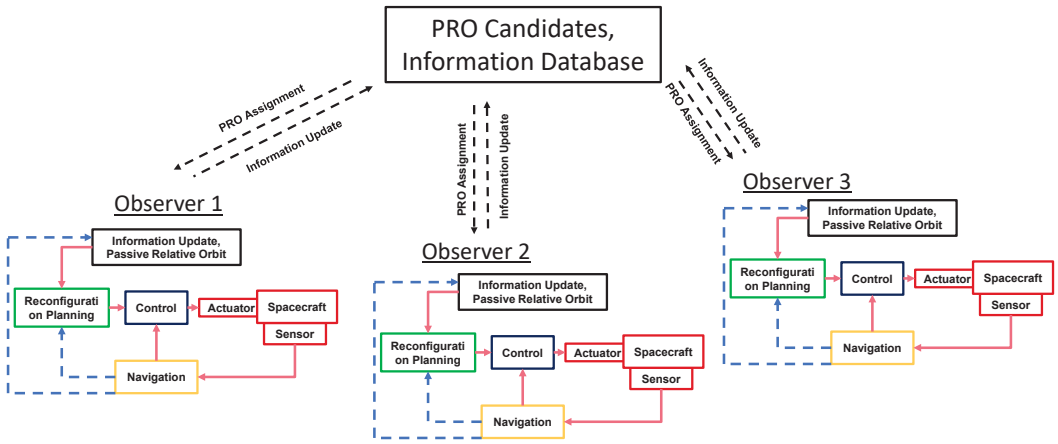


Figure 3.7: Centralized PRO candidate selection and information database implementation of the framework in Fig. 3.2 for inspection. The sampled PRO candidates and the information database are stored in the memory of the target spacecraft. The PRO candidates are then assigned to the observers based on coverage (information) and fuel optimality. The assigned candidates are communicated to the observers. The observers have a decentralized controller for initialization or reconfiguration to the assigned PROs. From the PRO, the observer spacecraft communicates the information update to the target spacecraft at a fixed time period for computing a new set of PRO candidates to inspect areas that have not been inspected earlier.

The distributed implementation of the architecture is shown in Fig. 3.8. The candidate selection can be done by one spacecraft and the resulting set of selected PROs can be communicated efficiently to the others using broadcast communication. It is also possible to compute this in a distributed fashion by using synchronization messages between the spacecraft. The database and the information update can be updated in a distributed fashion using broadcast communication, where each observer broadcasts its' current camera pose at a fixed frequency. The PRO assignment can be easily distributed by using, for example, distributed auctions [14]. Note that the PRO controller, the attitude planning, and control are decentralized, provided we have the full absolute pose estimates of the observers and the LVLH frame.

3.3 Inspection of a Target Spacecraft

In this section, we apply the proposed multi-level hierarchical GNC architecture and its algorithms towards inspection of the Cygnus spacecraft (target) in a low Earth orbit using multiple CubeSat observers. The choice of Cygnus as the target is motivated by a recent mission Seeker [4], where a single CubeSat was deployed from the target spacecraft and initialized in a relative orbit around Cygnus. The CubeSat was used to estimate the centroid of a bounding box around the target spacecraft

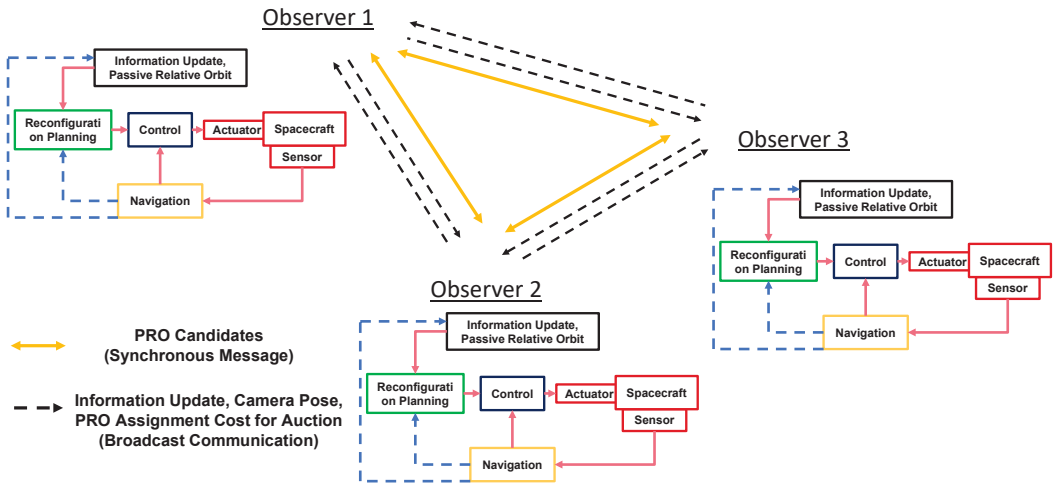


Figure 3.8: Distributed PRO candidate selection and information database implementation of the framework in Fig. 3.2 for inspection. The information database is stored on all the observers and evaluated using the updates communicated at fixed frequency from the neighbours. The PRO candidates are sampled in a distributed fashion using synchronous messages with the updated information cost. Optimal PRO assignment is done using distributed auction method discussed in [14].

using a neural network. With the potential applications of multi-spacecraft extension to this mission, we study the mission scenario where multiple CubeSats are deployed from the target spacecraft and initialized in safe stable relative orbits to inspect Cygnus. We begin by describing the problem setup, followed by implementation details, and finally present results for planning and control during different phases of the mission.

3.3.1 Inspection Scenario Setup

The observer spacecraft are of mass $m = 10\,\mathrm{kg}$ corresponding to a 6U-CubeSat ($20\mathrm{cm} \times 10\mathrm{cm} \times 30\mathrm{cm}$) and the thruster coefficient is $f_{\mathrm{thr}} = 200\,\mathrm{mN}$ for each element of the input scaling vector $\mathbf{u} \in [-1, 1]$ for the force per unit mass in (2.3). Each CubeSat is deployed using a Nanoracks [138] system with initial velocity between $0.5\,\mathrm{m/s}$ to $2\,\mathrm{m/s}$ from Cygnus. The target spacecraft is in an orbit with radius $r_0 = 6771\,\mathrm{km}$. The number of observers deployed from Cygnus is varied from 1 to 5 to study the design trade-off for the inspection mission. We assume that the observers are equipped with a reaction wheel configuration that is capable of performing attitude control for a maximum inertia $0.2\,\mathrm{kgm}^2$ in each direction and have a limit maximum torque of $\pm 2\,\mathrm{mNm}$ in each direction. For each case, the proposed architecture in Algorithm 1 is applied to initialize stable orbits and to

compute the progress of inspection in PROs over time. In the following test cases, we implement the proposed algorithm on a single computer (could be in the target spacecraft or in an observer spacecraft) and communicate the assigned PROs, the reconfiguration plan, and the information database to the observers.

3.3.2 Results and Discussion

In this section, we demonstrate the effectiveness of the proposed architecture in inspecting Cygnus using multiple spacecraft through orbits and reconfiguration planning based on the information cost function H. Additionally, we discuss in detail how the proposed architecture can be utilized as a design tool to determine the optimal number of observers needed and to validate the hardware selection for multi-spacecraft inspection missions.

Deployment and PRO Initialization

The algorithm is initialized when the CubeSats are deployed from Cygnus using a Nanorack. The initial deployment of the CubeSat plays a crucial role in the total fuel required for the mission – we design for this by using a nominal initial velocity condition that is $0.5\,\mathrm{m/s}$ in our algorithm to get an estimate of the fuel required to complete the mission. The initial position after drift is randomly sampled within $10\,\mathrm{m}$ to $20\,\mathrm{m}$ away from the Cygnus spacecraft. The average PRO initialization cost is computed to be 0.09 for zero initial velocity, 1.93 for the nominal deployment velocity of $0.5\,\mathrm{m/s}$, and 7.91 for the worst case deployment velocity. Due to the high cost and potential violation of HCW dynamics assumption for large initial velocity deployment ranging from $0.7\,\mathrm{m/s}$ to $2\,\mathrm{m/s}$, we recommend that the observers drift initially to get close to Cygnus over the next few orbits for minimum fuel initialization. This approach requires further analysis and is deferred to a future work.

Inspection with Two Observers

With a nominal initial deployment position ($10\,\mathrm{m}$ to $20\,\mathrm{m}$) and velocity ($0.5\,\mathrm{m/s}$) knowledge, we apply Algorithm 1 with two observers ($N = 2$) to inspect Cygnus. Figure 3.9 shows how the proposed architecture effectively coordinates the actions of the two observers to collectively inspect Cygnus over time. The two observers are initially reconfigured to a closest PRO set that minimizes H rapidly. The two observers then reconfigure twice during the entire mission to avoid information saturation. Note that we use a time-triggered scheduler to check if a reconfiguration

is required to inspect areas with higher variance. The observers might choose not to reconfigure if the current orbit is already optimal. This approach can be easily modified to have an information-based trigger for quicker maneuvering response to collect measurement on POIs of higher relative importance.

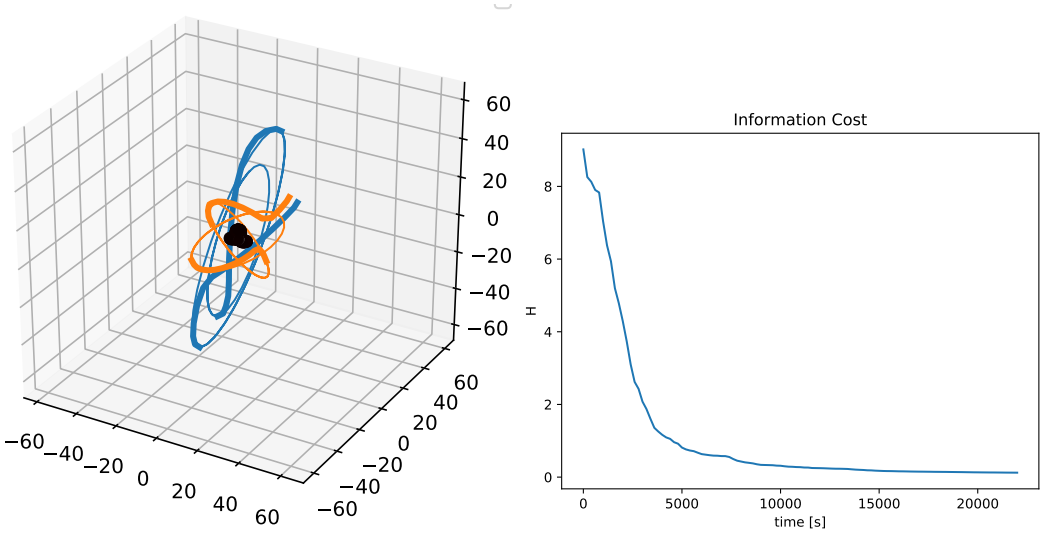


Figure 3.9: Inspection of Cygnus using two observers. Left: Trajectories of the two observers. The bold part shows the planned PRO reconfiguration. Right: The information cost converges to zero over time, because the entire target spacecraft is inspected.

Multi-Spacecraft Formation Design For Inspection

From a mission point of view, we want to design a formation that can inspect Cygus in minimum time and fuel efficient manner while ensuring safety. Using the PROs to generate candidate solutions already ensures minimal fuel requirement due to passive stability of orbits. We use fuel for intermediate reconfiguration between PROs, if required, which is guaranteed to be minimal from the SCP formulation in Problem 2 for reconfiguration. In order to design the optimal number of observers required for such a mission, we compare the information cost reduction by varying the formation size from 3 to 5 iteratively. The variation of information cost H and coverage over time for different formation sizes are shown in Fig. 3.10. From this figure, we can observe a significant time reduction in inspection when we use a formation of size 3 compared to when only using a single spacecraft. On the other hand, the coverage does not increase much with time when the formation size is increased from from 3 to 5. In Fig. 3.11, we show the minimum distance between the observers as a function of time for different formation sizes. For the formation

with 3 observers, the minimum distance varies from $5\,\mathrm{m}$ to $30\,\mathrm{m}$ as the mission evolves in time and for 5 observers, it varies from near $0\,\mathrm{m}$ to $15\,\mathrm{m}$. There is a safety hazard when using a formation with 5 observers with no significant improvement in performance compared to formation with size 3. From this analysis, we conclude that from a minimum time and safety point of view, the mission can be achieved optimally using a formation of 3 observers.

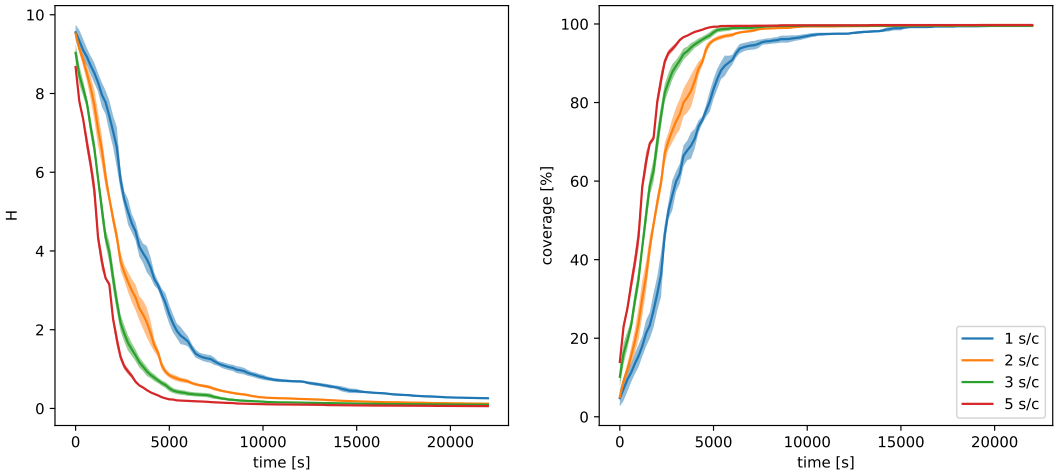


Figure 3.10: Information cost and POI coverage over time with varying number of observer spacecraft. The results are averaged over 5 trials with standard deviation shown as shaded area. Left: Using multiple observers allows to significantly reduce the time until a target information cost value is reached. For example, $H = 2$ is achieved in less than half the time when using 5 observers compared to when using one observer. Right: A lower information cost leads to a better coverage of POIs.

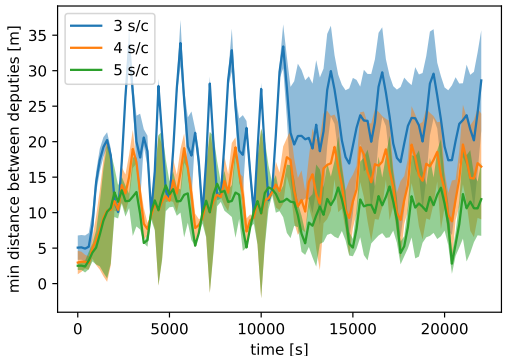


Figure 3.11: A plot showing comparison of minimum distance between observers when 3, 4, and 5 CubeSats are used for the inspection task as function of time. As expected, 3 observer configuration has larger separation compared to 5 observer configuration.

At a high-rate we run an attitude trajectory optimization and controller to smoothly

track the coarse pointing vectors computed using the ray casting database. Figure 3.12 shows a comparison of slerp interpolated trajectory, smoothed trajectory taking into account torque and momentum limits of the reaction wheel, and the controlled trajectory. We can observe that the smoothed trajectory does not saturate the reaction wheels, whereas the slerp trajectory leads to saturation by crossing the maximum torque limit of $2\,\mathrm{mNm}$ and jerk in the attitude motion of the observer. Note that this step is useful in designing the pointing requirement and reaction wheel desaturation strategy for each observer. For example, the observers around Cygnus initialized in a stable PRO have their period matched to that of Cygnus's orbital period. Due to this, the observers spend some time of the orbit in the dark and might not be able to inspect the Cygnus. The actuator limits can be used to design a partial orbit inspection strategy, where the observers only inspect when the pointing strategy is feasible for $30\,\%$ of the orbit. If the pointing plan is not feasible, then the system can desaturate the wheels when in the dark and inspect only when in the sunlight.

3.4 Experiments on the Robotic Spacecraft Simulator Testbed

We validate Algorithm 1 by optimally inspecting the four surfaces of a target spacecraft, as shown in Fig. 3.13, on the M-STAR (Multi-Spacecraft Testbed for Autonomy Research) platform described in Chapter 7. We assume that observers-2 and 3 are stationary and observer-1 is active. Observers-2 and 3 have inspected two surfaces (red normal in Fig. 3.13). Observer-1 communicates with the neighboring observers to sort and select the unobserved surfaces of the target spacecraft via information database. The information database includes a list of surfaces not inspected.

In Fig. 3.14, we show the implementation of the hierarchical Algorithm 1 for M-STAR. We use a higher-level planner to select an optimal and informative terminal state for observer-1. We sample the feasible state-space of the observer-1 around the target spacecraft for terminal states. We choose the terminal state that the observer has good visibility of the surface to be inspected from that state and requires minimum fuel to achieve that state. We use the terminal condition as a constraint in Problem 1 for planning a safe trajectory to approach and point an observer to inspect the surfaces that are not inspected by the neighbours. We execute the motion plan by using an exponentially stable nonlinear feedback controller [30].

We generate 800 terminal state space samples within the concentric circles defined

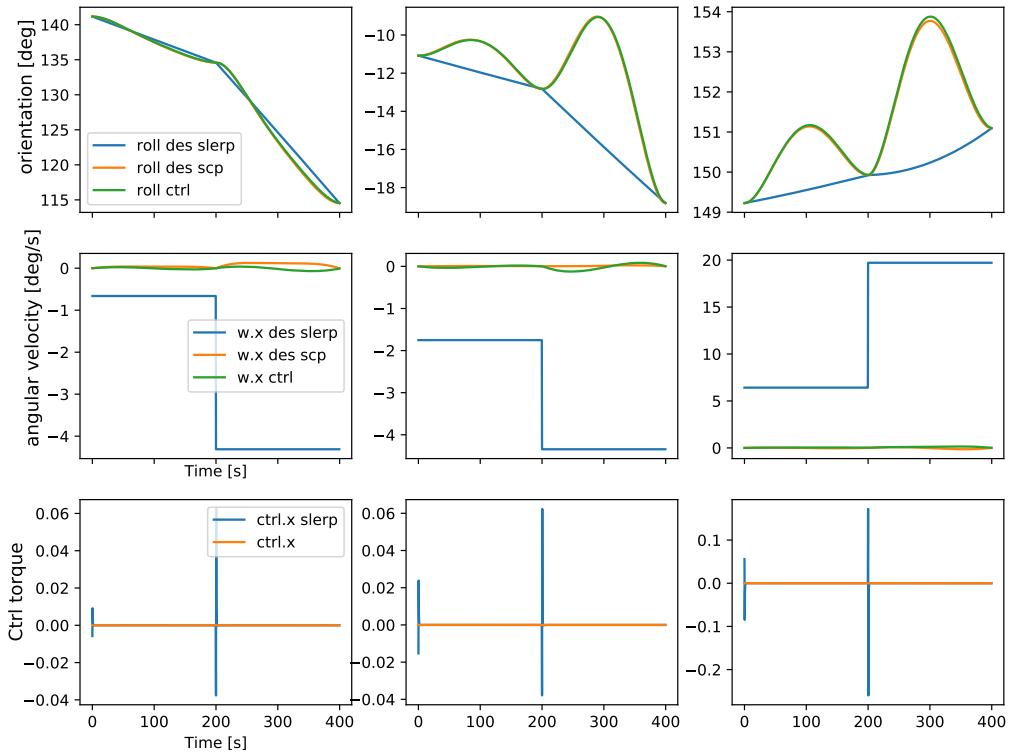


Figure 3.12: Attitude trajectory generated using the slerp interpolation and sequential convex programming. The control required to track the slerp trajectory and the smoothed trajectory are compared. The smoothed trajectory satisfies the torque limits of $\pm 2\,\mathrm{mNm}$, while tracking the slerp trajectory easily saturates the reaction wheels.

by the safety distance $\mathrm{d}_{\min} = 0.7\,\mathrm{m}$ and the visibility distance $\mathrm{d}_{\max} = 1.2\,\mathrm{m}$, as shown in Fig. 3.15a. We observe that 14 samples (see Fig. 3.15b) have good visibility of the surfaces to be inspected. In Figure 3.15c, we show the optimal trajectories computed for each of the terminal samples to estimate the cost-to-go. We compute the information cost using large prior w and update the information using (3.4) and (3.5). The cost-to-go and the information cost are scaled by using the maximum value of the 14 samples. We select the optimal and informative terminal state by maximizing the ratio of the scaled information cost and the cost-to-go, as shown in Fig. 3.15e. We show the optimal motion plan in Fig. 3.15d. The optimal trajectory is executed using an exponentially stable feedback controller.

Observer-1 updates the information database and replans to achieve coverage. In Fig. 3.16, we show the optimal trajectories generated using the hierarchical algorithm for inspecting all four surfaces of the target spacecraft. In this approach, we select the next best terminal condition by using a fixed number of state-space

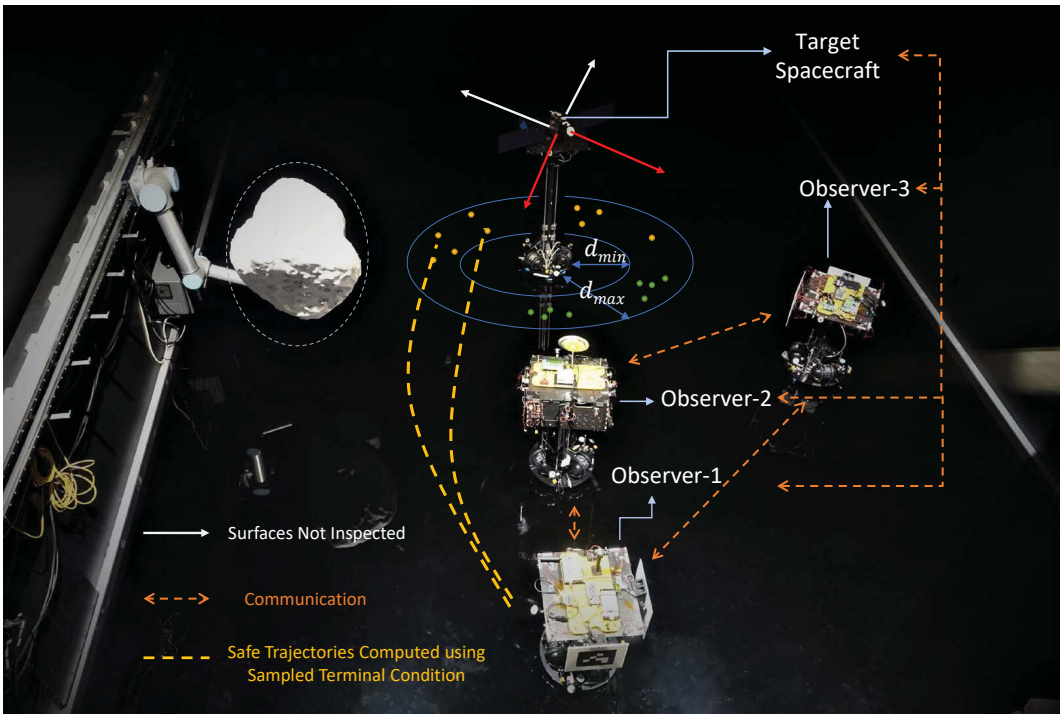


Figure 3.13: A picture of the information-based planning (Fig. 3.2) experiment on the multi-spacecraft testbed [30] for autonomy research.

samples at each inspection iteration. We guarantee coverage by replanning. To guarantee asymptotic optimality, we can use efficient sampling techniques.

3.5 Chapter Summary

In this chapter, we present a new guidance and control architecture that consists of multiple timescale modules for cooperatively observing and inspecting a target spacecraft in Earth orbit using multiple observer spacecraft. The proposed methodology hierarchically combines optimal trajectory design for orbit initialization and reconfiguration, attitude/pointing planning and control, and an optimal control formulation that incorporates an information cost for inspecting the target spacecraft. The information cost is used to cover the entire shape or structure of the target spacecraft and balances the trade-off between gathered data quality and fuel/energy cost. The orbit planning involves designing a safe and optimal trajectory to optimally insert each observer spacecraft into passively relative orbits around the target spacecraft for conducting the inspection task. To this end, we formulate the problem as a multi-spacecraft optimal control problem and solve it using sequential convex programming. The passively stable relative orbits are chosen in an offline phase based on the energy matching condition for stability and using an information gain

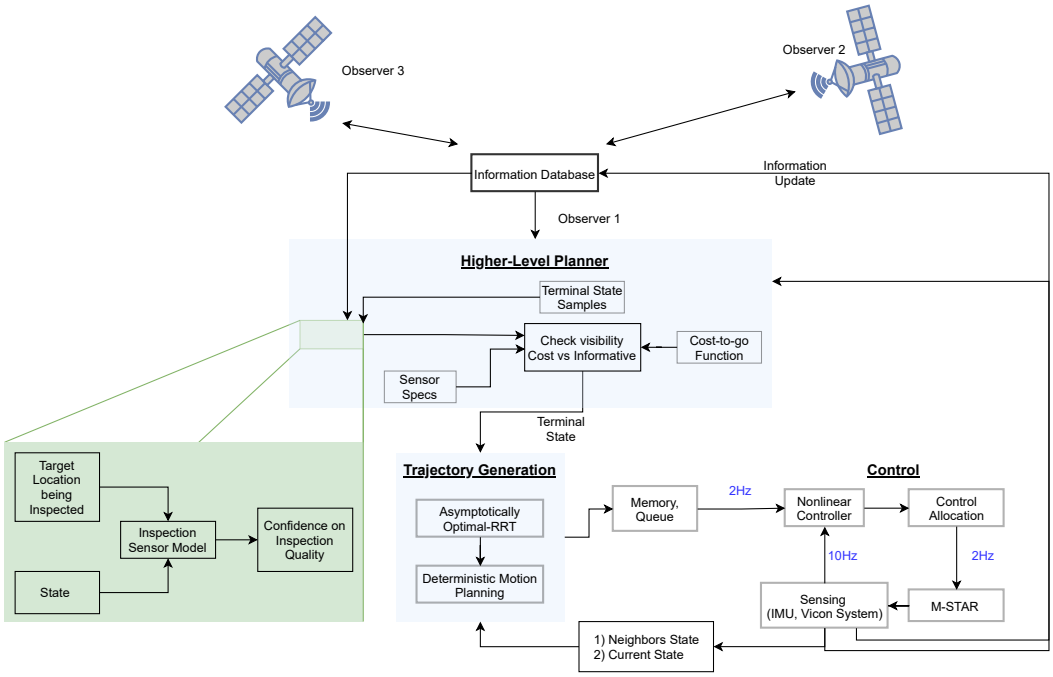
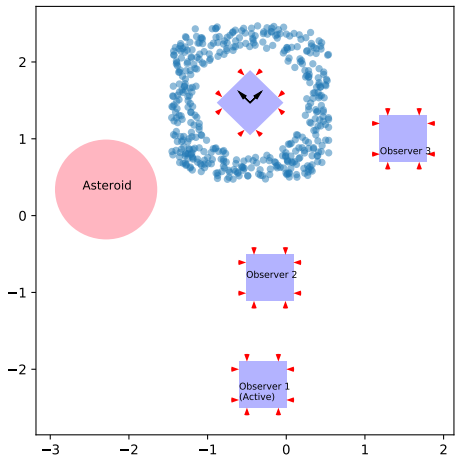
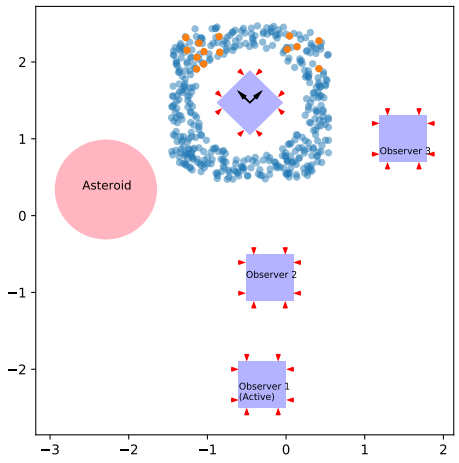


Figure 3.14: Block diagram showing the hierarchical planner as applied to the three degree-of-freedom spacecraft dynamics simulator.

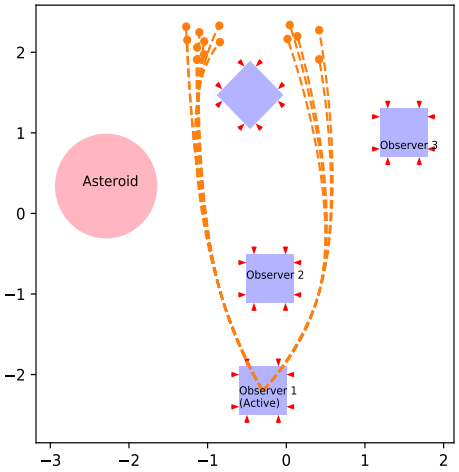
metric to inspect the maximum surface area of the target spacecraft. During the inspection, we confirm if an area of the target spacecraft has been inspected or not, using our novel raycasting database to efficiently check the visibility of POIs. Once the passive relative orbits are initialized, we compute a pointing control plan at fast rates to optimally reorient the spacecraft for inspection. The planned attitude trajectory is executed using a nonlinear feedback controller. The effectiveness of the proposed architecture is validated via numerical simulation of inspecting the target spacecraft in Earth orbit using multiple observer spacecraft. We demonstrate the proposed planning approach on a three-degree-of-freedom robotic spacecraft testbed for inspecting a target spacecraft. The methods presented in this chapter can be effectively used as a mission planning tool to compute and track multi-spacecraft observers on energy-efficient passive relative orbits for inspecting a spacecraft.



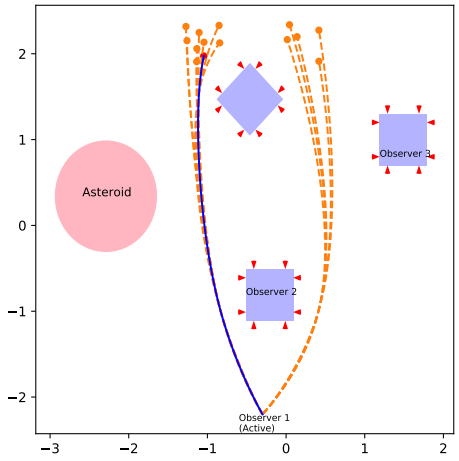
(a) State space samples around the target spacecraft for the observer-1.



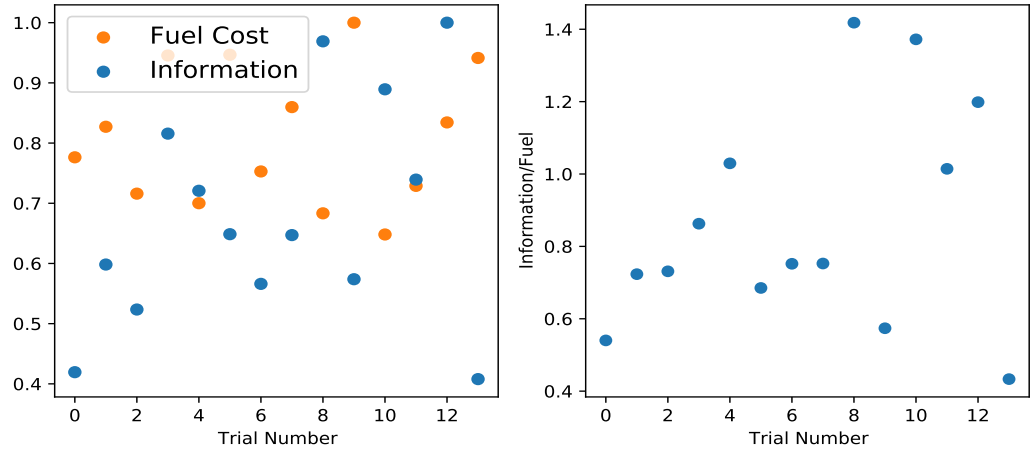
(b) Terminal state samples (orange) with good visibility.



(c) Observer 1 computes the cost-to-go using SCP, and the potential information using the information cost.

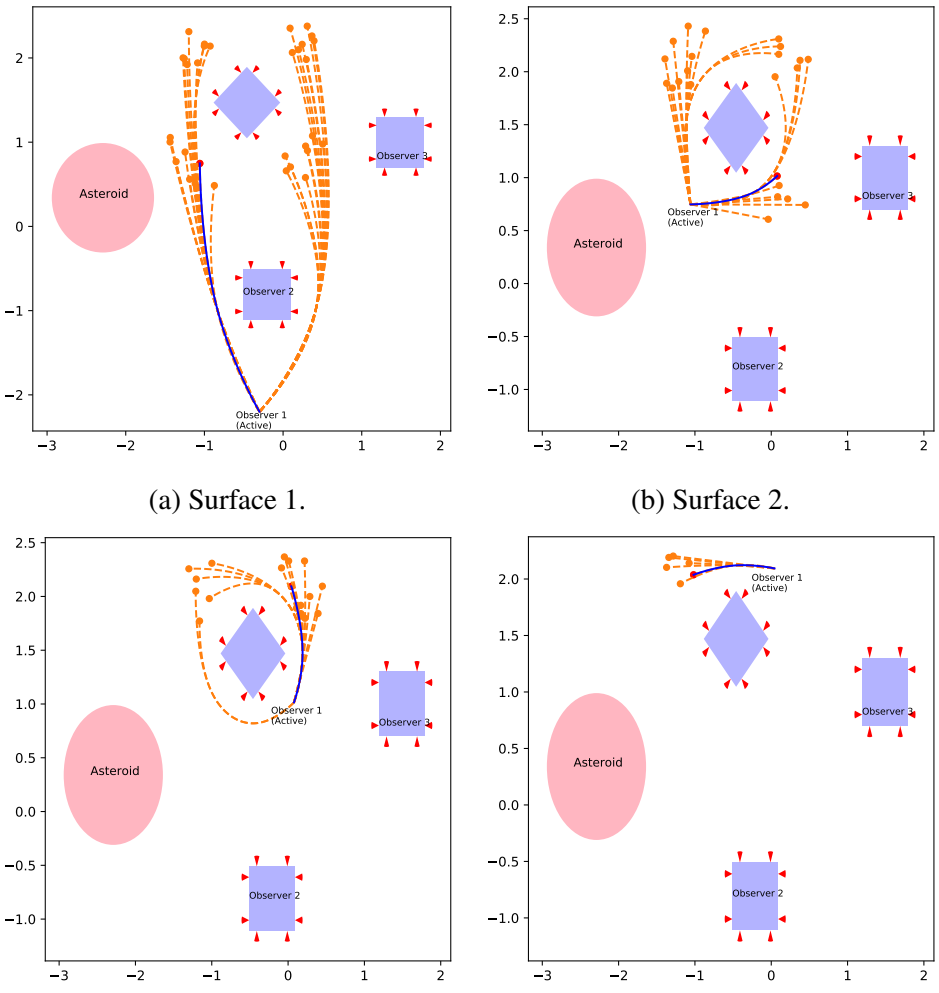


(d) Observer 1 selects an optimal informative trajectory from the pool of samples.



(e) Scaled information and fuel cost computed by the observer 1.

Figure 3.15: Output of Algorithm 1. Observer 1 is actively inspecting the target spacecraft using the hierarchical information-based planning algorithm.



(a) Surface 1.

(b) Surface 2.

(c) Surface 3.

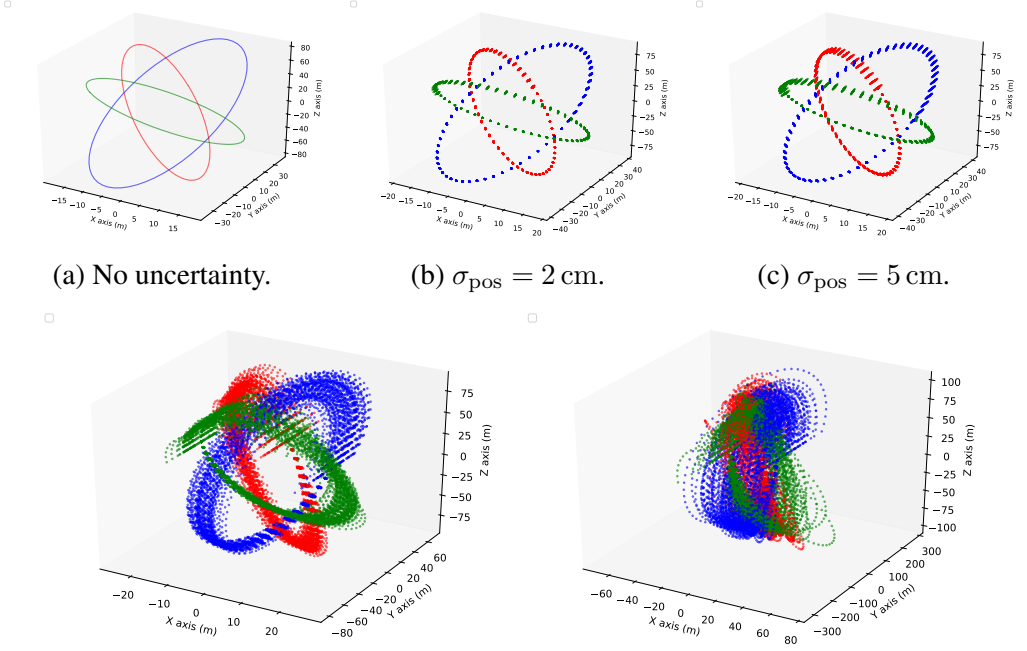
(d) Surface 4.

Figure 3.16: Information-based planning to inspect all the four surfaces of the target spacecraft using a single agent.

Chapter 4

TRAJECTORY OPTIMIZATION FOR CHANCE-CONSTRAINED NONLINEAR STOCHASTIC SYSTEMS

The PRO initialization and reconfiguration methods discussed in Chapter 2 and in [16] assume perfect knowledge of the state information and that there is no uncertainty in actuation. In Fig. 4.1, we show the effect of uncertainty in position information, σ_{pos}, and the actuation (ΔV), σ_{vel}, on the propagation of three deputy spacecraft. We propagate the PRO's by adding uncertainty to the position of the



(a) No uncertainty. (b) $\sigma_{\text{pos}} = 2\,\text{cm}$. (c) $\sigma_{\text{pos}} = 5\,\text{cm}$.

(d) $\sigma_{\text{vel}} = 2\,\text{mm/s}$. (e) $\sigma_{\text{vel}} = 5\,\text{mm/s}$.

Figure 4.1: Effect of uncertainty in the state estimate and actuation on the PRO maintenance, shown in LVLH frame. We show the PRO trajectories generated over 100 Monte-Carlo trials, by sampling the uncertainty in state and actuation.

deputy with respect to the LVLH frame, $[x_0 + \sigma_{\text{pos}}\xi_1, y_0 + \sigma_{\text{pos}}\xi_2, z_0 + \sigma_{\text{pos}}\xi_2]$, and by incorporating uncertainty in actuation as $[\dot{x}_0 + \sigma_{\text{vel}}\xi_4, \dot{y}_0 + \sigma_{\text{vel}}\xi_5, \dot{z}_0 + \sigma_{\text{vel}}\xi_6]$. We assume to have the knowledge of the initial position estimate, $[x_0, y_0, z_0]$, w.r.t the LVLH and compute the initial actuation $[\dot{x}_0, \dot{y}_0, \dot{z}_0]$ using the energy matching conditions derived in [16]. $\xi_i \sim \mathcal{N}(0,1)\ \forall i \in \{1,\ldots,6\}$. We observe in Fig. 4.1, that uncertainty in state estimation and actuation leads to deviation from the designed or-

bit. During proximity operations, such a dispersion could lead to a mission failure. In order to ensure safety, the trajectory optimization algorithms, discussed in Chapter 2, should incorporate uncertainty in dynamics and the neighboring deputy state. To this end, we formulate a stochastic optimal control problem under chance constraints and propose a new algorithm, shown in Fig. 1.2, to compute safe motion plans under uncertainty in the dynamics and obstacle state.

4.1 Problem Formulation

In this section, we present the stochastic optimal control problem formulation, preliminaries on the relaxations used for chance constraints, and the generalized polynomial chaos approach that forms a basis for constructing a surrogate deterministic optimal control problem.

4.1.1 Stochastic Nonlinear Optimal Control Problem

We consider the finite-horizon stochastic nonlinear optimal control (SNOC) problem with joint chance constraints in continuous time and continuous space. The SNOC problem minimizes an expectation cost function, that is a sum of a quadratic function in the random state variable $\mathbf{x}(t)$ and a convex norm of the control policy $\bar{\mathbf{u}}(t)$. The evolution of the stochastic process $\mathbf{x}(t)$ for all sampled paths is defined by a stochastic differential equation. The joint chance constraints guarantee constraint feasibility with a probability of $1-\epsilon$, where $\epsilon > 0$ and is chosen to be a small value (example: $\epsilon \in [0.001, 0.05]$) for better constraint satisfaction. The following optimal control problem is considered with the state distribution and control as the decision variables.

Problem 6. Chance-Constrained Stochastic Nonlinear Optimal Control.

$$J^*_{\text{SNOC}} = \min_{\mathbf{x}(t),\bar{\mathbf{u}}(t)} \mathbb{E}\left[\int_{t_0}^{t_f} J(\mathbf{x}(t),\bar{\mathbf{u}}(t))dt + J_f(\mathbf{x}(t_f))\right] \tag{4.1}$$

$$\text{s.t.} \quad d\mathbf{x} = f(\mathbf{x}(t),\bar{\mathbf{u}}(t))dt + g(\mathbf{x}(t),\bar{\mathbf{u}}(t))dw(t) \tag{4.2}$$

$$\Pr(\mathbf{x}(t) \in \mathcal{X}_{\mathcal{F}}) \geq 1-\epsilon \; \forall t \in [t_0, t_f] \tag{4.3}$$

$$\bar{\mathbf{u}}(t) \in \mathcal{U} \quad \forall t \in [t_0, t_f] \tag{4.4}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0 \quad \mathbf{x}_{t_f} \in \mathcal{X}_f, \tag{4.5}$$

where the cost functional J and the terminal cost J_f are defined as follows:

$$\begin{aligned} J(\mathbf{x}(t),\bar{\mathbf{u}}(t)) &= \mathbf{x}(t)^\top Q\mathbf{x}(t) + \|\bar{\mathbf{u}}\|_p, \text{ where } p \in \{1,2,\infty\}, \\ J_f(\mathbf{x}(t_f)) &= \mathbf{x}(t_f)^\top Q_f \mathbf{x}(t_f), \end{aligned} \tag{4.6}$$

where Q and Q_f are positive definite matrices. The $p-norm$ of a vector $\bar{\mathbf{u}}$ is defined as $\|\bar{\mathbf{u}}\|_p = (\sum_1^{d_u} |\bar{\mathbf{u}}_i|^p)^{\frac{1}{p}}$. The terminal set $\mathcal{X}_f$ is the set of allowed realization of the state $\mathbf{x}$ after propagation. The terminal constraints is applied as a probabilistic soft constraint to ensure feasibility of Problem 6. In the following, we define each of the aforementioned elements of Problem 6 and discuss convex approximations of linear and quadratic chance constraints.

Stochastic Differential Equation (SDE) [76]

The dynamics of the system is modeled as a controlled diffusion process with Itô assumptions. The random variable $\mathbf{x}(t)$ is defined on a probability space $(\Omega, \mathbb{F}, \Pr)$ where Ω is the sample space, $\mathbb{F}$ forms a σ-field with measure $\Pr$.

$$\begin{aligned} &d\mathbf{x}(t) = f(\mathbf{x}(t), \bar{\mathbf{u}}(t))dt + g(\mathbf{x}(t), \bar{\mathbf{u}}(t))dw(t), \\ &\Pr(|\mathbf{x}(t_0) - \mathbf{x}_0| = 0) = 1, \quad \forall t_0 \leq t \leq t_f < \infty, \end{aligned} \tag{4.7}$$

where: $f(.,.) : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}^{d_\mathbf{x}}$, $g(.,.) : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}^{d_\mathbf{x} \times d_\xi}$, and $w(t)$ is a d_ξ-dimensional Wiener process and the initial random variable $\mathbf{x}_0$ is independent of $w(t) - w(t_0)$ for $t \geq t_0$, and $dw(t) \sim \mathcal{N}(0, dt\mathbb{I})$. The sets $\mathcal{X} \subseteq \mathbb{R}^{d_\mathbf{x}}$ and $\mathcal{U} \subseteq \mathbb{R}^{d_u}$ are compact sets. We make the following assumptions to ensure the existence and uniqueness of a solution to the SDE.

Assumption 1. The functions $f(\mathbf{x}(t), \bar{\mathbf{u}}(t))$ and $g(\mathbf{x}(t), \bar{\mathbf{u}}(t))$ are defined and measurable on $\mathcal{X} \times \mathcal{U}$.

Assumption 2. Equation (4.7) has a unique solution $\mathbf{x}(t)$, that is continuous with probability 1, and $\exists$ a $K \in \mathbb{R}^{++}$ such that the following conditions are satisfied:
a) Lipschitz condition [76]: $\forall t \in [t_0, t_f]$, $s_j \in \mathcal{X} \times \mathcal{U},\ j = 1, 2$,

$$\|f(s_1) - f(s_2)\| + \|g(s_1) - g(s_2)\|_{\mathrm{F}} \leq K\|s_1 - s_2\|, \tag{4.8}$$

b) Restriction on growth: $\forall t \in [t_0, t_f]$, $s_1 \in \mathcal{X} \times \mathcal{U}$

$$\|f(s_1)\|^2 + \|g(s_1)\|_F^2 \leq K^2(1 + \|s_1\|^2). \tag{4.9}$$

We use the following definition to study the controllability of deterministic approximations of SDE (4.7).

Definition 1. The SDE (4.7) is ϵ_c-controllable [49]. For any initial state $\mathbf{x}_0 \in \mathcal{X}$, we can compute a sequence of control $\bar{\mathbf{u}}(t)\ \forall t \in [t_0, t_f]$ such that $\Pr(\|\mathbf{x} - \mathbf{x}(t_f)\|^2 \geq \delta \mid \mathbf{x}(t_0) = \mathbf{x}_0) \leq \epsilon_c$, where $\mathbf{x}(t_f)$ is the terminal state, $\delta > 0$ and $\epsilon_c > 0$ are small, and t_f is finite.

Control Policy. We assume that the control policy $\bar{\mathbf{u}}(t) \in \mathcal{U} \subseteq \mathbb{R}^{d_u}$ is deterministic and the set $\mathcal{U}$ is a convex set. The deterministic control policy is motivated by a hardware implementation strategy, where a state dependent Markov control policy defined on the compact set $\mathcal{X}$ is sampled for a value with highest probability (or) for the mean.

Note: We use the gPC method to project the SDE to an Ordinary Differential Equation (ODE) in a higher dimensional space for propagating the dynamics.

Chance Constraints [46]

In order to accommodate the unbounded uncertainty model in the dynamics, the feasible region $\mathcal{X}_{\mathcal{F}}$ defined as,

$$\mathcal{X}_{\mathcal{F}} = \{\mathbf{x}(t) \in \mathcal{X} : h_i(\mathbf{x}(t)) \leq 0 \; \forall \; i \; \in \; \{1, \ldots, m\}\}, \tag{4.10}$$

is relaxed to a chance constraint (CC) using the risk measure ϵ,

$$\mathcal{X}_{\mathrm{CC}} = \{\mathbf{x}(t) \in \mathcal{X} : \Pr(\mathbf{x}(t) \in \mathcal{X}_{\mathcal{F}}) \geq 1 - \epsilon\}, \tag{4.11}$$

with a guaranteed constraint satisfaction probability of $1 - \epsilon$. The constraint set $\mathcal{X}_{\mathcal{F}}$ is assumed to be the polytope $\mathcal{X}_{\mathcal{F}} = \{\mathbf{x} \in \mathcal{X} : \wedge_{i=1}^{m} a_i^{\top}\mathbf{x} + b_i \leq 0\}$ with m flat sides, or a quadratic constraint set $\mathcal{X}_{\mathcal{F}} = \{\mathbf{x} \in \mathcal{X} : \mathbf{x}^{\top} A \mathbf{x} \leq c\}$ for any realization $\mathbf{x}$ of the state. The joint chance constraint formulation of the polytopic constraint is of the form, $\Pr(\wedge_{i=1}^{m} a_i^{\top}\mathbf{x} + b_i \leq 0) \geq 1 - \epsilon$.

A convex relaxation of the individual chance constraint for an arbitrary distribution of the state vector $\mathbf{x}(t)$ due to the nonlinearity in the system is intractable, so an extension of the problem called Distributionally-Robust Chance Constraints (DRCC) given as follows,

$$\mathcal{X}_{\mathrm{DRCC}} = \{\mathbf{x}(t) \in \mathcal{X} : \inf_{\mathbf{x}(t) \sim (\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}})} \Pr(\mathbf{x}(t) \in \mathcal{X}_{\mathcal{F}}) \geq 1 - \epsilon\}, \tag{4.12}$$

where the chance constraint is satisfied for all distributions with known mean and variance of the decision variable is used. The set defined by the DRCC in (4.12) is a conservative approximation [47] of the chance constraint i.e., $\mathcal{X}_{\mathrm{DRCC}} \subseteq \mathcal{X}_{\mathrm{CC}}$.

a) *Distributionally-Robust Linear Chance Constraint (DRLCC)* [46]: Consider a single Linear Chance Constraint (LCC) with $a \in \mathbb{R}^{d_{\mathbf{x}}}$ and $b \in \mathbb{R}$:

$$\mathcal{X}_{\mathrm{LCC}} = \{\mathbf{x}(t) \in \mathcal{X} : \Pr(a^{\top}\mathbf{x}(t) + b \leq 0) \geq 1 - \epsilon\}. \tag{4.13}$$

Assuming that the mean $\mu_{\mathbf{x}}$ and the covariance $\Sigma_{\mathbf{x}}$ of $\mathbf{x}$ are known, a distributionally-robust constraint version of (4.13) is given as follows:

$$\mathcal{X}_{\mathrm{DRLCC}} = \{\mathbf{x}(t) \in \mathcal{X} : \inf_{\mathbf{x}(t)\sim(\mu_{\mathbf{x}},\Sigma_{\mathbf{x}})} \Pr(a^\top \mathbf{x}(t) + b \leq 0) \geq 1 - \epsilon\}. \tag{4.14}$$

Equivalently, (4.14) can be rewritten in the following deterministic form, which will be used to derive a second-order cone constraint for the DNOC in Section 4.2.5.

$$\mathcal{X}_{\mathrm{DRLCC}} = \{\mathbf{x}(t) \in \mathcal{X} : a^\top \mu_{\mathbf{x}}(t) + b + \sqrt{\frac{1-\epsilon}{\epsilon}}\sqrt{a^\top \Sigma_{\mathbf{x}} a} \leq 0\} \tag{4.15}$$

Lemma 3. *The set $\mathcal{X}_{\mathrm{DRLCC}}$ in (4.15) is a subset of $\mathcal{X}_{\mathrm{LCC}}$ defined in (4.13).*

Proof. See Theorem 3.1 in [46]. □

Risk Measure. The risk measure ϵ in (4.13) is assumed to be in the range $[0.001, 0.5]$. For small values of ϵ (< 0.001), the value $\sqrt{\frac{1-\epsilon}{\epsilon}}$ increases exponentially. This decreases the feasible space defined by the set $\mathcal{X}_{\mathrm{LCC}}$ drastically leading to numerical issues in the gPC-SCP method presented in this thesis. For handling the risk of the order of $1e-7$, as discussed in [139], the uncertainty in the system needs to be modeled accurately such that $\Sigma_{\mathbf{x}}$ is small (or) newer deterministic surrogate method needs to be developed to overcome the numerical instability.

If the dynamics are linear, the constraint in (4.15) is replaced with a tighter equivalent deterministic constraint given by the following inequality:

$$a^\top \mu_{\mathbf{x}}(t) + b + \mathrm{erf}(1-2\epsilon)\sqrt{a^\top \Sigma_{\mathbf{x}} a} \leq 0, \tag{4.16}$$

where the function erf(.) is defined as $\mathrm{erf}(\delta) = \frac{2}{\sqrt{\pi}} \int_0^\delta e^{-t^2} dt$ and $\forall \epsilon \in (0, 0.5)$. This constraint set is transformed to a second-order cone constraint in the gPC variables.

b) *Conservative Quadratic Chance Constraint (CQCC):* Lemma 4 presents a new conservative deterministic relaxation for the quadratic chance constraint that is used to bound the deviation of the random vector $\mathbf{x}(t)$ from the mean $\mu_{\mathbf{x}}(t)$.

Lemma 4. *The constraint set*

$$\mathcal{X}_{\mathrm{CQCC}} = \{\mathbf{x}(t) \in \mathcal{X} : \frac{1}{c} tr(Q\Sigma_{\mathbf{x}}) \leq \epsilon\} \tag{4.17}$$

is a conservative approximation of the original Quadratic Chance Constraint (QCC)

$$\mathcal{X}_{\mathrm{QCC}} = \{\mathbf{x} \in \mathcal{X} : \Pr((\mathbf{x} - \mu_{\mathbf{x}})^\top Q(\mathbf{x} - \mu_{\mathbf{x}}) \geq c) \leq \epsilon\} \tag{4.18}$$

i.e., $\mathcal{X}_{\mathrm{CQCC}} \subseteq \mathcal{X}_{\mathrm{QCC}}$, where $Q \in \mathbb{R}^{d_{\mathbf{x}} \times d_{\mathbf{x}}}$ is a positive definite matrix and $c \in \mathbb{R}^{++}$ and $\Sigma_{\mathbf{x}}$ is the covariance of the random variable $\mathbf{x}$.

Proof. We will prove that any random vector $\mathbf{x} \in \mathcal{X}$ that is in the set $\mathcal{X}_{\text{CQCC}}$ is also in the set $\mathcal{X}_{\text{QCC}}$ implying $\mathcal{X}_{\text{CQCC}} \subseteq \mathcal{X}_{\text{QCC}}$. The proof follows from the approach taken to prove the multivariate Chebyschev's inequality [140]. Let $F(\mathbf{x})$ be the Cumulative Distribution Function (CDF) of the random variable $\mathbf{x}$ and $v = \mathbf{x} - \mu_{\mathbf{x}}$.

$$\mathcal{G} = \{v \in \mathcal{X} : v^\top Q v \geq c\} \implies \frac{1}{c} v^\top Q v \geq 1 \ \ \forall v \in \mathcal{G}.$$

Using the definition of probability in terms of the CDF,

$$\begin{aligned}\Pr((\mathbf{x} - \mu_{\mathbf{x}}) \in \mathcal{G}) &\leq \frac{1}{c} \int_{v \in \mathcal{G}} v^\top Q v dF(v) \\ &\leq \frac{1}{c} \int_{v \in \mathcal{R}^n} v^\top Q v dF(v).\end{aligned}$$

Let q_{ij} be an element of matrix Q in the i^{th} row and j^{th} column, and v_i be the i^{th} element in the vector v. Using the expansion $v^\top Q v = \sum_{i=1}^{d_{\mathbf{x}}} \sum_{j=1}^{d_{\mathbf{x}}} q_{ij} v_i v_j$ in the inequality above, the integral is simplified.

$$\begin{aligned}\int_{v \in \mathbb{R}^n} v^\top Q v dF(v) &= \int_{v \in \mathbb{R}^v} \sum_{i=1}^{d_{\mathbf{x}}} \sum_{j=1}^{d_{\mathbf{x}}} q_{ij} v_i v_j dF(v) \\ &= \sum_{i=1}^{d_{\mathbf{x}}} \sum_{j=1}^{d_{\mathbf{x}}} q_{ij} \int_{v \in \mathbb{R}^{d_{\mathbf{x}}}} v_i v_j dF(v) \\ &= \text{tr}(Q \Sigma_{\mathbf{x}}).\end{aligned} \tag{4.19}$$

The quadratic chance constraint holds if (4.17) is satisfied, as $\Pr((\mathbf{x} - \mu_{\mathbf{x}}) \in \mathcal{G}) \leq \frac{1}{c}\text{tr}(Q\Sigma_{\mathbf{x}})$. Therefore, (4.17) is a conservative deterministic approximation of the quadratic chance constraint $\Pr((\mathbf{x}-\mu_{\mathbf{x}})^\top Q(\mathbf{x}-\mu_{\mathbf{x}}) \geq c) \leq \epsilon$ i.e., $\mathcal{X}_{\text{CQCC}} \subseteq \mathcal{X}_{\text{QCC}}$. Note that if ϵ is a design variable, the approximation can be made tight by solving an inner semi-definite program following the approach in [78]. □

Corollary 0.1. The constraint set $\frac{1}{c}\text{tr}(Q\Sigma_{\mathbf{x}}) + \frac{1}{c}(\mu_{\mathbf{x}}^\top Q \mu_{\mathbf{x}}) \leq \epsilon$ is a conservative approximation of the quadratic chance constraint $\Pr(\mathbf{x}^\top Q \mathbf{x} \geq c) \leq \epsilon\}$, where $Q \in \mathbb{R}^{d_{\mathbf{x}} \times d_{\mathbf{x}}}$ is a positive definite matrix and $c \in \mathbb{R}^{++}$ and $\Sigma_{\mathbf{x}}$ is the co-variance of the random variable $\mathbf{x}$.

Proof. The proof follows from Lemma 4. □

Note. Similar to linear chance constraints, having a small ϵ ($< 1e-3$) in the quadratic chance constraints might lead to numerical issues due to decreased feasible space. Further research needs to be conducted in this direction to provide such safety guarantees.

c) *Joint Chance Constraints (JCC)* [47]: The distributionally-robust joint chance constraints (DRJCC) for a polytope set is defined as, $\inf_{\mathbf{x}(t)\sim(\mu_{\mathbf{x}},\Sigma_{\mathbf{x}})} \Pr(\wedge_{i=1}^{m} a_i^\top \mathbf{x} + b_i \leq 0) \geq 1-\epsilon$. The joint constraints are split into multiple single chance constraints using Bonferroni's inequality [47] method as follows:

$$\begin{aligned} &\inf_{\mathbf{x}(t)\sim(\mu_{\mathbf{x}},\Sigma_{\mathbf{x}})} \Pr(\wedge_{i=1}^{m} a_i^\top \mathbf{x} + b_i \leq 0) \geq 1-\epsilon \\ \iff &\sup_{\mathbf{x}(t)\sim(\mu_{\mathbf{x}},\Sigma_{\mathbf{x}})} \Pr(\vee_{i=1}^{m} a_i^\top \mathbf{x} + b_i \geq 0) \leq \epsilon \\ \subseteq &\sum_{i=1}^{m} \sup_{\mathbf{x}(t)\sim(\mu_{\mathbf{x}},\Sigma_{\mathbf{x}})} \Pr(a_i^\top \mathbf{x} + b_i \geq 0) \leq \epsilon. \end{aligned} \tag{4.20}$$

If the probability distribution of $\mathbf{x}$ is Gaussian, then the JCC are split using Boole's inequality [21]. The total risk measure ϵ is allocated between each of the chance constraints in the summation such the $\sum_{i=1}^{m} \epsilon_i = \epsilon$ leading to m individual DRCC of the following form.

$$\inf_{\mathbf{x}(t)\sim(\mu_{\mathbf{x}},\Sigma_{\mathbf{x}})} \Pr(a_i^\top \mathbf{x} + b_i \leq 0) \geq 1-\epsilon_i. \tag{4.21}$$

We follow a naive risk allocation approach by equally distributing the risk measure ϵ among the m constraints such that $\epsilon_i = \frac{\epsilon}{m}$. Alternatively, optimal risk allocation [141] can be achieved using iterative optimization techniques. Using distributional robustness, Problem 6 is reformulated to the following Problem 7.

Problem 7. Distributionally-Robust Chance-Constrained Stochastic Nonlinear Optimal Control.

$$\begin{aligned} J^*_{\mathrm{DR-SNOC}} = \min_{\mathbf{x}(t),\bar{\mathbf{u}}(t)} \quad & \mathbb{E}\left[\int_{t_0}^{t_f} J(\mathbf{x}(t),\bar{\mathbf{u}}(t))dt + J_f(\mathbf{x}(t_f))\right] \\ \text{s.t.} \quad & (4.7),\ (4.15),\ (4.17),\ (4.4), and\ (4.5). \end{aligned}$$

Note: Given a risk measure ϵ, the constraints in Problem 7 are a function of mean $\mu_{\mathbf{x}}$ and covariance matrix $\Sigma_{\mathbf{x}}$ of the state at any time t. While this enables fast computation of chance constraints, it reduces the feasible space $\mathcal{X}_{\mathcal{F}}$. An optimal approach to trade off between the feasible space and computational complexity with theoretical guarantees seems infeasible due to the nonlinearity in the dynamics model. We present an empirical evidence that using distributional robustness approach does not lead to infeasibility in practical scenarios for both planning and control problems. For practical implementation, this approach should be integrated with system design to ensure feasibility. We will transform the SNOC problem to a DNOC problem by applying the generalized polynomial chaos expansion (see Section 2.5). This approach transforms the SNOC problem that is infinite dimensional

in state (stochastic state) and time, to a problem that is infinite dimension only in time. In the following section, we derive an approximate nonlinear ordinary differential equation system for the SDE in (4.7) using gPC expansion (see Section 2.5) and the Galerkin scheme. The DRCC are projected to the gPC coordinates x_{ij} leading to convex constraints.

4.2 Deterministic surrogate of the SNOC Problem

The stochastic nonlinear optimal control problem discussed in Section 4.1.1 is reformulated in terms of the coefficients of the gPC expansion, with decision variables as the gPC coefficients and the control $\bar{\mathbf{u}}$. In the following, we discuss the existence and uniqueness of a solution to the coupled Ordinary Differential Equations (ODE) obtained form gPC approximation of SDE, the cost function in the gPC space, and present the convex constraints for the gPC coefficients obtained from deterministic approximation of chance constraints. We present the convergence and feasibility theorem of the approximation at the end of this section.

4.2.1 Deterministic ODE Approximation of the SDE

The gPC expansion in (2.32) is applied for all the elements in the vector $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{d_\mathbf{x}}$ and the matrix representation using Kronecker product is given in the following, where $\mathrm{X} = \begin{bmatrix} x_{10} & \cdots & x_{1\ell} & \cdots & x_{d_\mathbf{x}0} & \cdots & x_{d_\mathbf{x}\ell} \end{bmatrix}^\top$ are gPC states.

$$\Phi(\xi) = \begin{bmatrix} \phi_0(\xi) & \cdots & \phi_\ell(\xi) \end{bmatrix}^\top \tag{4.22}$$

$$\mathbf{x} \approx \bar{\Phi}\mathrm{X}, \text{ where } \bar{\Phi} = \mathbb{I}_{d_\mathbf{x} \times d_\mathbf{x}} \otimes \Phi(\xi)^\top. \tag{4.23}$$

Consider the following Ito's integral form of the SDE in (4.7).

$$\mathbf{x}(t) = \mathbf{x}(t_0) + \int_{t_0}^{t} f(\mathbf{x}, \bar{\mathbf{u}})dt + \int_{t_0}^{t} g(\mathbf{x}, \bar{\mathbf{u}})dw. \tag{4.24}$$

The gPC projection of the above SDE is given by the following ODE.

$$\begin{aligned} x_{ij}(t) &= x_{ij}(t_0) + \int_{t_0}^{t} \bar{f}_{ij}(\mathrm{X}, \bar{\mathbf{u}})dt + \int_{t_0}^{t} \bar{g}_{ij}(\mathrm{X}, \bar{\mathbf{u}})\sqrt{dt} \\ \bar{f}_{ij} &= \frac{\int_{\mathbb{D}} \boldsymbol{\zeta}(\xi)\phi_j(\xi) f_i(\bar{\Phi}\mathrm{X}, \bar{\mathbf{u}})d\xi}{\langle \phi_j(\xi), \phi_j(\xi) \rangle}, \\ \bar{g}_{ij} &= \frac{\int_{\mathbb{D}} \boldsymbol{\zeta}(\xi)\phi_j(\xi) g_i(\bar{\Phi}\mathrm{X}, \bar{\mathbf{u}})\xi d\xi}{\langle \phi_j(\xi), \phi_j(\xi) \rangle}. \end{aligned} \tag{4.25}$$

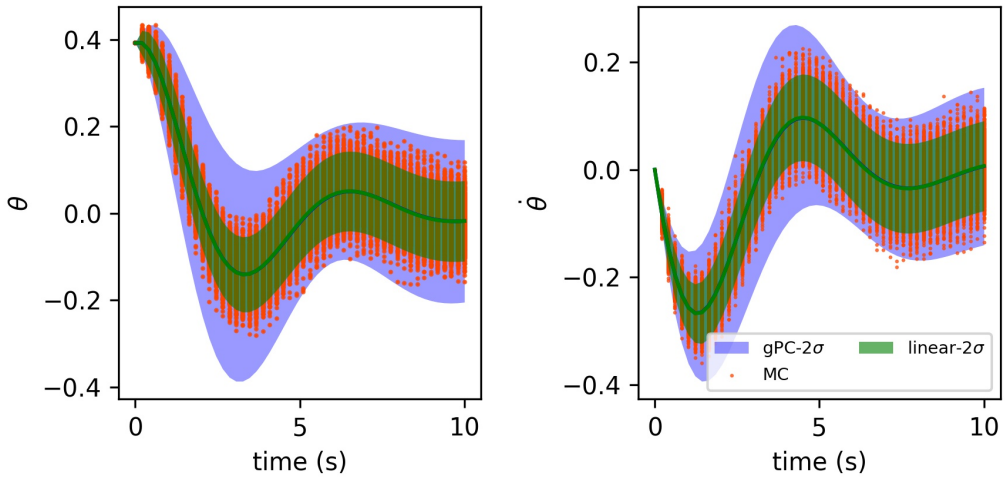


Figure 4.2: Example gPC propagation for a pendulum. The figure compares the mean and 2σ confidence computed using gPC Projection ($P_{\text{gPC}} = 1$), linear covariance propagation and Monte Carlo (MC) propagation of the simple pendulum dynamics $\ddot{\theta} = -\sin\theta - 0.8\dot{\theta} + \sqrt{0.001}\xi(t)$. It is observed that the gPC approximation overestimates the variance compared to MC and the linear covariance propagation underestimates the variance. The $P_{\text{gPC}} = 1$ projection corresponds to a Gaussian approximation that includes the cross correlation between the state and uncertainty.

The dynamics of the coefficients x_{ij} with the above notation is given in (4.27), where f_i and g_i are the i^{th} element of the vector f and i^{th} row of the matrix g respectively. We use the Euler-Maruyama discretization method of the SDE for time integration. The discrete time stochastic dynamics is given as follows:

$$\mathbf{x}[k+1] = \mathbf{x}[k] + f(\mathbf{x}[k], \bar{\mathbf{u}}[k])\Delta t + g(\mathbf{x}[k], \bar{\mathbf{u}}[k])\sqrt{\Delta t}\xi, \tag{4.26}$$

where $\mathbf{x}[k]$, $\bar{\mathbf{u}}[k]$ are the states and controls at time step k, Δt is the integration time interval, and ξ is a multivariate Gaussian distribution $\mathcal{N}(0, \mathbb{I})$. The discrete stochastic system is projected to a discrete deterministic system using gPC method.

$$\begin{aligned} x_{ij}[k+1] = x_{ij}[k] &+ \bar{f}_{ij}(\mathbf{X}[k], \bar{\mathbf{u}}[k])\Delta t \\ &+ \bar{g}_{ij}(\mathbf{X}[k], \bar{\mathbf{u}}[k])\sqrt{\Delta t} \end{aligned} \tag{4.27}$$

The full nonlinear discrete time ODE with the stacked vector X is given as follows:

$$\begin{aligned} \mathbf{X}[k+1] = &\mathbf{X}[k] + \bar{f}(\mathbf{X}[k], \bar{\mathbf{u}}[k], \Delta t) \\ &+ \bar{g}(\mathbf{X}[k], \bar{\mathbf{u}}[k], \sqrt{\Delta t}). \end{aligned} \tag{4.28}$$

Figure 4.2 shows an example of propagation using (4.28). While not discussed here, the projection is also applicable to a higher-order discretization methods [142]. The

sequential convex programming method used for trajectory optimization involves successive linearizations [15] of the dynamics about a given trajectory and discretization for time integration. In Proposition 1, we present the conditions for existence and uniqueness of the solution to the projected system. The existence and uniqueness of solution to the ODE surrogate ensure convergence of any Picard iteration scheme used for integration.

Proposition 1. *The ODE system (4.27) obtained using gPC approximation of the SDE has a solution and the solution is unique, for a given initial condition, assuming that the SDE satisfies the existence and uniqueness conditions in (4.8), (4.9) and the expectation*

$$K_{g_{ij}} = \frac{K}{k_j}\mathbb{E}(L_{g_j}(\xi)), K_{f_j} \quad = \frac{K}{k_j}\mathbb{E}(L_{f_j}(\xi)) \tag{4.29}$$

in (4.29) are bounded for each $j = 0, 1, \cdots, \ell$*, where:* $P = \begin{bmatrix} \bar{\Phi} & 0 \\ 0 & \mathbb{I} \end{bmatrix}$, $k_j = \langle \phi_j, \phi_j \rangle$, $L_{f_j}(\xi) = |\phi_j(\xi)|\|P\|_2$, $L_{g_j}(\xi) = L_{f_j}(\xi)|\phi_1(\xi)|$. *The constants* $K_{g_{ij}}$ *and* K_{f_j} *are the Lipschitz coefficients of the projected functions* $\bar{g}_{ij}$ *and* $\bar{f}_j$ *respectively.*

Proof. See Proposition 1 of [20]. □

While the projection operation preserves existence and uniqueness properties of the SDE, it might not conserve the controllability of the moments of the system. Following examples discuss on how the ϵ_c controllability in Definition 1 of the SDE effects the controllablility of the projected ODE system.

Example 1. Consider the linear SDE $dx = xdt + \bar{u}dt + \sqrt{dt}\xi$, where $x \in \mathbb{R}^1$, $\bar{u} \in \mathbb{R}^1$ and $\xi \sim \mathcal{N}(0, 1)$. Using the random variable ξ as the variable, we can construct the first order gPC expansion $x = x_0 + x_1\xi$ of the state with $x_0, x_1 \in \mathbb{R}^1$. The projected dynamics using the expansion is given as follows:

$$\begin{bmatrix} dx_0 \\ dx_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} dt + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \bar{u}dt + \begin{bmatrix} 0 \\ \sqrt{dt} \end{bmatrix}. \tag{4.30}$$

The dynamics of x_1 is decoupled from x_0 and the propagation is not influenced by the control $\bar{u}$. Notice that, even though the original SDE ($dx = xdt + \bar{u}dt$) is controllable, the projected system (4.30) is not fully controllable. The projection operation converts a SDE to an ODE in higher dimensions. Though this operation enables for fast uncertainty propagation, the linear projected system is underactuated and not fully controllable.

Remark 3. Using a stochastic state feedback of the form $u = -kx$ in Example 1, we get the closed-loop SDE $dx = (1-k)xdt + \sqrt{dt}\xi$. The gPC projection of the system is as follows:

$$\begin{bmatrix} dx_0 \\ dx_1 \end{bmatrix} = \begin{bmatrix} 1-k & 0 \\ 0 & 1-k \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} dt + \begin{bmatrix} 0 \\ \sqrt{dt} \end{bmatrix}. \tag{4.31}$$

Using a stochastic feedback, the state x_1 that corresponds to the variance of the SDE can be controlled.

Example 2. The gPC projection of the nonlinear SDE $dx = (x^2 + \bar{u})dt + \sqrt{dt}\xi$ using the expansion $x = x_0 + x_1\xi$ is given as follows:

$$\begin{bmatrix} dx_0 \\ dx_1 \end{bmatrix} = \begin{bmatrix} x_0^2 + x_1^2 + \bar{u} \\ 2x_0x_1 \end{bmatrix} dt + \begin{bmatrix} 0 \\ \sqrt{dt} \end{bmatrix}. \tag{4.32}$$

The projected system (4.32) is underactuated. In the case of nonlinear systems, the coupling between the dynamics of x_0 and x_1 allows for indirectly controlling the state x_1.

Remark 4. The gPC projected ODE system in (4.27) might not be fully controllable as discussed in Example 1. We choose soft terminal constraints on the variance of the state variable to ensure the feasibility of Problem 6 in accordance with Definition 1.

With above remarks on the controllability of the projected system, we proceed to construct finite dimensional approximation of the cost functional and chance-constraints to formulate the convex-constrained nonlinear deterministic optimal control problem.

4.2.2 Cost Function

Using the notation in (4.23), expectation of the cost functional in (4.6) is expressed in the gPC coefficients as follows:

$$\begin{aligned} J_{\mathrm{gPC}}(\mathbf{X}(t), \bar{u}(t)) &= \mathbf{X}(t)^\top Q_{\mathrm{gPC}} \mathbf{X}(t) + \|\bar{\mathbf{u}}\|_p, \\ J_{\mathrm{gPC}_f}(\mathbf{X}(t_f)) &= \mathbf{X}(t_f)^\top Q_{\mathrm{gPC}_f} \mathbf{X}(t_f), \end{aligned} \tag{4.33}$$

where $Q_{\mathrm{gPC}} = \mathbb{E}(\bar{\Phi}^\top Q \bar{\Phi})$ and $Q_{\mathrm{gPC}_f} = \mathbb{E}(\bar{\Phi}^\top Q_f \bar{\Phi})$. Since the gPC projection is a canonical transformation, we can prove that the projected matrix Q_{gPC} is positive definite.

Proposition 2. *The expectation matrix* $\mathbb{E}(\bar{\Phi}^\top \bar{\Phi})$ *is a positive definite matrix.*

Proof. We can prove the following equality by expanding the matrix multiplication.

$$\mathbb{E}(\bar{\Phi}^\top \bar{\Phi}) = \mathbb{I} \otimes \mathbb{E}(\Phi \Phi^\top) \tag{4.34}$$

The block matrix $\mathbb{E}(\Phi\Phi^\top)$ is positive definite as the functions ϕ_i used to construct the column vector Φ are orthogonal with respect to the density function ζ. Therefore, $\mathbb{E}(\bar{\Phi}^\top\bar{\Phi})$ is positive definite, since $\mathbb{E}(\Phi\Phi^\top)$ is positive definite. □

Lemma 5. *If Q is a positive definite matrix, then the expectation $Q_{\mathrm{gPC}} = \mathbb{E}(\bar{\Phi}^\top Q \bar{\Phi})$ is a positive definite matrix.*

Proof. Since Q is a positive definite matrix, we have $Q \succcurlyeq \lambda_{\min}(Q)\mathbb{I}$ where $\lambda_{\min}(Q) > 0$. The expectation $\mathbb{E}(\bar{\Phi}^\top Q\bar{\Phi})$ can be lower bounded as follows:

$$\mathbb{E}(\bar{\Phi}^\top Q\bar{\Phi}) \succcurlyeq \mathbb{E}(\bar{\Phi}^\top \lambda_{\min}(Q)\mathbb{I}\bar{\Phi}), \tag{4.35}$$

$$\succcurlyeq \lambda_{\min}(Q).\mathbb{E}(\bar{\Phi}^\top\bar{\Phi}). \tag{4.36}$$

Using Proposition 2 in (4.36), we conclude that $\mathbb{E}(\bar{\Phi}^\top Q\bar{\Phi})$ is a positive definite matrix. □

Corollary 0.2. If the polynomials ϕ_i used for gPC projection are Hermite polynomials, then $\mathbb{E}(\bar{\Phi}^\top Q\bar{\Phi}) \succcurlyeq \lambda_{\min}(Q)\mathbb{I}$.

Proof. We can prove Corollary 0.2 by using the fact that for Hermite polynomials $\mathbb{E}(\bar{\Phi}^\top\bar{\Phi}) \succcurlyeq \mathbb{I}$ in Lemma 5. □

4.2.3 Convex Approximation of the Chance Constraint

The deterministic approximations of the chance constraints discussed in Section 4.1.1 are expressed in terms of the gPC coefficients that define a feasible set for the deterministic optimal control problem with gPC coefficients as decision variables.

Lemma 6. *The second-order cone constraint given below*

$$(a^\top \otimes M)X + b + \sqrt{\frac{1-\epsilon}{\epsilon}}\sqrt{X^\top UNN^\top U^\top X} \leq 0 \tag{4.37}$$

is equivalent to the deterministic approximation of the DRLCC in (4.14) as $\ell \to \infty$*., where the matrices* M, U, N *are given by*

$$
\begin{aligned}
M &= \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}_{1\times(\ell+1)} \\
U &= \begin{bmatrix} a_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & a_{d_{\mathbf{x}}} \end{bmatrix} \otimes \mathbb{I}_{(\ell+1)\times(\ell+1)} \\
N &= \mathbb{1}_{d_{\mathbf{x}}\times d_{\mathbf{x}}} \otimes \mathbb{H};\ \mathbb{H} = \begin{bmatrix} 0 & \mathbb{O} \\ \mathbb{O} & \sqrt{\mathbb{E}(HH^\top)} \end{bmatrix} \\
\textit{where } H &= \begin{bmatrix} \phi_1(\xi) & \cdots & \phi_\ell(\xi) \end{bmatrix}^\top
\end{aligned}
\tag{4.38}
$$

and $\mathbb{1}$ *is a matrix with entries as* 1.

Proof. It is sufficient to prove that $(a^\top \otimes M) \approx a^\top \mu_{\mathbf{x}}$ and $\mathbf{X}^\top U N N^\top U^\top \mathbf{X} \approx a^\top \Sigma_{\mathbf{x}} a$ as $\ell \to \infty$. Invoking Lemma 1 and Remark 2, the polynomials of gPC coefficients can be replaced by mean and variable of the variable $\mathbf{x}$.

$$
\begin{aligned}
(a^\top \otimes M)\mathbf{X} &= \begin{bmatrix} a_1 M & a_2 M & \cdots & a_{d_{\mathbf{x}}} M \end{bmatrix} \mathbf{X} \\
&= a_1 x_{10} + a_2 x_{20} + \cdots + a_{d_{\mathbf{x}}} x_{dx0} \\
&\approx a^\top \mu_{\mathbf{x}}.
\end{aligned}
\tag{4.39}
$$

Equation (4.39) shows the steps involved to prove $(a^\top \otimes M) \approx a^\top \mu_{\mathbf{x}}$. Let us define a vector $p_i = \begin{bmatrix} x_{i0} & \bar{p}_i^\top \end{bmatrix}^\top$ where $\bar{p}_i = \begin{bmatrix} x_{i1} & \cdots & x_{i\ell} \end{bmatrix}^\top$.

$$
U^\top \mathbf{X} = \begin{bmatrix} a_1 p_1^\top & a_2 p_2^\top & \cdots & a_{d_{\mathbf{x}}} p_{d_{\mathbf{x}}}^\top \end{bmatrix}^\top \tag{4.40}
$$

$$
N N^\top U^\top \mathbf{X} = \begin{bmatrix} \mathbb{H} a_1 p_1 & \mathbb{H} a_2 p_2 & \cdots & \mathbb{H} a_{d_{\mathbf{x}}} p_{d_{\mathbf{x}}} \end{bmatrix} \tag{4.41}
$$

$$
\begin{aligned}
\mathbf{X}^\top U N N^\top U^\top \mathbf{X} &= \sum_{i=1}^{d_{\mathbf{x}}} \sum_{j=1}^{d_{\mathbf{x}}} a_i a_j p_i^\top \mathbb{H} p_j \\
&= \sum_{i=1}^{d_{\mathbf{x}}} \sum_{j=1}^{d_{\mathbf{x}}} a_i a_j \bar{p}_i^\top \mathbb{E}(HH^\top) \bar{p}_j \\
&\approx a^\top \Sigma_{\mathbf{x}} a.
\end{aligned}
\tag{4.42}
$$

Using this notation, the matrices in (4.37) are expanded as shown in (4.40), (4.41), and (4.42). Therefore the equivalence is proved by Lemma 1 as $\ell \to \infty$. □

Lemma 7. *The quadratic inequality*

$$\sum_{i=1}^{d_{\mathbf{x}}} \sum_{k=1}^{\ell} q_{ii} \langle \phi_k, \phi_k \rangle x_{ik}^2 \leq \epsilon c, \tag{4.43}$$

expressed in terms of the gPC coefficients is equivalent to the constraint in (4.17) as $\ell \to \infty$, where Q is a diagonal matrix with i^{th} diagonal element as q_{ii} and $\langle \phi_k, \phi_k \rangle = \int_{\mathbb{D}} \boldsymbol{\zeta}(\xi) \phi_k \phi_k d\xi$.

Proof. The deterministic approximation, $\text{tr}(Q\Sigma_{\mathbf{x}}) \leq c\epsilon$, of the QCC in (4.18) can be expand as follows.

$$\begin{aligned} \text{tr}(Q\Sigma_{\mathbf{x}}) \leq c\epsilon &\equiv \sum_{i=1}^{d_{\mathbf{x}}} q_{ii} \mathbb{E}((x_i - \mu_{x_i})^\top (x_i - \mu_{x_i})) \leq c\epsilon \\ &\equiv \sum_{i=1}^{d_{\mathbf{x}}} \sum_{j=1}^{\ell} q_{ii} \langle \phi_j, \phi_j \rangle x_{ij}^2 \leq c\epsilon \end{aligned} \tag{4.44}$$

The equivalence is proved by directly expanding the trace and using Remark 2 as shown in (4.44). □

Using the projected dynamics (4.25), the cost functional in (4.33), and the linear and quadratic chance constraints (4.37), (4.43) in gPC coefficients we can formulate the following distributionally-robust deterministic nonlinear optimal control problem with the gPC states X and the control $\bar{\mathbf{u}}$ as the decision variables.

Problem 8. Distributionally-Robust Deterministic Nonlinear Optimal Control Problem.

$$\begin{aligned} J^*_{\text{gPC}} = \min_{\text{X}(t), \bar{\mathbf{u}}(t)} \quad & \int_{t_0}^{t_f} J_{\text{gPC}}(\text{X}(t), \bar{\mathbf{u}}(t)) dt + J_{\text{gPC}_f}(\text{X}(t_f)) \\ \text{s.t.} \quad & (4.28),\ (4.37),\ (4.43) \\ & \bar{\mathbf{u}}(t) \in \mathcal{U} \quad \forall t \in [t_0, t_f] \\ & \text{X}(t_0) = \text{X}_0 \quad \text{X}(t_f) \in \mathcal{X}_{\text{X}_f}, \end{aligned}$$

where the projection of the initial condition $\mathbf{x}_0$ in gPC space is X_0. The terminal set $\mathcal{X}_{\text{X}_f}$ is constructed using a distributionally-robust polytope (or) a conservative ellipsoid approximation of the set $\mathbf{x}_f$ in Problem 6 with probabilistic guarantees using Lemmas 6 and 7. We make the following observations about distributional robustness and gPC projection discussed above for transforming Problem 6 to Problem 8.

- The infinite-dimensional optimal control problem in state space and time, as described in Problem 6, is projected to Problem 8, that is finite dimensional in space and infinite-dimensional in time.
- The ODE approximation of the SDE using gPC projection diverges over long horizon problem, (or) when the uncertainty effecting the SDE has large variance, (or) when the uncertainty model has large gradients with respect to state and control. Multi-element gPC method can be used to over come this divergence due to finite dimensional approximation. The structure of the proposed constraint reformulation is invariant to multi-element gPC method.
- The choice of the terminal set used in Problem 6 is restricted due to the ϵ_c-controllability of the SDE. We use soft constraints on the terminal state to ensure the feasibility of both Problem 6 and 8.
- The projected cost functional preserves the positive definite property of the quadratic cost used in Problem 6.
- The linear and quadratic chance constraints for a given risk measure are second-order cone and semi definite constraints respectively in the gPC coefficients.

Problem 8 (DNOC) enables the use of techniques like psuedo-spectral method, and sequential convex programming for solving Problem 8 (SNOC). We use sequential convex programming to solve Problem 8 and apply this technique to compute safe and optimal motion plans under uncertainty. The model predictive extension of gPC-SCP is applied for controlling a nonlinear robotic system under uncertainty and safety constraints.

4.2.4 gPC-SCP: Generalized Polynomial Chaos-Based Sequential Convex Programming

We formulate the gPC-SCP problem by constructing a sequential convex programming (SCP) approximation of Problem 8 with gPC state (X) and control ($\bar{\mathbf{u}}$) as the decision variables. The convex program is then solved iteratively using interior point method till a convergence criteria is satisfied and projected back to the probability space from the gPC space to compute a solution of Problem 7.

The SCP problem formulation involves two steps: 1) discretizing the continuous time optimal control problem to a discrete time optimal control problem, and 2)

convexifing the non-convex constraints and cost function about a nominal initial state and control trajectory. Following this approach, the projected integral cost functional (4.33), the nonlinear dynamics (4.28), and the second-order cone constraint (4.37) and semi-definite constraint (4.43) are discretized using first-order hold approach for T time steps between the time horizon $[t_0, t_f]$ with gPC state and control as the decision variables.

At iteration i, the cost functional, constraints (4.37) and (4.43), and feasible control set $\mathcal{U}$ are convex. The discretized gPC dynamics in (4.28) is a nonlinear equality constraint at each time step. We convexify the nonlinear dynamics (4.28) by linearizing it about the state and control trajectory $S^{(i-1)} = \{\mathbf{X}^{(i-1)}, \bar{\mathbf{u}}^{(i-1)}\}$ computed at $(i-1)^{th}$ iteration. The linearized equations form a set of linear constraints on the state and control action as follows:

$$\begin{aligned}\mathbf{X}^{(i)}[k+1] = \mathbf{X}^{(i)}[k] + A^{(i)}[k]\mathbf{X}^{(i)}[k] + B^{(i)}[k]\bar{\mathbf{u}}^{(i)}[k] \\ + Z^{(i)}[k], \quad \text{where} : k \in \{1, \ldots, T-1\}.\end{aligned} \tag{4.45}$$

$$A^{(i)} = \frac{\partial(\bar{f}+\bar{g})}{\partial \mathbf{X}}\Big|_{S^{(i-1)}}; \quad B^{(i)} = \frac{\partial(\bar{f}+\bar{g})}{\partial \bar{\mathbf{u}}}\Big|_{S^{(i-1)}}$$

$$\begin{aligned}Z^{(i)} = \bar{f}(S^{(i-1)}, \Delta t) + \bar{g}(S^{(i-1)}, \sqrt{\Delta t}) \\ - A^{(i)}\mathbf{X}^{(i-1)} - B^{(i)}\bar{\mathbf{u}}^{(i-1)}.\end{aligned} \tag{4.46}$$

The gPC-SCP problem at iteration i, after discretization and convexification is given in the following Problem 9.

Problem 9. gPC-SCP: Generalized Polynomial Chaos-based Sequential Convex Programming.

$$\min_{\mathbf{X}^{(i)}, \bar{\mathbf{u}}^{(i)}} \quad \sum_{k=1}^{T-1} J_{\text{gPC}}(\mathbf{X}^{(i)}[k], \bar{\mathbf{u}}^{(i)}[k])\Delta t + J_{\text{gPC}_f}(\mathbf{X}^{(i)}[T])$$

s.t. Projected Dynamics :(4.45)

Constraints : $\{(4.37),\ (4.43)\}$

$\bar{\mathbf{u}}^{(i)}[k] \in \mathcal{U} \quad \forall\, k \in \{1, \ddots, T-1\}$

$\mathbf{X}^{(i)}[1] = \mathbf{X}_0 \quad \mathbf{X}^{(i)}[T] \in \mathcal{X}_{\mathbf{X}_f}$

$$\|\mathbf{X}^{(i)}[k] - \mathbf{X}^{(i-1)}[k]\|_2^2 \leq \alpha_{\mathbf{x}}\beta \;\forall\, k \in \{1, \ldots, T\} \tag{4.47}$$

$$\|\bar{\mathbf{u}}^{(i)}[k] - \bar{\mathbf{u}}^{(i-1)}[k]\|_2^2 \leq \alpha_{\mathbf{u}}\beta \;\forall\, k \in \{1, \ldots, T-1\}. \tag{4.48}$$

Problem 9 shows the SCP formulation at i^{th}, given a nominal trajectory $S^{(i-1)} = \{\mathbf{X}^{(i-1)}, \bar{\mathbf{u}}^{(i-1)}\}$ computed at $(i-1)^{\text{th}}$ iteration with the constraint set at each time step k and iteration i, where $\mathcal{X}_{\mathrm{X}_f}$ is the projected terminal constraint. The nominal trajectory S^0 at $i = 1$, used to initialize gPC-SCP, is computed using a deterministic trajectory optimization for the nominal dynamics $\dot{\mathbf{x}} = f(\mathbf{x}, \bar{\mathbf{u}})$, that ignores the uncertainty affecting the system. For motion planning problem, the nominal trajectory S^0 is computed using kino-dynamic motion planning algorithms like asymptotically-optimal rapidly exploring random trees [51].

An additional trust region constraint on the gPC state (4.47) and control (4.48) are used to ensure the convergence and feasibility of the SCP as $i \to \infty$, where $\alpha_{\mathbf{x}} > 0$, $\alpha_{\mathbf{x}} > 0$, and $\beta \in (0, 1)$. The choice of β ensures the convergence of the trust region as the number of iterations increases. This acts as a convergence criteria, while ensuring that the search space is small. The trust region on gPC state in (4.47) can be equivalently understood as probabilistic constraint of the form $\Pr(\|\mathbf{x}^i - \mathbf{x}^{i-1}\| \leq \alpha_{xp}) \geq 1 - \epsilon_t$, where $\alpha_{\mathbf{x}}$ is a function of α_{xp} using the quadratic projection discussed in Lemma 7. The SCP algorithm is known to converge to the KKT point of the DNOC problem under mild conditions. For detailed analysis on convergence, see [14, 15, 48]. We ensure feasibility of gPC-SCP: 1) by using stochastic reachable terminal sets, as discussed in [80], that are constructed using the linearized approximation of the dynamics, and 2) by increasing the trust region in-loop with $\beta > 1$ when an in-feasibility occurs.

4.2.5 Sub-Optimality and Convergence

In this section, we study the optimality of Problem 8 and show that Problem 8 computes a sub-optimal solution to Problem 6. We make a two step approximation of Problem 6 by using distributional robustness to formulate Problem 7 with known mean and variance of the state and then use gPC propagation to construct the deterministic optimal control Problem 8 that is solved using SCP. In Lemma 8, we prove the sub-optimality of the optimal cost J^*_{SNOC} of Problem 6 compared to the optimal cost $J^*_{\text{DR}-\text{SNOC}}$ of Problem 7 that is distributionally robust.

Lemma 8. *The optimal solution of Problem 7 is a sub-optimal solution of Problem 6, i.e.,* $J^*_{\text{SNOC}} \leq J^*_{\text{DR}-\text{SNOC}}$.

Proof. The constraint set $\mathcal{X}_{\text{DRLCC}}$ and $\mathcal{X}_{\text{CQCC}}$ in Problem 7 are a subset of the constraint set $\mathcal{X}_{\text{LCC}}$ and $\mathcal{X}_{\text{QCC}}$ of Problem 6 respectively. Therefore, $J^*_{\text{SNOC}} \leq$

$J^*_{\mathrm{DR-SNOC}}$ as the feasible space of Problem 6 is larger than the feasible space of Problem 7. □

Problem 8 (DNOC) computed via gPC projection converges asymptotically to Problem 7 (SNOC). The following theorem discuss the conditions for convergence.

Theorem 1. *The surrogate deterministic nonlinear optimal control Problem 8 with convex constraints is a sub-optimal surrogate for the stochastic nonlinear optimal control Problem 6 with following being true:*
***(a)** In the case with no chance constraints, the cost $|J^*_{\mathrm{gPC}} - J^*| \to 0$ as $\ell \to \infty$*
***(b)** In the case with linear and quadratic chance constraints, any feasible solution of Problem 8 is a feasible solution of Problem 6 as $\ell \to \infty$ and $J^*_{\mathrm{SNOC}} \leq J^*_{\mathrm{gPC}}$, assuming that a feasible solution exists.*

Proof. **Case (a)**: It is sufficient to prove that the cost function and the dynamics are exact as $\ell \to \infty$. Using the Kronecker product notation, due to Lemma 1, we have the following:

$$\|\mathbf{x} - \bar{\Phi}\mathrm{X}\|_{\mathcal{L}_2} \to 0 \text{ as } \ell \to \infty \tag{4.49}$$

$$(4.49) \implies \|\dot{\mathbf{x}} \to \bar{\Phi}\dot{\mathrm{X}}\|_{\mathcal{L}_2} \to 0 \text{ as } \ell \to \infty \tag{4.50}$$

$$\begin{aligned} (4.49) \implies |J_{\mathrm{gPC}} - J| &\to 0 \text{ as } \ell \to \infty \\ |J_{\mathrm{gPC}_f} - J_f| &\to 0 \text{ as } \ell \to \infty \end{aligned} \tag{4.51}$$

From (4.50), and (4.51) we conclude that the optimal value $|J^*_{\mathrm{gPC}} - J^*| \to 0$ as $\ell \to \infty$, since the cost function, the dynamics and the initial and terminal conditions are exact as $\ell \to \infty$.
Case (b): Consider the sets $\mathcal{X}_{\mathrm{LgPC}}$, and $\mathcal{X}_{\mathrm{QgPC}}$ defined below.

$$\mathcal{X}_{\mathrm{LgPc}} = \{\mathbf{x} \in \mathcal{X} : x \approx \bar{\Phi}\mathrm{X} \text{ where } \mathrm{X} \in (4.37)\} \tag{4.52}$$

$$\mathcal{X}_{\mathrm{QgPC}} = \{\mathbf{x} \in \mathcal{X} : \mathbf{x} \approx \bar{\Phi}\mathrm{X} \text{ where } \mathrm{X} \in (4.43)\} \tag{4.53}$$

Using Lemmas 6 and 7, we have the approximate convex constraints converge to the deterministic equivalent of the distributionally robust chance constraint as $\ell \to \infty$.

$$\begin{aligned} \mathit{Lemma}\ 6 &\implies \mathcal{X}_{\mathrm{LgPc}} \to \mathcal{X}_{\mathrm{DRLCC}} \text{ as } \ell \to \infty \\ \mathit{Lemma}\ 7 &\implies \mathcal{X}_{\mathrm{QgPC}} \to \mathcal{X}_{\mathrm{CQCC}} \text{ as } \ell \to \infty \end{aligned} \tag{4.54}$$

Using Lemmas 3 and 4, we have the following:

$$\begin{aligned} Lemma\ 3 &\implies \mathcal{X}_{\mathrm{DRLCC}} \subseteq \mathcal{X}_{\mathrm{LCC}} \\ Lemma\ 4 &\implies \mathcal{X}_{\mathrm{CQCC}} \subseteq \mathcal{X}_{\mathrm{QCC}} \end{aligned} \tag{4.55}$$

$$\{(4.54),\ (4.55)\} \implies \begin{cases} \mathcal{X}_{\mathrm{LgPc}} \subseteq \mathcal{X}_{\mathrm{LCC}} \text{ as } \ell \to \infty \\ \mathcal{X}_{\mathrm{QgPC}} \subseteq \mathcal{X}_{\mathrm{QCC}} \text{ as } \ell \to \infty \end{cases} \tag{4.56}$$

Combining (4.54) and (4.55), we can conclude that (4.56) holds as $\ell \to \infty$. This proves that if a feasible solution exists for Problem 8 then it is a feasible solution of Problem 6 as $\ell \to \infty$. Using Lemma 8, as $\ell \to \infty$ we have $J^*_{\mathrm{gPC}} \to J^*_{\mathrm{DR-SNOC}}$. This implies that $J^*_{\mathrm{SNOC}} \leq J^*_{\mathrm{gPC}}$. □

The above theorem proves the consistency of the gPC projection method as $\ell \to \infty$. The asymptotic convergence of the cost and the chance constraints constraints is achieved with large number of the polynomials ϕ. This leads to a deterministic optimal control problem with size $\ell d_{\mathbf{x}}$. The choice of ℓ depends on the number of uncertainty in the system and the nature of the state distribution. A computationally efficient approach is to use $P_{\mathrm{gPC}} = 2$ for generating the functions ϕ used in the projection. This replicates the computational efficiency of linear covariance propagation techniques, while ensuring the convexity of the chance constraints in gPC space. We study the chance constraint formulation for collision checking under uncertainty in dynamics and obstacle locations in the following section using distributional robustness and gPC projection.

4.3 Chapter Summary

We present a generalized polynomial chaos-based sequential convex programming method for safe and optimal motion planning and control under uncertainty in dynamics and constraints. The method uses generalized polynomial chaos projection and distributional robustness to compute a convex subset of the multi-model state-dependent chance constraints, and a high-fidelity deterministic surrogate of the stochastic dynamics and the cost functional. The surrogate deterministic optimal problem is a finite-dimensional approximation of the stochastic optimal control problem and enables the use of sequential convex programming for trajectory optimization. We study the controllability of the surrogate deterministic dynamics and propose a terminal constraint to ensure the feasibility of the surrogate optimal control problem. We prove the asymptotic convergence of the surrogate problem to the

stochastic optimal control problem. The asymptotic convergence property of the deterministic surrogate allows for achieving a greater degree of safety.

Chapter 5

GUIDANCE AND CONTROL UNDER UNCERTAINTY

We formulate a motion planning problem to incorporate uncertainty in dynamics and obstacles using the gPC-SCP method proposed in Chapter 4. We derive provably convex collision constraints using the linear and quadratic chance constraints and integrate them with the gPC-SCP method to compute safe motion plans under uncertainty. We extend the gPC-SCP method for tracking control using a stochastic model predictive formulation.

5.1 Motion Planning Under Uncertainty

The motion planning problem is to compute an optimal and safe trajectory ($\mathbf{x} \in \mathcal{X}_{\mathcal{F}}$) for the SDE in (4.7) from an initial state $\mathbf{x}_0 \in \mathcal{X}$ to the terminal set $\mathcal{X}_f \subseteq \mathcal{X}$ on a given map with static obstacles. In the following, we derive a chance constraint formulation of the collision constraint and terminal state constraint. The chance constraints are used to formulate a SNOC problem as described in Problem 6. The SNOC problem is then projected to the gPC space for solving via SCP method. At each SCP iteration, the collision constraints are approximated as linear chance constraint around the nominal trajectory and form a second-order cone constraint in gPC state X as discussed in Lemma 6. The terminal set is defined as a soft constraint on an ellipsoidal set and forms a semi-definite constraint in gPC states as discussed in Lemma 7.

In the following, we first discuss the linear chance constraint formulation for collision checking with a deterministic obstacle and then extend it to include the uncertainty in obstacle locations for SCP. We prove that the approximation is a subset of the original nonlinear chance constraint. We then discuss the chance constraint formulation of the terminal set constraint. The chance constraint formulations for collision checking and terminal set are used to design a motion planning algorithm that integrates an asymptotically-optimal sampling based planner [51] with the gPC-SCP method for computing a safe trajectory under uncertainty.

5.1.1 Collision Checking with Deterministic Obstacles

We derive a second-order cone constraint approximation of the circular obstacle in the gPC coordinates under the uncertainty in dynamics at any point in time $t \in$

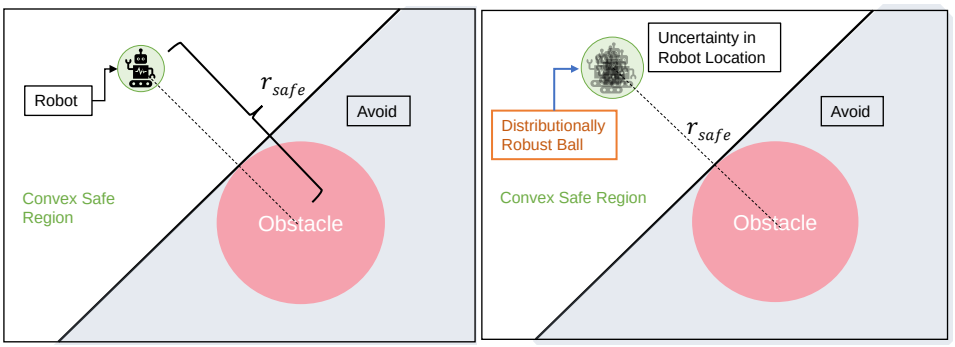


Figure 5.1: An illustration of the convex linear constraint used for collision checking in deterministic SCP at a particular instant in time is shown on the left. The linear chance constraint under stochastic dynamics is projected to gPC space forming a second-order cone constraint. The cone constraint is visualized as a robustness bound on the robot's state as shown in the figure on right.

$[t_0, t_f]$. The approximation involves two steps. We first derive a conservative linear chance constraint approximation of the nonlinear collision chance constraint. In the second step we project the linear chance constraint to a second order cone constraint in the gPC coordinates. Let the state of the obstacle be $\bar{\mathbf{p}}_{\text{obs}}$ at time t and the radius of the obstacle be r_{obs}. The collision chance constraint at any time t for a robot with the state distribution $\mathbf{x}$ and radius r_{rob} is given as follows:

$$\Pr\left(\|\mathbf{C}(\mathbf{x} - \bar{\mathbf{p}}_{\text{obs}})\|_2 \geq r_{\text{rob}} + r_{\text{obs}}\right) \geq 1 - \epsilon_{\text{col}}, \tag{5.1}$$

where the matrix $\mathbf{C}$ is used to compute the position of the obstacle and the robot given the states $\bar{\mathbf{p}}_{\text{obs}}$ and $\mathbf{x}$ respectively. The probability of collision is tuned using the risk measure $\epsilon_{\text{col}} \in [0.001, 0.1]$. In the following theorem, we prove that, given a nominal state distribution trajectory $\mathbf{x}_{\text{nom}}$, we can compute a conservative linear chance constraint approximation of the collision constraint (5.1). The nominal trajectory $\mathbf{x}_{\text{nom}}$ can computed by using a deterministic planner without considering the uncertainty in the dynamics.

Theorem 2. *The linear chance constraint,*

$$\begin{aligned}\Pr\Big((\bar{\mathbf{x}}_{\text{nom}} - \bar{\mathbf{p}}_{\text{obs}})^\top \mathbf{C}^\top \mathbf{C}(\mathbf{x} - \bar{\mathbf{p}}_{\text{obs}}) \\ \geq r_{\text{safe}}\|\mathbf{C}(\bar{\mathbf{x}}_{\text{nom}} - \bar{\mathbf{p}}_{\text{obs}})\|_2\Big) \geq 1 - \epsilon_{\text{col}},\end{aligned} \tag{5.2}$$

in robot state distribution $\mathbf{x}$ *is a conservative approximation of the nonlinear collision chance constraint in* (5.1) *at any time* $t \in [t_0, t_f]$*, where* $\bar{\mathbf{x}}_{\text{nom}}$ *is a realization of the nominal state distribution* $\mathbf{x}_{\text{nom}}$ *of the robot that satisfies* (5.1)*,* $\bar{\mathbf{p}}_{\text{obs}}$ *is the state of the obstacle, and* $r_{\text{safe}} = r_{\text{rob}} + r_{\text{obs}}$.

Proof. Consider the set $\mathcal{X}_{\text{free}}$, defined as $\mathcal{X}_{\text{free}} = \{\mathbf{x} : \|\mathbf{C}(\mathbf{x} - \bar{\mathbf{p}}_{\text{obs}})\|_2 \geq r_{\text{safe}}\}$. Given a nominal trajectory $\mathbf{x}_{\text{nom}}$, the set $\mathcal{X}_s$ defined as $\mathcal{X}_s = \{\mathbf{x} : (\bar{\mathbf{x}}_{\text{nom}} - \bar{\mathbf{p}}_{\text{obs}})^\top \mathbf{C}^\top \mathbf{C}(\mathbf{x} - \bar{\mathbf{p}}_{\text{obs}}) \geq r_{\text{safe}} \|\mathbf{C}(\bar{\mathbf{x}}_{\text{nom}} - \bar{\mathbf{p}}_{\text{obs}})\|_2\}$ is such that $\mathcal{X}_s \subseteq \mathcal{X}_{\text{free}}$. For a proof of $\mathcal{X}_s \subseteq \mathcal{X}_{\text{free}}$ see [15]. Figure 5.1 shows an example of the set $\mathcal{X}_s$ and $\mathcal{X}_{\text{free}}$, where the hyperplane used for linearization of the circular constraint leads to reduced feasible space. We can construct an indicator function $I_{\text{free}}(\mathbf{x})$ such that $I_{\text{free}}(\mathbf{x}) = 1$ if $\mathbf{x} \in \mathcal{X}_{\text{free}}$ and $I_{\text{free}}(\mathbf{x}) = 0$ otherwise. Similarly, indicator $I_s(\mathbf{x})$ is such that $I_s(\mathbf{x}) = 1$ if $\mathbf{x} \in \mathcal{X}_s$ and $I_s(\mathbf{x}) = 0$ otherwise.

$$\begin{aligned} &\text{Since } \mathcal{X}_s \subseteq \mathcal{X}_{\text{free}},\ \mathbf{x} \in \mathcal{X}_s \implies \mathbf{x} \in \mathcal{X}_{\text{free}}, \\ &\text{and } I_s(\mathbf{x}) = 1 \implies I_{\text{free}}(\mathbf{x}) = 1. \end{aligned} \tag{5.3}$$

Therefore, if $\mathbb{E}(I_s) \geq 1 - \epsilon_{\text{col}}$, then $\mathbb{E}(I_{\text{free}}) \geq 1 - \epsilon_{\text{col}}$ with at least $1 - \epsilon_{\text{col}}$ probability. Note that, $\Pr(\mathbf{x} \in \mathcal{X}_{\text{free}}) = \mathbb{E}(I_{\text{free}}(\mathbf{x}))$ and $\Pr(\mathbf{x} \in \mathcal{X}_s) = \mathbb{E}(I_s(\mathbf{x}))$. This implies that if the chance constraint in (5.2) is satisfied with probability $1 - \epsilon_{\text{col}}$, then the constraint in (5.1) is satisfied with at least a probability of $1 - \epsilon_{\text{col}}$. The distributional robustness approach can be visualized, as shown in Fig. 5.1, as a robust ball around the robot's state for collision checking using the convex feasible subset $\mathcal{X}_s$ of the non-convex feasible space $\mathcal{X}_{\text{free}}$. □

Remark 5. The constraint (5.2) is of the form $\Pr(a^\top \mathbf{x} + b \leq 0) \geq 1 - \epsilon_{\text{col}}$, where $a = -(\bar{\mathbf{x}}_{\text{nom}} - \bar{\mathbf{p}}_{\text{obs}})^\top \mathbf{C}^\top \mathbf{C}$, and $b = (\bar{\mathbf{x}}_{\text{nom}} - \bar{\mathbf{p}}_{\text{obs}})^\top \mathbf{C}^\top \mathbf{C} \bar{\mathbf{p}}_{\text{obs}} + r_{\text{safe}} \|\mathbf{C}(\bar{\mathbf{x}}_{\text{nom}} - \bar{\mathbf{p}}_{\text{obs}})\|_2$. Using the Lemma 2, we formulate a second-order cone constraint that is used in the SCP problem for collision checking.

5.1.2 Collision Checking with Stochastic Obstacles

We extend the linear chance constraint formulation in (5.2) to include uncertainty in obstacle state. Let the obstacle state distribution be $\mathbf{p}_{\text{obs}} \sim \mathcal{N}(\mu_{\mathbf{p}}, \Sigma_p)$, where $\mu_{\mathbf{p}}$ is the mean, Σ_p is the variance matrix, and the radius of obstacle is r_{obs}.

Assumption 3. The obstacle state distribution $\mathbf{p}$ is uncorrelated to the state distribution $\mathbf{x}$ of the robot.

The collision chance constraint at any time t for a state distribution $\mathbf{x}$ and radius r_{rob} is given as follows:

$$\Pr\left(\|\mathbf{C}(\mathbf{x} - \mathbf{p}_{\text{obs}})\|_2 \geq r_{\text{rob}} + r_{\text{obs}}\right) \geq 1 - \epsilon_{\text{col}}, \tag{5.4}$$

where both $\mathbf{x}$ and $\mathbf{p}_{\text{obs}}$ are random variables, unlike in (5.1).

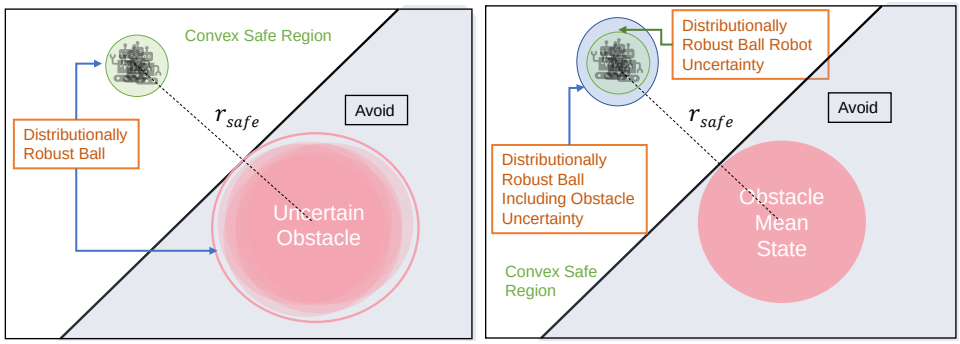


Figure 5.2: An illustration of the second-order cone constraint used for collision checking with uncertainty in dynamics and the obstacle position at an instant in time is shown on the right. For a given risk of collision probability ϵ, the uncertainty in obstacle position is visualized as an additional uncertainty in the robots state.

Theorem 3. *The linear chance constraint,*

$$\Pr\Big((\bar{\mathbf{x}}_{\text{nom}} - \bar{\mathbf{p}}_{\text{obs}})^\top \mathbf{C}^\top \mathbf{C}(\mathbf{x} - \mathbf{p}_{\text{obs}}) \\ \geq r_{\text{safe}} \|\mathbf{C}(\bar{\mathbf{x}}_{\text{nom}} - \bar{\mathbf{p}}_{\text{obs}})\|_2\Big) \geq 1 - \epsilon_{\text{col}}, \tag{5.5}$$

in robot state distribution $\mathbf{x}$ *and obstacle state distribution* $\mathbf{p}_{\text{obs}}$ *is a conservative approximation of the nonlinear collision chance constraint in* (5.4) *at any time* $t \in [t_0, t_f]$*, where* $\bar{\mathbf{x}}_{\text{nom}}$ *is a realization of the nominal state distribution* $\mathbf{x}_{\text{nom}}$ *of the robot,* $\bar{\mathbf{p}}_{\text{obs}}$ *is a realization of the obstacle state distribution* $\mathbf{p}_{\text{obs}}$*, and* $r_{\text{safe}} = r_{\text{rob}} + r_{\text{obs}}$.

Proof. Consider the sets $\mathcal{X}_{\text{free}}$ and $\mathcal{X}_s$, defined as $\mathcal{X}_{\text{free}} = \{\mathbf{x} : \|\mathbf{C}(\mathbf{x} - \mathbf{p}_{\text{obs}})\|_2 \geq r_{\text{safe}}$ and $\mathcal{X}_s = \{\mathbf{x} : (\bar{\mathbf{x}}_{\text{nom}} - \bar{\mathbf{p}}_{\text{obs}})^\top \mathbf{C}^\top \mathbf{C}(\mathbf{x} - \mathbf{p}_{\text{obs}}) \geq r_{\text{safe}} \|\mathbf{C}(\bar{\mathbf{x}}_{\text{nom}} - \bar{\mathbf{p}}_{\text{obs}})\|_2\}$ respectively, where $\bar{\mathbf{p}}_{\text{obs}}$ is a sample from the obstacle state distribution $\mathbf{p}_{\text{obs}}$. In the constraint $\mathcal{X}_s$, $(\mathbf{x} - \mathbf{p}_{\text{obs}})$ is the decision variable. Note that, for any realization of the state $\bar{\mathbf{x}}$ and the $\bar{\mathbf{p}}_{\text{obs}}$ we have $\mathcal{X}_s \subseteq \mathcal{X}_{\text{free}}$ (see [15] for the proof). Using the arguments in Theorem 2, the constraint (5.5) is a conservative approximation of the constraint (5.4). As shown in Fig. 5.2, the uncertainty in obstacle is projected as an additional uncertainty in robot's state for collision checking using the hyperplane approximation. □

Remark 6. The constraint (5.5) is a linear chance constraint of the form $\Pr(a^\top(\mathbf{x} - \mathbf{p}_{\text{obs}}) + b \leq 0) \geq 1 - \epsilon_{\text{col}}$, where $a = -(\bar{\mathbf{x}}_{\text{nom}} - \bar{\mathbf{p}}_{\text{obs}})^\top \mathbf{C}^\top \mathbf{C}$, $b = r_{\text{safe}} \|\mathbf{C}(\bar{\mathbf{x}}_{\text{nom}} - \bar{\mathbf{p}}_{\text{obs}})\|_2$, and $\mathbf{p}_{\text{obs}} \sim \mathcal{N}(\mu_{\mathbf{p}}, \Sigma_p)$. In this case, the distributionally-robust deterministic surrogate is computed for the stacked state $\mathbf{x}_c = [\mathbf{x}^\top \mathbf{p}_{\text{obs}}^\top]^\top$, that includes both

robot and the obstacle state. The surrogate constraint is given as follows:

$$a^\top \mu_{\mathbf{x}} - a^\top \mu_{\mathbf{p}} + b + \sqrt{\tfrac{1-\epsilon_{\text{col}}}{\epsilon_{\text{col}}}} \sqrt{a^\top \Sigma_{\mathbf{x}} a + a^\top \Sigma_{\mathbf{p}} a} \leq 0. \tag{5.6}$$

Using Lemma 6, the inequality constraint in moments is transformed to a second-order cone constraint in terms of the gPC states X of the robot dynamics.

Remark 7. For correlated obstacle state $\mathbf{p}$ and robot state $\mathbf{x}$, with the cross correlation matrix Σ_{xp}, the deterministic surrogate of (5.4) is given as follows:

$$\begin{aligned} &a^\top \mu_{\mathbf{x}} - a^\top \mu_{\mathbf{p}} + b + \\ &\sqrt{\tfrac{1-\epsilon_{\text{col}}}{\epsilon_{\text{col}}}} \sqrt{a^\top \Sigma_{\mathbf{x}} a + 2a^\top \Sigma_{\mathbf{xp}} a + a^\top \Sigma_{\mathbf{p}} a} \leq 0. \end{aligned} \tag{5.7}$$

The derivation uses the stacked state $\mathbf{x}_c$, as shown in Remark 6.

Remark 8. Theorem 3 can be applied for safe multi-agent reconfiguration under uncertainty by replacing the obstacle state $\mathbf{p}_{\text{obs}}$ with the neighbouring robots state. The robots communicate the moments used in (5.7) for collision checking with the neighbouring agents.

5.1.3 Terminal Constraint

The terminal constraint is defined as an ellipsoidal set $(\mathbf{x} - \bar{\mathbf{x}}_f)^\top Q_{\mathcal{X}_f} (\mathbf{x} - \bar{\mathbf{x}}_f) \leq c_f$ around a terminal point $\bar{\mathbf{x}}_f$, where $Q_{\mathcal{X}_f}$ is a positive definite matrix. The chance constraint formulation of the terminal set involves two steps: 1) constraining the mean of the terminal point as $\mu_f = \bar{\mathbf{x}}_f$ and 2) formulating the quadratic chance constraint $\Pr((\mathbf{x} - \bar{\mathbf{x}}_f)^\top Q_{\mathcal{X}_f} (\mathbf{x} - \bar{\mathbf{x}}_f) \leq c_f) \geq 1 - \epsilon_f$ around the mean μ_f with risk measure ϵ_f of not reaching the terminal set. We use the conservative deterministic constraint discussed in Lemma 4, that bounds the variance of the state. The terminal constraints are summarized as follows:

$$\mu_f = \bar{\mathbf{x}}_f, \quad \frac{1}{c_f} \text{tr}(Q_{\mathcal{X}_f} \Sigma_{\mathbf{x}_f}) \leq \epsilon_f, \tag{5.8}$$

where μ_f is the mean and $\Sigma_{\mathbf{x}_f}$ is the variance of the terminal state. The conservative approximations we presented in this Section are a trade-off between the knowledge of moments available and the computational speed achieved by convex constraints. For a linear SDE with obstacles whose uncertainty is described by Gaussian distribution, a tighter equivalent deterministic surrogate constraints can be derived using the covariance propagation technique for uncertainty propagation and the inverse cumulative distribution function for Gaussian distribution.

5.1.4 Motion Planning Algorithm

For motion planning, we integrate the deterministic approximations discussed in Sections 5.1.1, 5.1.2, and 5.1.3 with the asymptotically-optimal rapidly exploring random trees [51] (AO-RRT) algorithm. Following Algorithm 2, outlines the motion planning method using gPC-SCP for a dynamical system under uncertainty.

Algorithm 2: Distributionally-Robust Motion Planning.

Input: Map, obstacle location, $\mathbf{x}_0$, $\mathcal{X}_f$, Δt, ℓ,
Input: Uncertainty model of $g(\mathbf{x}, \bar{\mathbf{u}})$ in SDE (4.7).
Output: Optimal and safe state distribution $\mathcal{X}_{\text{sol}} = \{\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_T\}$ and control input $\mathcal{U}_{\text{sol}} = \{\bar{\mathbf{u}}_0, \bar{\mathbf{u}}_1, ..., \bar{\mathbf{u}}_{T-1}\}$.

▷*Stage 1: gPC Projection.*
Problem 8 ← **gPC Projection**
  Formulate the collision constraint using Theorems 2 and 3,
  Formulate the terminal set $\mathcal{X}_f$ using (5.8),
  Project the SDE using (4.25),
  Project the collision constraint using Lemma 6,
  Project the terminal set using Lemma 7,
  Setup and project cost function using (4.33),
  return: Problem 8 in gPC space.
Problem 9 ← **`Linearize`** (**`Discretize`** (Problem 8))
Save Problem 9.
▷*Stage 2: Compute a nominal trajectory using AO-RRT.*
$\{\mathcal{X}^0_{\text{sol}}, \mathcal{U}^0_{\text{sol}}, T\}$ ← **`AO-RRT`** $(\mathbf{x}_0, \mathcal{X}_f, \Delta t, \dot{\mathbf{x}} = f(\mathbf{x}, \bar{\mathbf{u}}))$ `/* For detailed implementation of AO-RRT see [51]. */`
▷*Stage 3: gPC-SCP.*
$\{\mathcal{X}_{\text{sol}}, \mathcal{U}_{\text{sol}}\}$ ←**`SCP`** (Problem 9,$\{\mathcal{X}^0_{\text{sol}}, \mathcal{U}^0_{\text{sol}}, T\}$) `/* The sequential convex programming (SCP) approach is described in Section 4.2.4. */`

Algorithm 2 has three stages. In Stage 1, we formulate the linear chance constraint for collision checking and the quadratic chance constraint for the terminal constraint respectively. Using the chance constraints, we setup Problem 7 (SNOC) and project it to Problem 8 (DNOC). We formulate the gPC-SCP in Problem 9 by discretizing Problem 8. In Stage 2, we use AO-RRT to compute an initial feasible trajectory $\{\mathcal{X}^0_{\text{sol}}, \mathcal{U}^0_{\text{sol}}\}$ for the nominal dynamics $\dot{x} = f(\mathbf{x}, \bar{\mathbf{u}})$. In Stage 3, the feasible trajectory is then used to initialize the SCP iterations in gPC-SCP, that optimizes for the uncertainty in dynamics. The output of stage 3 is a safe and optimal state trajectory in gPC space. Using the gPC polynomials in (4.23), the gPC space trajec-

tory is projected back to the state space distribution to output $\{\mathcal{X}_{\text{sol}}, \mathcal{U}_{\text{sol}}\}$ in line 12 of Algorithm 2. Note that RRT in AO-RRT can be replaced with sparse tree [143] algorithm for improved speed and with RRT* [144] for optimality. We discuss the application of Algorithm 2 in Section 5.3.

5.2 Tracking Control Using Stochastic Model Predictive Control

We derive a stochastic model predictive control (SMPC) algorithm using gPC-SCP for the nonlinear control affine system defined as follows:

$$d\mathbf{x} = f(\mathbf{x})dt + B(\mathbf{x})\bar{\mathbf{u}}dt + g(\mathbf{x}, \bar{\mathbf{u}})dw, \tag{5.9}$$

to track a desired state and control trajectory $(\bar{\mathbf{x}}_{\text{des}}(t), \bar{\mathbf{u}}_{\text{des}}(t))$, that is at least $\mathcal{C}^2$ continuous and defined $\forall t \in [t_0, t_f]$. In the SMPC approach, we solve Problem 7 for a fixed time horizon $[t_0, t_h]$, where $t_0 \leq t_{\mathcal{S}} < t_h < t_f$, and apply the control input at the time t_0. The control input $\bar{\mathbf{u}}$ is state dependent, i.e., $\bar{\mathbf{u}} = \bar{\mathbf{u}}(\mathbf{x})$, as the SMPC problem is solved at each sample time $t_{\mathcal{S}}$ with an estimate of the current state $\bar{\mathbf{x}}(t_{\mathcal{S}})$ as an input. The SMPC approach discussed below will ensure safety of the system in real-time at the control stage.

5.2.1 Continuous-time SMPC for Reference Tracking

We first present the continuous-time SMPC problem and then discuss the conditions for convergence and stability in terms of probability. The finite-horizon SMPC problem for tracking a desired trajectory $(\bar{\mathbf{x}}_{\text{des}}, \bar{\mathbf{u}}_{\text{des}})$ in the error state $\delta\mathbf{x} = \mathbf{x} - \bar{\mathbf{x}}_{\text{des}}$ and the control $\delta\bar{\mathbf{u}} = \bar{\mathbf{u}} - \bar{\mathbf{u}}_{\text{des}}$ is given by the following Problem 10. The desired trajectory $(\bar{\mathbf{x}}_{\text{des}}(t), \bar{\mathbf{u}}_{\text{des}}(t))$ is computed for the nominal dynamics $d\mathbf{x} = f(\mathbf{x})dt + B(\mathbf{x})\bar{\mathbf{u}}dt$ by using a deterministic motion planning algorithm. Note that the desired trajectory is still a feasible trajectory for the SDE (5.9). Hence, the desired trajectory could be unsafe in the presence of a white noise in the dynamics. We assume that the full state information is available to the controller.

Problem 10. Continuous-Time SMPC.

$$J^*_{\mathcal{S}_0} = \min_{\delta\mathbf{x}, \delta\bar{u}} \mathbb{E}\left[\int_{t_0}^{t_h} J_{\mathcal{S}}(\delta\mathbf{x}(t), \delta\bar{\mathbf{u}})dt + J_{\mathcal{S}_f}(\delta\mathbf{x}(t_h))\right] \tag{5.10}$$

$$\text{s.t. } d\delta\mathbf{x} = \Delta f dt + \Delta B dt + g(\mathbf{x}, \bar{\mathbf{u}})dw \tag{5.11}$$

$$\mathbf{x} \in \mathcal{X}_{\mathcal{S}_{\text{safe}}}, \forall t \in [t_0, t_h],\ \mathbf{x}(t_h) \in \mathcal{X}_{\mathcal{S}_f} \tag{5.12}$$

$$\mathbb{E}(\mathbf{x}(t_0)) = \mu_{\mathbf{x}_0},\ \bar{\mathbf{u}} \in \mathcal{U}, \tag{5.13}$$

where $\Delta f(\mathbf{x}, \bar{\mathbf{x}}_{\text{des}}) = f(\mathbf{x}) - f(\bar{\mathbf{x}}_{\text{des}})$, $\Delta B(\mathbf{x}, \bar{\mathbf{u}}, \bar{\mathbf{x}}_{\text{des}}, \bar{\mathbf{u}}_{\text{des}}) = B(\mathbf{x})\bar{\mathbf{u}} - B(\bar{\mathbf{x}}_{\text{des}})\bar{\mathbf{u}}_{\text{des}}$, $\mathbf{x}(t_h)$ is the terminal state, and $\bar{\mathbf{x}}_0$ is an estimate of the system's state at t_0. We

solve Problem 10 at each time $t_{\mathcal{S}} = t_0 + k\Delta t$ for the horizon $[t_{\mathcal{S}}, t_h + k\Delta t]$, where Δt is the sampling time interval and $k \in \mathbb{Z}^+$ is the time step respectively. The cost functional $J_{\mathcal{S}}$ in (5.10) is defined as follows:

$$J_{\mathcal{S}}(\delta\mathbf{x}, \delta\bar{\mathbf{u}}) = \delta\mathbf{x}^\top Q \delta\mathbf{x} + \delta\bar{\mathbf{u}}^\top R \delta\bar{\mathbf{u}}, \tag{5.14}$$

where Q and R are positive definite matrices. The safe set $\mathcal{X}_{\mathcal{S}_{\text{safe}}}$ in (5.12) is defined using joint linear chance constraints:

$$\mathcal{X}_{\mathcal{S}_{\text{safe}}} = \{\mathbf{x} | \Pr(\wedge_i a_i^\top \mathbf{x} + b_i \leq 0) \geq 1 - \epsilon_i \; \forall t \in [t_0, t_f]\}. \tag{5.15}$$

The terminal constraint set $\mathcal{X}_{\mathcal{S}_f}$ in (5.12) is defined using a quadratic chance constraint as follows:

$$\mathcal{X}_{\mathcal{S}_f} = \{\mathbf{x} \mid \Pr(\delta\mathbf{x}_{t_h}^\top Q_{\mathcal{X}_f} \delta\mathbf{x}_{t_h} \leq c_f) \geq 1 - \epsilon_f\}, \tag{5.16}$$

where $\delta\mathbf{x}_{t_h} = \mathbf{x}(t_h) - \bar{\mathbf{x}}_{\text{des}}(t_h)$ and $Q_{\mathcal{X}_f}$ is a positive definite matrix. The matrix $Q_{\mathcal{X}_f}$ and the bound c_f in the quadratic constraint (5.16) are designed to be a sub-level set of the positive control invariant and reachable set of the dynamics (5.11) in Problem 10. Using a reachable set as the terminal constraint guarantees the feasibility of Problem 10 at each sample time $t_{\mathcal{S}} \in [t_0, t_f]$. The terminal cost $J_{\mathcal{S}_f}$ enables tracking of the desired trajectory by ensuring the stability of SMPC (Problem 10), as discussed in Section 5.2.2.

5.2.2 Convergence and Stability

The control problem is to track the desired trajectory $(\bar{\mathbf{x}}_{\text{des}}, \bar{\mathbf{u}}_{\text{des}})$, i.e., $\lim_{t\to\infty} \mathbb{E}(\|\mathbf{x} - \bar{\mathbf{x}}_{\text{des}}\|_2^2) \leq c_t$, while ensuring that $\mathbf{x} \in \mathcal{X}_{\mathcal{S}_{\text{safe}}}$, where c_t is an upper bound on the tracking error. The finite-horizon stochastic model predictive closed-loop system might be unstable. To guarantee tracking of the desired trajectory and the stability of the system (5.11), the terminal cost function $J_{\mathcal{S}_f}$ should represent the truncated cost of the infinite-horizon optimal control problem. An approach to achieve stability (see [145, 146]) in a deterministic nonlinear model predictive control is to have a control Lyapunov function as the terminal cost. We use a Stochastic Control Lyapunov Function (SCLF) [50, 147], as the terminal cost for guaranteeing the stability of the SDE system (5.11). This is an extension of the approach discussed in [82] and [83] for discrete-time Markov decision process. We make the following assumptions to ensure feasibility and constraint satisfaction of the Problem 10 for studying the convergence and stability of the closed-loop system.

Assumption 4. The SDE in (5.11) is ϵ_c- controllable, as stated in Definition 1, to the terminal state $\delta\mathbf{x} = 0$.

Assumption 5. The Problem 10 is initialized at the state $\mathbf{x}_0$ such that $\mathbf{x}_0 - \bar{\mathbf{x}}_{\text{des}}(t_0) \in \mathcal{X}_{ci} \subseteq \mathcal{X}$, where $\mathcal{X}_{ci}$ is a stochastic control invariant set of the error dynamics (5.11) and $0 \in \mathcal{X}_{ci}$. A detailed discussion on the set-invariance of a controlled stochastic system can be found in [81, 148, 149].

Assumption 6. The constraint satisfaction at each sampling time, $t_{\mathcal{S}}$ is achieved by using the constraint tightening approach. In the constraint tightening approach, as described in [80, 150], an inner-level optimization problem is used to compute an optimal risk measure ϵ that leads to a feasible Problem 10.

The cost functional $J_{\mathcal{S}}$ is a function of the quadratic polynomials in the error state $\delta\mathbf{x}$ and control $\delta\bar{\mathbf{u}}$ and satisfies the lower bounded as discussed in Remark 9.

Remark 9. The cost functional $J_{\mathcal{S}}$ is convex in $[\delta\mathbf{x}, \delta\bar{\mathbf{u}}]$, and is lower bounded at any time $t \in [t_0, t_f]$ as follows:

$$J_{\mathcal{S}} \geq \min\{\lambda_{\min}(Q), \lambda_{\min}(R)\}(\|\delta\mathbf{x}\|_2 + \|\delta\bar{\mathbf{u}}\|_2) \; \forall \, t, \tag{5.17}$$

where $\lambda_{\min}(Q)$ and $\lambda_{\min}(R)$ are the minimum eigenvalues of the positive definite matrices Q and R respectively.

In the following Assumption 7, we introduce the properties of the terminal cost $J_{\mathcal{S}_f}$ that guarantee the convergence of the cost and stability of the closed-loop system. Along with a SCLF-based terminal cost, we use a terminal constraint that is a subset of the control invariant set of the SDE (5.11) as discussed below.

Assumption 7. The terminal cost $J_{\mathcal{S}_f}$ from (5.10) and the terminal constraint set $\mathcal{X}_{\mathcal{S}_f}$ satisfy the following conditions.

(A1) $\mathcal{X}_{\mathcal{S}_f} \subseteq \mathcal{X}_{ci} \subset \mathcal{X}_{\text{safe}}$, where $\mathcal{X}_{ci}$ is defined in Assumption 5 and $0 \in \mathcal{X}_{\mathcal{S}_f}$

(A2) The terminal cost $J_{\mathcal{S}_f}(\delta\mathbf{x})$ is of the following form:

$$J_{\mathcal{S}_f} = \gamma\delta\mathbf{x}^\top M(\mathbf{x}, t)\delta\mathbf{x}, \; \gamma > 0, \tag{5.18}$$

is a contraction metric (see [28, 88]) and is uniform bounded as follows:

$$c_{l_f}\|\delta\mathbf{x}\|^2 \leq J_{\mathcal{S}_f}(\delta\mathbf{x}) \leq c_{u_f}\|\delta\mathbf{x}\|^2, \tag{5.19}$$

where $c_{l_f} > 0$ and $c_{u_f} > 0$, and satisfies the following inequality:

$$M(\mathbf{x},t)\frac{\partial F}{\partial \mathbf{x}} + \left(\frac{\partial F}{\partial \mathbf{x}}\right)^{\top} M(\mathbf{x},t) + \frac{d}{dt}M(\mathbf{x},t) \leq -2\alpha M(\mathbf{x},t) \tag{5.20}$$

where $F = \Delta f + \Delta B$.

(A3) We assume that $\mathrm{tr}(g\nabla_{xx}J_{\mathcal{S}_f}g^{\top}) \leq c_v \; \forall (\mathbf{x},\bar{\mathbf{u}}) \in \mathcal{X} \times \mathcal{U}$.

(A4) The terminal cost $J_{\mathcal{S}_f}$ satisfies the following inequality:

$$\mathcal{L}J_{\mathcal{S}_f} \leq -2\gamma\alpha\delta x M(x,t)\delta x + c_v. \tag{5.21}$$

If we choose γ such that $2\alpha\gamma\delta\mathbf{x}M(\mathbf{x},t)\delta\mathbf{x} \geq J_{\mathcal{S}}$, then

$$\mathcal{L}J_{\mathcal{S}_f}(\delta\mathbf{x}) + J_{\mathcal{S}}(\delta\mathbf{x},\delta\bar{\mathbf{u}}) \leq c_v, \tag{5.22}$$

where the operator $\mathcal{L}$ is defined as $\mathcal{L}(J_{\mathcal{S}_f}) = \partial_t J_{\mathcal{S}_f} + \nabla_x J_{\mathcal{S}_f}^{\top}\Delta f + \nabla_x J_{\mathcal{S}_f}^{\top}\Delta B + \frac{1}{2}\mathrm{tr}(\nabla_{xx}J_{\mathcal{S}_f}g(\mathbf{x},\bar{\mathbf{u}})g^{\top}(\mathbf{x},\bar{\mathbf{u}}))$, $J_{\mathcal{S}_f}(\delta\mathbf{x})$ is the terminal state cost, $J_{\mathcal{S}}(\delta\mathbf{x},\delta\bar{\mathbf{u}})$ is defined in (5.10), and c_v is defined in the Assumption $(A3)$.

(A5) For the SMPC feedback control $\bar{\mathbf{u}}_{\mathcal{S}}$, the cost $J_{\mathcal{S}}$ is uniformly bounded as follows:

$$c_l\|\delta\mathbf{x}_{\mathcal{S}}\|^2 \leq J_{\mathcal{S}}(\delta\mathbf{x}_{\mathcal{S}},\delta\bar{\mathbf{u}}_{\mathcal{S}}) \leq c_u\|\delta\mathbf{x}_{\mathcal{S}}\|^2. \tag{5.23}$$

where $c_l > 0$ and $c_u > 0$. The coefficients c_l and c_u are a function of the eigenvalues of the positive definite matrices Q and R.

In Theorem 4, we prove that the optimal cost $J^*_{\mathcal{S}_k}$ decreases with the time step k, i.e., $J^*_{\mathcal{S}_{k+1}} \leq J^*_{\mathcal{S}_k}$, provided that Problem 10 satisfies the Assumptions 4, 5, 6, and 7. We show that a decreasing optimal cost implies that the SMPC is a stabilizing controller.

Theorem 4. *Suppose that Problem 10 satisfies Assumptions 4, 5, 6, and 7, the terminal cost $J_{\mathcal{S}_f}$ is a control contraction metric as defined in [147], and $J_{\mathcal{S}_f}$, $J_{\mathcal{S}}$ satisfy the inequality (5.22). Then,*

$$J^*_{\mathcal{S}_{k+1}} \leq J^*_{\mathcal{S}_k} + c_v, \tag{5.24}$$

$$J^*_{\mathcal{S}_k} = \mathbb{E}\left(\int_{t_0+k\Delta t}^{t_h+k\Delta t} J_{\mathcal{S}}(\delta\mathbf{x}^*,\delta\bar{\mathbf{u}}^*)dt\right) + \mathbb{E}(J_{\mathcal{S}_f}(\delta\mathbf{x}^*(t_h+k\Delta t))), \tag{5.25}$$

where $J^*_{\mathcal{S}_k}$ is the optimal cost and $(\delta\mathbf{x}^*, \delta\bar{\mathbf{u}}^*)$ is the optimal feasible trajectory of Problem 10 for the time horizon $[t_0 + k\Delta t, t_h + k\Delta t]$, Δt and k are the sampling time interval and time step respectively. Furthermore, the control $\bar{\mathbf{u}}_{\mathcal{S}}$ computed using the SMPC (Problem 10) is an exponential stabilizing control for the error dynamics in (5.11).

Proof. We first prove that the optimal cost of the finite-horizon SMPC (Problem 10) decreases with the time step k and then show that the decreasing optimal cost implies stability of the closed-loop system. Without loss of generality, we prove the inequality (5.24) for $k = 0$. This result can be extended to any k by moving the time horizon of Problem 10.

(a) Decreasing Cost: Let the optimal trajectory computed using the SMPC at the sampling time $t_0 = 0$ and $t_0 = \Delta t$ be $(\delta\mathbf{x}^*_0, \delta\bar{\mathbf{u}}^*_0), \forall t \in [0, t_h]$ and $(\delta\mathbf{x}^*_1, \delta\bar{\mathbf{u}}^*_1), \forall t \in [\Delta t, t_h + \Delta t]$, respectively. Assume that the trajectory $(\delta\mathbf{x}^*_0, \delta\bar{\mathbf{u}}^*_0), \forall t \in [0, t_h]$ is appended with the trajectory $(\delta\mathbf{x}^*_{c_0}, \delta\bar{\mathbf{u}}^*_{c_0}), \forall t \in [t_h, t_h + \Delta t]$, that is computed using an asymptotically stable controller satisfying the Assumption 7. The optimal costs for $k = 0$ and $k = 1$ are defined as $J^*_{\mathcal{S}_0}(\delta\mathbf{x}^*_0, \delta\bar{\mathbf{u}}^*_0)$ and $J^*_{\mathcal{S}_1}(\delta\mathbf{x}^*_1, \delta\bar{\mathbf{u}}^*_1)$, respectively. Using (5.25), we can show the following equality:

$$\begin{aligned} J_{\mathcal{S}_1}(\delta\mathbf{x}^*_0, \delta\bar{\mathbf{u}}^*_0) = J^*_{\mathcal{S}_0}(\delta\mathbf{x}^*_0, \delta\bar{\mathbf{u}}^*_0) - \mathbb{E}\left(\int_0^{\Delta t} J_{\mathcal{S}}(\delta\mathbf{x}^*_0, \delta\bar{\mathbf{u}}^*_0)dt\right) \\ + \mathbb{E}\left(\int_{t_h}^{t_h+\Delta t} J_{\mathcal{S}}(\delta\mathbf{x}^*_{c_0}, \delta\bar{\mathbf{u}}^*_{c_0})dt\right) \\ + \mathbb{E}(J_{\mathcal{S}_f}(\delta\mathbf{x}^*_{c_0}(t_h + \Delta t))) - \mathbb{E}(J_{\mathcal{S}_f}(\delta\mathbf{x}^*_{c_0}(t_h))). \end{aligned} \tag{5.26}$$

Since the cost $J_{\mathcal{S}}$ is quadratic in state and control, the term $-\mathbb{E}(\int_0^{\Delta t} J_{\mathcal{S}}(\delta\mathbf{x}^*_0, \delta\bar{\mathbf{u}}^*_0)dt)$ is ≤ 0. Using this, the above equality reduces to the following inequality.

$$\begin{aligned} J_{\mathcal{S}_1}(\delta\mathbf{x}^*_0, \delta\bar{\mathbf{u}}^*_0) \leq J^*_{\mathcal{S}_0}(\delta\mathbf{x}^*_0, \delta\bar{\mathbf{u}}^*_0) \\ + \mathbb{E}\left(\int_{t_h}^{t_h+\Delta t} J_{\mathcal{S}}(\delta\mathbf{x}^*_{c_0}, \delta\bar{\mathbf{u}}^*_{c_0})dt\right) \\ + \mathbb{E}(J_{\mathcal{S}_f}(\delta\mathbf{x}^*_{c_0}(t_h + \Delta t))) - \mathbb{E}(J_{\mathcal{S}_f}(\delta\mathbf{x}^*_{c_0}(t_h))). \end{aligned}$$

By conditioning the expectation operation for a known $\delta\mathbf{x}^*_{c_0}(t_h)$, we get the follow-

ing inequality:

$$\begin{aligned} J_{\mathcal{S}_1}(\delta\mathbf{x}_0^*, \delta\bar{\mathbf{u}}_0^*) \leq\ & J_{\mathcal{S}_0}^*(\delta\mathbf{x}_0^*, \delta\bar{\mathbf{u}}_0^*) \\ & + \mathbb{E}\left(\int_{t_h}^{t_h+\Delta t} J_{\mathcal{S}}(\delta\mathbf{x}_{c_0}^*, \delta\bar{\mathbf{u}}_{c_0}^*)dt\right) \\ & + \mathbb{E}(J_{\mathcal{S}_f}(\delta\mathbf{x}_{c_0}^*(t_h+\Delta t))) - J_{\mathcal{S}_f}(\delta\mathbf{x}_{c_0}^*(t_h)). \end{aligned} \tag{5.27}$$

By Dynkin's formula (see [49, 151]), we have the following equality:

$$\begin{aligned} &\mathbb{E}\left(J_{\mathcal{S}_f}(\delta\mathbf{x}_{c_0}^*(t_h+\Delta t))) - J_{\mathcal{S}_f}(\delta\mathbf{x}_{c_0}^*(t_h)\right) \\ &= \mathbb{E}\left(\int_{t_h}^{t_h+\Delta t} \mathcal{L}J_{\mathcal{S}_f}(\delta\mathbf{x}_{c_0}^*)dt\right). \end{aligned} \tag{5.28}$$

Using the Dynkin's equality (5.28) in the inequality (5.27), we have the following inequality:

$$\begin{aligned} & J_{\mathcal{S}_1}(\delta\mathbf{x}_0^*, \delta\bar{\mathbf{u}}_0^*) \leq J_{\mathcal{S}_0}^*(\delta\mathbf{x}_0^*, \delta\bar{\mathbf{u}}_0^*) \\ & + \mathbb{E}\left(\int_{t_h}^{t_h+\Delta t} \left(J_{\mathcal{S}}(\delta\mathbf{x}_{c_0}^*, \delta\bar{\mathbf{u}}_{c_0}^*) + \mathcal{L}J_{\mathcal{S}_f}(\delta\mathbf{x}_{c_0}^*)\right) dt.\right) \end{aligned} \tag{5.29}$$

Using (5.22) in (5.29), we have

$$J_{\mathcal{S}_1}(\delta\mathbf{x}_0^*, \delta\bar{\mathbf{u}}_0^*) \leq J_{\mathcal{S}_0}^*(\delta\mathbf{x}_0^*, \delta\bar{\mathbf{u}}_0^*) + c_v. \tag{5.30}$$

The trajectory $(\delta\mathbf{x}_0^*, \delta\bar{\mathbf{u}}_0^*)$ is a sub-optimal trajectory for the horizon $[\Delta t, t_h + \Delta t]$. Therefore, $J_{\mathcal{S}_1}^* \leq J_{\mathcal{S}_0}^* + c_v$. Since we use linear operations in proving $J_{\mathcal{S}_1}^* \leq J_{\mathcal{S}_0}^* + c_v$, we can extend the inequality to any $k \in \mathbb{Z}^+$ by simply moving the time horizon. Thus, we have $J_{\mathcal{S}_{k+1}}^* \leq J_{\mathcal{S}_k}^* + c_v$.

(b) Stability: The control $\delta\bar{\mathbf{u}}_{\mathcal{S}}$ computed using Problem 10 satisfies the inequality in (5.22) as stated in Assumption 7. Using the uniform bounds in (5.19) and (5.23), the SCLF inequality is simplified as follows for the feedback control $\bar{\mathbf{u}}_{\mathcal{S}}$:

$$\mathcal{L}J_{\mathcal{S}_f}(\delta\mathbf{x}_{\mathcal{S}}) \leq -\frac{c_l}{c_{u_f}} J_{\mathcal{S}_f}(\delta\mathbf{x}_{\mathcal{S}}) + c_v. \tag{5.31}$$

Applying the Dynkin's formula, we have the following bound.

$$\mathbb{E}(\|\delta\mathbf{x}_{\mathcal{S}}(t)\|^2) \leq \frac{c_{u_f}}{c_{l_f}} \mathbb{E}(\|\delta\mathbf{x}_{\mathcal{S}}(0)\|^2) e^{-\frac{c_l}{c_{u_f}}t} + \frac{c_v c_{u_f}}{c_{l_f} c_l} \tag{5.32}$$

As $t \to \infty$, the expectation of the tracking error is bounded by $\frac{c_v c_{u_f}}{c_{l_f} c_l}$. Therefore, the closed-loop system is exponentially stable, assuming that Assumptions 4, 5, 6, and 7 are satisfied when formulating Problem 10. □

Note that the convergence and stability of the SMPC controller depend on the choice of the terminal cost $J_{\mathcal{S}_f}$. The cost functional $J_{\mathcal{S}_f}$ and the metric $M(\mathbf{x}, t)$ defined in Assumption 7 are designed by using the control contraction metric discussed in [147]. In the following section, we discuss the implementation of the SMPC controller using gPC-SCP.

5.2.3 SMPC using gPC-SCP

We formulate the SMPC (Problem 10) such that recursive feasibility, constraint satisfaction, and stability is guaranteed, as discussed in Section 5.2.2. We then solve the SMPC problem by using the gPC approach and the gPC-SCP problem. In the following Algorithm 3, we discuss the stochastic model predictive control algorithm for tracking a given desired trajectory $(\bar{\mathbf{x}}_{\text{des}}, \bar{\mathbf{u}}_{\text{des}})$. We generate the desired trajectory for the nominal trajectory using a deterministic motion planner.

Algorithm 3: Stochastic Model Predictive Control.

Input: Obstacle location, $\mathbf{x}_0$, $\mathcal{X}_f$, Δt, T,
Input: Desired trajectory $(\bar{\mathbf{x}}_{\text{des}}, \bar{\mathbf{u}}_{\text{des}})$, uncertainty model of $g(\mathbf{x}, \bar{\mathbf{u}})$ in SDE (4.7).
Output: Safe control input $\mathcal{U}_{\text{sol}} = \{\bar{\mathbf{u}}_0, \bar{\mathbf{u}}_1, ..., \bar{\mathbf{u}}_{T-1}\}$ to track $(\bar{\mathbf{x}}_{\text{des}}, \bar{\mathbf{u}}_{\text{des}})$.

Compute Problem 9 for Problem 10, as discussed in Algorithm 2.
while *The terminal set $\mathcal{X}_f$ not reached* **do**
    $\mathcal{U}_{\text{sol}} \leftarrow$ **SCP** (Problem 9,$(\bar{\mathbf{x}}_{\text{des}}, \bar{\mathbf{u}}_{\text{des}}), T, \Delta t, x_0)$
    Apply $\bar{\mathbf{u}}_0$ to the system (4.7)
    Update $\mathbf{x}_0$ using sensor information
    Update $(\bar{\mathbf{x}}_{\text{des}}, \bar{\mathbf{u}}_{\text{des}})$ based on Δt

In the offline stage, we compute the projected gPC-SCP problem for Problem 10. Given a desired trajectory $(\bar{\mathbf{x}}_{\text{des}}, \bar{\mathbf{u}}_{\text{des}})$, time interval Δt and the number of time steps T, we solve the gPC-SCP at each time step k and apply the control $\bar{\mathbf{u}}(t_0 + k\Delta t)$ to the system. We apply the control $\bar{\mathbf{u}}(t_0 + k\Delta t)$ to the system, until the terminal set $\mathcal{X}_f$ is reached. Note that the terminal set $\mathcal{X}_f$ is large large enough to accommodated for the tracking error bound discussed in Theorem 4. In the following Section 5.3, we discuss the implementation of Algorithms 2 and 3 on a three degree-of-freedom robot.

5.3 Simulations and Experiments on Spacecraft Dynamics Simulator

We apply the gPC-SCP method in Algorithm 2 to design safe and optimal motion plans, and the SMPC Algorithm 3 to track a nominal unsafe trajectory for the three

degree-of-freedom robotic spacecraft simulator dynamics [30]. For the spacecraft simulator dynamics, we conduct an empirical study via simulation to demonstrate the safety provided by Algorithms 2 and 3 in comparison to the Gaussian collision constraint [21, 22, 25, 52, 74]. We ran the simulations on an Ubuntu machine with the configuration: 7^{th} generation Intel Core i7 process, and 16 GB RAM. We solve the SCP problem using CVXpy [152] and ECOS [153] solver. We then validate the experimental results on the spacecraft simulator hardware platform, where the motion planning and control is computed on a NVIDIA Jetson TX2 Computer.

5.3.1 Robotic Spacecraft Dynamics Simulator [30]

The Caltech's M-STAR (Multi-Spacecraft Testbed for Autonomy Research) is shown in Fig. 5.3. The testbed floats on an ultra-precise epoxy floor using linear air bearings to achieve 3DOF friction-less motion. The M-STAR is equipped with eight thrusters for position (x, y) and yaw angle (θ) control. The dynamics of the robot is given as follows:

$$d\mathbf{x} = f(\mathbf{x}, \bar{\mathbf{u}})dt + \sigma g(\mathbf{x}, \bar{\mathbf{u}})dw, \tag{5.33}$$

where $\mathbf{x} = [x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}]^\top$, $\bar{\mathbf{u}} \in \mathbb{R}^8$, $\sigma \in \mathbb{R}$. The functions $f(\mathbf{x}, \bar{\mathbf{u}})$ and $g(\mathbf{x}, \bar{\mathbf{u}})$ are given below:

$$f(\mathbf{x}, \bar{\mathbf{u}}) = \begin{bmatrix} \mathbb{I}_{3\times 3} & 0 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ B(m, \mathrm{I}, l, \theta)\bar{\mathbf{u}} \end{bmatrix}, \tag{5.34}$$

$$g(\mathbf{x}, \bar{\mathbf{u}}) = \mathrm{blkdiag}\{0, B(m, I, l, \theta)\bar{\mathbf{u}}\}. \tag{5.35}$$

The control effort $\bar{\mathbf{u}}$ is constrained to be $0 \leq \bar{\mathbf{u}} \leq 0.45\,\mathrm{N}$, and $B(m, \mathrm{I}, l, \theta) \in \mathbb{R}^{3\times 8}$ is the control allocation matrix (see Chapter 7), where $m = 10\,\mathrm{kg}$ and $\mathrm{I} = 1.62\,\mathrm{kg\,m\,s^{-2}}$ are the mass and the inertia matrix, and $l = 0.4\,\mathrm{m}$ is the moment arm. The uncertainty $\sigma g(\mathbf{x}, \bar{\mathbf{u}})$ stems due to viscous friction between the robot and the flat floor, drift due to gravity gradient, and uncertainty in thruster actuation. We choose $\sigma = 0.1$, this value encompasses all the above forms of uncertainty. With this model, we study the convergence, collision checking, and trajectory tracking discussed in Theorems 1, 2, 3, and 4.

5.3.2 Simulation

Consider the map shown in Fig. 5.4. We design a safe and optimal trajectory, $J = \|\bar{\mathbf{u}}\|_2$, from the initial state $\mathbb{E}(\mathbf{x}_0) = 0$ to the terminal state $\mathbb{E}(\mathbf{x}_f) = [0.3, 2.3, 0, 0, 0, 0]^\top$, while avoiding the obstacle located at $\mathbf{p}_{\mathrm{obs}} = [0.3, 1, 0, 0, 0, 0]^\top$ with radius $r_{\mathrm{safe}} =$

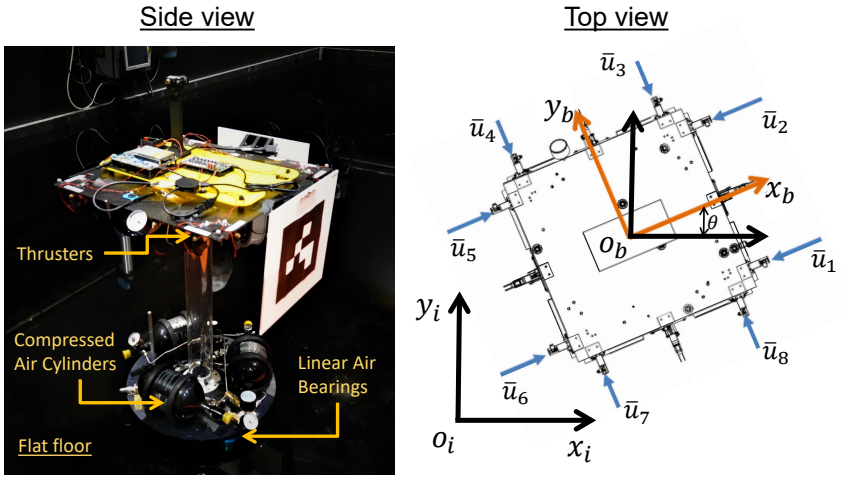


Figure 5.3: The top and side view of the Caltech's robotic spacecraft dynamics simulator.

$0.5\,\mathrm{m}$. We formulate the collision constraint using Theorems 2 and 3 and bound the terminal variance using a slack variable to ensure feasibility. In Fig. 5.4, we

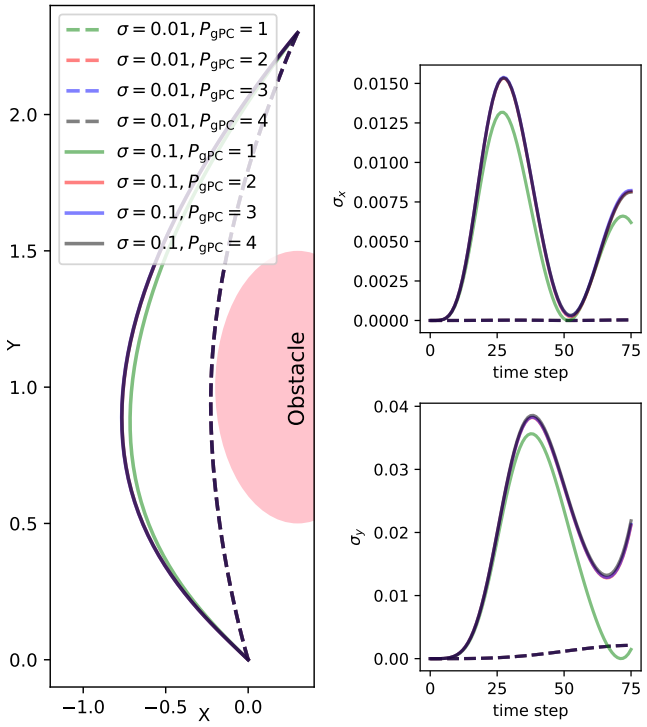


Figure 5.4: The figure demonstrates convergence of the mean and the variance (σ_x, σ_y) of the states (x, y) with increasing P_{gPC} for $\sigma = \{0.01, 0.1\}$.

show the mean and variance of the position of M-STAR, (x, y), computed using Algorithm 2. We compare the mean and variance computed for gPC polynomial

degree $P_{\text{gPC}} = \{1, 2, 3, 4\}$ and variance $\sigma = \{0.01, 0.1\}$ with $g(\mathbf{x}, \bar{\mathbf{u}}) = [0, B\bar{\mathbf{u}}]^\top$. The convergence of mean and variance with increasing P_{gPC}, implies convergence with respect to ℓ, validating Theorem 1. Since, there are no known methods to compute global optimal solution for Problem 6, we cannot comment on the sub-optimality of the solution. For the case with low variance $\sigma = 0.01$, we observe that gPC polynomial with degree $P_{\text{gPC}} = 1$ are sufficient for computing the mean and the variance accurately. While for the large variance $\sigma = 0.1$, we need gPC polynomials with degree $P_{\text{gPC}} = 2$. We can use $P_{\text{gPC}} = 1$ with large variance in dynamics for the following two case: 1) short-horizon planning, and 2) iterative planning with closed-loop state information updates. We use gPC polynomials with degree $P_{\text{gPC}} = 2$ in the following analysis for motion planning and $P_{\text{gPC}} = 1$ for SMPC-based tracking control.

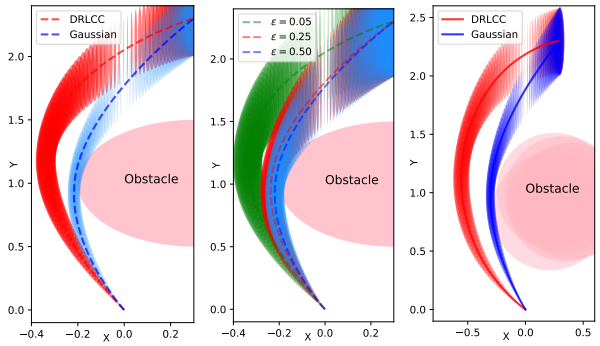


Figure 5.5: Left: We compare the probabilistic safe trajectories computed using distributionally robust and Gaussian collision chance constraint. Center: We compare the trajectories for various risks $(\epsilon = 0.05, 0.25, 0.5)$ of collision constraint violation. Right: We demonstrate collision checking under uncertainty in both robot dynamics and obstacle location.

Motion Planning

In Fig. 5.5, we show the mean and 2-σ confidence level of the trajectories computed using distributionally robust collision checking and Gaussian confidence-based collision checking for a risk measure $\epsilon = 0.05$. We observe that increasing the risk of collision (ϵ) in planning formulation from 0.05 to 0.5 reduces the safety in the trajectories. As shown in Fig. 5.5, that the trajectories generated using DRLCC are safer for both deterministic and stochastic obstacle ($\Sigma_p = 1e - 4$), in comparison to Gaussian constraint.

We validate the safety of the motion plans computed using gPC-SCP by tracking a sampled trajectory with the exponentially stable controller designed in [30] for the

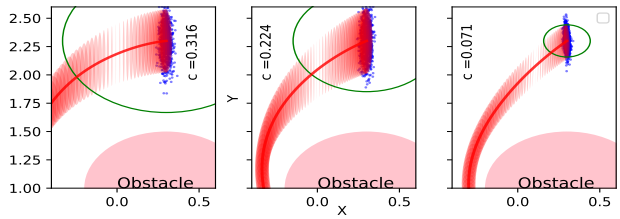


Figure 5.6: We compare the trajectories generated for different sizes $c = \{0.316, 0.224, 0.071\}$ (shown as the green circle) of the terminal set. We show the terminal state of the robot (blue), when a nominal trajectory (sampled from the probabilistic trajectory) is executed using an exponentially stable controller.

nominal dynamics. We sample a trajectory $\mathbf{x}$ by using the projection equation $\mathbf{x} = \Phi^T(\xi)X$. The gPC-SCP algorithm computes the gPC coordinates X, we compute $\mathbf{x}$ by randomly sampling the multivariate normal distribution ξ. Using the motion plans shown in Fig. 5.5 as an input to the controller, we get the following number of collisions over 1000 trials. The number of collisions with DRLCC constraint

	DRLCC	Gaussian
Deterministic Obstacle	2	66
Stochastic Obstacle ($\Sigma_p = 1e-4$)	5	182

Table 5.1: Number of collisions over 1000 trails.

for the risk measure $\epsilon = \{0.05, 0.25, 0.5\}$ are $\{0, 23, 181\}$, respectively. The goal reaching of the robot for various terminal variance size is shown in Fig. 5.6. For the terminal variance with $c = 0.071$, the robot violates the constraint 49 times over 1000 trials. Note that, although the DRLCC constraint performs better than the Gaussian constraint for nonlinear dynamical system, it reduces the feasible space of the optimization problem. For a given dynamical system and obstacle map, a trade-off analysis between distributional robustness and risk measure ϵ is required to ensure that the feasible space is non-empty.

Tracking Control using Stochastic Model Predictive Control

We apply the SMPC method described in Section 5.2 to track a trajectory designed using nominal dynamics $d\mathbf{x} = f(\mathbf{x}, \bar{\mathbf{u}})dt$ with deterministic obstacles for a fixed horizon of t_f =25 s and Δt =0.25 s. Since, the uncertainty in the dynamics and the obstacle location is not considered in the design of the nominal trajectory, it could be unsafe during operation. Algorithm 3 will enable safe operation, provided that Assumption 7 are satisfied. We use the Lyapunov function defined in [30] as

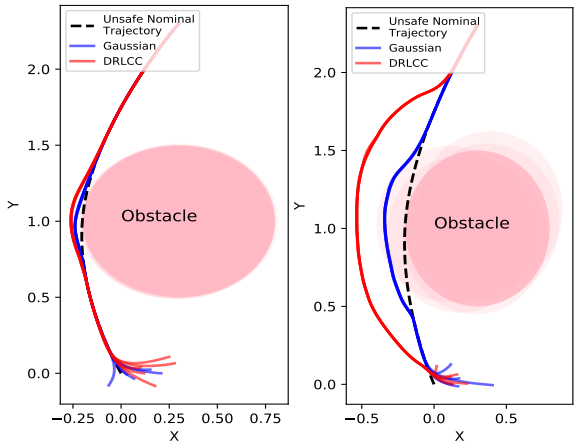


Figure 5.7: We show trajectory tracking and safety under uncertainty in dynamics and obstacle location in real-time over 5 trials by using the proposed stochastic model predictive controller. We compare the distributionally robust (DRLCC) collision constraint with the Gaussian collision constraint. Left: In the case with small uncertainty ($\Sigma_p = 1e-4$), both constraints perform safe tracking. Right: With large uncertainty ($\Sigma_p = 1e-2$), the DRLCC provides safety in all the trials, while Gaussian collision constraint fails in 2 trials.

the terminal cost $J_{\mathcal{S}_f}$, that satisfies Assumption 7 provided that the cost-to-go is defined as $Q = \mathbb{I}$ and $R = \mathbb{I}$. We formulate the terminal constraint set as discussed in Section 5.1.3. The terminal variance is bounded using a slack variable for feasibility.

In Fig. 5.7, we compare the tracking performance with $\sigma = 0.1$ and uncertainty in obstacle position $\Sigma_p = \{0.01, 0.1\}$. We observe that for $\Sigma_p = 0.01$, both the proposed distributionally robust approach and Gaussian collision checking are safe for $t_h = 5\,\mathrm{s}$. In the case with $\Sigma_p = 0.1$, the Gaussian approach has 2 failure over 5 trials for time horizon $t_h = 5\,\mathrm{s}$. We observe that the appropriate choice of T in Problem 10 and the cost-to-go function depend on the terminal cost $J_{\mathcal{S}_f}$, the size of the invariant set around the nominal trajectory and the uncertainty in the system and the environment. If the uncertainty is large, then the time horizon needs to be large to ensure safety. While this validates Theorem 4, further research needs to be conducted towards construction of probabilistic invariant sets for nonlinear systems to apply SMPC method for non-Hamiltonian systems.

5.3.3 Experiments

We apply Algorithm 2 for the scenario shown in Fig. 1.2 using the closed-loop described in Fig. 5.8 to design and execute safe plans for SS-1 in Fig. 1.2 under

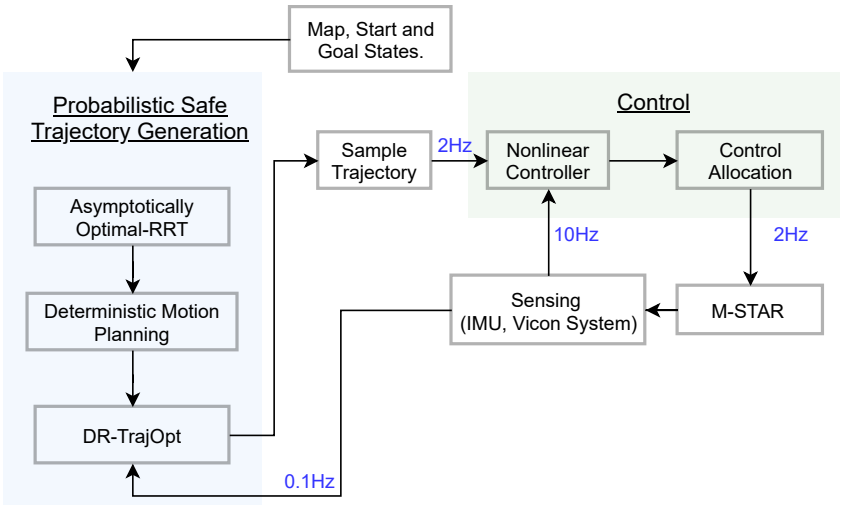


Figure 5.8: The guidance, navigation, and control loop used for planning a distributionally-robust safe trajectory using gPC-SCP and controlling the 3 DOF spacecraft simulators.

uncertainty in dynamics and obstacle location. This scenario is relevant to the low-earth orbit, on-orbit, servicing application discussed in [27]. Please see [30], for details on sensing module to estimate full-state, control and control allocation algorithm. We use the location of the obstacles SS-2, SS-3, SS-4, and Asteroid shown in Fig. 5.9, and the uncertainty in position of the obstacles $\Sigma_p = 1e-4$ as an input to Algorithm 2. The initial state and terminal state of SS-1 are $\mathbb{E}(\mathbf{x}_0) = [-0.9, -2.3, 0, 0, 0, 0]^\top$ and $\mathbb{E}(\mathbf{x}_f) = [0, 2.3, 0, 0, 0, 0]^\top$, respectively.

In Fig. 5.9, we present the results for 10-trials of the closed-loop tracking experiment. We compute an initial anytime trajectory using AO-RRT and optimize it for nominal dynamics. We use the optimized solution as an input for the gPC-SCP method. We observe that the method is biased towards the initial trajectory. As shown, the gPC-SCP method outputs a safe trajectory. We use the mean of the output trajectory as a reference trajectory for the controller. As shown in Fig. 5.9, the uncertainty in the model leads to drift in the system. The gPC-SCP method provides a safe trajectory for control by accommodating for the uncertainty in dynamics. We observe one failure out of the 10 trials of closed-loop tracking. The failure was because of a large disturbance torque on SS-1 due to a damaged floor. Out of the 10 trials, 7 closed-loop tracking trials reached the expected terminal set. This demonstrates the efficacy of the gPC-SCP method.

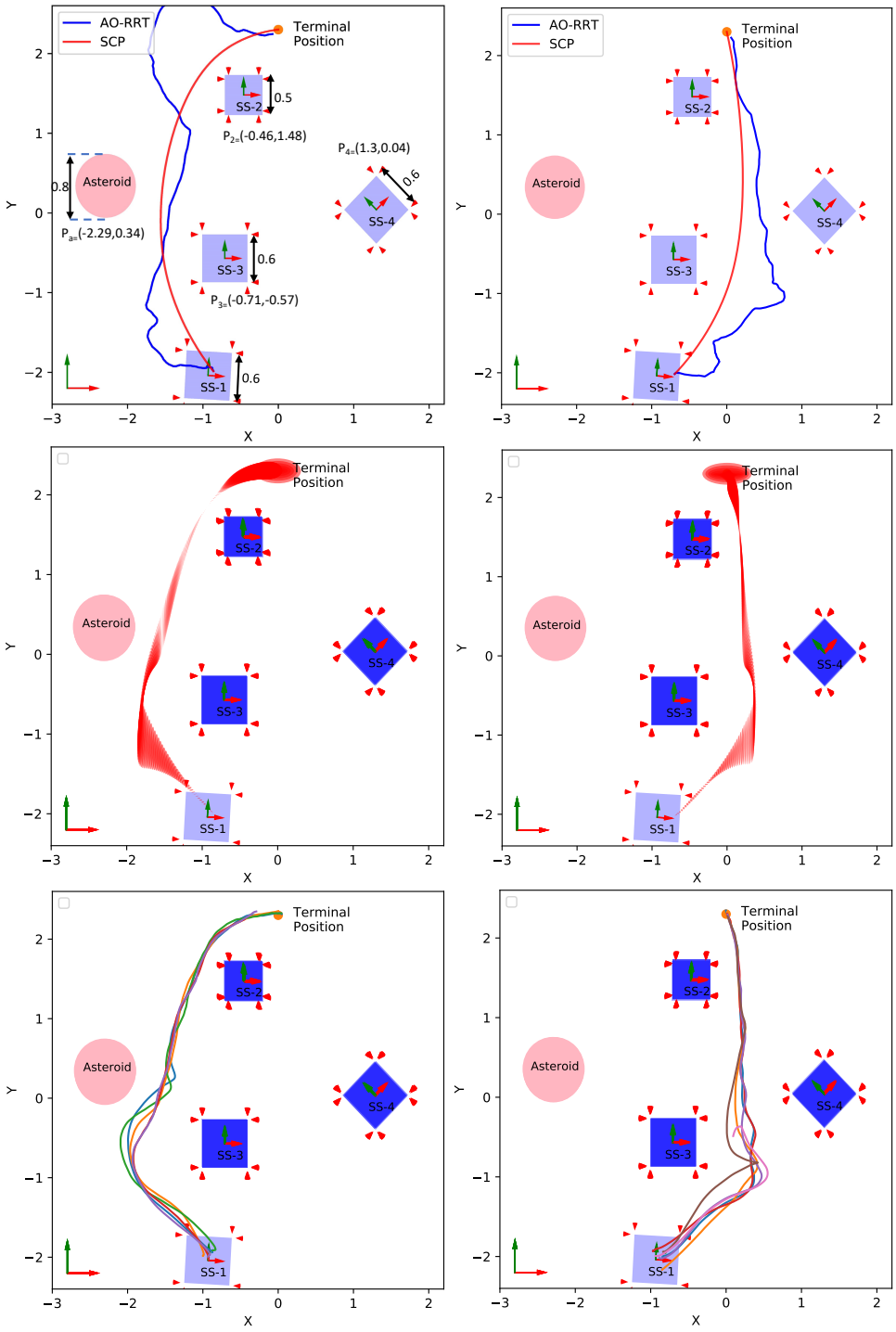


Figure 5.9: We show the output of the gPC-SCP method at each stage of Algorithm 2 and 10 trials of closed-loop trajectory tracking by using an exponentially stable feedback controller designed in [30]. Top: We show the output of AO-RRT for 5000 nodes and the SCP for the nominal dynamics. Middle: We show the probabilistic safe trajectory generated using the gPC-SCP method with a risk measure of $\epsilon = 0.05$ for collision checking. Bottom: We observe one failure in the 10 trials of the closed-loop trajectory execution.

5.4 Chapter Summary

We derive deterministic surrogate convex constraints for collision checking with deterministic and stochastic obstacles. Using these constraints, we integrate the gPC-SCP method with a sampling-based motion planning algorithm to compute safe motion plans under uncertainty in dynamics and obstacle location. We extend this method to design a stochastic model predictive control for safely tracking a nominal trajectory which is computed using a deterministic motion planning algorithm by ignoring the uncertainty. We prove the convergence and stability of the stochastic model predictive controller. We validate our approach in simulations and on the robotic spacecraft simulator hardware and demonstrate a higher success rate in ensuring the safety of motion trajectories compared to a Gaussian approximation of the chance constraints.

Chapter 6

SAFE EXPLORATION AND LEARNING

Robots deployed in the real world often operate in unknown or partially-known environments. We have incorporated the partial knowledge in dynamics and environment as stochastic uncertainty in the motion planning and control algorithms proposed in Chapters 4 and 5. These algorithms assume prior knowledge of the uncertainty model. Safe exploration is an efficient approach to collect ground truth data by safely interacting with the environment to build the uncertainty model in the guidance and control loop.

In this chapter, we present an episodic learning and control algorithm for safe exploration, as shown in Fig. 1.3, that integrates learning, stochastic trajectory planning, and rollout for active and safe data collection. *Rollout* is defined as executing the computed safe trajectory and policy using a stable feedback controller.

The planning problem is formulated as an Information-cost Stochastic Nonlinear Optimal Control (Info-SNOC) problem that maximizes exploration and minimizes the control effort. Safety constraints are formulated as chance constraints. The propagation of uncertainty in the dynamic model and chance constraints in Info-SNOC are addressed by projecting the problem to the generalized polynomial chaos (gPC) space, as derived in Chapter 4, and computing a distributionally robust convex approximation, as described in Chapter 5. By building on this work, we derive a sequential convex optimization solution to the Info-SNOC problem to plan a pool of sub-optimal safe and information-rich trajectories with the learned approximation of the dynamics. A sample of the trajectory pool is used as an input to the rollout stage to collect new data. To ensure real-time safety, the nonlinear feedback controller with a safety filter used in the rollout stage certifies bounded stochastic stability [54]. The new data is used to learn an improved dynamic model.

6.1 Problem Formulation

In this section, we discuss the preliminaries of the learning method used for modelling the unknown dynamics, the formulation of the Info-SNOC problem, and the deterministic surrogate projection using gPC.

6.1.1 Robust Regression For Learning

An exploration step in active data collection for learning dynamics is regarded as a covariate shift problem. Covariate shift is a special case of distribution shift between training and testing data distributions. In particular, we aim to learn the unknown part of the dynamics $g(\mathbf{x}, \bar{\mathbf{u}})$ from state $\mathbf{x}$ and control $\bar{\mathbf{u}}$ defined in Sec. 6.1.2. The covariate shift assumption indicates the conditional dynamics distribution given the states and controls remains the same while the input distribution of training ($\mathrm{Pr}_s(\mathbf{x}, \bar{\mathbf{u}})$) is different from the target input distribution ($\mathrm{Pr}_t(\mathbf{x}, \bar{\mathbf{u}})$). Robust regression is derived from a *min–max* adversarial estimation framework, where the estimator minimizes a loss function and the adversary maximizes the loss under statistical constraints. The resulting Gaussian distributions provided by this learning framework are given below. For more technical details, we refer the readers to [55, 154]. The output Gaussian distribution takes the form $\mathcal{N}(\mu_g, \Sigma_g)$ with mean μ_g and variance Σ_g:

$$\begin{aligned} \Sigma_g(\mathbf{x}, \bar{\mathbf{u}}, \theta_2) &= \left(2\tfrac{\mathrm{Pr}_s(\mathbf{x},\bar{\mathbf{u}})}{\mathrm{Pr}_t(\mathbf{x},\bar{\mathbf{u}})}\theta_2 + \Sigma_0^{-1}\right)^{-1}, \\ \mu_g(\mathbf{x}, \bar{\mathbf{u}}, \theta_1) &= \Sigma_g(\mathbf{x}, \bar{\mathbf{u}}, \theta_2)\left(-2\tfrac{\mathrm{Pr}_s(\mathbf{x},\bar{\mathbf{u}})}{\mathrm{Pr}_t(\mathbf{x},\bar{\mathbf{u}})}\theta_1\phi(\mathbf{x}, \bar{\mathbf{u}}) + \mu_0\Sigma_0^{-1}\right), \end{aligned} \tag{6.1}$$

where $\mathcal{N}(\mu_0, \Sigma_0)$ is a non-informative (i.e., large Σ_0) base distribution, and $\phi(\mathbf{x}, \bar{\mathbf{u}})$ is the feature function that is learned using neural networks from data. The model parameters θ_1 and θ_2 are selected by maximizing the target log likelihood. The density ratio $\frac{\mathrm{Pr}_s(\mathbf{x},\bar{\mathbf{u}})}{\mathrm{Pr}_t(\mathbf{x},\bar{\mathbf{u}})}$ is estimated from data beforehand. Robust regression can handle multivariate outputs with correlation efficiently by incorporating neural networks and predicting a multivariate Gaussian distribution, whereas traditional methods like Gaussian process regression suffer from high-dimensions and require heavy tuning of kernels [154].

6.1.2 Optimal and Safe Planning Problem

In this section, we present the finite-time chance-constrained stochastic optimal control problem formulation [20] used to design an information-rich trajectory. The optimization has control effort and terminal cost as performance objectives, and the safety is modelled as a joint chance constraint of a set $\mathcal{F}$ defined by a polytope or a

quadratic constraint. The full stochastic optimal control problem is as follows:

$$J^* = \min_{\mathbf{x}(t),\bar{\mathbf{u}}(t)} \mathbb{E}\left[\int_{t_0}^{t_f} J(\mathbf{x}(t),\bar{\mathbf{u}}(t))dt + J_f(\mathbf{x}(t),\bar{\mathbf{u}}(t))\right] \tag{6.2}$$

$$\text{s.t.} \quad \dot{\mathbf{x}}(t) = f(\mathbf{x}(t),\bar{\mathbf{u}}(t)) + \hat{g}(\mathbf{x}(t),\bar{\mathbf{u}}(t)) \tag{6.3}$$

$$\Pr(\mathbf{x}(t) \in \mathcal{F}) \geq 1-\epsilon, \quad \forall t \in [t_0, t_f] \tag{6.4}$$

$$\bar{\mathbf{u}}(t) \in \mathcal{U} \quad \forall t \in [t_0, t_f] \tag{6.5}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0 \quad \mathbb{E}(\mathbf{x}(t_f)) = \mu_{\mathbf{x}_f}, \tag{6.6}$$

where $\mathbf{x}(t) \in \mathcal{X} \subseteq \mathbb{R}^{d_\mathbf{x}}$ denotes the state of the dynamics, $\mathbf{x}_0$ and $\mathbf{x}_f$ are the initial and the terminal states respectively, the control $\bar{\mathbf{u}} \in \mathcal{U} \subseteq \mathbb{R}^{d_u}$ is deterministic, $\hat{g}$ is the learned probabilistic model, and $\mathbb{E}$ is the expectation operator. The modelling assumptions and the problem formulation will be elaborated in the following sections.

Dynamical Model

The $\hat{g}$ term of (6.3) is the estimated model of the unknown g term of the original dynamics:

$$\dot{\bar{\mathbf{p}}} = f(\bar{\mathbf{p}},\bar{\mathbf{u}}) + \underbrace{g(\bar{\mathbf{p}},\bar{\mathbf{u}})}_{\text{unknown}}, \tag{6.7}$$

where the state $\bar{\mathbf{p}} \in \mathcal{X}$ is now considered deterministic, and the functions $f : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^{d_\mathbf{x}}$ and $g : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^{d_\mathbf{x}}$ are Lipschitz with respect to $\bar{\mathbf{p}}$ and $\bar{\mathbf{u}}$. The control set $\mathcal{U}$ is convex and compact.

Remark 10. The maximum entropy distribution with the known mean μ_g and covariance matrix Σ_g of the random variable $\hat{g}$ is the Gaussian distribution $\mathcal{N}(\mu_g, \Sigma_g)$.

The learning algorithm computes the mean vector $\mu_g(\mu_\mathbf{x}, \bar{\mathbf{u}})$ and the covariance matrix $\Sigma_g(\mu_\mathbf{x}, \bar{\mathbf{u}})$ estimates of $g(\mathbf{x}, \bar{\mathbf{u}})$ that are functions of the mean $\mu_\mathbf{x}$ of the state $\mathbf{x}$ and control $\bar{\mathbf{u}}$. Due to Remark 10, the unknown bias term is modeled as a multivariate Gaussian distribution $\hat{g}(\mu_\mathbf{x}, \bar{\mathbf{u}}) \sim \mathcal{N}(\mu_g, \Sigma_g)$. The estimate $\hat{g}$ in (6.3) can be expressed as

$$\hat{g} = B(\mu_\mathbf{x}, \bar{\mathbf{u}})\theta(t) + \mu_g(\mu_\mathbf{x}, \bar{\mathbf{u}}), \tag{6.8}$$

where $\theta(t) \sim \mathcal{N}(0, dt\mathbb{I})$ i.i.d and $B(\mu_\mathbf{x}, \bar{\mathbf{u}})B^\top(\mu_\mathbf{x}, \bar{\mathbf{u}}) = \Sigma_g(\mu_\mathbf{x}, \bar{\mathbf{u}})$. Using (6.8), (6.7) can be written in standard Ito stochastic differential equation (SDE) form as given below, where $\theta(t)dt = dw$.

$$dx = f(\mathbf{x}, \bar{\mathbf{u}})dt + \mu_g(\mu_\mathbf{x}, \bar{\mathbf{u}})dt + B(\mu_\mathbf{x}, \bar{\mathbf{u}})dw \tag{6.9}$$

The existence and uniqueness of a solution to the SDE for a given initial distribution $\mathbf{x}_0$ and control trajectory $\bar{\mathbf{u}}$ such that $\Pr(|\mathbf{x}(t_0) - \mathbf{x}_0)| = 0) = 1$ with measure $\Pr$, is guaranteed by the *Lipschitz condition* (4.8) and *Restriction on Growth condition*(4.9). The approximate system (6.9) is assumed to be controllable in the given feasible space.

Cost Functional

The integrand cost functional $J = J_{\mathcal{C}} + J_{\mathcal{I}}$ includes two objectives: a) exploration cost $J_{\mathcal{I}}$ for achieving the maximum value of information for learning the unknown dynamics g, and b) performance cost $J_{\mathcal{C}}$ (e.g., minimizing the control effort). The integrand cost functional $J_{\mathcal{C}}$ for fuel optimality, which is convex in $\bar{\mathbf{u}}$, is given as

$$J_{\mathcal{C}} = \|\bar{\mathbf{u}}\|_s \quad \text{where } \; s \in \{1, 2, \infty\} \tag{6.10}$$

One example of $J_{\mathcal{I}}$ is the following variance-based information cost using each i^{th} diagonal element σ_i^2 in Σ_g.

$$J_{\mathcal{I}} = -\sum_{i=1}^{n} \sigma_i(\mu_{\mathbf{x}}, \bar{\mathbf{u}}) \tag{6.11}$$

The information cost $J_{\mathcal{I}}$ in (6.11) is a functional of the mean $\mu_{\mathbf{x}}$ of the state $\mathbf{x}$ and control $\bar{\mathbf{u}}$ at time t. By minimizing the cost $J_{\mathcal{I}}$, we maximize the information [105] available in the trajectory $\mathbf{x}$ to learn the unknown model g. The terminal cost functional J_f is quadratic in the state $\mathbf{x}$, $J_f = \mathbf{x}(t_f)^\top Q_f \mathbf{x}(t_f)$, where Q_f is a positive semi-definite function.

State and Safety Constraints

Safety is defined as a constraint on the state space $\mathbf{x}$, $\mathbf{x}(t) \in \mathcal{F}$ at time t. The safe set $\mathcal{F}$ is relaxed by formulating a joint chance constraint with risk of constraint violation as

$$\Pr(x \in \mathcal{F}) \geq 1 - \epsilon. \tag{6.12}$$

The constant ϵ is called the risk measure of a chance constraint. We consider the polytopic constraint set $\mathcal{F}_{\text{lin}} = \{\mathbf{x} \in \mathcal{X} : \wedge_{i=1}^{k} a_i^\top \mathbf{x} + b_i \leq 0\}$ with k flat sides and a quadratic constraint set $\mathcal{F}_{\text{quad}} = \{\mathbf{x} \in \mathcal{X} : \mathbf{x}^\top Q \mathbf{x} \leq c\}$ for any realization $\mathbf{x}$ of the state as described in Chapter 4. For the rest of the chapter, we consider the following individual chance-constrained problem:

$$\begin{aligned} (\mathbf{x}^*, \bar{\mathbf{u}}^*) = &\underset{\mathbf{x}(t), \bar{\mathbf{u}}(t)}{\text{argmin}} \quad \mathbb{E}\left[\int_{t_0}^{t_f} ((1-\rho) J_{\mathcal{C}} + \rho J_{\mathcal{I}}) dt + J_f\right], \\ &\text{s.t.} \{(6.9),\ (4.15),\ (4.18),\ (6.5),\ (6.6)\}. \end{aligned} \tag{6.13}$$

that is assumed to have a feasible solution with $\rho \in [0, 1]$. In Sec. 6.2, we present the Info-SNOC algorithm by using the projected deterministic optimal control problem described in Chapter 4.

6.2 Info-SNOC Main Algorithm

We present the main algorithm for the architecture shown in Fig. 1.3 that integrates the learning method and the Info-SNOC method with rollout. We discuss an iterative solution method to Info-SNOC by projecting (6.13) to the gPC space. We formulate a deterministic optimal control problem in the gPC space and solve it using SCP [14, 15, 20] method. The gPC projection of (6.13) is given by the following equation:

Problem 11.

$$\begin{aligned}(\mathbf{X}^*, \bar{\mathbf{u}}^*) = &\underset{\mathbf{X}(t), \bar{\mathbf{u}}(t)}{\operatorname{argmin}} \quad \left[\int_{t_0}^{t_f} ((1-\rho) J_{\mathcal{C}} + \rho J_{\mathcal{I}}) dt + \mathbb{E} J_{gPC_f}\right] \\ &\text{s.t. } \{(4.28),\ (4.37),\ (4.43),\ (6.5)\} \\ &\mathbf{X}(t_0) = \mathbf{X}_0, \quad \mathbf{X}(t_f) = \mathbf{X}_f.\end{aligned} \tag{6.14}$$

We solve Problem 11 by using the gPC-SCP approach described in Section 4.2.4. The information cost functional $J_{\mathcal{I}}$ from (6.11) is expressed as a function of X by using the polynomial representation [20] of $\mu_{\mathbf{x}}$ in terms of X. Let $S = \{\mathrm{X}, \bar{\mathbf{u}}\}$ and the cost $J_{\mathcal{I}}$ is linearized around a feasible nominal trajectory $S^0 = \{\mathrm{X}^0, \bar{\mathbf{u}}^0\}$ to derive a linear convex cost functional $J_{d\mathcal{I}}$:

$$J_{d\mathcal{I}} = -\sum_{i=1}^{n} \left(\sigma_i(S^0) + \frac{\partial \sigma_i}{\partial S}\Big|_{S^0} (S - S^0)\right). \tag{6.15}$$

We use the convex approximation $J_{d\mathcal{I}}$ as the information cost in the SCP formulation of the optimal control problem in (6.14). In the gPC space, we split the problem into two cases: a) $\rho = 0$ that computes a performance trajectory, and b) $\rho \in (0, 1]$ that computes information trajectory to have stable iterations. The main algorithm is outlined below.

Algorithm

We use an initial estimate of the model (6.8) learned from data generated by a known safe control policy, and a *nominal trajectory* $(\mathbf{x}^o, \bar{\mathbf{u}}^o)$ computed using deterministic SCP [15] with nominal model to initialize Algorithm 4. The stochastic model and the chance constraints are projected to the gPC state space, which is in

line 2 of Algorithm 4. The projected dynamics is linearized around the nominal trajectory and used as a constraint in the SCP. The projection step is only needed in the first epoch. The projected system can be directly used for epoch > 1. The current estimated model is used to solve (6.14) using SCP, in line 7 with $\rho = 0$, for a performance trajectory $(\mathbf{x}_p, \bar{\mathbf{u}}_p)$. The output $(\mathbf{x}_p, \bar{\mathbf{u}}_p)$ of this optimization is used as initialization to the Info-SNOC problem obtained by setting $\rho \in (0, 1]$ to compute the information trajectory $(\mathbf{x}_\mathcal{I}, \bar{\mathbf{u}}_\mathcal{I})$. The trajectory $(\mathbf{x}_\mathcal{I}, \bar{\mathbf{u}}_\mathcal{I})$ is then sampled for a safe motion plan $(\bar{\mathbf{p}}_{\text{des}}, \bar{\mathbf{u}}_{\text{des}})$ in line 9, that is used for rollout, in line 10, to collect more data for learning. The SCP step is performed in the gPC space X. After each SCP step, the gPC space coordinates X are projected back to the random variable $\mathbf{x}$ space. The Info-SNOC problem outputs a trajectory of random variable $\mathbf{x}$ with finite variance at each epoch.

Algorithm 4: Info-SNOC using SCP [14] and gPC [20]

Input:

1. Initial Safe Set Data, Feasible Nominal Trajectory$(\mathbf{x}^o, \bar{\mathbf{u}}^o)$
2. gPC Projection as discussed in Section 4.2
3. Linearize the gPC cost and dynamics, see Section 4.2.4

1 epoch = 1,
2 **while** *Learning Criteria Not Satisfied* **do**
3 Learn $g \sim \mathcal{N}(\mu_g, \Sigma_g)$ using Robust Regression,
4 $(\mathbf{x}_p, \bar{\mathbf{u}}_p)$ = **SCP** $((\mathbf{x}^o, \bar{\mathbf{u}}^o), \rho = 0)$, using (6.14),
5 $(\mathbf{x}_\mathcal{I}, \bar{\mathbf{u}}_\mathcal{I})$ = **SCP** $((\mathbf{x}_p, \bar{\mathbf{u}}_p), \rho \in (0, 1])$, using (6.14),
6 Sample $(\mathbf{x}_\mathcal{I}, \bar{\mathbf{u}}_\mathcal{I})$ for $(\bar{\mathbf{p}}_{\text{des}}, \bar{\mathbf{u}}_{\text{des}})$,
7 Rollout using sample $(\bar{\mathbf{p}}_{\text{des}}, \bar{\mathbf{u}}_{\text{des}})$ and $\mathbf{u}_c$,
8 Data collection during rollout,
9 epoch $\leftarrow$ epoch + 1

Convergence and Optimality

The information trajectory $(\mathbf{x}_\mathcal{I}, \bar{\mathbf{u}}_\mathcal{I})$ computed using SCP with the approximate linear information cost $J_{d\mathcal{I}}$ (6.15) is a sub-optimal solution of (6.14) with the optimal cost value $J^*_{d\mathcal{I}}$. Therefore, the optimal cost of (6.14) given by $J^*_\mathcal{I}$ is bounded above by $J^*_{d\mathcal{I}}$, $J^*_\mathcal{I} \leq J^*_{d\mathcal{I}}$. For the Info-SNOC algorithm, we cannot guarantee the convergence of SCP iterations to a Karush-Kuhn-Tucker point using the method in [14, 103] due to the non-convexity of $J_\mathcal{I}$. Due to the non-convex cost function $J_\mathcal{I}$, the linear approximation $J_{d\mathcal{I}}$ of the cost $J_\mathcal{I}$ can potentially lead to numerical

instability in SCP iterations. Finding an initial performance trajectory, $\rho = 0$, and then optimizing for information, $\rho \in (0, 1]$, is observed to be numerically stable compared to directly optimizing for the original cost functional $J = J_{\mathcal{C}} + J_{\mathcal{I}}$.

Feasibility

The initial phases of learning might lead to a large covariance Σ_g due to the insufficient data, resulting in an infeasible optimal control problem. To overcome this, we use two strategies: 1) explore the initial safe set till we find a feasible solution to the problem, and 2) use slack variables on the terminal condition to approximately reach the goal accounting for a large variance.

6.2.1 Rollout Policy Implementation

The information trajectory $(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})$ computed using the Info-SNOC algorithm is sampled for a pool of motion plans $(\bar{\mathbf{p}}_{\text{des}}, \bar{\mathbf{u}}_{\text{des}})$. The trajectory pool is computed by randomly sampling the multivariate Gaussian distribution θ and transforming it using the gPC expansion $\mathbf{x}(\theta) \approx \bar{\Phi}(\theta)\mathrm{X}$. For any realization $\bar{\theta}$ of θ, we get a deterministic trajectory $\bar{\mathbf{p}}_{\text{des}} = \bar{\Phi}(\bar{\theta})\mathrm{X}$ that is ϵ safe with respect to the distributionally robust chance constraints. The trajectory $(\bar{\mathbf{p}}_{\text{des}}, \bar{\mathbf{u}}_{\text{des}})$ is executed using the closed-loop control law $\mathbf{u}_c = \mathbf{u}_c(\bar{\mathbf{p}}, \bar{\mathbf{p}}_{\text{des}}, \bar{\mathbf{u}}_{\text{des}})$ for rollout, where $\bar{\mathbf{p}}$ is the current state. To ensure real-time safety during the initial stages of exploration, a safe control policy $\mathbf{u}_c$ is computed using the control barrier function-based safety filter. The properties of the control law $\mathbf{u}_c$ and the safety during rollout are studied in the following section.

6.3 Analysis

In this section, we present the main theoretical results analyzing the following two questions: 1) at any epoch i how do learning errors translate to safety violation bounds during rollout, and 2) under what assumptions is the multivariate robust regression a consistent learning method as epoch $\to \infty$. The analysis proves that if the Info-SNOC algorithm computes a motion plan with finite variance, then learning is consistent and implies safety during rollout.

Assumption 8. The projected problem (6.14) computes a feasible trajectory to the original problem (6.13). The assumption is generally true if we choose a sufficient number of polynomials [20], for the projection operation.

Assumption 9. The probabilistic inequality $\Pr(\|g(\bar{\mathbf{p}}, \bar{\mathbf{u}}) - \mu_g(\bar{\mathbf{p}}, \bar{\mathbf{u}})\|_2^2 \leq c_1) \geq 1 - \epsilon_{\ell\text{b}}$ holds, where $\epsilon_{\ell\text{b}}$ is small, for the same input $(\bar{\mathbf{p}}, \bar{\mathbf{u}})$ to the original model g,

and the mean μ_g of the learned model. Using this inequality, we can say that the following bounds hold with high probability:

$$\|g(\bar{\mathbf{p}}, \bar{\mathbf{u}}) - \mu_g(\bar{\mathbf{p}}, \bar{\mathbf{u}})\|_2^2 \leq c_1, \mathrm{tr}\left(\Sigma_g\right) \leq c_2, \tag{6.16}$$

where $c_2 = c_1 \epsilon_{\ell \mathrm{b}}$. As shown in [154], the mean predictions made by the model learned using robust regression is bounded by c_1, which depends on the choice of the function class for learning. The variance prediction Σ_g is bounded by design. With Assumptions 8 and 9, the analysis is decomposed into the following three subsections.

6.3.1 State Error Bounds During Rollout

The following assumptions are made on the nominal system $\dot{\bar{\mathbf{p}}} = f(\bar{\mathbf{p}}, u)$ to derive the state tracking error bound during rollout.

Assumption 10. There exists a globally exponentially stable (i.e., finite-gain $\mathcal{L}_p$ stable) tracking control law $\mathbf{u}_c = \mathbf{u}_c(\bar{\mathbf{p}}, \bar{\mathbf{p}}_{\mathrm{des}}, \bar{\mathbf{u}}_{\mathrm{des}})$ for the nominal dynamics $\dot{\bar{\mathbf{p}}} = f(\bar{\mathbf{p}}, \mathbf{u}_c)$. The control law $\mathbf{u}_c$ satisfies the property $\mathbf{u}_c(\bar{\mathbf{p}}_{\mathrm{des}}, \bar{\mathbf{p}}_{\mathrm{des}}, \bar{\mathbf{u}}_{\mathrm{des}}) = \bar{\mathbf{u}}_{\mathrm{des}}$ for any sampled trajectory $(\bar{\mathbf{p}}_{\mathrm{des}}, \bar{\mathbf{u}}_{\mathrm{des}})$ from the information trajectory $(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})$. At any time t the state $\bar{\mathbf{p}}$ satisfies the following inequality, when the closed-loop control $\mathbf{u}_c$ is applied to the nominal dynamics,

$$M(\bar{\mathbf{p}}, t)\tfrac{\partial f}{\partial \bar{\mathbf{p}}} + \left(\tfrac{\partial f}{\partial \bar{\mathbf{p}}}\right)^{\top} M(\bar{\mathbf{p}}, t) + \tfrac{d}{dt} M(\bar{\mathbf{p}}, t) \leq -2\alpha M(\bar{\mathbf{p}}, t),$$

where $f = f(\bar{\mathbf{p}}, \mathbf{u}_c(\bar{\mathbf{p}}, \bar{\mathbf{p}}_{\mathrm{des}}, \bar{\mathbf{u}}_{\mathrm{des}}))$, $\alpha > 0$, $M(\bar{\mathbf{p}}, t)$ is a uniformly positive definite matrix with $(\lambda_{\min}(M)\|\bar{\mathbf{p}}\|^2 \leq \bar{\mathbf{p}}^{\top} M(\bar{\mathbf{p}}, t)\bar{\mathbf{p}} \leq \lambda_{\max}(M)\|\bar{\mathbf{p}}\|^2)$, and $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and minimum eigenvalues.

Assumption 11. The unknown model g satisfies the bound $\|g\left(\bar{\mathbf{p}}, \mathbf{u}_c\right) - g(\bar{\mathbf{p}}_{\mathrm{des}}, \bar{\mathbf{u}}_{\mathrm{des}})\|_2^2 \leq c_3$.

Assumption 12. The probability density ratio $\frac{\zeta_{\mathbf{x}_{\mathcal{I}}(t)}}{\zeta_{\mathbf{x}_{\mathcal{I}}(0)}} \leq r$ is bounded, where the functions $\zeta_{\mathbf{x}_{\mathcal{I}}(0)}$ and $\zeta_{\mathbf{x}_{\mathcal{I}}(t)}$ are the probability density functions of $\mathbf{x}_{\mathcal{I}}$ at time $t = 0$ and t respectively.

Lemma 9. *Given that the estimated model* (6.8) *satisfies the Assumption 9, and the systems (6.7) and* (6.9) *satisfy Assumptions 10, 11, 12, if the control* $\mathbf{u}_c = \mathbf{u}_c(\bar{\mathbf{p}}, \mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})$ *is applied to the system* (6.7)*, then the following condition holds at time* t

$$\begin{aligned} \mathbb{E}_{\mathbf{x}_{\mathcal{I}}(t)}(\|\bar{\mathbf{p}} - \mathbf{x}_{\mathcal{I}}\|_2^2) \leq\ & \tfrac{\lambda_{\max}(M)}{2\lambda_{\min}(M)\alpha_m}(c_1 + c_2 + c_3)r \\ & + \tfrac{\lambda_{\max}(M)r}{\lambda_{\min}(M)} \mathbb{E}\left(\|\bar{\mathbf{p}}(0) - \mathbf{x}_{\mathcal{I}}(0)\|^2\right) e^{-2\alpha_m t}, \end{aligned} \tag{6.17}$$

where $(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})$ *is computed from* (6.14) *and* $\alpha_m = (\alpha - 1)$. *The states* $\bar{\mathbf{p}} \in \mathcal{X}$, *and* $\mathbf{x}_{\mathcal{I}} \in \mathcal{X}$ *are feasible trajectories of the deterministic dynamics (6.7) and the SDE (6.9) for the initial conditions* $\bar{\mathbf{p}}(0) \in \mathcal{X}$ *and* $\mathbf{x}_{\mathcal{I}}(0) \in \mathcal{X}$ *respectively at* $t \geq t_0$.

Proof. Consider the following original system $(\bar{\mathbf{p}}, \bar{\mathbf{u}}_c)$ propagated with the control $\mathbf{u}_c = \mathbf{u}_c(\bar{\mathbf{p}}, \mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})$, and the learned dynamical system that satisfies that information trajectory $(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})$.

$$\begin{aligned} d\bar{\mathbf{p}} &= f(\bar{\mathbf{p}}, \mathbf{u}_c)dt + g(\bar{\mathbf{p}}, \mathbf{u}_c)dt \\ d\mathbf{x}_{\mathcal{I}} &= f(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})dt + \mu_g(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})dt + B(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})dw \end{aligned} \tag{6.18}$$

We define an augmented system using $\mathbf{y} = [\bar{\mathbf{p}}, \mathbf{x}_{\mathcal{I}}]^\top$.

$$d\mathbf{y} = \begin{bmatrix} f(\bar{\mathbf{p}}, \mathbf{u}_c) \\ f(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}}) \end{bmatrix} dt + \begin{bmatrix} g(\bar{\mathbf{p}}, \mathbf{u}_c) \\ \hat{g}(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}}) \end{bmatrix} dt + \begin{bmatrix} 0 & 0 \\ 0 & B(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}}) \end{bmatrix} \begin{bmatrix} dw_1 \\ dw \end{bmatrix} \tag{6.19}$$

Consider the distance metric $\mathcal{V} = e^\top M(\bar{\mathbf{p}}, t)e$, where $e = \bar{\mathbf{p}} - \mathbf{x}_{\mathcal{I}}$. The infinitesimal generator $\mathcal{L}$ using the Itô assumption is given by,

$$\begin{aligned} \mathcal{L}\mathcal{V} =& e^\top \frac{d}{dt} M(\bar{\mathbf{p}}, t)e \\ &+ 2e^\top M(\bar{\mathbf{p}}, t) \left(f(e + \mathbf{x}_{\mathcal{I}}, \mathbf{u}_c) - f(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})\right) \\ &+ 2e^\top M(\bar{\mathbf{p}}, t) \left(g\left(e + \mathbf{x}_{\mathcal{I}}, \mathbf{u}_c\right) - \mu_g(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})\right) \\ &+ \operatorname{tr}(B(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})^\top M(\bar{\mathbf{p}}, t)B(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})), \end{aligned}$$

where $\frac{d}{dt}M_{ij} = \frac{\partial M_{ij}}{\partial t} + \left(\frac{\partial M_{ij}}{\partial \bar{\mathbf{p}}}\right)^\top f(e + \mathbf{x}_{\mathcal{I}}, \mathbf{u}_c)$. Using the bound,

$$\operatorname{tr}(B(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})^\top M(\bar{\mathbf{p}}, t)B(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})) \leq \lambda_{\max}(M)c_2 \tag{6.20}$$

and $\mathbf{u}_c$ is designed such that $\mathbf{u}_c(\bar{\mathbf{p}} = \mathbf{x}_{\mathcal{I}}, \mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}}) = \bar{\mathbf{u}}_{\mathcal{I}}$. For a fixed $t > 0$, using the argument in [155], we can find $c \in [\bar{\mathbf{p}}, \mathbf{x}_{\mathcal{I}}]$ such that

$$\begin{aligned} \mathcal{L}\mathcal{V} \leq\ & e^\top \frac{d}{dt} M(\bar{\mathbf{p}}, t)e + e^\top M \frac{\partial f}{\partial \bar{\mathbf{p}}}\Big|_{\bar{\mathbf{p}}=c} e + e^\top \left(\frac{\partial f}{\partial \bar{\mathbf{p}}}\right)^\top\Big|_{\bar{\mathbf{p}}=c} Me, \\ &+ 2e^\top M(\bar{\mathbf{p}}, t) \left(g\left(e + \mathbf{x}_{\mathcal{I}}, \mathbf{u}_c\right) - g(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})\right) \\ &+ 2e^\top M(\bar{\mathbf{p}}, t) \left(g(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}}) - \mu_g(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})\right) + \lambda_{\max}(M)c_2. \end{aligned}$$

$$\begin{aligned} \mathcal{L}\mathcal{V} \leq\ & -2\alpha e^\top M(\bar{\mathbf{p}}, t)e \\ &+ 2e^\top M(\bar{\mathbf{p}}, t) \left(g\left(e + \mathbf{x}_{\mathcal{I}}, \mathbf{u}_c\right) - g(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})\right) \\ &+ e^\top M(\bar{\mathbf{p}}, t)e \\ &+ \lambda_{\max}(M) \| \left(g(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}}) - \mu_g(\mathbf{x}_{\mathcal{I}}, \bar{\mathbf{u}}_{\mathcal{I}})\right) \|_2^2 + \lambda_{\max}(M)c_2 \end{aligned}$$

Using completion of squares,

$$\begin{aligned}\mathcal{L}\mathcal{V} \leq & -2(\alpha-1)e^\top M(\bar{\mathbf{p}},t)e \\ & + \lambda_{\max}(M)\| \left(g\left(e+\mathbf{x}_\mathcal{I},\mathbf{u}_c\right) - g(\mathbf{x}_\mathcal{I},\bar{\mathbf{u}}_\mathcal{I})\right)\|_2^2 \\ & + \lambda_{\max}(M)(c_1+c_2).\end{aligned}$$

We assume that $\|g\left(e+\mathbf{x}_\mathcal{I},\mathbf{u}_c\right) - g(\mathbf{x}_\mathcal{I},\bar{\mathbf{u}}_\mathcal{I})\|_2^2 \leq c_3$.

$$\mathcal{L}\mathcal{V} \leq -2\alpha_m\mathcal{V} + \lambda_{\max}(M)(c_1+c_2+c_3). \tag{6.21}$$

Applying Dynkin's formula, the bound on the density ratio, and using the bounds on M we have the following bound:

$$\begin{aligned}\mathbb{E}_{\mathbf{x}(0)}(\|\bar{\mathbf{p}}-\mathbf{x}_\mathcal{I}\|_2^2) \leq & \frac{\lambda_{\max}(M)}{\lambda_{\min}(M)}\mathbb{E}\left(\|\bar{\mathbf{p}}(0)-\mathbf{x}_\mathcal{I}(0)\|^2\right)e^{-2\alpha_m t} \\ & + \frac{\lambda_{\max}(M)(c_1+c_2+c_3)}{2\lambda_{\min}(M)\alpha_m}\end{aligned} \tag{6.22}$$

$$\begin{aligned}\mathbb{E}_{\mathbf{x}_\mathcal{I}(t)}(\|\bar{\mathbf{p}}-\mathbf{x}_\mathcal{I}\|_2^2) \leq & \tfrac{\lambda_{\max}(M)}{2\lambda_{\min}(M)\alpha_m}(c_1+c_2+c_3)r \\ & + \tfrac{\lambda_{\max}(M)r}{\lambda_{\min}(M)}\mathbb{E}\left(\|\bar{\mathbf{p}}(0)-\mathbf{x}_\mathcal{I}(0)\|^2\right)e^{-2\alpha_m t}\end{aligned} \tag{6.23}$$

Therefore, expectation of the tracking error is bounded. □

Lemma 9 states that the expected mean squared error $\mathbb{E}(\|\bar{\mathbf{p}}-\mathbf{x}_\mathcal{I}\|^2)$ is bounded by $\frac{\lambda_{\max}(M)(c_1+c_2+c_3)r}{2\alpha_m\lambda_{\min}(M)}$ as $t\to\infty$ when the control law $\mathbf{u}_c$ is applied to the dynamics in (6.7). The bounded tracking performance leads to constraint violation, which is studied in the next section.

6.3.2 Safety Bounds

The safety of the original system (6.7) for the linear and quadratic chance constraints during rollout with a controller $\mathbf{u}_c$ discussed in Sec. 6.3.1 is analyzed in Theorems 5 and 6.

Theorem 5. *Given a feasible solution* $(\mathbf{x},\bar{\mathbf{u}}_\mathbf{x})$ *of (6.13), with the quadratic chance constraint* $\Pr((\mathbf{x}-\mu_\mathbf{x})^\top Q(\mathbf{x}-\mu_\mathbf{x}) \geq c) \leq \frac{\mathbb{E}((\mathbf{x}-\mu_\mathbf{x})^\top Q(\mathbf{x}-\mu_\mathbf{x}))}{c} \leq \epsilon_q$*, the trajectory* $\bar{\mathbf{p}}$ *of the deterministic dynamics* (6.7) *satisfies the following inequality at any time* t*:*

$$(\bar{\mathbf{p}}-\mu_\mathbf{x})^\top Q(\bar{\mathbf{p}}-\mu_\mathbf{x}) \leq \lambda_{\max}(Q)\mathbb{E}_\mathbf{x}\left(\|\bar{\mathbf{p}}-\mathbf{x}\|_2^2\right), \tag{6.24}$$

where $\mathbb{E}\left(\|\bar{\mathbf{p}}-\mathbf{x}\|_2^2\right)$ *is bounded as defined in Lemma 9.*

Proof. Consider the expectation of the ellipsoidal set $(\bar{\mathbf{p}} - \mu_{\mathbf{x}})^\top Q(\bar{\mathbf{p}} - \mu_{\mathbf{x}})$. Using $\bar{\mathbf{p}} - \mu_{\mathbf{x}} = \bar{\mathbf{p}} - \mathbf{x} + \mathbf{x} - \mu_{\mathbf{x}}$, the expectation of the set can be expressed as follows:

$$\begin{aligned} &\mathbb{E}\left((\bar{\mathbf{p}} - \mu_{\mathbf{x}})^\top Q(\bar{\mathbf{p}} - \mu_{\mathbf{x}})\right) = \mathbb{E}\left((\bar{\mathbf{p}} - \mathbf{x})^\top Q(\bar{\mathbf{p}} - \mathbf{x})\right) \\ &+\mathbb{E}\left((\mathbf{x} - \mu_{\mathbf{x}})^\top Q(\mathbf{x} - \mu_{\mathbf{x}})\right) + 2\mathbb{E}\left((\bar{\mathbf{p}} - \mathbf{x})^\top Q(\mathbf{x} - \mu_{\mathbf{x}})\right). \end{aligned} \tag{6.25}$$

Using the following equality:

$$\mathbb{E}\left((\bar{\mathbf{p}} - \mathbf{x})^\top Q(\mathbf{x} - \mu_{\mathbf{x}})\right) = -\mathbb{E}\left((\mathbf{x} - \mu_{\mathbf{x}})^\top Q(\mathbf{x} - \mu_{\mathbf{x}})\right),$$

and $\mathbb{E}\left((\mathbf{x} - \mu_{\mathbf{x}})^\top Q(\mathbf{x} - \mu_{\mathbf{x}})\right) \geq 0$ in (6.25), we obtain the constraint bound in (6.24). Using the feedback tracking bound (6.17) in (6.24), we can show that the constraint violation bound is a function of learning bounds c_1, c_2, and c_3. □

Note that if the learning method converges, i.e., $c_1 \to 0$, $c_2 \to 0$, and $c_3 \to 0$, then $\bar{\mathbf{p}} \to \mu_{\mathbf{x}}$. The quadratic constraint violation in (6.24) depends on the tracking error and the size of the ellipsoidal set described by Q.

Theorem 6. *Given a feasible solution* $(\mathbf{x}, \bar{\mathbf{u}}_{\mathbf{x}})$ *of (6.13) with* $\inf_{\mathbf{x}\sim(\mu_{\mathbf{x}},\Sigma_{\mathbf{x}})} \Pr(a^\top \mathbf{x} + b \leq 0) \geq 1 - \epsilon_\ell$*, the trajectory* $\bar{\mathbf{p}}$ *of the deterministic dynamics* (6.7)*, with control* $\mathbf{u}_c$*, satisfies the following inequality at any time* t*:*

$$\inf_{\mathbf{x}\sim(\mu_{\mathbf{x}},\Sigma_{\mathbf{x}})} \Pr\left(a^\top \bar{\mathbf{p}} + b \leq \delta_\ell(\mathbf{x})\right) \geq 1 - \epsilon_\ell, \tag{6.26}$$

where $\delta_\ell(\mathbf{x}) = \|a\|_2 \mathbb{E}_{\mathbf{x}}(\|(\bar{\mathbf{p}} - \mathbf{x})\|_2) - \|a\|_2 \sqrt{\frac{1-\epsilon_\ell}{\epsilon_\ell}} \sqrt{c_4}$*,* $\mathbb{E}_{\mathbf{x}}(\|(\bar{\mathbf{p}} - \mathbf{x}\|_2)$ *is bounded as defined in* (6.17) *and* $\mathrm{tr}(\Sigma_{\mathbf{x}}) = c_4$*.*

Proof. From Lemma 3, the feasible solution $(\mathbf{x}, \bar{\mathbf{u}}_{\mathbf{x}})$ satisfies the equivalent condition $\mathcal{P}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}) \leq 0$, where $\mathcal{P}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}) = a^\top \mu_{\mathbf{x}} + b + \sqrt{\frac{1-\epsilon_\ell}{\epsilon_\ell}} \sqrt{a^\top \Sigma_{\mathbf{x}} a}$ for the risk measure ϵ_ℓ, mean $\mu_{\mathbf{x}}$, and covariance $\Sigma_{\mathbf{x}}$. Consider the similar condition for the actual trajectory $\bar{\mathbf{p}}(t)$, $\mathcal{P}(\mu_{\bar{\mathbf{p}}}, \Sigma_{\bar{\mathbf{p}}})$:

$$\begin{aligned} &\mathcal{P}(\mu_{\bar{\mathbf{p}}}, \Sigma_{\bar{\mathbf{p}}}) = a^\top \mu_{\bar{\mathbf{p}}} + b + \sqrt{\frac{1-\epsilon_\ell}{\epsilon_\ell}} \sqrt{a^\top \Sigma_{\bar{\mathbf{p}}} a} \\ &= a^\top \mu_{\mathbf{x}} + a^\top (\mu_{\bar{\mathbf{p}}} - \mu_{\mathbf{x}}) + b + \sqrt{\frac{1-\epsilon_\ell}{\epsilon_\ell}} \sqrt{a^\top \Sigma_{\bar{\mathbf{p}}} a} \\ &+ \sqrt{\frac{1-\epsilon_\ell}{\epsilon_\ell}} \sqrt{a^\top \Sigma_{\mathbf{x}} a} - \sqrt{\frac{1-\epsilon_\ell}{\epsilon_\ell}} \sqrt{a^\top \Sigma_{\mathbf{x}} a} \end{aligned} \tag{6.27}$$

Note that since the system (6.7) is deterministic, we have $\mu_{\bar{\mathbf{p}}} = \bar{\mathbf{p}}$ and $\Sigma_{\bar{\mathbf{p}}} = 0$. Using $\mathcal{P}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}) \leq 0$, the right hand side of the above inequality reduces to the following:

$$\mathcal{P}(\mu_{\bar{\mathbf{p}}}, \Sigma_{\bar{\mathbf{p}}}) \leq a^\top(\bar{\mathbf{p}} - \mu_{\mathbf{x}}) - \sqrt{\frac{1-\epsilon_\ell}{\epsilon_\ell}}\sqrt{a^\top \Sigma_{\mathbf{x}} a}. \tag{6.28}$$

Using the decomposition $\Sigma_{\mathbf{x}} = \tilde{G}^\top \tilde{G}$, Cauchy-Schwarz's inequality, and Jensen's inequality, we have $a^\top(\mu_{\bar{\mathbf{p}}} - \mu_{\mathbf{x}}) \leq \|a\|_2 \|(\bar{\mathbf{p}} - \mu_{\mathbf{x}})\|_2 \leq \|a\|_2 \mathbb{E}_{\mathbf{x}}(\|(\bar{\mathbf{p}} - \mathbf{x})\|_2)$. Using the sub-multiplicative property of ℓ_2-norm in the inequality above, we have $\mathcal{P}(\mu_{\bar{\mathbf{p}}}, \Sigma_{\bar{\mathbf{p}}}) \leq \|a\|_2 \left(\mathbb{E}(\|(\bar{\mathbf{p}} - \mathbf{x})\|_2) - \sqrt{\frac{1-\epsilon_\ell}{\epsilon_\ell}} \|\tilde{G}\|_{\mathrm{F}}\right)$. Assuming that $\mathrm{tr}(\Sigma_{\mathbf{x}}) = c_4$ in the above inequality, we have

$$\mathcal{P}(\mu_{\bar{\mathbf{p}}}, \Sigma_{\bar{\mathbf{p}}}) \leq \|a\|_2 \mathbb{E}(\|(\bar{\mathbf{p}} - \mathbf{x})\|_2) - \|a\|_2 \sqrt{\frac{1-\epsilon_\ell}{\epsilon_\ell}} \sqrt{c_4}.$$

The above inequality is equivalent to the probabilistic linear constraint in (6.26). The bound $\delta_\ell(\mathbf{x}) = \|a\|_2 \mathbb{E}(\|(\bar{\mathbf{p}} - \mathbf{x})\|_2) - \|a\|_2 \sqrt{\frac{1-\epsilon_\ell}{\epsilon_\ell}} \sqrt{c_4}$ is a function of the learning bounds in (6.16) by substituting the feedback tracking bound in (6.17). □

The linear constraint is offset by δ_ℓ leading to constraint violation of the original formulation (4.14). Note that, if $c_1 \to 0$, $c_2 \to 0$, $c_3 \to 0$, and $c_4 \to 0$, then $\delta_\ell \to 0$. In order to ensure real-time safety during trajectory tracking, we use a high gain control for disturbance attenuation with safety filter augmentation for constraint satisfaction.

6.3.3 Consistency

Data is collected during the rollout of the nonlinear system to learn a new model for the next epoch. For epoch e, the predictor $\hat{g}^e$ follows a multivariate Gaussian distribution $\mathcal{N}(\mu_g^e, \Sigma_g^e)$ and g is the empirical true data. We assume that set $\mathcal{X}_e \subset \mathcal{X}$ generated by the optimization problem in (6.13) for the first e iterations is a discretization of $\mathcal{X}$. Assuming that there exists a global optimal predictor $\hat{g}^*$ in the function class $\mathcal{G}$ that can achieve the best error ϵ^* at each epoch e:

$$\max_{\mathbf{x} \in \mathcal{X}_e} \|g - \hat{g}^*\|_2 = \epsilon^*, \tag{6.29}$$

the consistency of the learning algorithm is given by the convergence of the regret r_e, which is defined as the Euclidean distance between the predictor $\hat{g}^e$ and the optimal predictor $\hat{g}^*$:

$$r_e = \|\hat{g}^e - \hat{g}^*\|_2 \to 0, \text{ as } e \to \infty. \tag{6.30}$$

We prove the consistency of Algorithm 4 in Theorem 7.

Theorem 7. *If Assumption 9 holds with the maximum prediction error* $\max_{\mathbf{x}\in\mathcal{X}_e} \|g-\mu_g^e\|_2^2 \triangleq c_1^e$ *at epoch* e*, then the regret* $r_e = \|\hat{g}^e - \hat{g}^*\|_2$ *of Algorithm 4 achieves the bound:*

$$r_e^2 \leq c_1^e + dC^2 + \epsilon^{*2} \text{ with probability } 1 - |\mathcal{X}_e|\delta_e, \tag{6.31}$$

$$\text{where } \delta_e \triangleq (C^d \prod_p^d \Delta_p)^{-1} \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_g^e|^{\frac{1}{2}}} e^{-\frac{1}{2}C^2\sum_{p=1}^d \Delta_p} \tag{6.32}$$

with the output dimension d*,* $\Delta_p \triangleq \sum_{q=1}^d m_{qp} > 0$*,* $\forall p \in \{1,2,\ldots,d\}$*, and* $\mathcal{M} = (\Sigma_g^e)^{-1}$ *with its* (q,p)*-th element* (m_{qp})*. Furthermore, the regret* $r_e \to 0$ *as* $e \to \infty$*.*

Proof. Using the inequality $\Pr(\hat{g}^e - \mu_g^e \geq C\mathbf{e}) < \delta_e$, $\forall \mathbf{x} \in \mathcal{X}_e$, since $\hat{g}^e \sim \mathcal{N}(\mu_g^e, \Sigma_g^e)$, we can bound r_e as follows:

$$r_e^2 \leq \|\mu_g^e + C\mathbf{e} - \hat{g}^*\|_2^2, \tag{6.33}$$

where δ_e is defined in (6.32), and $\mathbf{e}$ is unit vector in d dimensions. Hence, C is a function of $|\mathcal{X}_e|\delta_e$. This is the tail probabilities inequality [156] of multivariate Gaussian distributions. The error r_e^2 is then bounded using the empirical prediction error $\|\mu_g^e - g\|_2^2$ and the best prediction error (6.29) as $r_e^2 \leq \|\mu_g^e + C\mathbf{e} - g\|_2^2 + \|g - \hat{g}^*\|_2^2$. By using the triangular inequality, we get the bound in (6.31) as follows:

$$r_e^2 \leq \|\mu_g^e - g\|_2^2 + dC^2 + \epsilon^{*2} \leq c_1^e + dC^2 + \epsilon^{*2}. \tag{6.34}$$

At any epoch e, we have $\mathbf{x} \in \mathcal{X}_e$, $|\|\mu_g^e\|_2^2 + \mathrm{tr}(\Sigma_g^e) - \|g\|_2^2| \leq \omega_e$ [154], where ω_e is a hyper-parameter that is associated with model selection in robust regression. Using $\mathrm{tr}(\Sigma_g^e) \leq c_2^e$, (6.33) becomes $r_e^2 \leq \omega_e + c_2^e + dC^2 + \epsilon^{*2}$.

Assuming that the ground truth model is in the function class $\mathcal{G}$, we have $\epsilon^* = 0$. With infinite data, the hyper parameter $\omega_e = 0$ and $c_2^e \to 0$ at each data point. For achieving the same value of δ_e, $C \to 0$ with shrinking Σ_g^e. This implies that for large data, i.e., as $e \to \infty$, we have $r_e \to 0$. The regret r_e is bounded as shown in (6.31). Therefore, Algorithm 4 is consistent with high probability. Note that the Info-SNOC approach outputs trajectories of finite variance at epoch e, given that $\mathrm{tr}(\Sigma_g^e)$ is bounded. □

During the initial exploration phase, the finite dimensional approximation in the gPC projection might add to the error bounds discussed in Theorems 5 and 6 leading to safety violation. This approximation error is automatically incorporated as

residual dynamics in the stochastic dynamics when collecting data during rollout and learned using robust regression. Thus ensuring that safety is guaranteed with increasing data.

6.4 Simulation and Discussion

We test the Info-SNOC framework shown in Fig. 1.3 on the three degree-of-freedom robotic spacecraft simulator [30] dynamics with an unknown friction model and an over-actuated thruster configuration that is used for a real spacecraft. The dynamics of $\mathbf{x} = (x, y, \theta)^T$ with respect to an inertial frame is given as:

$$\ddot{\mathbf{x}} = f(\theta, B, \bar{\mathbf{u}}) + g(\dot{\mathbf{x}}, \dot{y}, \dot{\theta}),\ f = \mathrm{blkdiag}(R(\theta), 1)B\bar{\mathbf{u}},$$

where $R(\theta) \in SO(2)$. The states $(x, y) \in \mathbb{R}^2$, $\theta \in [0, 2\pi)$ denote position and orientation respectively. The function g is unknown, and assumed to be linear viscous damping in the simulations. The control effort $\bar{\mathbf{u}} \in \mathbb{R}^8$ is constrained to be $0 \leq \bar{\mathbf{u}} \leq 1$, and $B(m, I, l, b) \in \mathbb{R}^{3\times 8}$ is the control allocation matrix where m =17 kg and $I = 2\,\mathrm{kg\,m\,s^{-2}}$ are the mass and the inertia matrix, and $l = b = 0.4\,\mathrm{m}$ is the moment arm.

The unknown function $g = \mathrm{diag}([-0.02, -0.02, -0.002])\dot{\mathbf{x}}$ is modeled as a multivariate Gaussian distribution to learn from data using robust regression. To get an initial estimate of the model, we explore a small safe set around the initial condition and collect data. For the following test cases, we collect 40 data points to have a feasible optimal control problem (6.13) in the planning stage as discussed in Sec. 6.2. The algorithm is initialized with a *nominal trajectory* computed using deterministic SCP under nominal dynamics.

Info-SNOC Results

The learned dynamics is used to design safe trajectories for $40\,\mathrm{s}$ using the Info-SNOC algorithm for Scenarios 1 and 2 as shown in Figs. 6.1a and 6.2a, respectively. Scenario 1 has a wall at $y = 6\,\mathrm{m}$ and a circular obstacle of radius $2.5\,\mathrm{m}$ at $(5, -0.3)$, and scenario 2 has two circular obstacles of radius $0.6\,\mathrm{m}$ at $(5, -0.5)$ and $(10, 0.5)$ as collision constraints respectively. The obstacles in both scenarios are transformed to linear chance constraints and the terminal constraint is transformed to quadratic chance constraint as discussed in [20], with a risk measure of $\epsilon_\ell = \epsilon_q = 0.05$. The Info-SNOC algorithm is applied with $\rho = 0$ ($\mathcal{L}_1$-norm control cost) and $\rho = 1$ (information cost). We compare the mean $\mu_{\mathbf{x}}$ and 2σ-confidence ellipse around $\mu_{\mathbf{x}}$ of the trajectories with the nominal trajectory. We observe that

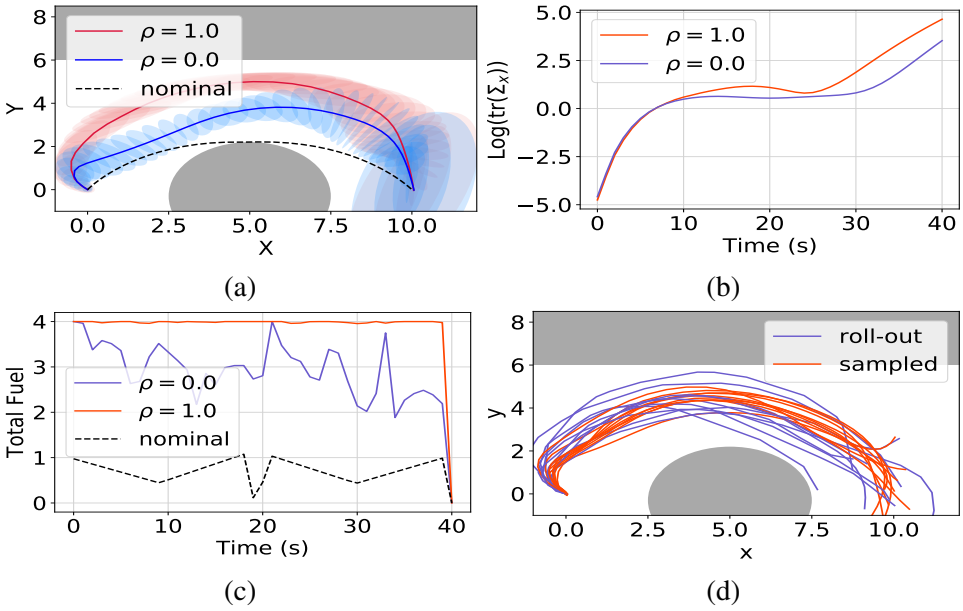
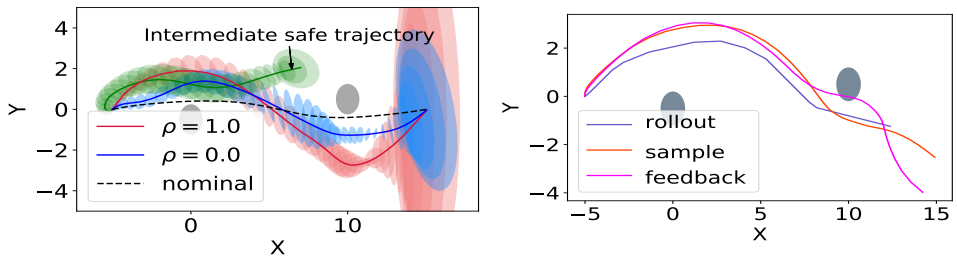


Figure 6.1: Info-SNOC applied to Scenario 1. In Fig. (a), we show the motion plan along with the 2σ confidence in position of the performance trajectory ($\rho = 0$) and the information trajectory ($\rho = 1$) computed using Info-SNOC, and the nominal trajectory computed using SCP under nominal dynamics. In Fig. (b), we show the trace of $\Sigma_{\mathbf{x}}$ w.r.t time. The information trajectory ($\rho = 1$) has higher $\Sigma_{\mathbf{x}}$ compared to the performance trajectory ($\rho = 0$). We compare total open-loop fuel computed at each time step in Fig. (c), and in Fig. (d) we demonstrate collision avoidance during exploration for 20 trials of rollout using a safety augmented stable controller.

for the $\rho = 1$, the safe trajectory explores larger state-space compared to the $\rho = 0$ case, which corresponds to the fuel-optimal trajectory. The total control effort at each time is shown in Fig. 6.1c. It shows that information trajectory uses more energy compared to the performance trajectory. The extra fuel is used to explore the domain for improving the model. The terminal variance $\Sigma_{\mathbf{x}}(t_f)$ in both scenarios is large due to the correlation among the multiple dimensions of g that are predicted by the learning algorithm.

Safe Rollout

The safe trajectories in Figs. 6.1a and 6.2a are sampled following the method discussed in Sec. 6.2.1 and the rollout is performed using the controller designed in [30] that satisfies Lemma 9. The sample trajectories and rollout trajectories are shown in red and blue respectively in Figs. 6.1d and 6.2b. The sampled trajectory is safe with the risk measure of collision $\epsilon_\ell = 0.05$ around the obstacles. Rollout trajectories, with a feedback controller, collide with the obstacles due to the



(a) (b)

Figure 6.2: Info-SNOC applied to Scenario 2. In Fig. (a), we show a comparison of the performance trajectory ($\rho = 0$), the information trajectory ($\rho = 1$), and an intermediate safe trajectory (green) computed using Info-SNOC and the nominal trajectory computed using deterministic SCP under nominal dynamics. In Fig. (b), we compare a sampled trajectory with the trajectories generated by feedback tracking and rollout with a safety filter.

following two reasons: 1) the learning bounds (6.16) lead to constraint violation as discussed in Theorems 5 and 6, and 2) the state-dependent uncertainty model $\mathcal{N}(\mu_g, \Sigma_g)$ might predict large Σ_g that can saturate the actuators. Saturated actuators cannot compensate for the unmodelled dynamics. In order to ensure safety, we augment the feedback controller with a real-time safety augmentation using barrier-function -based quadratic program [157]. Using this filter, the blue rollout trajectories are diverted from obstacles, as seen in Figs. 6.1d and 6.2b, avoiding constraint violation.

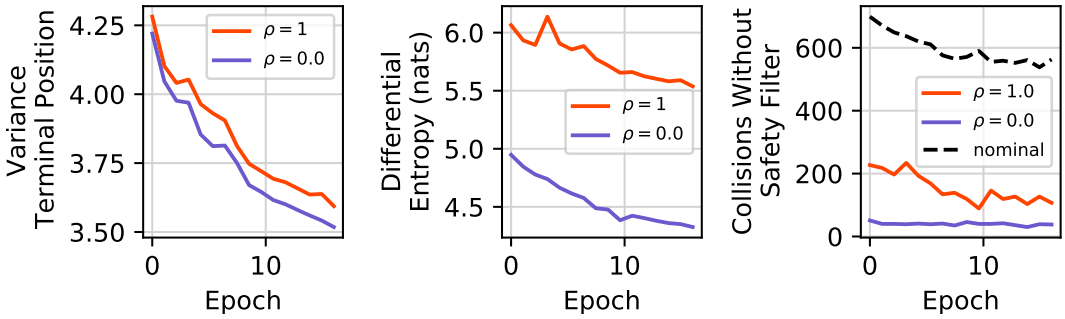


Figure 6.3: Performance over epochs for Scenario 1. Left: we show decrease in the terminal position variance over epochs demonstrating improved goal reaching with epoch. Center: the differential entropy of the prediction variance Σ_g for information trajectory ($\rho = 1$) is larger compared to the performance trajectory ($\rho = 0$). Right: the number of collisions during rollout for 1000 trials decrease as the learning converges, validating Theorem 6.

Consistency

The data collected during rollout is appended to the earlier data to learn a new model. Figure 6.3 shows improvement in control performance (control cost for $\rho = 0.0$) with increasing number of epochs. The prediction variance Σ_g decreases from $1e-2$ to $0.3e-2$ over 15 epochs. The main assumption for consistency, which states that the variance of the trajectory computed using (6.13) is bounded and decreasing, is satisfied, thereby demonstrating the correctness of Theorem 7. We observe that the differential entropy of the information trajectory is higher (i.e., contains more information about g) than the performance trajectory by design.

We observed that $\Sigma_{\mathbf{x}}(t_f)$ of the information trajectory $(\rho = 1)$ computed using Info-SNOC decreases from 104.11 to 61.46 over 15 epochs, by applying the framework in Fig. 1.3 for Scenario 1, demonstrating the increase in probability of reaching the quadratic terminal set. This validates the Theorem 5. The number of collisions of the rollout trajectory without a safety filter over a 1000 trials at each epoch is shown in Fig. 6.3. The number of collisions decrease from 227 to 107 for the information trajectory, from 51 to 38 for the performance trajectory, and from 699 to 563 for the nominal trajectory over 15 epochs. This increase in the probability of the linear chance constraint satisfaction with epoch validates the Theorem 6. In the first epoch, we observed that the rollout using the nominal trajectory leads to 69.9% collisions over 1000 trials, using the information trajectory leads to 22% collisions, and using the performance trajectory leads to less than 5.1% collisions, demonstrating the effectiveness of Info-SNOC.

6.5 Chapter Summary

We present a new method of learning-based safe exploration and motion planning by solving information-cost stochastic optimal control using a partially learned non-linear dynamical model. The variance prediction of the learned model is used as the information cost, while the safety is formulated as distributionally robust chance constraints. The problem is then projected to the generalized polynomial chaos space and solved using sequential convex programming. We use the Info-SNOC method to compute a safe and information-rich pool of trajectories for rollout using an exponentially stable controller with a safety filter augmentation for safe data collection. We analyze the probability of constraint violation for both linear and quadratic constraints. We show that the safety constraints are satisfied for rollout under learned dynamics, as the learned model converges to the optimal predictor

over epochs. The consistency of the learning method using the Info-SNOC algorithm is proven under mild assumptions.

The episodic learning framework was applied to the robotic spacecraft model to explore the state space and learn the friction under collision constraints. We compute a pool of safe and optimal trajectories using the Info-SNOC algorithm for a learned spacecraft model under collision constraints and discuss an approach for rollout using a stable feedback control law to collect data for learning. We validate the consistency of robust regression method and the safety guarantees by showing the reduction in variance of the learned model predictions and the number of collisions over 15 epochs respectively.

Chapter 7

MULTI-SPACECRAFT TESTBED FOR AUTONOMY RESEARCH

The information-based guidance and control architecture proposed in Chapter 3 and the gPC-SCP method derived in Chapters 4 and 5 are tested on the Multi-Spacecraft Testbed for Autonomy Research (M-STAR). In the following, we describe the design, construction, and control of the M-STAR testbed.

7.1 The Spacecraft Dynamics Simulator Facility at Caltech

7.1.1 Overview of the Facility

The spacecraft simulator facility requires the following three components to be operational: the epoxy flat floor, the compressed air filling station, and the M-STAR. The epoxy flat floor is a high precision floor with flatness within 0.001 inches for frictionless translation of the spacecraft dynamics simulator using three flat air-bearing pads. Figure 7.1 shows the facility with multiple M-STAR spacecraft simulators and protection for collisions on the outer edge of the floor. The full 6-DOF spacecraft simulator can be seen in the middle with two 3-DOF simulators on the sides.The second component, the filling station, is composed of an industrial air compressor and two 6,000 psi storage tanks. The filling station is used to fill the on-board air cylinders that supply air to the flat air bearings, spherical air bearing, and 16 on-off non-latching solenoid valves that act as thrusters on the simulator. The M-STAR shown in Fig. 7.2 acts as the dynamic simulation platform for a smallsat and includes all the necessary on-board sensors, actuator systems, and computing to achieve full 6-DOF control. The pose of the spacecraft simulator is estimated using 14 motion capture cameras mounted on the ceiling of the facility. In the following section, we elaborate on the subsystem hardware of the simulator.

7.1.2 M-STAR Spacecraft Simulator Hardware

The Caltech Aerospace Robotics and Controls Lab's 6-DOF spacecraft dynamics simulator for spacecraft formation control research was designed to accommodate up to a 12U CubeSat as a payload. The floating test bed simulates 5-DOF dynamic motion and 1-DOF kinematic motion along the gravity direction, with translation and attitude stages decoupled via a spherical air bearing. The translation stage

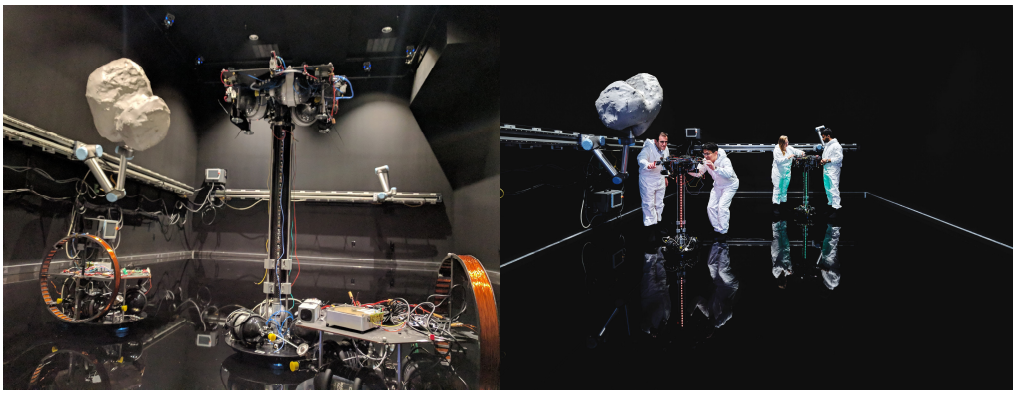


Figure 7.1: Multiple 6-DOF M-STAR spacecraft at Caltech's Aerospace Robotics and Control Lab.

floats frictionlessly on the precision floor using three flat round air bearings. The attitude stage has a hemispherical air bearing ball that floats frictionlessly on the cup mounted at the top of the linear actuator on the translation stage. Tables 7.1 and 7.2 list the hardware components on both the translation stage and attitude stages respectively. The hardware on each stage is divided into three subsystems: 1) mechanical, including structural and pneumatic components; 2) electrical, including power, computing, and low level controller boards; and 3) actuation, to impart torque or impulse in the required degree of freedom. Each of these components plays an essential role in achieving torque-free controlled motion.

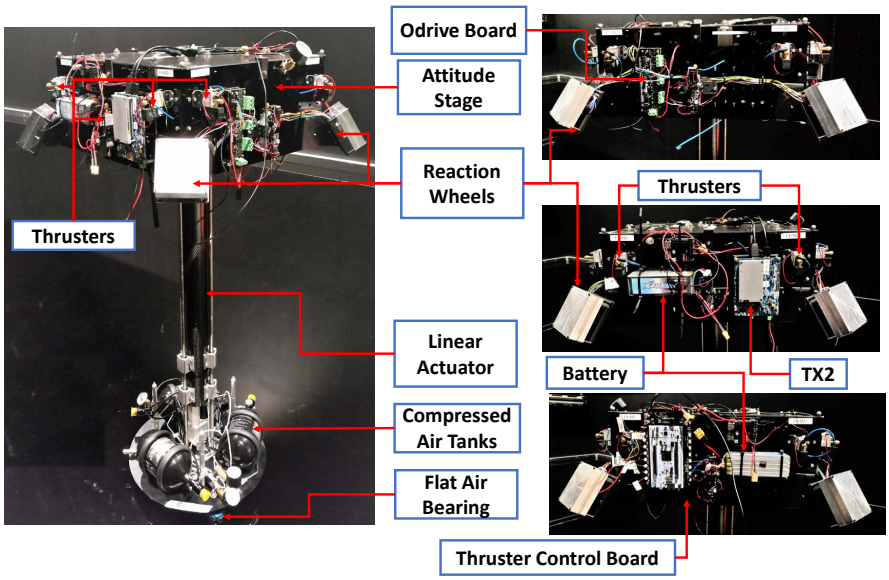


Figure 7.2: M-STAR spacecraft dynamics simulator.

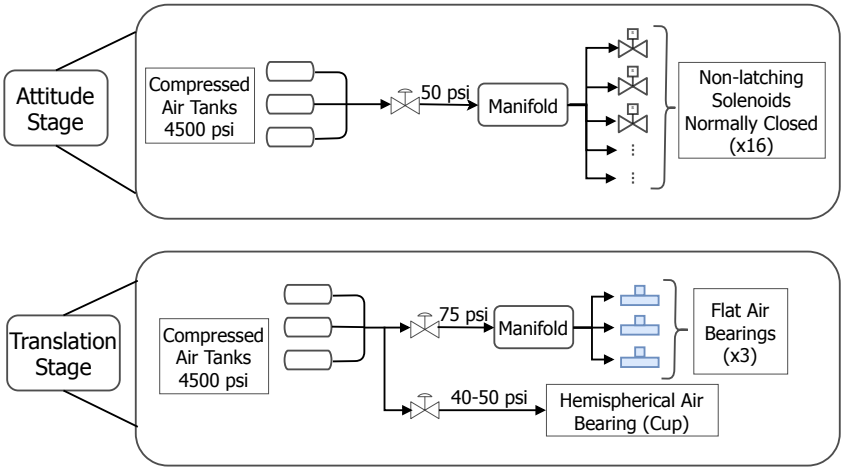


Figure 7.3: Flowchart of pneumatic system on translation and attitude stage.

Translation Stage.

The translation stage provides frictionless in-plane motion for the whole simulator using three linear flat round air bearings. It consists of three compressed air cylinders running at 4500 psi, a spherical air bearing cup, pneumatic components for pressure regulation, and tubing required to supply air for the bearings. The pneumatic system on the translation stage is shown in Fig. 7.3. In addition, it is equipped with a linear actuator, a brushless DC linear motor for achieving motion in the gravity direction with supporting control electronics.

The different operation modes of operation (3-DOF, 4-DOF, 5-DOF, and 6-DOF) can be achieved as follows:

- 3-DOF: spherical air bearing turned off and linear actuator replaced with a passive tube
- 4-DOF: spherical air bearing turned off
- 5-DOF: linear actuator replaced with a passive tube
- 6-DOF: all actuators active

This provides flexibility in operation and allows the construction of algorithms with increased complexity. The compressed air storage tanks' capacity was designed to achieve at least 25-30 minutes of flotation time at the operating pressure in 6-DOF mode. Several custom-designed add-ons can be incorporated on the translation stage such as docking ports and reaction wheels for yaw control.

Subsystem	Component
Mechanical	NewWay Air Bearing Compressed Air Cylinders Structure Design Spherical Air bearing Regulator
Actuator	Progressive Automation Linear Actuator
Electronics and Power	Battery Linear Actuator Controller Raspberry Pi

Table 7.1: List of components on the translation stage.

Attitude Stage.

The attitude stage structure was designed using carbon fiber composites and honeycomb materials, optimized to provide a flotation time of up to 30 minutes with a payload of 12 kilograms. It has a box structure and acts as a platform for a potential payload, such as a 12U CubeSat. The attitude stage structure has the hemispherical ball of the air bearing pair and floats on the translation stage to provide 3-DOF frictionless attitude motion. This stage has 16 on-off non-latching solenoids with custom made nozzles and four in-house reaction wheels as actuators. The power distribution board for the attitude stage and the low-level controller for the thrusters are designed at Caltech. The schematic of the pneumatic subsystem for supplying regulated compressed air to the thrusters is shown in Fig. 7.3. It includes three compressed air cylinders, a regulator, and a manifold for air distribution. The regulated pressure is supplied to all the thrusters through the manifold to maintain the pressure across them. The operating pressure of the thrusters is decided based on experimental characterization of the solenoids. The electrical subsystem of the attitude stage is shown in Fig. 7.4. We chose an NVIDIA Jetson TX2 as the main computer to run the GNC and perception algorithms. The computer communicates the control signal to the low level boards as shown in Fig. 7.4. The subsystem components of the stages are listed in Table 7.2.

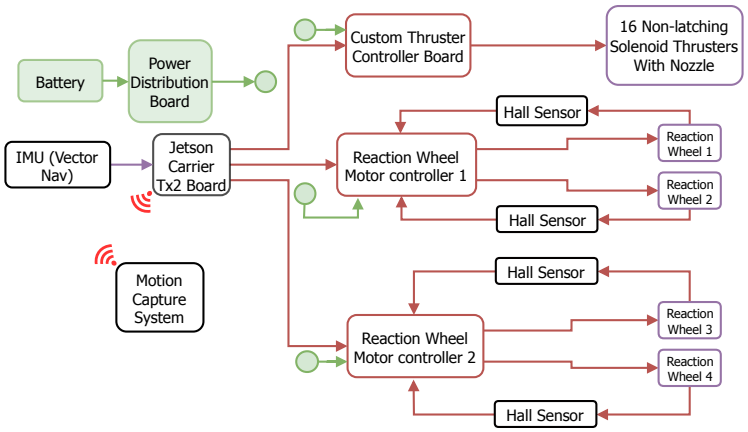


Figure 7.4: Attitude stage architecture.

Subsystem	Component
Mechanical	Structure Nozzles Pneumatics
Actuator	Thrusters Reaction Wheel Assembly
Electronics and Power	Battery Power Distribution Board Thruster Control Board ODRIVE Reaction Wheel Driver Maxon Motor Reaction Wheel Motor NVIDIA Jetson TX2 Computer

Table 7.2: List of components on the attitude stage.

Reaction Wheel Sizing and Manufacturing.

Custom reaction wheels were designed for the spacecraft simulator to achieve certain nominal performance specifications. The principle axis of inertia of the whole simulator including a 12U CubeSat payload, shown in (7.1), was estimated using CAD software. The nominal torque and angular momentum requirements for attitude control corresponding to the estimated principle inertia of the spacecraft is shown in Table 7.3.

$$J = \begin{bmatrix} 1.19 & 0 & 0 \\ 0 & 1.24 & 0 \\ 0 & 0 & 1.43 \end{bmatrix} [\text{kgm}^2] \tag{7.1}$$

Nominal torque X-axis [Nm]	0.069
Nominal torque Y-axis [Nm]	0.072
Nominal torque Z-axis [Nm]	0.044
Angular momentum X-axis [Nms]	0.2077
Angular momentum Y-axis [Nms]	0.2164
Angular momentum Z-axis [Nms]	0.4492

Table 7.3: Nominal torque and angular momentum of the spacecraft.

The flywheel was made out of brass, fabricated using a CNC lathe for better tolerances. The motor selected was an EC frameless motor from Maxon Motor, which has a decoupled stator and the rotor, leading to an increased lifetime. The entire structure (flywheel, shaft, and motor) was constrained between two ball bearings to reduce vibration as shown in Fig. 7.5.

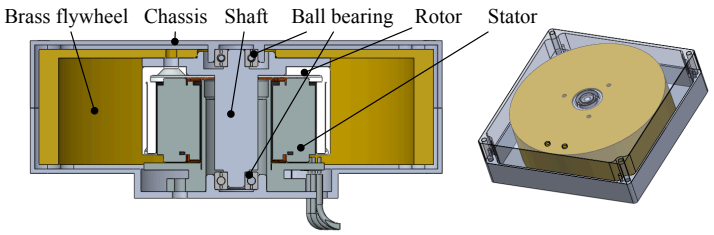


Figure 7.5: Section view of Caltech's custom-made reaction wheels.

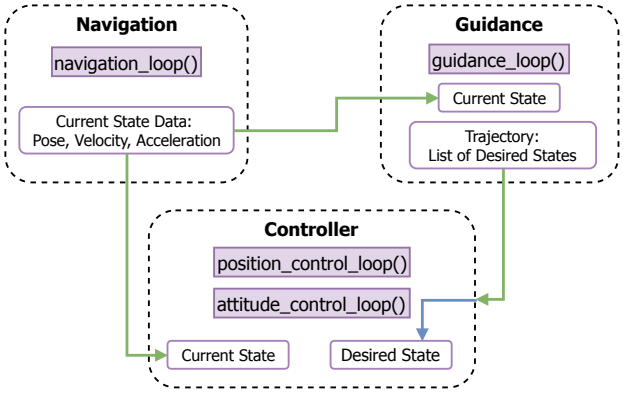


Figure 7.6: Software architecture design.

7.1.3 M-STAR Software Architecture

The software for the simulator was designed to allow for interchangeable guidance, navigation, and control modules. The architecture is implemented in C++ using abstract base classes for the three modules, with virtual loop functions for subclasses

to implement. As illustrated in Fig. 7.6, navigation subclasses are responsible for generating updated state data for the guidance system and controller. The guidance system maintains a trajectory of desired states, from which the controller selects a target state for the current time step and implements the required dynamics. The current experimental setup features waypoint guidance, motion capture camera based navigation, and the 5-DOF controller outlined in the next section. However, these could respectively be swapped for an arbitrary motion-planning algorithm, pose feedback from integrated on board sensor data, and controllers for the four configurations of the simulator.

The architecture is built on Robotic Operating System (ROS) framework, which allows for each loop to be scheduled at a unique rate that can be changed at run time. Data from other modules is automatically fetched before each loop runs. ROS also provides a messaging architecture for communicating with peripheral boards, the ability to create unique launch configurations for different module setups, and test logging.

7.2 Dynamics and Control

Each M-STAR has two links coupled using a spherical air bearing as a joint. This system can be modelled as a three dimensional pendulum on a floating platform with a ball joint to provide 3-DOF rotation of the pendulum (modelling the attitude of the spacecraft) and 2-DOF planar motion of the floating platform. Constraints on the 3D pendulum motion due to mechanical interference between the attitude stage and the translation stage are shown in Table 7.4. The coordinate systems used for deriving the kinematics and dynamics of the system are shown in Fig. 7.7. The inertial reference frame on the test floor is defined by the coordinate system (X_i, Y_i, Z_i) with origin O_i . A non-rotating reference frame (X_{ib}, Y_{ib}, Z_{ib}) that is parallel to the inertial frame, is attached to the attitude stage with origin O_b at the center of the hemispherical bearing to define the orientation of the attitude stage. The attitude stage dynamics are derived in terms of the angular rates in the body frame (X_b, Y_b, Z_b) at origin O_b. Before proceeding to the discussion on the dynamics and control implementation, the attitude representation used for describing the motion of the 3D pendulum in $SO(3)$ space is discussed.

Pitch (rotation about X_{ib})	$\pm 45^\circ$
Roll (rotation about Y_{ib})	$\pm 45^\circ$
Yaw (rotation about Z_{ib})	$\pm 180^\circ$

Table 7.4: Constraints on the angular motion of the attitude stage.

7.2.1 Attitude Kinematics

The attitude of the 3D pendulum can be represented by any attitude representations including quaternions [128], Modified Rodrigues Parameters (MRPs) [128], and $SO(3)$ rotation matrix. For example, the MRPs, $\mathbf{p}_{\text{mrp}} = [p_1, p_2, p_3]^\top$, $\mathbf{p}_{\text{mrp}} \in \mathbb{R}^3$ are stereographic projections of the unit quaternions [128], $\mathbf{q} \in \mathbb{H}$, where $\mathbb{H}$ is the Hamiltonian space and have a bijective mapping to the quaternion sphere are used here. The attitude representation in MRPs takes into account the unit norm of the quaternions. The attitude kinematics equation is given using the body angular rates $\boldsymbol{\omega} \in \mathbb{R}^3$. The kinematics of MRPs are given as follows:

$$\begin{aligned}
&\dot{\mathbf{p}}_{\text{mrp}} = \mathbf{Z}(\mathbf{p}_{\text{mrp}})\boldsymbol{\omega}, \\
&\mathbf{Z}(\mathbf{p}_{\text{mrp}}) = \frac{1}{2}\left(\mathbb{I}_3\left(\frac{1-\mathbf{p}_{\text{mrp}}^T\mathbf{p}_{\text{mrp}}}{2}\right) + \mathbf{p}_{\text{mrp}}\mathbf{p}_{\text{mrp}}^T + \mathbf{S}(\mathbf{p}_{\text{mrp}})\right), \\
&\mathbf{S}(\mathbf{p}_{\text{mrp}}) = \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix}.
\end{aligned} \tag{7.2}$$

The rotation matrix $R(\mathbf{p}_{\text{mrp}})^\top$ to transform from the frame (X_{ib}, Y_{ib}, Z_{ib}) to the body frame(X_b, Y_b, Z_b) in terms of the MRPs is given as

$$R(\mathbf{p}_{\text{mrp}})^\top = \mathbb{I}_{3\times 3} - \frac{4(1-\mathbf{p}_{\text{mrp}}^T\mathbf{p}_{\text{mrp}})}{(1+\mathbf{p}_{\text{mrp}}^T\mathbf{p}_{\text{mrp}})^2}\mathbf{S}(\mathbf{p}_{\text{mrp}}) + \frac{8}{(1+\mathbf{p}_{\text{mrp}}^T\mathbf{p}_{\text{mrp}})^2}\mathbf{S}(\mathbf{p}_{\text{mrp}})^2. \tag{7.3}$$

The transformation $R(\mathbf{p}_{\text{mrp}})$ is used in mapping the external force due to thrusters in the body frame to the inertial frame for controlling the translation dynamics.

7.2.2 Nonlinear Dynamic Model

The dynamics of the 5-DOF system with the velocity $\mathbf{v}_b$ at the centre of rotation of the attitude stage and angular rates of the attitude stage $\boldsymbol{\omega}$ in body frame (X_b, Y_b, Z_b) is given in the (7.5), where, r_{cg} is the center of gravity offset from the center of rotation of the attitude stage in the body frame coordinates, J is the mass moment of inertia of the attitude stage about the center of rotation in the body frame, $R(\mathbf{p}_{\text{mrp}})$ is defined in (7.3), (x, y) is the planar location of the center of rotation from the inertial frame origin, m_a is the mass of the attitude stage, and m_t is the mass of

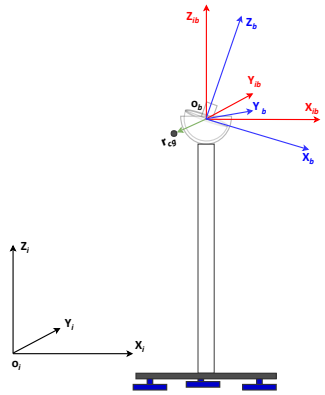


Figure 7.7: Coordinate Systems used for the derivation of the dynamic model.

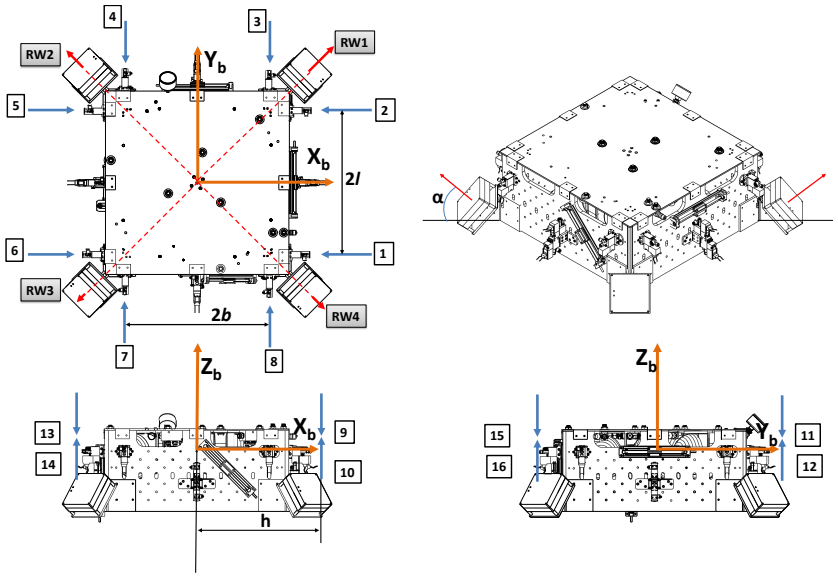


Figure 7.8: Attitude Stage with actuator configuration and nomenclature in the body frame.

the translation stage. In the following equations, $a_1 = [1;0;0]$, $a_2 = [0;1;0]$ and $a_3 = [0;0;1]$ are unit vectors in the reference frame (X_{ib}, Y_{ib}, Z_{ib}).

$$\dot{\mathbf{p}}_{\mathrm{p}} = R(\mathbf{p}_{\mathrm{mrp}})\mathbf{v}_b \quad \text{where} \quad \mathbf{p}_{\mathrm{p}} = (x, y, 0)^\top, \quad D = (a_1^\top; a_2^\top; 0) \tag{7.4}$$

$$M_b(\mathbf{p}_{\mathrm{mrp}}) \begin{bmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\mathbf{v}}_b \end{bmatrix} + C_b \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{v}_b \end{bmatrix} + H_b = \boldsymbol{\tau}_b \tag{7.5}$$

$$M_b(\mathbf{p}_{\mathrm{mrp}}) = \begin{bmatrix} J & m_a \mathbf{S}(r_{cg}) R(\mathbf{p}_{\mathrm{mrp}})^\top D R(\mathbf{p}_{\mathrm{mrp}}) \\ m_a \left(\mathbf{S}(r_{cg}) R(\mathbf{p}_{\mathrm{mrp}})^\top D R(\mathbf{p}_{\mathrm{mrp}})\right)^\top & (m_a + m_t) \end{bmatrix} \tag{7.6}$$

$$C_b = \begin{bmatrix} -\mathbf{S}(J\boldsymbol{\omega}) & m_a\mathbf{S}(r_{cg})R(\mathbf{p}_{\mathrm{mrp}})^\top DR(\mathbf{p}_{\mathrm{mrp}})\mathbf{S}(\boldsymbol{\omega}) \\ -m_aR(\mathbf{p}_{\mathrm{mrp}})^\top DR(\mathbf{p}_{\mathrm{mrp}})\mathbf{S}(\boldsymbol{\omega})\mathbf{S}(r_{cg}) & (m_t+m_a)\mathbf{S}(\boldsymbol{\omega}) \end{bmatrix} \tag{7.7}$$

$$H_b = \begin{bmatrix} -m_ag\mathbf{S}(r_{cg})R(\mathbf{p}_{\mathrm{mrp}})^\top a_3 \\ 0 \end{bmatrix} \tag{7.8}$$

The control inputs to the system are represented by $\boldsymbol{\tau}_b = [\begin{smallmatrix}\boldsymbol{\tau}_f \\ \boldsymbol{\tau}_t\end{smallmatrix}]$, which include forces due to thrusters $\boldsymbol{\tau}_f$ and torques $\boldsymbol{\tau}_t$ due to thrusters and reaction wheels in body frame. The control design is done in body frame. The forces computed in body frame $\boldsymbol{\tau}_f$ are transformed to forces in inertial frame $\boldsymbol{\tau}_p = R(\mathbf{p}_{\mathrm{mrp}})\boldsymbol{\tau}_f$ for implementation of the position control law. The implementation of the transformation and the influence of thrusters in the body frame on the position dynamics in inertial frame is discussed in the following sections. In the body frame, for the 5-DOF dynamics in (7.5) it can be proved that $\dot{M}_b - (C_b + C_b^T) = 0$ and that $\dot{M}_b - 2C_b$ is a skew-symmetric matrix. The matrix form in (7.5) will be used in the following section to derive a controller that globally exponentially tracks a given position and almost globally exponentially tracks an attitude trajectory.

7.2.3 Control Design for Full Nonlinear Dynamics

The objective of the control design is to ensure that the 5-DOF of M-STAR, $[\mathbf{p}_{\mathrm{p}}, \mathbf{p}_{\mathrm{mrp}}]$ given in (7.4), exponentially tracks a given trajectory $[\mathbf{p}_{\mathrm{p_{des}}}, \mathbf{p}_{\mathrm{mrp_{des}}}] \in \mathcal{C}^2([0,\infty])$. The following theorem states the nonlinear control law and proves the global exponential stability of the closed-loop system in (7.11). Here the variables $s_{\boldsymbol{\omega}} = \boldsymbol{\omega} - \boldsymbol{\omega}_r$ and $s_{\mathbf{v}} = \mathbf{v}_b - \mathbf{v}_{b_r}$ define the states for virtual dynamics. The variables $\boldsymbol{\omega}_r$ and $\mathbf{v}_{b_r}$ define the reference signal computed from filtered desired states dynamics given in the following (7.9).

$$\begin{aligned} \boldsymbol{\omega}_r &= \mathbf{Z}^{-1}(\mathbf{p}_{\mathrm{mrp}})\dot{\mathbf{p}}_{\mathrm{mrp_{des}}}(t) + \mathbf{Z}^{-1}(\mathbf{p}_{\mathrm{mrp}})\Lambda_{\boldsymbol{\omega}}(\mathbf{p}_{\mathrm{mrp_{des}}}(t) - \mathbf{p}_{\mathrm{mrp}}) \\ \mathbf{v}_{b_r} &= R^\top\dot{\mathbf{p}}_{\mathrm{p_{des}}}(t) + R^\top\Lambda_{\mathbf{v}}(\mathbf{p}_{\mathrm{p_{des}}}(t) - \mathbf{p}_{\mathrm{p}}) \end{aligned} \tag{7.9}$$

Theorem 8. *The closed-loop system in terms of virtual states $s_{\boldsymbol{\omega}}, s_{\mathbf{v}}$, given in (7.11), with the control law (7.9–7.10), is globally exponentially stable in the sense of the Euclidean norm, assuming the feedback gains $K_{\boldsymbol{\omega}}, K_{\mathbf{v}}, \Lambda_{\boldsymbol{\omega}}, \Lambda_{\mathbf{v}} > 0$ and the inertia matrix M_b is positive definite and uniformly bounded with lower bound $\lambda_{\min}$ and*

upper bound $\lambda_{\max}$.

$$\boldsymbol{\tau}_b = M_b \begin{bmatrix} \dot{\boldsymbol{\omega}}_r \\ \dot{\mathbf{v}}_{b_r} \end{bmatrix} + C_b \begin{bmatrix} \boldsymbol{\omega}_r \\ \mathbf{v}_{b_r} \end{bmatrix} + H_b - \begin{bmatrix} K_{\boldsymbol{\omega}} & 0 \\ 0 & K_{\mathbf{v}} \end{bmatrix} \begin{bmatrix} s_{\boldsymbol{\omega}} \\ s_{\mathbf{v}} \end{bmatrix} \tag{7.10}$$

$$M_b \begin{bmatrix} \dot{s}_{\boldsymbol{\omega}} \\ \dot{s}_{\mathbf{v}} \end{bmatrix} + C_b \begin{bmatrix} s_{\boldsymbol{\omega}} \\ s_{\mathbf{v}} \end{bmatrix} + \begin{bmatrix} K_{\boldsymbol{\omega}} & 0 \\ 0 & K_{\mathbf{v}} \end{bmatrix} \begin{bmatrix} s_{\boldsymbol{\omega}} \\ s_{\mathbf{v}} \end{bmatrix} = 0 \tag{7.11}$$

Proof. The inertia matrix M_b, due to the properties of positive definiteness and uniform boundedness, is used to compute the norm $\mathcal{V} = \frac{1}{2} \left[\begin{smallmatrix} \delta s_{\boldsymbol{\omega}} \\ \delta s_{\mathbf{v}} \end{smallmatrix} \right]^\top M_b \left[\begin{smallmatrix} \delta s_{\boldsymbol{\omega}} \\ \delta s_{\mathbf{v}} \end{smallmatrix} \right]$, for Lyapunov-like stability analysis [158, 159]. The closed-loop system in (7.11) has two particular solutions $[s_{\boldsymbol{\omega}}, s_{\mathbf{v}}] = 0$ and $[s_{\boldsymbol{\omega}}, s_{\mathbf{v}}] = s_e = [\boldsymbol{\omega} - \boldsymbol{\omega}_r, \mathbf{v}_b - \mathbf{v}_{b_r}]$. We perform a squared length analysis using the norm, after obtaining the infinitesimal distance δs_e at fixed time. The derivative of the squared length is given in the following equation.

$$\dot{\mathcal{V}} = \frac{1}{2} \begin{bmatrix} \delta s_{\boldsymbol{\omega}} \\ \delta s_{\mathbf{v}} \end{bmatrix}^\top \dot{M}_b \begin{bmatrix} \delta s_{\boldsymbol{\omega}} \\ \delta s_{\mathbf{v}} \end{bmatrix} + \begin{bmatrix} \delta s_{\boldsymbol{\omega}} \\ \delta s_{\mathbf{v}} \end{bmatrix}^\top M_b \begin{bmatrix} \dot{\delta} s_{\boldsymbol{\omega}} \\ \dot{\delta} s_{\mathbf{v}} \end{bmatrix} \tag{7.12}$$

Using the closed-loop dynamics and $s^\top \left(\dot{M}_b - 2C_b \right) s = 0$, on the right hand side of the above equation we get the following.

$$\begin{aligned} \dot{\mathcal{V}} &= \frac{1}{2} \begin{bmatrix} \delta s_{\boldsymbol{\omega}} \\ \delta s_{\mathbf{v}} \end{bmatrix}^\top \left(\dot{M}_b - 2C_b - 2 \begin{bmatrix} K_{\boldsymbol{\omega}} & 0 \\ 0 & K_{\mathbf{v}} \end{bmatrix} \right) \begin{bmatrix} \delta s_{\boldsymbol{\omega}} \\ \delta s_{\mathbf{v}} \end{bmatrix} \\ &= - \begin{bmatrix} \delta s_{\boldsymbol{\omega}} \\ \delta s_{\mathbf{v}} \end{bmatrix}^\top \begin{bmatrix} K_{\boldsymbol{\omega}} & 0 \\ 0 & K_{\mathbf{v}} \end{bmatrix} \begin{bmatrix} \delta s_{\boldsymbol{\omega}} \\ \delta s_{\mathbf{v}} \end{bmatrix} \end{aligned} \tag{7.13}$$

With the inertia matrix bounds and $K = \min\{K_{\boldsymbol{\omega}}, K_{\mathbf{v}}\}$, we get the following inequality.

$$\dot{\mathcal{V}} \leq -\frac{2K}{\lambda_{\max}\{M_b\}} \mathcal{V} \tag{7.14}$$

Using Comparison Lemma [158] and the bounds on the inertia matrix M_b, we obtain:

$$\begin{aligned} \mathcal{V}(t) &\leq \exp\left(-\frac{2K}{\lambda_{\max}\{M_b\}} t \right) \mathcal{V}(0) \\ \left\| \left[\begin{smallmatrix} \delta s_{\boldsymbol{\omega}} \\ \delta s_{\mathbf{v}} \end{smallmatrix} \right] \right\|_2 &\leq \sqrt{\frac{\lambda_{\max}\{M_b\}}{\lambda_{\min}\{M_b\}}} \exp\left(-\frac{K}{\lambda_{\max}\{M_b\}} t \right) \left\| \left[\begin{smallmatrix} \delta s_{\boldsymbol{\omega}}(0) \\ \delta s_{\mathbf{v}}(0) \end{smallmatrix} \right] \right\|_2 \end{aligned} \tag{7.15}$$

It follows from contraction-based incremental stability analysis [159, 160] that all the system trajectories of the closed-loop system in (7.11) converge exponentially fast with rate $\frac{2K}{\lambda_{\max}\{M_b\}}$, i.e. $\delta s_e \to 0$, which implies $[\boldsymbol{\omega} - \boldsymbol{\omega}_r, \mathbf{v}_b - \mathbf{v}_{b_r}] \to 0$. With converged virtual dynamics, from the definition of reference trajectories in (7.9), we get the following equations.

$$\begin{aligned}\boldsymbol{\omega} - \boldsymbol{\omega}_r &= \mathbf{Z}^{-1}(\mathbf{p}_{\text{mrp}})(\dot{\mathbf{p}}_{\text{mrp}} - \dot{\mathbf{p}}_{\text{mrp}_{\text{des}}}(t)) + \mathbf{Z}^{-1}(\mathbf{p}_{\text{mrp}})\Lambda_{\boldsymbol{\omega}}(\mathbf{p}_{\text{mrp}}(t) - \mathbf{p}_{\text{mrp}_{\text{des}}}(t)) \\ \mathbf{v}_b - \mathbf{v}_{b_r} &= R^\top(\dot{\mathbf{p}}_{\text{p}} - \dot{\mathbf{p}}_{\text{p}_{\text{des}}}(t)) + R^\top\Lambda_{\mathbf{v}}(\mathbf{p}_{\text{p}}(t) - \mathbf{p}_{\text{p}_{\text{des}}})\end{aligned} \tag{7.16}$$

From (7.16), it is clear that the attitude trajectory and the position trajectory converge globally exponentially fast to the desired trajectory as $[\boldsymbol{\omega} - \boldsymbol{\omega}_r, \mathbf{v}_b - \mathbf{v}_{b_r}] \to 0$. The closed-loop virtual dynamics with a bounded disturbance at input $d(t) = \left[\begin{smallmatrix} d_{\boldsymbol{\omega}}(t) \\ d_{\mathbf{v}}(t) \end{smallmatrix}\right]$ is as following:

$$M_b \begin{bmatrix} \dot{s}_{\boldsymbol{\omega}} \\ \dot{s}_{\mathbf{v}} \end{bmatrix} + C_b \begin{bmatrix} s_{\boldsymbol{\omega}} \\ s_{\mathbf{v}} \end{bmatrix} + \begin{bmatrix} K_{\boldsymbol{\omega}} & 0 \\ 0 & K_{\mathbf{v}} \end{bmatrix} \begin{bmatrix} s_{\boldsymbol{\omega}} \\ s_{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} d_{\boldsymbol{\omega}}(t) \\ d_{\mathbf{v}}(t) \end{bmatrix}. \tag{7.17}$$

The Lyapunov derivative with the closed-loop system as in (7.17) can be simplified to following equation using Cauchy-Schwarz inequality and the bounds on the inertia matrix M_b.

$$\dot{\mathcal{V}} = -\frac{2K}{\lambda_{\max}\{M_b\}}\mathcal{V} + \sqrt{\frac{2\mathcal{V}}{\lambda_{\min}\{M_b\}}}\|d(t)\|_2 \tag{7.18}$$

Using the transformation $\mathcal{W} = \sqrt{\mathcal{V}}$, followed by application of Comparison Lemma [158] and the uniform bounds on the inertia matrix M_b, we get the bounds on the norm of the virtual coordinates, where $\sup_{t\geq 0}|d(t)| \leq \gamma_\infty$.

$$\begin{aligned}\left\| \left[\begin{smallmatrix} \delta s_{\boldsymbol{\omega}} \\ \delta s_{\mathbf{v}} \end{smallmatrix}\right] \right\|_2 \leq \sqrt{\frac{\lambda_{\max}\{M_b\}}{\lambda_{\min}\{M_b\}}} \exp\left(-\frac{K}{\lambda_{\max}\{M_b\}}t\right) \left\| \left[\begin{smallmatrix} \delta s_{\boldsymbol{\omega}}(0) \\ \delta s_{\mathbf{v}}(0) \end{smallmatrix}\right] \right\|_2 \\ + \frac{\lambda_{\max}\{M_b\}\gamma_\infty}{\lambda_{\min}\{M_b\}K}\left(1 - \exp\left(-\frac{Kt}{\lambda_{\max}\{M_b\}}\right)\right)\end{aligned} \tag{7.19}$$

By taking a limit $t \to \infty$, we get the bounds on the virtual states $\left\| \left[\begin{smallmatrix} \delta s_{\boldsymbol{\omega}} \\ \delta s_{\mathbf{v}} \end{smallmatrix}\right] \right\|_2 \to \frac{\lambda_{\max}\{M_b\}\gamma_\infty}{\lambda_{\min}\{M_b\}K}$. Thus the exponentially stable closed-loop virtual dynamics, (7.11), is finite-gain $\mathcal{L}_p$ stable and Input-to-State Stable (ISS) for a bounded disturbance $d \in \mathcal{L}_p$ at the input [159]. □

7.2.4 Control Implementation

For the control implementation, it is assumed that the attitude stage is coarsely balanced with small r_{cg}. Equation (7.20) shows the decoupled translation dynamics

in inertial frame and rotational dynamics in body frame with small center of gravity offset. The terms in the dynamics corresponding to the r_{cg} act as a bounded disturbance at the input $d(t) = \left[\begin{smallmatrix} d_w(t) \\ d_p(t) \end{smallmatrix}\right]$ for small accelerations.

$$\begin{bmatrix} J & 0 & 0 \\ 0 & m_a + m_t & 0 \\ 0 & 0 & m_a + m_t \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\omega}} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega} \times J\boldsymbol{\omega} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -m_a g \mathbf{S}(r_{cg}) R(\mathbf{p}_{\text{mrp}})^\top a_3 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_r \\ \boldsymbol{\tau}_p \end{bmatrix} + \begin{bmatrix} d_w(t) \\ d_p(t) \end{bmatrix} \tag{7.20}$$

$$\begin{aligned} d_w(t) &= -m_a \mathbf{S}(r_{cg}) R(\mathbf{p}_{\text{mrp}})^\top a_1 \ddot{x} - m_a \mathbf{S}(r_{cg}) R(\mathbf{p}_{\text{mrp}})^\top a_2 \ddot{y} \\ d_p(t) &= \begin{bmatrix} -m_a (\mathbf{S}(r_{cg}) R(\mathbf{p}_{\text{mrp}})^\top a_1)^\top \dot{\boldsymbol{\omega}} - m_a a_1^\top R(\mathbf{p}_{\text{mrp}}) \mathbf{S}(\boldsymbol{\omega})^2 r_{cg} \\ -m_a (\mathbf{S}(r_{cg}) R(\mathbf{p}_{\text{mrp}})^\top a_2)^\top \dot{\boldsymbol{\omega}} - m_a a_2^\top R(\mathbf{p}_{\text{mrp}}) \mathbf{S}(\boldsymbol{\omega})^2 r_{cg} \end{bmatrix} \end{aligned} \tag{7.21}$$

A hierarchical control law was implemented with an inner attitude control loop and an outer position control loop because of the timescale separation between the two dynamics, (7.20). Given a desired position trajectory, $[x_{\text{des}}(t), y_{\text{des}}(t)] \in \mathbb{R}^2$, and attitude trajectory represented in MRPs, $\mathbf{p}_{\text{mrp}_{\text{des}}}(t) \in \mathbb{R}^3$, the control law presented below exponentially tracks both position and attitude trajectories using smooth control inputs for the decoupled dynamics for no disturbance. In the case with a bounded disturbance at the input, the closed-loop system is finite-gain $\mathcal{L}_p$ stable. The control input to the position dynamics is simplified from (7.10) and is given by (7.22).

$$\boldsymbol{\tau}_p = (m_t + m_a) \begin{bmatrix} \ddot{x}_{\text{des}} \\ \ddot{y}_{\text{des}} \end{bmatrix} - K_d \begin{bmatrix} \dot{x} - \dot{x}_{\text{des}} \\ \dot{y} - \dot{y}_{\text{des}} \end{bmatrix} - K_p \begin{bmatrix} x - x_{\text{des}} \\ y - y_{\text{des}} \end{bmatrix} \tag{7.22}$$

$$(m_t + m_a) \begin{bmatrix} \ddot{x} - \ddot{x}_{\text{des}} \\ \ddot{y} - \ddot{y}_{\text{des}} \end{bmatrix} - K_d \begin{bmatrix} \dot{x} - \dot{x}_{\text{des}} \\ \dot{y} - \dot{y}_{\text{des}} \end{bmatrix} - K_p \begin{bmatrix} x - x_{\text{des}} \\ y - y_{\text{des}} \end{bmatrix} = d_p(t) \tag{7.23}$$

The closed-loop position dynamics with the control law in (7.22) are given in (7.23). The gain values K_d and K_p are chosen to achieve the required position tracking performance. The attitude controller in (7.24) is exponentially stable [161] with no disturbance and tracks a given desired attitude trajectory that is $\mathcal{C}^2$ continuous. It can be shown that this control law is simplified form of the controller proposed

in (7.10). The nonlinear controller is finite-gain $\mathcal{L}_p$ stable with bounded disturbance at the input.

$$\begin{aligned}\boldsymbol{\tau}_r &= J\dot{\boldsymbol{\omega}}_r - \mathbf{S}(J\boldsymbol{\omega})\boldsymbol{\omega}_r - K_r(\boldsymbol{\omega} - \boldsymbol{\omega}_r) - m_a g\mathbf{S}(r_{cg})R(\mathbf{p}_{\mathrm{mrp}})^\top a_3 \\ \boldsymbol{\omega}_r &= \mathbf{Z}^{-1}(\mathbf{p}_{\mathrm{mrp}})\dot{\mathbf{p}}_{\mathrm{mrp}_{\mathrm{des}}}(t) + \mathbf{Z}^{-1}(\mathbf{p}_{\mathrm{mrp}})\Lambda_r(\mathbf{p}_{\mathrm{mrp}_{\mathrm{des}}(t)} - \mathbf{p}_{\mathrm{mrp}})\end{aligned} \tag{7.24}$$

$$J(\dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_r) - \mathbf{S}(J\boldsymbol{\omega})(\boldsymbol{\omega} - \boldsymbol{\omega}_r) - K_r(\boldsymbol{\omega} - \boldsymbol{\omega}_r) = d_w(t) \tag{7.25}$$

The closed-loop attitude dynamics are given in the (7.25). The matrices Λ_r and K_r are positive definite and are chosen to achieve required tracking performance. The control laws presented above compute control signals which are at least $\mathcal{C}^2$ continuous and the number of control inputs are collocated with the states. Considering the overactuated design of the simulator and the impulse actuation of the thrusters, a transformation from the continuous control signal to the thruster on-off times is required to achieve equivalent performance with non-smooth control inputs. In the following two sections, we discuss the actuator models for thrusters and reaction wheels to make this transformation, along with the influence matrices due to the location of the actuators.

7.2.5 Thruster Model and Influence Matrix

Influence Matrix.

Equations (7.22) and (7.24) give force and torque inputs that need to be applied collocated with the five degrees of freedom of the system. The spacecraft has 16 thrusters mounted in the configuration shown in Fig. 7.8, with thrusters 1-8 used for position and yaw angle control, and 9-16 used for roll and pitch angle control. The collocated force and torque inputs from the control law are transformed to the force input requirements on each of the 16 actuators through control allocation using an influence matrix. For the position controller, the following is the actuator input to control input mapping called the influence matrix.

$$\boldsymbol{\tau}_p = R(\mathbf{p}_{\mathrm{mrp}})\mathbf{B}_p\mathbf{f}_{p_{\mathrm{thr}}} \tag{7.26}$$

In the equation 7.26, $R(\mathbf{p}_{\mathrm{mrp}})$ transforms the actuator input in the body frame to the inertial frame. $\mathbf{B}_p$ corresponds to the influence matrix given by (7.27) for position control. The force vector, $\mathbf{f}_{p_{\mathrm{thr}}} = [f_1\ f_2\ f_3\ f_4\ f_5\ f_6\ f_7\ f_8]^\top$, acts as the input to the spacecraft dynamics simulator thrusters mounted for position and yaw control. The actuator numbering is shown in Fig. 7.8.

$$\mathbf{B}_p = \begin{bmatrix} -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{7.27}$$

For attitude control, the thruster force to control input mapping is given as follows.

$$\boldsymbol{\tau}_r = \mathbf{B}_r \mathbf{f}_{\text{thr}} \quad \text{where } \mathbf{B}_r = [\mathbf{B}_1 \ \mathbf{B}_2] \text{ and } \mathbf{f}_{\text{thr}} = \begin{bmatrix} \mathbf{f}_{p_{\text{thr}}}^\top & \mathbf{f}_{a_{\text{thr}}}^\top \end{bmatrix}^\top \tag{7.28}$$

where $\mathbf{f}_{a_{\text{thr}}} = [f_9 \ f_{10} \ f_{11} \ f_{12} \ f_{13} \ f_{14} \ f_{15} \ f_{16}]^\top$. Also, see Fig. 7.8 for the thruster numbering and nomenclature of ℓ, b, and h.

$$\begin{aligned} \mathbf{B}_1 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\ell & \ell & -b & b & -\ell & \ell & -b & b \end{bmatrix}, \\ \mathbf{B}_2 &= \begin{bmatrix} 0 & 0 & h & -h & 0 & 0 & -h & h \\ h & -h & 0 & 0 & -h & h & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \end{aligned} \tag{7.29}$$

Control Allocation.

The control allocation scheme for the position controller computes the 8 dimensional thruster forces F_1 given the transformation matrix $R(\mathbf{p}_{\text{mrp}})$ and the influence matrix $\mathbf{B}_p$. A generalized right psuedo-inverse solution to the control allocation problem that minimizes the $\mathcal{L}^2$-norm of the control effort is given by $\mathbf{f}_{p_{\text{thr}}} = \mathbf{B}_p^\top \left(\mathbf{B}_p \mathbf{B}_p^\top\right)^{-1} R(\mathbf{p}_{\text{mrp}})^{-1} \boldsymbol{\tau}_p$ and weighted pseudo-inverse is given in (7.30), with a diagonal weighing matrix $\mathbf{W}$. We use this algorithm for real-time implementation.

$$\mathbf{f}_{p_{\text{thr}}} = \mathbf{W}(\mathbf{B}_p \mathbf{W})^\top \left(\mathbf{B}_p \mathbf{W} (\mathbf{B}_p \mathbf{W})^\top\right)^{-1} R(\mathbf{p}_{\text{mrp}})^{-1} \boldsymbol{\tau}_p \tag{7.30}$$

The elements of the diagonal matrix $\mathbf{W}$ can be chosen to take into account actuator saturation limits. For example, given the maximum $u_{\max}$ and minimum $u_{\min}$ thrust that can be produced by the thruster f_1, we choose the corresponding diagonal element in $\mathbf{W}$ as $\frac{1}{|u_{\max} - u_{\min}|}$. For attitude control using thrusters and reaction wheels, the same approach can be used for computing the actuator force.

Thruster Firing Time.

The continuous actuator force computed using the control allocation scheme needs to be transformed to the thruster firing times because the thrusters on the spacecraft simulator are on-off non-latching solenoids. The on time of the thrusters is controlled using a PWM signal with the duty cycle mapped to the on time requirements. Consider a PWM signal with frequency f_{pwm} with duty cycle corresponding

to firing time Δt, and continuous force F_r that needs to be applied by a thruster at time step t. Let F_a be the force applied by the thruster when open/on and the control loop frequency be f_{cl}. It is assumed that control frequency is the same as the PWM signal frequency. The firing time is given in the following equation.

$$\Delta t = \frac{F_r}{f_{\mathrm{cl}} F_a} \tag{7.31}$$

The equation above assumes that the thruster produces the same force for all firing times. To verify this claim and validate the model, an experimental setup was built as shown in Fig. 7.9. In the following section, we discuss the details of the experimental setup and the thruster model obtained from experiments.

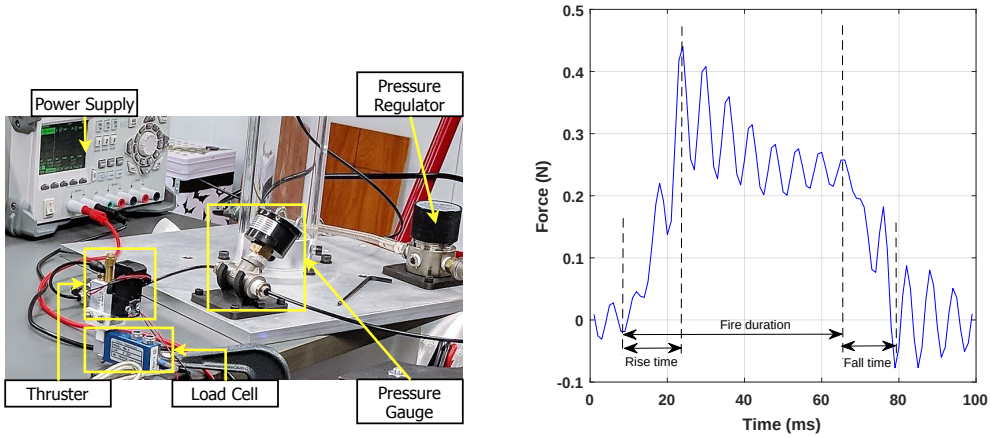
Experimental Characterization of the Solenoidal Thruster.

The experimental setup built to characterize the performance of the solenoidal thrusters includes a thruster mounted on a load cell with a regulated power supply. An instrumental amplifier is used to amplify the load cell output voltage, and the amplified voltage is sampled by a dSPACE MicroLabBox at $1\,\mathrm{kHz}$. The thruster is supplied with pressure-regulated compressed air at 40 psi, 50 psi, or 60 psi, the three operating pressures tested. The thrusters are fired for a multiple of $10\,\mathrm{ms}$ between $10\,\mathrm{ms}$ and $80\,\mathrm{ms}$, and the load cell force is recorded as a function of time for the duration of firing, as shown in Fig. 7.9b.

Figure 7.10a shows the experimental data and the linear fit of the average force produced by the thruster for varying firing time. The force produced is not constant due to the nonlinear behaviour of the thruster. The impulse produced with varying firing time is linear as seen in Fig. 7.10b. For the control implementation, we use the linear fit equations in Table 7.5 to compute the firing time of a thruster for a given force F_r that needs to be applied at time t with control loop frequency f_{cl}.

Operating Pressure (psi)	Fit equation
40	Δt = 7.863 $\frac{F_r}{f_{\mathrm{cl}}}$ -0.009727
50	Δt =4.829$\frac{F_r}{f_{\mathrm{cl}}}$ -0.007686
60	Δt =3.51$\frac{F_r}{f_{\mathrm{cl}}}$ -0.006035

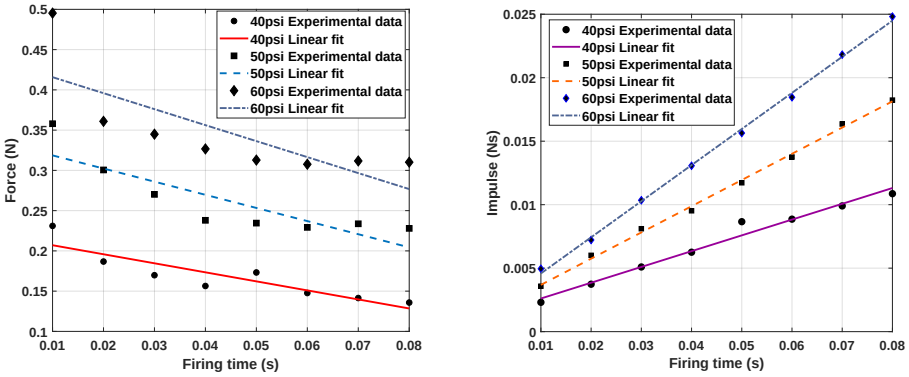
Table 7.5: Linear fit equations for firing time computation for a given control signal F_r at time t and control frequency f_{cl}.



(a) Thruster characterization experimental setup.

(b) Example force output for $50\,\mathrm{ms}$ fire duration.

Figure 7.9: Thruster characterization setup and sample results recorded by the load cell, showing rise time and fall time.



(a) Thruster firing time vs. average force.

(b) Thruster firing time vs. average impulse.

Figure 7.10: Experimental data and linear fit of average force and impulse vs. thruster firing time at 40 psi, 50 psi, and 60 psi operating pressure.

7.2.6 Reaction Wheel Configuration and Model

The simulator is equipped with four reaction wheels for attitude control arranged in a pyramid configuration (see Fig. 7.8). The angle α_{w} made by the axis of the wheel and the (X_b, Y_b) plane is chosen to have maximum momentum storage [162], $\alpha_{\mathrm{w}} = 35.26°$. The overactuated configuration will be used to study the fault detection, isolation, and recovery of reaction wheels, which is a major source of failure [163] in flight missions. The attitude dynamics with four reaction wheels in the pyramid configuration and no gravity torques is given in (7.32). The influence matrix is

given by G in (7.33).

$$J\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times J\boldsymbol{\omega} = -GJ_{\boldsymbol{\omega}}\dot{\boldsymbol{\omega}} - \boldsymbol{\omega} \times GJ_{\boldsymbol{\omega}}\boldsymbol{\omega} \tag{7.32}$$

$$G = \begin{bmatrix} c(\alpha_{\mathrm{w}})c(45^\circ) & -c(\alpha_{\mathrm{w}})c(45^\circ) & -c(\alpha_{\mathrm{w}})c(45^\circ) & c(\alpha_{\mathrm{w}})c(45^\circ) \\ c(\alpha_{\mathrm{w}})s(45^\circ) & c(\alpha_{\mathrm{w}})s(45^\circ) & -c(\alpha_{\mathrm{w}})s(45^\circ) & -c(\alpha_{\mathrm{w}})s(45^\circ) \\ s(\alpha_{\mathrm{w}}) & s(\alpha_{\mathrm{w}}) & s(\alpha_{\mathrm{w}}) & s(\alpha_{\mathrm{w}}) \end{bmatrix} \tag{7.33}$$

$$J_{\boldsymbol{\omega}} = \begin{bmatrix} J_{w1} & 0 & 0 & 0 \\ 0 & J_{w2} & 0 & 0 \\ 0 & 0 & J_{w3} & 0 \\ 0 & 0 & 0 & J_{w4} \end{bmatrix} \tag{7.34}$$

In the above equation, J is the mass moment of inertia including the four wheels, $J_{\boldsymbol{\omega}}$ is a diagonal matrix with the mass moment of inertia of the wheels about the rotation axis, $\boldsymbol{\omega} = [\boldsymbol{\omega}_1\ \boldsymbol{\omega}_2\ \boldsymbol{\omega}_3\ \boldsymbol{\omega}_4]^\top$ is the rotation speed of the each of the four wheels, and $s(\cdot), c(\cdot)$ denote the sine and cosine of a given angle, respectively. For the numbering and location of the wheels with respect to body frame see Fig. 7.8. The term $-GJ_{\boldsymbol{\omega}}\dot{\boldsymbol{\omega}}$ is the control input to the attitude dynamics. The attitude controller in (7.24) is modified to cancel the cross-coupling term $-\boldsymbol{\omega} \times GJ_{\boldsymbol{\omega}}\boldsymbol{\omega}$ by feeding the wheel speed to the control law. The final control law is given in the following equation.

$$\boldsymbol{\tau}_b = M_b \begin{bmatrix} \dot{\boldsymbol{\omega}}_r \\ \dot{\mathbf{v}}_{b_r} \end{bmatrix} + C_b \begin{bmatrix} \boldsymbol{\omega}_r \\ \mathbf{v}_{b_r} \end{bmatrix} + H_b - \begin{bmatrix} \mathbf{S}(GJ_{\boldsymbol{\omega}}\boldsymbol{\omega}) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_r \\ \mathbf{v}_{b_r} \end{bmatrix} - \begin{bmatrix} K_{\boldsymbol{\omega}} & 0 \\ 0 & K_{\mathbf{v}} \end{bmatrix} \begin{bmatrix} s_{\boldsymbol{\omega}} \\ s_{\mathbf{v}} \end{bmatrix} \tag{7.35}$$

The wheel torques can be computed using the generalized pseudo-inverse from the control inputs. The reaction wheels are designed to run at the nominal speeds $[-2500, 2500, -2500, 2500]$ rpm, which is the null space of the influence matrix G, to avoid excitation of the attitude dynamics. The speed control of the wheel is done using a Hall sensor integrated into the selected Maxon motor.

7.2.7 Hardware Implementation of the Hierarchical Control Law

In this section, we elaborate on the implementation of the hierarchical control law discussed earlier. The schematic of the control law is shown in Fig. 7.11. The attitude control is done in the inner-loop with control frequency between $80 - 100$ Hz using reaction wheels. The thrusters can be used to do coarse attitude control, or desaturate the reaction wheels. The X, Y position controller is done using

thrusters, it is coupled with the attitude dynamics by a rotation matrix to map the actuator force in the body frame to the inertial frame. The position dynamics are slow compared to the attitude dynamics, so it is run as an outer-loop with feedback on position data for control computations and attitude data for control allocation at control frequency between $1 - 10$ Hz.

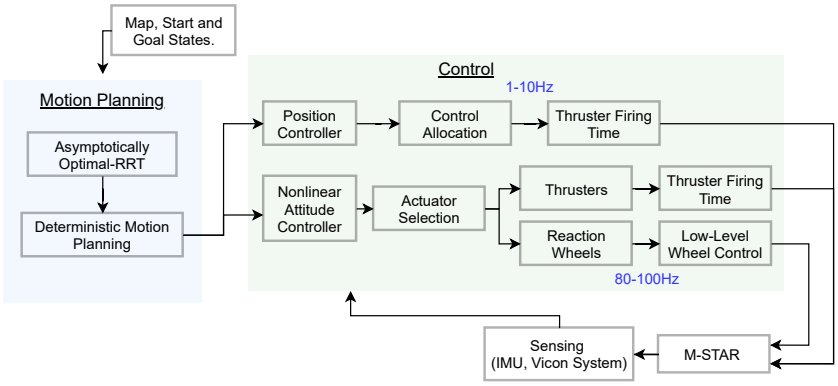


Figure 7.11: Closed-loop control implementation for the 6DOF simulator.

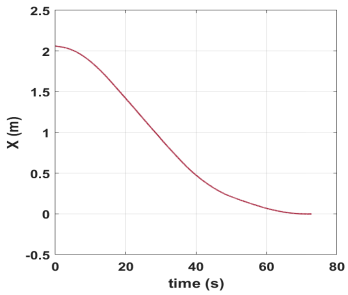
7.3 Experiments

In this section, we present the preliminary experimental results for the position tracking controller discussed earlier. Here, we try to track a step input and demanding harmonic trajectories using the control law, control allocation and firing time schemes developed in this Chapter. The position and orientation data of the simulator is measured using the motion capture camera system running at 100 Hz. The thrusters are operated at 50 psi. The tracking results are discussed in the following section. In a recent work [164], we demonstrated in-orbit spacecraft assembly using the controller discussed in section 7.2.7.

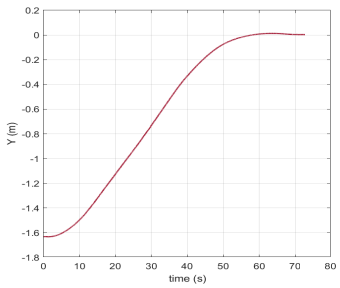
7.3.1 Results

Figures 7.12 and 7.13 show preliminary results of waypoint reaching experiments. The task for the controller was to reach origin of the inertial frame and stay there until a further command was communicated. The controller performs well for the two presented cases. The current position controller can be easily extended for tracking a trajectory with coarse way points. The steady-state error in both of the cases was less than the assigned value of 5cm. In this particular test the yaw angle attitude was coarsely maintained around 0, except when the system faced perturbations from uneven flow and varying pressure in the pressure manifold that supplies air to the thrusters, which caused a couple on the simulator due to firing forces that

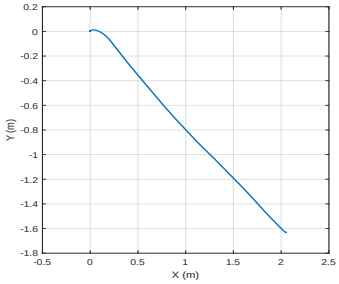
do not balance. Further investigation into characterizing the viscous friction due to air gap between the simulator and the epoxy floor, and the dead zone of the thrusters needs to be done to improve the performance of the controller.



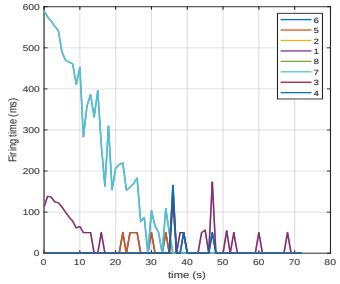
(a) x position (m) vs. time (s).



(b) y position (m) vs. time (s).

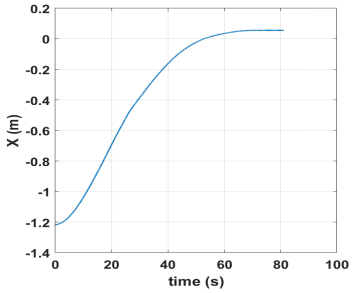


(c) (x,y) trajectory (m).

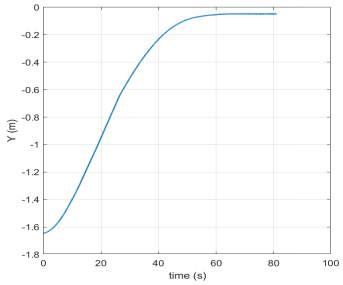


(d) Thruster firing time vs. time.

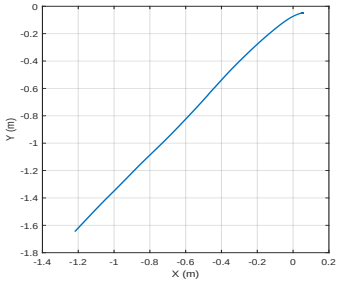
Figure 7.12: Closed-loop waypoint reaching experimental result- test case 1.



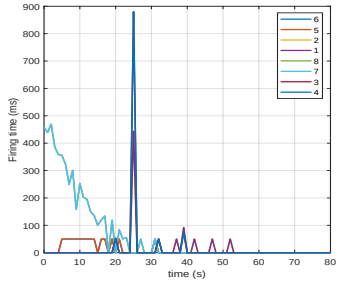
(a) x position (m) vs. time (s).



(b) y position (m) vs. time (s).



(c) (x,y) trajectory in m.



(d) Thruster firing time vs. time.

Figure 7.13: Closed-loop waypoint reaching experimental result- test case 2.

7.4 Chapter Summary

In this chapter, we discussed the hardware development of a 6-DOF robotic spacecraft simulator M-STAR for testing formation guidance, navigation and control algorithms. The simulator has 6-DOF with translation and attitude stages decoupled using a spherical air bearing. The translation stage floats on the epoxy flat floor using three flat round air bearings. The hardware architecture of M-STAR and its subsystems including mechanical structure, pneumatic system for flat air bearings, spherical air bearing required to achieve frictionless and disturbance torque free motion of the simulator were discussed in detail. The low level control architecture for thrusters and reaction wheels was mentioned for controlling the dynamics.

A nonlinear dynamic model of M-STAR was presented by modelling the system as a 3D pendulum on a floating platform. A hierarchical model-based control law for the nonlinear system was discussed for tracking a given position and attitude trajectory. A generalized pseudo-inverse control allocation scheme, with a thruster actuator model developed using experiments, was used to implement the control law in a ROS based software framework for testing position control. Future work will focus on multi-agent guidance and control experiments exercising all five M-STARs.

Chapter 8

CONCLUSION

In this thesis, we integrated an information cost with the traditional optimal control problems to demonstrate on-orbit coordinated inspection and safe exploration. We designed efficient algorithms for motion planning and control under uncertainty. In Chapter 2, we studied the fundamentals of formation flying, optimal orbit initialization and reconfiguration, and generalized polynomial chaos approach for uncertainty propagation.

In Chapter 3, we presented an information-based guidance and control architecture that consists of multiple timescale modules for cooperatively observing and inspecting a target spacecraft. The proposed methodology integrates optimal orbit initialization and reconfiguration, attitude planning and control, and a discrete information-cost optimal orbit selection for inspecting the target spacecraft. The information cost and a replanning strategy were used to guarantee the coverage of the target spacecraft and balance between the gathered data quality and control cost. During the inspection, we confirm if an area of the target spacecraft has been inspected or not, using our novel raycasting database to efficiently check the visibility of POIs. The updated database was used to monitor the progress of inspection. The effectiveness of the proposed architecture was validated by inspecting a target spacecraft in Earth orbit using multiple observer spacecraft. We demonstrate the proposed planning approach on a three-degree-of-freedom robotic spacecraft testbed for inspecting a target spacecraft. Using the architecture, we developed an effective mission design tool that incorporates autonomy for adaptation and safety.

The computational complexity of the information-based guidance and control architecture presented in Chapter 3 is upper bounded by the complexity of the assignment problem. We use a preliminary Hungarian method [136] to demonstrate the architecture. Distributed assignment methods [14] can be used to improve the speed of assignment and the overall architecture for online implementation.

In Chapter 4, we presented a generalized polynomial chaos-based sequential convex programming method for safe and optimal motion planning and control under uncertainty in dynamics and constraints. We used generalized polynomial chaos projection and distributional robustness to compute a convex subset of the multi-model

state-dependent chance constraints, and a high-fidelity deterministic surrogate of the stochastic dynamics and the cost functional. We proposed an algorithm to solve the surrogate deterministic optimal problem using sequential convex programming for trajectory optimization. We studied the controllability of the surrogate deterministic dynamics and proposed approaches to ensure the feasibility of the optimization problem. We proved the asymptotic convergence of the surrogate problem to the stochastic optimal control problem, thereby demonstrating the suboptimality of the proposed approach.

In Chapter 5, we derived deterministic surrogate convex constraints for collision checking with deterministic and stochastic obstacles. We integrated the gPC-SCP method with these constraint formulations and derived a motion planning algorithm to compute safe motion plans under uncertainty in dynamics and obstacle location. We extended this method to design a stochastic model predictive control for safely tracking a nominal trajectory which was computed by ignoring the uncertainty. We proved the convergence and stability of the stochastic model predictive controller and validated our approach in simulations and on the robotic spacecraft simulator hardware. We demonstrated a higher success rate in the safety of motion trajectories compared to a Gaussian approximation of the chance constraints.

The gPC-SCP method discussed in Chapter 5 can guarantee safety for risk measure $\epsilon \in [0.001, 0.5]$. For small risk measure ϵ ($< 1e-3$), the convex deterministic surrogate derived in this thesis might be overly conservative and might reduce the feasibility significantly, leading to the infeasibility of the optimization problem. Further research needs to be conducted in this direction to handle low-risk $\epsilon < 1e-7$.

In Chapter 6, we presented the Info-SNOC algorithm for safe exploration. In this method, we integrate a learning-based model and motion planning by solving information-cost stochastic optimal control using a partially learned nonlinear dynamical model and formulated safety as distributionally robust chance constraints. We solved the Info-SNOC problem using the gPC-SCP method descrived in Chapters 4 and 5. We used the Info-SNOC method to compute a safe and information-rich pool of trajectories for rollout using an exponentially stable controller with a safety filter augmentation for safe data collection. We analyzed safety during rollout under learned dynamics and showed that the learned model converges to the optimal predictor over epochs. The safe exploration architecture presented in Chapter 6 is limited by the range of risk measures that can be used in the gPC-SCP method

and the speed of the learning method used to learn and predict the unknown interaction. Fast Bayesian learning methods (for example, [165]) can be incorporated to improve the learning speed.

In Chapter 7, we described the hardware development of the 6-DOF M-STAR testbed. We discussed the construction and modularity of the platform that enables flexible formation guidance, navigation, and control algorithm testing. We presented the dynamic modelling, control algorithm development and implementation in software and on hardware.

8.1 Future Work

Mission Design for Autonomy

As discussed in Section 3.3, the information-based autonomy architecture could be an essential tool for mission design. While the architecture in Chapter 3 attempts to study mission design from a safety vs. information perspective, there is still significant work to design other aspects of autonomy. In particular, this architecture could prove useful to answer questions such as "what, how, and when to communicate?" and "can we achieve mission success under partial subsystem faults?". Furthermore, the experimental result provided in Section 3.4, can be applied to ground robotic systems for novelty detection for efficient safe exploration.

Fast Approximations For Motion Planning Under Uncertainty

Although the offline gPC projection approach described in Section 4.2.5 allows for motion planning and control under uncertainty in both dynamics and environment, it is computationally expensive for large degree-of-freedom systems. Alternatively, there is scope to use efficient numerical integration methods to develop an implementation that avoids the Galerkin projection. Furthermore, there is need to extend the proposed algorithms for multi-agent motion planning and control for safe on-orbit reconfiguration.

BIBLIOGRAPHY

[1] Andrew M Long, Matthew G Richards, and Daniel E Hastings. On-orbit servicing: a new value proposition for satellite design and operation. *Journal of Spacecraft and Rockets*, 44(4):964–976, 2007.

[2] Joshua P Davis, John P Mayberry, and Jay P Penn. On-orbit servicing: Inspection, repair, refuel, upgrade, and assembly of satellites in space. report, The Aerospace Corporation, 2019.

[3] Danielle Piskorz and Karen L. Jones. On-orbit assembly of space assets: A path to affordable and adaptable space infrastructure. report, The Aerospace Corporation, 2018.

[4] Brian Banker and Roger Askew. Seeker 1.0: Prototype robotic free flying inspector mission overview. In *33rd Annual Conference on Small Satellites*, 2019.

[5] Robert B. Friend. Orbital express program summary and mission overview. In Richard T. Howard and Pejmun Motaghedi, editors, *Sensors and Systems for Space Applications II*, volume 6958, pages 11 – 21. International Society for Optics and Photonics, SPIE, 2008.

[6] Rebecca Foust, Soon-Jo Chung, and Fred Hadaegh. Autonomous in-orbit satellite assembly from a modular heterogeneous swarm using sequential convex programming. In *AIAA/AAS Astrodynamics Specialist Conference*, 2016.

[7] Amir Rahmani, Saptarshi Bandyopadhyay, Federico Rossi, Michael Wolf, Jean-Pierre de la Croix, and Josh Vander Hook. Swarm of space vehicles and future opportunities. report, NASA Jet Propulsion Lab, 2018.

[8] Kiril Solovey and Dan Halperin. On the hardness of unlabeled multi-robot motion planning. *The International Journal of Robotics Research*, 35(14): 1750–1759, 2016.

[9] Jeffrey Kane Johnson. On the relationship between dynamics and complexity in multi-agent collision avoidance. *Autonomous Robots*, 42(7):1389–1404, 2018.

[10] Bruce Yost, Elwood Agasid, Roland Burton, Roberto Carlino, Gregory Defouw, Andres D. Perez, Andres D. Karacalıoğlu, Benjamin Klamm, Abraham Rademacher, James Schalkwyck, Rogan Shimmin, Julia Tilles, and Sasha Weston. Small spacecraft technology state of the art. report, NASA Ames Research Center, 2018.

[11] Angadh Nanjangud, Peter C Blacker, Saptarshi Bandyopadhyay, and Yang Gao. Robotics and ai-enabled on-orbit operations with future generation of small satellites. *Proceedings of the IEEE*, 106(3):429–439, 2018.

[12] Fred Y Hadaegh, Soon-Jo Chung, and Harish M Manohara. On development of 100-gram-class spacecraft for swarm applications. *IEEE Systems Journal*, 10(2):673–684, 2014.

[13] Saptarshi Bandyopadhyay, Rebecca Foust, Giri P Subramanian, Soon-Jo Chung, and Fred Y Hadaegh. Review of formation flying and constellation missions using nanosatellites. *Journal of Spacecraft and Rockets*, 53(3): 567–578, 2016.

[14] Daniel Morgan, Giri P Subramanian, Soon-Jo Chung, and Fred Y Hadaegh. Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming. *The International Journal of Robotics Research*, 35(10):1261–1285, 2016.

[15] Daniel Morgan, Soon-Jo Chung, and Fred Y Hadaegh. Model predictive control of swarms of spacecraft using sequential convex programming. *Journal of Guidance, Control, and Dynamics*, 37(6):1725–1740, 2014.

[16] Daniel Morgan, Soon-Jo Chung, Lars Blackmore, Behcet Acikmese, David Bayard, and Fred Y Hadaegh. Swarm-keeping strategies for spacecraft under j2 and atmospheric drag perturbations. *Journal of Guidance, Control, and Dynamics*, 35(5):1492–1506, 2012.

[17] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006. doi: 10.1017/CBO9780511546877.

[18] Mark W Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot modeling and control*, volume 3. Wiley New York, 2006.

[19] Kemin Zhou, John Comstock Doyle, and Keith Glover. *Robust and optimal control*, volume 40. Prentice hall New Jersey, 1996.

[20] Yashwanth Kumar Nakka and Soon-Jo Chung. Trajectory optimization for chance-constrained nonlinear stochastic systems. In *IEEE Conference on Decision and Control (CDC)*, pages 3811–3818, 2019.

[21] Lars Blackmore, Masahiro Ono, Askar Bektassov, and Brian C Williams. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *IEEE Trans. Robot.*, 26(3):502–517, 2010.

[22] Lars Blackmore, Masahiro Ono, and Brian C Williams. Chance-constrained optimal path planning with obstacles. *IEEE Trans. Robot.*, 27(6):1080–1094, 2011.

[23] N. E. Du Toit and J. W. Burdick. Robot motion planning in dynamic, uncertain environments. *IEEE Trans. Robot.*, 28(1):101–115, 2012. doi: 10.1109/TRO.2011.2166435.

[24] H. Tsukamoto and S. J. Chung. Robust controller design for stochastic nonlinear systems via convex optimization. *IEEE Trans. Autom. Control*, pages 1–1, 2020. doi: 10.1109/TAC.2020.3038402.

[25] Hai Zhu and Javier Alonso-Mora. Chance-constrained collision avoidance for mavs in dynamic environments. *IEEE Trans. Robot. Autom. Lett.*, 4(2): 776–783, 2019.

[26] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Integrated task and motion planning in belief space. *Int.J. Robot. Res.*, 32(9-10):1194–1227, 2013.

[27] Yashwanth Kumar K. Nakka, Wolfgang Hönig, Changrak Choi, Alexei Harvard, Amir Rahmani, and Soon-Jo Chung. Information-based guidance and control architecture for multi-spacecraft on-orbit inspection. In *AIAA Scitech 2021 Forum, Guidance Navigation and Control Conference*. doi: 10.2514/6.2021-1103. URL `https://arc.aiaa.org/doi/abs/10.2514/6.2021-1103`.

[28] 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 Trans. Robot. Autom. Lett.*, 6(2):389–396, 2021. doi: 10.1109/LRA.2020.3044033.

[29] K. P. Wabersich, L. Hewing, A. Carron, and M. N. Zeilinger. Probabilistic model predictive safety certification for learning-based control. *IEEE Trans. Autom. Control*, pages 1–1, 2021. doi: 10.1109/TAC.2021.3049335.

[30] Yashwanth Kumar Nakka, Rebecca C Foust, Elena Sorina Lupu, David B Elliott, Irene S Crowell, Soon-Jo Chung, and Fred Y Hadaegh. Six degree-of-freedom spacecraft dynamics simulator for formation control research. In *AAS/AIAA Astrodyn. Specialist Conf.*, pages 3367–3387, 2018.

[31] Jack Ridderhof and Panagiotis Tsiotras. Uncertainty quantication and control during mars powered descent and landing using covariance steering. In *AIAA GNC Conf.*, page 0611, 2018.

[32] Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural lander: Stable drone landing control using learned dynamics. In *Proc. IEEE Int. Conf. Robot. Autom.*, 2019.

[33] Guanya Shi, Wolfgang Hönig, Xichen Shi, Yisong Yue, and Soon-Jo Chung. Neural-swarm2: Planning and control of heterogeneous multirotor swarms using learned interactions. *IEEE Trans. Robot.*, 2021.

[34] Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural lander: Stable drone landing control using learned dynamics. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 9784–9790, 2019.

[35] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *Found. Comput. Math.*, pages 1–47, 2017.

[36] Chris J Ostafew, Angela P Schoellig, Timothy D Barfoot, and Jack Collier. Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. *J. Field Robot.*, 33(1):133–152, 2016.

[37] Ali Punjani and Pieter Abbeel. Deep learning helicopter dynamics models. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 3223–3230, 2015.

[38] H Levine, S Shaklan, and J Kasting. Terrestrial planet finder coronagraph science and technology definition team (stdt) report. *JPL document D-34923 (Pasadena: JPL)*, 2006.

[39] R. J. Barton. Mimo architectures for efficient communication in space. *arXiv preprint arXiv:1601.03664*, 2016.

[40] Sanda Mandutianu, Fred Hadaegh, and Paul Elliot. Multi-agent system for formation flying missions. In *Aerospace Conference, 2001, IEEE Proceedings.*, volume 6, pages 2793–2802. IEEE, 2001.

[41] Soon-Jo Chung, Aditya Avinash Paranjape, Philip Dames, Shaojie Shen, and Vijay Kumar. A survey on aerial swarm robotics. *IEEE Transactions on Robotics*, 34(4):837–855, 2018.

[42] Dongbin Xiu and George Em Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comp.*, 24(2):619–644, 2002.

[43] Dongbin Xiu. Fast numerical methods for stochastic computations: a review. *Com. in Comp. physics*, 5(2-4):242–272, 2009.

[44] Roger G Ghanem and Pol D Spanos. *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.

[45] Arkadi Nemirovski and Alexander Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2007.

[46] Giuseppe Carlo Calafiore and Laurent El Ghaoui. On distributionally robust chance-constrained linear programs. *Journal of Optimization Theory and Applications*, 130(1):1–22, 2006.

[47] Steve Zymler, Daniel Kuhn, and Berç Rustem. Distributionally robust joint chance constraints with second-order moment information. *Mathematical Programming*, 137(1-2):167–198, 2013.

[48] Daniel Morgan, Soon-Jo Chung, and Fred Hadaegh. Spacecraft swarm guidance using a sequence of decentralized convex optimizations. In *AIAA/AAS Astro. Spec. Conf.*, page 4583, 2012.

[49] Harold J Kushner. Stochastic stability and control. Technical report, Brown Univ Providence RI, 1967.

[50] Rafail Khasminskii. *Stochastic stability of differential equations*, volume 66. Springer Science & Business Media, 2011.

[51] K. Hauser and Y. Zhou. Asymptotically optimal planning by feasible kinodynamic planning in a state–cost space. *IEEE Trans. Robot.*, 32(6):1431–1443, 2016. doi: 10.1109/TRO.2016.2602363.

[52] Noel E Du Toit and Joel W Burdick. Probabilistic collision checking with chance constraints. *IEEE Trans. Robot.*, 27(4):809–815, 2011.

[53] Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-based model predictive control for safe exploration. In *Proc. IEEE Conf. Decis. Control*, pages 6059–6066, 2018.

[54] Ashwin P Dani, Soon-Jo Chung, and Seth Hutchinson. Observer design for stochastic nonlinear systems via contraction-based incremental stability. *IEEE Trans. Autom. Control*, 60(3):700–714, 2014.

[55] Xiangli Chen, Mathew Monfort, Anqi Liu, and Brian D Ziebart. Robust covariate shift regression. In *Artif. Intell. Stat.*, pages 1270–1279, 2016.

[56] Tor A Johansen and Thor I Fossen. Control allocation—a survey. *Automatica*, 49(5):1087–1103, 2013.

[57] Marc Bodson. Evaluation of optimization methods for control allocation. *Journal of Guidance, Control, and Dynamics*, 25(4):703–711, 2002.

[58] Allen Chen. Propulsion system characterization for the spheres formation flight and docking testbed. master's thesis, 2002.

[59] Dehann Fourie, Brent Tweddle, Steve Ulrich, and Alvar Saenz Otero. Vision-based relative navigation and control for autonomous spacecraft inspection of an unknown object. In *AIAA guidance, navigation, and control (GNC) conference*, page 4759, 2013.

[60] Dehann Fourie, Brent E Tweddle, Steve Ulrich, and Alvar Saenz-Otero. Flight results of vision-based navigation for autonomous spacecraft inspection of unknown objects. *Journal of Spacecraft and Rockets*, 51(6):2016–2026, 2014.

[61] Brent E Tweddle and Alvar Saenz-Otero. Relative computer vision-based navigation for small inspection spacecraft. *Journal of Guidance, Control, and Dynamics*, 38(5):969–978, 2015.

[62] M Sabatini, R Volpe, and GB Palmerini. Centralized visual based navigation and control of a swarm of satellites for on-orbit servicing. *Acta Astronautica*, 2020.

[63] Vincenzo Capuano, Kyunam Kim, Alexei Harvard, and Soon-Jo Chung. Monocular-based pose determination of uncooperative space objects. *Acta Astronautica*, 166:493–506, 2020.

[64] Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

[65] Nikolay Atanasov, Jerome Le Ny, Kostas Daniilidis, and George J Pappas. Information acquisition with sensing robots: Algorithms and error bounds. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6447–6454. IEEE, 2014.

[66] Stephen L Smith, Mac Schwager, and Daniela Rus. Persistent robotic tasks: Monitoring and sweeping in changing environments. *IEEE Transactions on Robotics*, 28(2):410–426, 2011.

[67] Nikolay Atanasov, Jerome Le Ny, Kostas Daniilidis, and George J Pappas. Decentralized active information acquisition: Theory and application to multi-robot slam. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4775–4782. IEEE, 2015.

[68] Mac Schwager, Philip Dames, Daniela Rus, and Vijay Kumar. A multi-robot control policy for information gathering in the presence of unknown hazards. In *Robotics research*, pages 455–472. Springer, 2017.

[69] Mac Schwager, Daniela Rus, and Jean-Jacques E. Slotine. Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment. *I. J. Robotics Res.*, 30(3):371–383, 2011.

[70] Emanuel Todorov and Weiwei Li. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proc. of American Control Conference*, pages 300–306, 2005.

[71] Jur Van Den Berg, Sachin Patil, and Ron Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *Int.J. Robot. Res.*, 31(11):1263–1278, 2012.

[72] Jack Ridderhof, Kazuhide Okamoto, and Panagiotis Tsiotras. Nonlinear uncertainty control with iterative covariance steering. In *IEEE 58th Conference on Decision and Control*, pages 3484–3490, 2019.

[73] Giuseppe C Calafiore and Lorenzo Fagiano. Stochastic model predictive control of lpv systems via scenario optimization. *Automatica*, 49(6):1861–1866, 2013.

[74] Lucas Janson, Edward Schmerling, and Marco Pavone. Monte carlo motion planning for robot trajectory optimization under uncertainty. In *Robotics Research*, pages 343–361. Springer, 2018.

[75] G. C. Calafiore and L. Fagiano. Robust model predictive control via scenario optimization. *IEEE Trans. Autom. Control*, 58(1):219–224, 2013. doi: 10.1109/TAC.2012.2203054.

[76] Ludwig Arnold. *Stochastic differential equations*. John Wiley & Sons, 1974.

[77] Manuel Castillo-Lopez, Philippe Ludivig, Seyed Amin Sajadi-Alamdari, Jose Luis Sanchez-Lopez, Miguel A Olivares-Mendez, and Holger Voos. A real-time approach for chance-constrained motion planning with dynamic obstacles. *IEEE Trans. Robot. Autom. Lett.*, 5(2):3620–3625, 2020.

[78] Lieven Vandenberghe, Stephen Boyd, and Katherine Comanor. Generalized chebyshev bounds via semidefinite programming. *SIAM review*, 49(1):52–64, 2007.

[79] T. Lew, R. Bonalli, and M. Pavone. Chance-constrained sequential convex programming for robust trajectory optimization. In *Proc. Eu. Control Conf.*, pages 1871–1878, 2020. doi: 10.23919/ECC51009.2020.9143595.

[80] Lukas Hewing and Melanie N Zeilinger. Stochastic model predictive control for linear systems using probabilistic reachable sets. In *IEEE Conf. on Dec. and Control*, pages 5182–5188, 2018.

[81] Lukas Hewing, Andrea Carron, Kim P Wabersich, and Melanie N Zeilinger. On a correspondence between probabilistic and robust invariant sets for linear systems. In *IEEE Euro. Control Conf.*, pages 1642–1647, 2018.

[82] Debasish Chatterjee and John Lygeros. On stability and performance of stochastic predictive control techniques. *IEEE Trans. Autom. Control*, 60 (2):509–514, 2014.

[83] Joel A Paulson, Stefan Streif, and Ali Mesbah. Stability for receding-horizon stochastic model predictive control. In *IEEE American Control Conference*, pages 937–943, 2015.

[84] Ali Mesbah, Stefan Streif, Rolf Findeisen, and Richard D Braatz. Stochastic nonlinear model predictive control with probabilistic constraints. In *IEEE American control conference*, pages 2413–2419, 2014.

[85] Vinay A Bavdekar and Ali Mesbah. Stochastic nonlinear model predictive control with joint chance constraints. *IFAC-PapersOnLine*, 49(18):270–275, 2016.

[86] Ashkan M Jasour and Brian C Williams. Sequential chance optimization for flow-tube based control of probabilistic nonlinear systems. In *IEEE 58th Conf. on Dec. and Con.*, pages 5392–5399, 2019.

[87] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.

[88] Quang-Cuong Pham, Nicolas Tabareau, and Jean-Jacques Slotine. A contraction theory approach to stochastic incremental stability. *IEEE Trans. Autom. Control*, 54(4):816–820, 2009.

[89] Ali Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Cont. Sys. Mag.*, 36(6):30–44, 2016.

[90] Franz S Hover and Michael S Triantafyllou. Application of polynomial chaos in stability and control. *Automatica*, 42(5):789–795, 2006.

[91] James Fisher and Raktim Bhattacharya. Stability analysis of stochastic systems using polynomial chaos. In *Proc. American Control Conference*, pages 4250–4255, 2008.

[92] Kwang-Ki Kim, Dongying Erin Shen, Zoltan K Nagy, and Richard D Braatz. Wiener's polynomial chaos for the analysis and control of nonlinear dynamical systems with probabilistic uncertainties [historical perspectives]. *IEEE Control Systems Magazine*, 33(5):58–67, 2013.

[93] George I Boutselis, Yunpeng Pan, Gerardo De La Tore, and Evangelos A Theodorou. Stochastic trajectory optimization for mechanical systems with parametric uncertainties. *arXiv preprint arXiv:1705.05506*, 2017.

[94] James Fisher and Raktim Bhattacharya. Optimal trajectory generation with probabilistic system uncertainty using polynomial chaos. *Journal of Dynamic Systems, Measurement, and Control*, 133(1):014501, 2011.

[95] Eddie Buehler. *Efficient Uncertainty Propagation for Stochastic Model Predictive Control*. PhD thesis, UC Berkeley, 2017.

[96] Anil Aswani, Humberto Gonzalez, S Shankar Sastry, and Claire Tomlin. Provably safe and robust learning-based model predictive control. *Automatica*, 49(5):1216–1226, 2013.

[97] Chris J Ostafew, Angela P Schoellig, and Timothy D Barfoot. Robust constrained learning-based NMPC enabling reliable mobile robot path tracking. *Int. J. Robot. Res.*, 35(13):1547–1563, 2016.

[98] Lukas Hewing, Alexander Liniger, and Melanie N Zeilinger. Cautious NMPC with Gaussian process dynamics for autonomous miniature race cars. In *Euro. Control Conf.*, pages 1341–1348, 2018.

[99] Ali Mesbah. Stochastic model predictive control with active uncertainty learning: a survey on dual control. *Ann. Rev. in Cont.*, 45:107–117, 2018.

[100] Allen Wang, Xin Huang, Ashkan Jasour, and Brian Williams. Fast risk assessment for autonomous vehicles using learned models of agent futures. In *Robotics: Sci. Systems*, 2020.

[101] Alexander Aronovich Feldbaum. Dual control theory. *Autom. Remote Control*, 21(9):874–1039, 1960.

[102] Y Cheng, S Haghighat, and Stefano Di Cairano. Robust dual control MPC with application to soft-landing control. In *Proc. Amer. Control Conf.*, pages 3862–3867, 2015.

[103] Quoc Tran Dinh and Moritz Diehl. Local convergence of sequential convex programming for nonconvex optimization. In *Recent Adv. Opt. & its App. Engr.*, pages 93–102. Springer, 2010.

[104] Mona Buisson-Fenet, Friedrich Solowjow, and Sebastian Trimpe. Actively learning Gaussian process dynamics. *arXiv:1911.09946*, 2019.

[105] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proc. Int. Conf. Mach. Learning*, pages 1015–1022, 2010.

[106] Jana L. Schwartz, Mason A. Peck, and Christopher D. Hall. Historical Review of Air-Bearing Spacecraft Simulators. *Journal of Guidance, Control, and Dynamics*, 26(4):513–522, 2003. ISSN 0731-5090. doi: 10.2514/2.5085. URL `http://arc.aiaa.org/doi/10.2514/2.5085`.

[107] Youngho Eun, Chandeok Park, and Sang-Young Park. Design and Development of Ground-Based 5-DOF Spacecraft Formation Flying Testbed. *AIAA Modeling and Simulation Technologies Conference*, pages 1–7, January 2016. doi: 10.2514/6.2016-1668. URL `http://arc.aiaa.org/doi/10.2514/6.2016-1668`.

[108] Jana Lyn Schwartz. *The distributed spacecraft attitude control system simulator: from design concept to decentralized control*. PhD thesis, Virginia Tech, 2004.

[109] David W Miller, A Saenz-Otero, J Wertz, A Chen, G Berkowski, C Brodel, S Carlson, D Carpenter, S Chen, S Cheng, et al. Spheres: a testbed for long duration satellite formation flying in micro-gravity conditions. In *Proceedings of the AAS/AIAA space flight mechanics meeting*, pages 167–179. Clearwater, Florida, January, 2000.

[110] Soon-Jo Chung, David W Miller, and Olivier L de Weck. Argos testbed: study of multidisciplinary challenges of future spaceborne interferometric arrays. *Optical Engineering*, 43(9):2156–2168, 2004.

[111] Brij N. Agrawal and Richard E. Rasmussen. Air-bearing-based satellite attitude dynamics simulator for control software research and development. 4366(831):204–214, 2001. ISSN 0277786X. doi: 10.1117/12.438072. URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=912729.

[112] D Gallardo and R Bevilacqua. Six Degrees of Freedom Experimental Platform for Testing Autonomous Satellites Operations. *Proceedings of the 8th International Conference on Guidance, Navigation and Control*, 2011.

[113] K. Saulnier, D. Pérez, R. C. Huang, D. Gallardo, G. Tilton, and R. Bevilacqua. A six-degree-of-freedom hardware-in-the-loop simulator for small spacecraft. *Acta Astronautica*, 105(2):444–462, 2014. ISSN 00945765. doi: 10.1016/j.actaastro.2014.10.027. URL http://dx.doi.org/10.1016/j.actaastro.2014.10.027.

[114] Daniel P. Scharf, Jason A. Keim, and Fred Y. Hadaegh. Flight-like ground demonstrations of precision maneuvers for spacecraft formations - Part II. *IEEE Systems Journal*, 4(1):96–106, 2010. ISSN 19328184. doi: 10.1109/JSYST.2010.2044281.

[115] Martin W Regehr, Ahmet B Acikmese, Asif Ahmed, M Aung, KC Clark, P MacNeal, J Shields, G Singh, R Bailey, C Bushnell, et al. The formation control testbed. In *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, volume 1, pages 557–564. IEEE, 2004.

[116] Richard Zappulla, Josep Virgili-Llop, Costantinos Zagaris, Hyeongjun Park, and Marcello Romano. Dynamic air-bearing hardware-in-the-loop testbed to experimentally evaluate autonomous spacecraft proximity maneuvers. *Journal of Spacecraft and Rockets*, 54(4):825–839, 2017.

[117] Panagiotis Tsiotras. Astros: A 5dof experimental facility for research in space proximity operations. *Advances in the Astronautical Sciences*, 151: 717–730, 01 2014.

[118] Daniele Gallardo, Riccardo Bevilacqua, and Richard Rasmussen. Advances on a 6 degrees of freedom testbed for autonomous satellites operations. In *AIAA Guidance, Navigation, and Control Conference*, page 6591, 2011.

[119] Markus Wilde, Brian Kaplinger, Tiauw Go, Hector Gutierrez, and Daniel Kirk. Orion: A simulation environment for spacecraft formation flight, capture, and orbital robotics. In *Aerospace Conference, 2016 IEEE*, pages 1–14. IEEE, 2016.

[120] David C Sternberg, Christopher Pong, Nuno Filipe, Swati Mohan, Shawn Johnson, and Laura Jones-Wilson. Jet propulsion laboratory small satellite dynamics testbed simulation: On-orbit performance model validation. *Journal of Spacecraft and Rockets*, 55(2):322–334, 2017.

[121] Sasi Prabhakaran Viswanathan, Amit Sanyal, and Lee Holguin. Dynamics and control of a six degrees of freedom ground simulator for autonomous rendezvous and proximity operation of spacecraft. In *AIAA Guidance, Navigation, and Control Conference*, page 4926, 2012.

[122] Markus Schlotterer, Eviatar Edlerman, Federico Fumenti, Pini Gurfil, Stephan Theil, and Hao Zhang. On-ground testing of autonomous guidance for multiple satellites in a cluster. In *Proceedings of the 8th International Workshop on Satellite Constellations and Formation Flying*, 06 2015.

[123] Soon-Jo Chung and David W Miller. Propellant-free control of tethered formation flight, part 1: Linear control and experimentation. *Journal of Guidance, Control, and Dynamics*, 31(3):571–584, 2008.

[124] Marco Ciarcia, Roberto Cristi, and Marcello Romano. Experimental emulation of the scaled clohessy-wiltshire dynamics on a flat air-bearing testbed. In *AIAA Guidance, Navigation, and Control Conference*, page 1047, 2017.

[125] Kyle Alfriend, Srinivas Rao Vadali, Pini Gurfil, Jonathan How, and Louis Breger. *Spacecraft formation flying: Dynamics, control and navigation*, volume 2. Elsevier, 2009.

[126] John L Junkins and Hanspeter Schaub. *Analytical mechanics of space systems*. American Institute of Aeronautics and Astronautics, 2009.

[127] Rebecca Foust, Soon-Jo Chung, and Fred Y Hadaegh. Optimal guidance and control with nonlinear dynamics using sequential convex programming. *Journal of Guidance, Control, and Dynamics*, 43(4):633–644, 2020.

[128] F Landis Markley and John L Crassidis. *Fundamentals of spacecraft attitude determination and control*, volume 33. Springer, 2014.

[129] Saptarshi Bandyopadhyay, Soon-Jo Chung, and Fred Y. Hadaegh. Nonlinear attitude control of spacecraft with a large captured object. *Journal of Guidance, Control, and Dynamics*, 39(4):754–769, 2016.

[130] Robert H Cameron and William T Martin. The orthogonal development of non-linear functionals in series of fourier-hermite functionals. *Annals of Math.*, pages 385–392, 1947.

[131] Tillmann Mühlpfordt, Rolf Findeisen, Veit Hagenmeyer, and Timm Faulwasser. Comments on truncation errors for polynomial chaos expansions. *IEEE Cont. Sys. Let.*, 2(1):169–174, 2017.

[132] Géraud Blatman and Bruno Sudret. Adaptive sparse polynomial chaos expansion based on least angle regression. *J. of Comp. Physics*, 230(6):2345–2367, 2011.

[133] Sergey Oladyshkin and Wolfgang Nowak. Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. *Reliability Engineering & System Safety*, 106:179–190, 2012.

[134] Yijun Xu, Lamine Mili, and Junbo Zhao. A novel polynomial-chaos-based kalman filter. *IEEE Signal Processing Letters*, 26(1):9–13, 2018.

[135] Mac Schwager, Brian J. Julian, Michael Angermann, and Daniela Rus. Eyes in the sky: Decentralized control for the deployment of robotic camera networks. *Proceedings of the IEEE*, 99(9):1541–1561, 2011.

[136] Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. ISSN 1931-9193.

[137] Benjamin Bernhard, Changrak Choi, Amir Rahmani, Soon-Jo Chung, and Fred Hadaegh. Coordinated motion planning for on-orbit satellite inspection using a swarm of small-spacecraft. In *IEEE Aerospace Conference*, pages 1–13, 2020.

[138] Nanoracks cygnus deployment documentation. `https://nanoracks.com/products/external-cygnus-deployment/`, 2020. Accessed: 2020-07-06.

[139] Richard Cheng, Richard M Murray, and Joel W Burdick. Limits of probabilistic safety guarantees when considering human uncertainty. *arXiv preprint arXiv:2103.03388*, 2021.

[140] Xinjia Chen. A new generalization of chebyshev inequality for random vectors. *arXiv preprint arXiv:0707.0805*, 2007.

[141] M. Ono and B. C. Williams. Iterative risk allocation: A new approach to robust model predictive control with a joint chance constraint. In *2008 47th IEEE Conf. on Decis. and Control*, pages 3427–3432, Dec 2008. doi: 10.1109/CDC.2008.4739221.

[142] Eckhard Platen and Nicola Bruti-Liberati. *Numerical solution of stochastic differential equations with jumps in finance*, volume 64. Springer Science & Business Media, 2010.

[143] Yanbo Li, Zakary Littlefield, and Kostas E Bekris. Asymptotically optimal sampling-based kinodynamic planning. *Int.J. Robot. Res.*, 35(5):528–564, 2016.

[144] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *Int.J. Robot. Res.*, 30(7):846–894, 2011.

[145] Ali Jadbabaie, Jie Yu, and John Hauser. Unconstrained receding-horizon control of nonlinear systems. *IEEE Trans. Autom. Control*, 46(5):776–783, 2001.

[146] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[147] Hiroyasu Tsukamoto, Soon-Jo Chung, and Jean-Jacques E. Slotine. Neural stochastic contraction metrics for learning-based control and estimation. *IEEE Control Systems Letters*, 5(5):1825–1830, 2021. doi: 10.1109/LCSYS.2020.3046529.

[148] Lora Billings and Ira B Schwartz. Identifying almost invariant sets in stochastic dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 18(2):023122, 2008.

[149] Ernesto Kofman, José A De Doná, and Maria M Seron. Probabilistic set invariance and ultimate boundedness. *Automatica*, 48(10):2670–2676, 2012.

[150] Matthias Lorenzen, Fabrizio Dabbene, Roberto Tempo, and Frank Allgöwer. Constraint-tightening and stability in stochastic model predictive control. *IEEE Trans. Autom. Control*, 62(7):3165–3177, 2016.

[151] Borisovich Dynkin. *Markov processes*. Springer, 1965.

[152] Steven Diamond and Stephen Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *J. of Mach. Learn. Res.*, 17(1):2909–2913, 2016.

[153] Alexander Domahidi, Eric Chu, and Stephen Boyd. Ecos: An socp solver for embedded systems. In *IEEE Euro. Control Conf.*, pages 3071–3076, 2013.

[154] Anqi Liu, Guanya Shi, Soon-Jo Chung, Anima Anandkumar, and Yisong Yue. Robust regression for safe exploration in control. *arXiv:1906.05819*, 2019.

[155] Alexey Pavlov, Alexander Pogromsky, Nathan van de Wouw, and Henk Nijmeijer. Convergent dynamics, a tribute to boris pavlovich demidovich. *Sys. & Cont. Letters*, 52(3-4):257–261, 2004.

[156] I Richard Savage. Mills' ratio for multivariate normal distributions. *J. Res. Nat. Bur. Standards Sect. B*, 66:93–96, 1962.

[157] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *Euro. Control Conf.*, pages 3420–3431, 2019.

[158] Hassan K Khalil. Noninear systems. *Prentice-Hall, New Jersey*, 2(5):5–1, 1996.

[159] Soon-Jo Chung, Saptarshi Bandyopadhyay, Insu Chang, and Fred Y Hadaegh. Phase synchronization control of complex networks of lagrangian systems on adaptive digraphs. *Automatica*, 49(5):1148–1161, 2013.

[160] Winfried Lohmiller and Jean-Jacques E Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.

[161] Saptarshi Bandyopadhyay, Soon-Jo Chung, and Fred Y Hadaegh. Nonlinear attitude control of spacecraft with a large captured object. *Journal of Guidance, Control, and Dynamics*, 39(4):754–769, 2016.

[162] F Landis Markley, Reid G Reynolds, Frank X Liu, and Kenneth L Lebsock. Maximum torque and momentum envelopes for reaction wheel arrays. *Journal of Guidance, Control, and Dynamics*, 33(5):1606–1614, 2010.

[163] R. Cowen. The wheels come off kepler. *Nature*, 497(7450):417–418, 2013.

[164] Rebecca C. Foust, E. Sorina Lupu, Yashwanth K. Nakka, Soon-Jo Chung, and Fred Y. Hadaegh. Autonomous in-orbit satellite assembly from a modular heterogeneous swarm. *Acta Astronautica*, 169: 191–205, 2020. ISSN 0094-5765. doi: https://doi.org/10.1016/j.actaastro.2020.01.006. URL https://www.sciencedirect.com/science/article/pii/S0094576520300060.

[165] Lu Gan, Ray Zhang, Jessy W Grizzle, Ryan M Eustice, and Maani Ghaffari. Bayesian spatial kernel smoothing for scalable dense semantic mapping. *IEEE Robotics and Automation Letters*, 5(2):790–797, 2020.

[166] G C Anderson, D A Quinn, G A Beals, J D Nelson, and G S Nurre. An overview of the hubble space telescope pointing control system design and operation. In *AIAA Guidance, Navigation, and Control and Co-located Conferences*, Hilton Head Island, Aug. 1992.

[167] Luis Meza, Frank Tung, Satya Anandakrishnan, Victor Spector, and Tupper Hyde. Line of sight stabilization for the james webb space telescope. *Advances in the Astronautical Sciences*, 121:15–28, 2005.

[168] M. Knapp, M. Smith, C. Pong, J. Nash, and S. Seager. ExoplanetSat: High precision photometry for exoplanet transit detections in a 3U. In *IAC 16th Symposium on Small Satellite Missions*, 2012. IAC-12-B4.

[169] Exo-c imaging nearby worlds. *https*: *//exoplanets.nasa.gov/stdt/Exo-C_InterimReport.pdf*.

[170] M Guelman, Anton Kogan, Amir Livne, Meir Orenstein, and Harald Michalik. Acquisition and pointing control for inter-satellite laser communications. *Aerospace and Electronic Systems, IEEE Transactions on*, 40(4):1239–1248, 2004.

[171] Saptarshi Bandyopadhyay, Rebecca Foust, Giri P. Subramanian, Soon-Jo Chung, and Fred Y. Hadaegh. Review of formation flying and constellation missions using nanosatellites. *Journal of Spacecraft and Rockets*, 53(3): 567–578, 2016.

[172] Soon-Jo Chung, David W. Miller, and Olivier L. de Weck. Argos testbed: study of multidisciplinary challenges of future spaceborne interferometric arrays. *Optical Engineering*, 43(9):2156–2167, 2004.

[173] Lars Blackmore, Emmanuell Murray, Daniel P Scharf, Mimi Aung, David Bayard, Paul Brugarolas, Fred Hadaegh, Allan Lee, Mark Milman, Sam Sirlin, et al. Instrument pointing capabilities: Past, present, and future. In *AAS Guidance and Control Conference*, Breckenridge, Colorado, Feb. 2011.

[174] Christian M Chilan, Daniel R Herber, Yashwanth Kumar Nakka, Soon-Jo Chung, and James T Allison. Co-design of strain-actuated solar arrays for precision pointing and jitter reduction. *AIAA Journal*, pages 1–16, 2017.

[175] Oscar Alvarez-Salazar, Jack Aldrich, Nuno Filipe, James Allison, and Soon-Jo Chung. Strain actuated solar-arrays for precision pointing of spacecraft. In *AAS Guidance, Navigation, and Control Conference*, Breckenridge, Colorado, Feb. 2016.

[176] Christian M Chilan, Daniel R Herber, Yashwanth Kumar Nakka, Soon-Jo Chung, and James T Allison. Co-design of strain-actuated solar arrays for precision pointing and jitter reduction. In *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0162, 2016.

[177] L Meirovitch. *Dynamics and Control of Structures*. John Wiley & Sons, 1990.

[178] D. H. Hodges and G. A. Pierce. *Introduction to Structural Dynamics and Aeroelasticity*. Cambridge University Press, 2 edition, 2011.

[179] L. Meirovitch and T. Stemple. Hybrid equations of motion for flexible multibody systems using quasicoordinates. *Journal of Guidance, Control, and Dynamics*, 18(4):678–688, – 1995.

[180] John L. Junkins and Youdan Kim. *Introduction to Dynamics and Control of Flexible Structures*. AIAA, 1 edition, 1993.

[181] J. Piedboeuf. Six methods to model a flexible beam rotating in the vertical plane. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 3, pages 2832–2839 vol.3, 2001.

[182] R. Ryan T. R. Kane and A. K. Banerjee. Dynamics of a cantilever beam attached to a moving base. *Journal of Guidance, Control, and Dynamics*, 10 (2):139–151, – 1987.

[183] V. J. Modi, A. C. Ng, and F. Karray. Nonlinear dynamics and control of orbiting structures: an approach with applications. In *Proceedings of IEEE International Conference on Control and Applications*, pages 1–6 vol.1, Sep 1993. doi: 10.1109/CCA.1993.348315.

[184] Mark W. Spong. *Underactuated mechanical systems*, pages 135–150. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[185] Soon-Jo Chung, Jean-Jacques E. Slotine, and David W. Miller. Propellant-free control of tethered formation flight, part 2: Nonlinear underactuated control. *Journal of Guidance, Control, and Dynamics*, 31(5), sep 2008.

[186] Daniel R Herber, Jason W. McDonald, Oscar S. Alvarez-Salazar, Girish Krishnan, and James T. Allison. Reducing spacecraft jitter during satellite reorientation maneuvers via solar array dynamics. In *AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, number AIAA 2014-3278, Atlanta, GA, USA, June 2014.

[187] M J Balas. Active control of flexible systems. *Journal of Optimization Theory and Applications*, 25(3):415–436, July 1978.

[188] L. Meirovitch and H. Baruh. On the problem of observation spillover in self-adjoint distributed-parameter systems. *Journal of Optimization Theory and Applications*, 39(2):269–291, Feb 1983. ISSN 1573-2878. doi: 10.1007/BF00934533. URL `https://doi.org/10.1007/BF00934533`.

[189] W. He and S. S. Ge. Dynamic modeling and vibration control of a flexible satellite. *IEEE Transactions on Aerospace and Electronic Systems*, 51(2): 1422–1431, April 2015. ISSN 0018-9251. doi: 10.1109/TAES.2014.130804.

[190] Gangbing Song and Brij N Agrawal. Vibration suppression of flexible spacecraft during attitude control. *Acta Astronautica*, 49(2):73–83, 2001.

[191] P. D. Christofides and P. Daoutidis. Finite-dimensional control of parabolic pde systems using approximate inertial manifolds. In *Proceedings of the 36th IEEE Conference on Decision and Control*, volume 2, pages 1068–1073 vol.2, Dec 1997. doi: 10.1109/CDC.1997.657588.

[192] Ruth F. Curtain and Hans Zwart. *An Introduction to Infinite-dimensional Linear Systems Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94475-3.

[193] Miroslav Krstic and Andrey Smyshlyaev. *Boundary control of PDEs: A course on backstepping designs*, volume 16. Siam, 2008.

[194] Z. Artstein. Linear systems with delayed controls: A reduction. *IEEE Transactions on Automatic Control*, 27(4):869–879, Aug 1982. ISSN 0018-9286. doi: 10.1109/TAC.1982.1103023.

[195] W. Kwon and A. Pearson. Feedback stabilization of linear systems with delayed control. *IEEE Transactions on Automatic Control*, 25(2):266–269, Apr 1980. ISSN 0018-9286.

[196] M. Krstic. Compensating a string pde in the actuation or sensing path of an unstable ode. *IEEE Transactions on Automatic Control*, 54(6):1362–1368, June 2009. ISSN 0018-9286.

[197] S. Tang and C. Xie. Stabilization of a coupled pde-ode system by boundary control. In *49th IEEE Conference on Decision and Control*, pages 4042–4047, Dec 2010. doi: 10.1109/CDC.2010.5718141.

[198] Miroslav Krstic. Compensating actuator and sensor dynamics governed by diffusion pdes. *Systems & Control Letters*, 58:372–377, 2009.

[199] Gian Antonio Susto and Miroslav Krstic. Control of pde–ode cascades with neumann interconnections. *Journal of the Franklin Institute*, 347(1):284–314, 2010.

[200] R. Olfati-Saber. Nonlinear control and reduction of underactuated systems with symmetry. i. actuated shape variables case. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 5, pages 4158–4163 vol.5, 2001.

[201] J.-J. E. Slotine and W. Li. On the adaptive control of robot manipulators. *Int. J. Rob. Res.*, 6(3):49–59, September 1987. ISSN 0278-3649. doi: 10.1177/027836498700600303. URL `http://dx.doi.org/10.1177/027836498700600303`.

[202] Jean-Jacques E Slotine and Weiping Li. *Applied nonlinear control*. Pearson, Upper Saddle River, NJ, 1991.

[203] Raymond L Bisplinghoff, Holt Ashley, and Robert L Halfman. *Aeroelasticity*. Addison-Wesley, 1995.

[204] A. A. Paranjape, S.-J. Chung, H. H. Hilton, and A. Chakravarthy. Dynamics and performance of tailless micro aerial vehicle with flexible articulated wings. *AIAA Journal*, 50(5):1177–1188, May 2012.

[205] Aditya A. Paranjape, Jinyu Guan, Soon-Jo Chung, and Miroslav Krstic. Pde boundary control for flexible articulated wings on a robotic aircraft. *IEEE Transactions on Robotics*, 29(3):625–640, June 2013.

[206] S S Rao and M Sunar. Piezoelectricity and its use in disturbance sensing and control of flexible structures: A survey. *Applied Mechanics Reviews*, 47(4):113–123, April 1994.

[207] T. Bailey and J. E. Hubbard Jr. Distributed piezoelectric-polymer active vibration control of a cantilever beam. *Journal of Guidance, Control, and Dynamics*, 8(5):605–611, September 1985.

[208] E F Crawley and J D Luis. Use of piezoelectric actuators as elements of intelligent structures. *AIAA Journal*, 25(10):1373–1385, October 1987.

[209] J L Fanson and J C Chen. Structural control by the use of piezoelectric active members. In *NASA-Langley Research Center NASA/DOD Control/Structures Interaction Technology*, pages 809–829, 1986.

[210] S. O. M. Moheimani and A. J. Fleming. *Piezoelectric Transducers for Vibration Control and Damping*. Springer, 1 edition, 2006.

[211] O. S. Alvarez-Salazar and K. Iliff. Destabilizing effects of rate feedback on strain actuated beams. *Journal of Sound and Vibration*, 221(2):289–307, March 1999.

[212] Edward F Crawley and Eric H Anderson. Detailed models of piezoceramic actuation of beams. *Journal of Intelligent Material Systems and Structures*, 1(1):4–25, 1990.

[213] Y K Nakka. Nonlinear attitude control of spacecraft with strain-actuated solar arrays. M.S. Thesis, University of Illinois at Urbana-Champaign, Urbana, IL, USA, August 2016.

[214] Inna Sharf. Geometric stiffening in multibody dynamics formulations. *Journal of Guidance, Control, and Dynamics*, 18(4):882–890, –August 1995.

Appendix A

NONLINEAR ATTITUDE CONTROL OF A SPACECRAFT WITH DISTRIBUTED ACTUATION OF SOLAR ARRAYS

A.1 Introduction

Space observatories require precise attitude control and pointing accuracy to achieve desired diffraction-limited optical performance. For example, the Hubble and James Webb Space Telescopes require 7 milliarcsec [166] and 15 milliarcsec [167] pointing accuracies, respectively. A range of emerging science missions such as Asteria [168], Exo-C [169] and technologies such as laser communication [170], precision formation flying [171], and interferometric imaging [172], place stringent requirements on pointing error and stability. In these state-of-the-art space observatories, precision imaging involves two stages [169, 173]: (i) coarse stage instrument pointing, which is achieved using the primary spacecraft Attitude Control System (ACS), and (ii) fine stage precision control and jitter reduction. Coarse stages are typically built from Reaction Wheel Assemblies (RWAs), because hydrazine thrusters are too disruptive for precision imaging [167]. In this case, however, the RWA system, which drives the ACS, ironically becomes the dominant source of instrument jitter and imaging performance degradation. To avoid these issues, the RWA disturbance must be eliminated at its source, so that an ultra-quiet, spacecraft actuation system (UQSAS), i.e., one that does not produce severe unwanted disturbances as a side effect, can be put in its place. Recently, electric propulsion systems have been proposed as viable UQSAS options, but these actuation systems are clearly fuel-limited and are typically designed as low-thrust propulsion systems that do not have enough control authority to support a high-bandwidth pointing control loop. In contrast, Strain-Actuated Solar Arrays (SASA) have been recently introduced as a propellant-free, UQSAS alternative [174, 175] that can support a high-bandwidth pointing control loop assuming a high force-density piezoelectric-based actuation system.

Given the potential for SASA to serve as a viable UQSAS alternative, several studies have been conducted thus far. In [175], replacing the RWA with SASA to achieve spacecraft slewing, attitude control, and momentum management was investigated. In [174, 176], a design optimization of both the control architecture and structural geometry was reported. In contrast to earlier work, this ar-

ticle presents a complete derivation of the spacecraft-SASA dynamics in which the infinite-dimensional nature of the flexible appendage is modeled using Euler-Bernoulli beam theory [177, 178]. More specifically, the nonlinear dynamics of the open-loop plant is modeled as an ODE-PDE system with the ODE describing spacecraft single-axis rigid-body rotation, and the PDE describing the spatially continuous flexible dynamics of the solar array, which includes an allocation for a multi-input distributed piezoelectric actuation system. This study includes a relevant control-structure interaction stability analysis, simulation, and experimental validation. The salient feature of the proposed approach is that the piezoelectric actuators induce bending deflections that are strategically manipulated by a novel underactuated control law to effectively yield the needed array-to-bus coupling (reaction torques) required by the assumed precision pointing tracking control system realization (that is derived herein).

A literature survey aimed at extracting an appropriate open-loop plant model for model-based control design approach revealed several candidate modeling approaches, including those based directly on classical methods such as the Newton-Euler method, Lagrange's equation, Hamilton's principle, and the principal of virtual work, where the spacecraft and flexible appendages are modeled as a rigid/flexible multi-body system [179–182]. These approaches combine finite and infinite-dimensional dynamics via a coupled ODE-PDE system of nonlinear equations, which can be linearized as in [177, 180, 181, 183]. In particular, [182] accounted for rotation-induced centrifugal stiffening in the context of a linearized analysis. In this paper, the linear model for a circular bus with flexible appendages in [180] is extended with axial stiffness in the beam due to rotation and gyroscopic terms using the explicit generalization of Lagrange's equation for infinite dimension systems [180]. As such, the resulting system dynamics becomes coupled, nonlinear, and underactuated in nature [184, 185] with control inputs derived exclusively from the underlying SASA actuation system. This is noteworthy because earlier studies of SASA, including [186], modeled the solar array with a finite-dimensional rigid-body approximation, in contrast to the infinite dimensional PDE model that is used herein. The importance of improving model fidelity in this manner is underscored by the fact that finite-dimensional controllers are prone to spillover [187, 188] when applied to the actual infinite dimensional system for which they were designed.

Numerous previous studies [180, 189, 190] have focused on the control-structure interaction problem for vibration suppression of flexible appendages. However, this paper is the first of its kind where a beam is driven by distributed actuators

such that the underactuated spacecraft bus achieves attitude guidance and control objectives. In either case, a canonical system of ODE-PDE equations of motion is generated, for which additional processing takes on either of the following two standard methods: (i) the ODE-PDE system is discretized into an ODE system using either Galerkin [177, 178, 191] or operator-theoretic methods [192], or (ii) The ODE-PDE form is left intact [193] for analysis. As mentioned previously, discretization in the first method leads to spillover issues [187, 188]; while keeping PDEs intact avoids this problem.

It is worth noting that ODE-PDE cascaded systems, similar to the SASA system generated herein, have been studied previously in the literature. Published results in this area depend heavily on the type of PDE classification. Indeed, separate results have been obtained for first-order wave equation [194, 195], second-order wave equation [196], and parabolic equation [197, 198]. In each of these cases, the control input entering the system at the boundary can assume either a Dirichlet [196, 198] or a Neumann interconnection [199]. In these works, however, the ODE-PDE interaction is strictly unidirectional with regards to how the PDE couples with the ODE system. In contrast, [197] presents an ODE-PDE system where this assumption is relaxed so that the more complete modeling case of bidirectional coupling can be considered. With regards to control, a PDE backstepping method [193, 196, 198] and boundary controller are able to provide infinite-dimensional full-state feedback where the gain kernels are used to compensate for PDE dynamics while stabilization of the entire system is achieved. Similarly, the ODE-PDE SASA system considered herein is bidirectionally coupled with a distributed control input. The open-loop model's inherent passivity [18], and kinetic symmetry [200] are exploited in the construction of the nonlinear ODE-PDE attitude control law. Specifically, the nonlinear controller derivation proposed in [201, 202] is adapted to the ODE-PDE under-actuated SASA system proposed herein. In doing so, the resultant closed-loop dynamics with the beam dynamics canceled by a feed-forward term are governed by a virtual dynamical system that converges exponentially to a stable manifold in the sense of spatial $\mathcal{L}_2$ norm. In the case with partial or no cancellation of the beam dynamics, finite-gain time-signal ℓ_p stability of the closed-loop system is proved by formulating the uncanceled beam dynamics as a disturbance term. From an end-to-end point of view, the controller actuates the flexible beam to achieve satellite slewing using the coupled ODE-PDE dynamics.

One of the major objectives of this paper is to experimentally demonstrate real-time closed-loop SASA (ODE-PDE) attitude tracking control in the laboratory. In order

to do so, it was necessary to discretize the infinite-dimensional ODE-PDE plant dynamics and control law. Towards this end, Galerkin's method [177, 178, 180, 203] was used to form ODE systems from the PDE dynamics. The discretized control law was tested on the simulated plant dynamics before attempting implementation on the actual laboratory SASA testbed. The testbed consists of a one-degree-of-freedom cylinder (spacecraft bus) with flexible beams (solar arrays) symmetric about the cylinder's axis of rotation. Strain actuation in the beams is achieved by bonding piezoelectric actuators asymmetrically about the axis of the cylinder.

The main contribution of this paper is the design of a new nonlinear ODE-PDE control law using the coupled structure in the dynamics to perform precision attitude control with detailed stability proofs. Notably, the proposed ODE-PDE control design captures all the modes of the flexible beam, thereby avoiding the spillover problem. Stable trajectory tracking via the nonlinear distributed control law is successfully demonstrated by analysis, numerical simulations, and experimental tests of the closed-loop system. The actuator model is validated by comparing the tip deflection of a composite cantilever beam (aluminum beam with surface-bonded PZT) with those obtained from simulations and experiments. The piezoelectric ceramic used in the actuators is lead zirconate titanate, which is abbreviated PZT based on its chemical formula: $\mathrm{Pb}[\mathrm{Zr}_x\mathrm{Ti}_{1-x}]\mathrm{O}_3,\ (0 \leq x \leq 1)$. It is shown that the maximum controlled attitude rotation achieved by bending the solar array is constrained both by the inertia of the solar array and the bus, and by the maximum stress that can be applied to the solar array without compromising its structural integrity.

The paper is organized as follows. The kinematics, dynamics of the system, and actuator model of the composite beam (PZT and beam) are discussed in Section A.2. Nonlinear ODE-PDE control law design and the associated stability analysis are presented in Section A.3. Numerical implementation and results for trajectory tracking and slewing is given in Section A.4. Experimental results that validate the actuator model and real-time trajectory tracking of the bus rotation are given in Section A.5. Concluding remarks are given in Section A.6.

A.2 Dynamics of a 1-DOF Satellite with SASA

The spacecraft with asymmetric SASAs is modeled as a cylinder with flexible appendages that are fixed symmetrically to the rotational axis of the cylinder as shown in Fig. A.1. The dynamics of the system include planar rotation of the spacecraft bus and bending in the flexible composite beam due to strain-actuation. Our modeling approach uses explicit generalization of Lagrange's equations [180]. The equations

of motion of the ODE-PDE system can also be derived by following the approach discussed in [204, 205]. Before proceeding to the Euler-Lagrange equations, we discuss briefly the system kinematics, physical properties of the composite beam, and the actuator model for strain-actuation using PZTs.

A.2.1 Kinematics

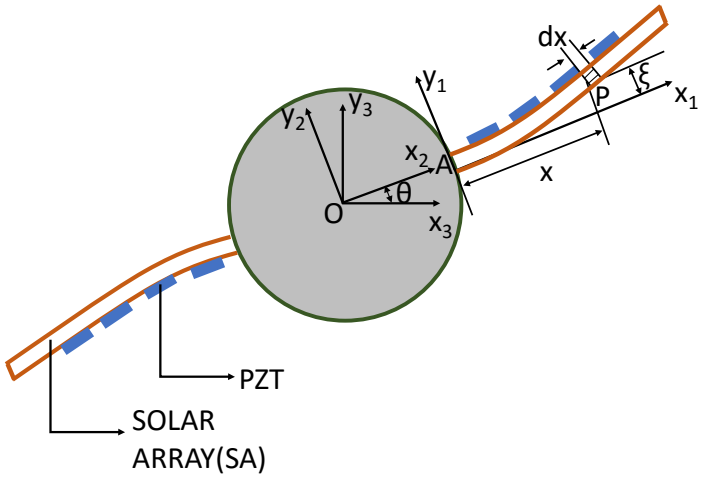


Figure A.1: 1-DOF cylinder and flexible solar array model.

The spacecraft body is modeled as a cylinder of radius r, and the solar array is modeled as a composite beam of length ℓ, with PZT actuators bonded on the beam surface. It is assumed that the beam does not undergo any longitudinal vibration. The coordinate systems used in the derivation of kinematics are shown in Fig. A.1. In the (x_1, y_1) coordinate system, the location of a point P on an element dx of the beam is given by $R_{P/A} = [x, \xi]$, where $\xi(x,t) : [0,\ell] \times \mathbb{R}^+ \to \mathbb{R}$ is the displacement due to bending in y_1 direction. The spacecraft body rotation angle about origin O is $\theta(t)$, where $\theta(t) : \mathbb{R}^+ \to [-\pi, \pi]$. The position and velocity of P with respect to origin of the spacecraft O are given by (A.1) and (A.2), respectively. These equations are used to compute ξ using x_3 and y_3.

$$\begin{bmatrix} x_3 \\ y_3 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} r + x \\ \xi \end{bmatrix} \tag{A.1}$$

The velocity kinematics are given by:

$$\begin{bmatrix} \dot{x}_3 \\ \dot{y}_3 \end{bmatrix} = \begin{bmatrix} -\sin(\theta) & -\cos(\theta) \\ \cos(\theta) & -\sin(\theta) \end{bmatrix} \begin{bmatrix} (r + x)\dot{\theta} + \dot{\xi} \\ \xi\dot{\theta} \end{bmatrix} \tag{A.2}$$

A.2.2 Mass Per Unit Length and Total Rigidity of the Composite Beam

The physical properties of the solar array, such as mass per unit length and location of the neutral axis that plays an important role in the evolution of the system dynamics, are functions of the spatial variable x due to the composite nature of the beam. The mass per unit length of the composite beam is given in (A.3).

$$m_R(x) = m_b + k(x)m_p,\ m_b = \rho_b A_b,\ m_p = \rho_p A_p \tag{A.3}$$

where ρ_b and ρ_p denote the beam and the PZT densities, respectively. Also, A_b and A_p are the cross sectional areas of the beam and the PZT, respectively. Note that the function $k\,(x) = 1$ at the locations where the PZT is bonded, and $k\,(x) = 0$ otherwise.

$$m_R(x) = m_b + k(x)m_p + \sum_{i=1}^{n_v} m_v \delta_d(x - \ell_{v_i}) \tag{A.4}$$

For simulations and experiments described in this paper, we use (A.4) for the mass per unit length, which takes into account the effect of point masses placed on the beam for a particular sensing system we used. The point mass is m_v, δ_d is the Dirac delta function, ℓ_{v_i} is the distance to the i-th point mass on the beam from the root, and n_v is the number of point masses.

The physical, structural, and geometric properties of the PZTs and the beam are different. As a result, the elastic neutral axis of the composite beam is offset from the geometric centroid. The distance between the top surface of the composite beam to the neutral axis h_n, as shown in Fig. A.2a, is given as:

$$h_n = \frac{E_p t_p^2 + E_b t_b\,(t_b + 2t_p)}{2\,(E_p t_p + E_b t_b)} \tag{A.5}$$

where E_b, I_b, t_b, E_p, I_p, and t_p are the Young's modulus, area moment of inertia about neutral axis, and thickness of the beam and the PZT, respectively.

The total beam rigidity EI_t at the locations where PZTs are bonded, with area moment of inertia calculated about the new neutral axis, is given by:

$$EI_t = E_b I_b + E_p I_p$$
$$where \quad I_b = \frac{w t_b^3}{12} + w t_b \left(t_p + \frac{t_b}{2} - h_n\right)^2, \quad I_p = \frac{w t_p^3}{12} + w t_p \left(h_n - \frac{t_p}{2}\right)^2 \tag{A.6}$$

The total beam rigidity EI_b at locations with $k(x) = 0$ (where no PZT actuator is attached) is given by following equation:

$$EI_b = \frac{E_b w t_b^3}{12} \tag{A.7}$$

In the dynamic model, we use the variable $EI(x)$ for rigidity of the beam, which is a function of spatial position x.

A.2.3 Actuator Model

Piezoelectric materials undergo uniform strain when an electrical potential difference is applied across their electrodes [206]. Due to this and other desirable properties, PZT actuators are commonly used in design of intelligent structures, active vibration control, and strain-actuated beams [187, 207–211]. PZT patches can be bonded to the surface of a solar array, or embedded within the solar array substructure [208, 212]. References [208, 212] discuss actuator models for the SASA configurations described above, with and without perfect bonding. Herein, a quasi-static actuator model based on the surface bonded PZT model [210] is assumed, including some needed error corrections in the final model.

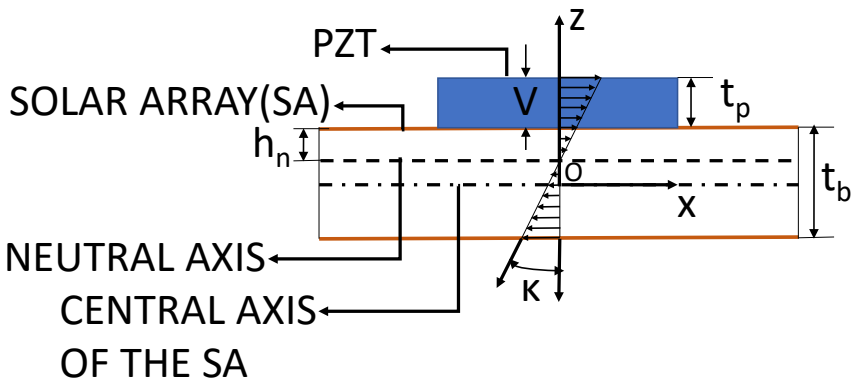
The following actuator model provides the mathematical framework for modeling moments applied to the beam due to a voltage V applied across the PZT bonded to the beam surface. In deriving the model, it is assumed that the bonding between the PZT and the beam is perfect, and that the composite beam has constant width 'w'. Here, the PZT is bonded only on the top surface and the strain distribution 'ϵ' along the cross section of the composite beam is assumed to be linear (see Fig. A.2a):

$$\epsilon = \kappa z + \epsilon_0 \tag{A.8}$$

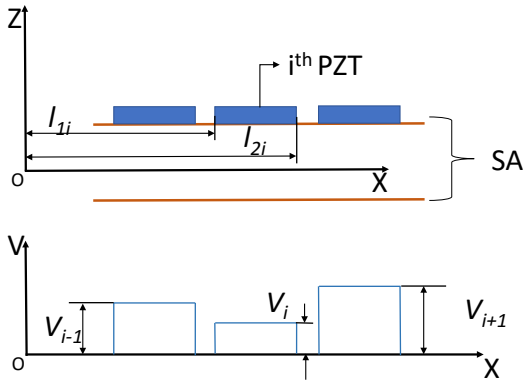
where κ is the slope of the strain distribution due to bending, and ϵ_0 is the beam extension due to the neutral axis offset. Force and moment equilibria are used to evaluate the expressions for κ and ϵ_0 (for detailed derivation, see [213]). The static model derived here is used in the dynamic model, assuming that the process is quasi-static in nature. In the following, ϵ_p is the strain of an unconstrained PZT when a voltage 'V' is applied across it. This strain is given by: $\epsilon_p = \frac{d_{zx}V}{t_p}$, [210]. The piezoelectric coefficient d_{zx} is the ratio of strain in the x-direction when an electric field is applied across the PZT in the z-direction.

$$\kappa = \frac{6E_bE_p\epsilon_p t_b t_p(t_b + t_p)}{E_b^2t_b^4 + 4E_bE_pt_b^3t_p + 6E_bE_pt_b^2t_p^2 + 4E_bE_pt_bt_p^3 + E_p^2t_p^4} \tag{A.9}$$

$$\epsilon_0 = \frac{\epsilon_p E_p t_p(E_p t_p^3 + E_b t_b^3)}{E_b^2t_b^4 + 4E_bE_pt_b^3t_p + 6E_bE_pt_b^2t_p^2 + 4E_bE_pt_bt_p^3 + E_p^2t_p^4} \tag{A.10}$$



(a) Strain distribution.



(b) Distributed Piezo.

Figure A.2: Distributed Piezo and and strain distribution across the composite beam.

The moment in the beam, due to the uniform PZT strain produced by the applied voltage V, is given by

$$M_b = E_b I_b \kappa = cV,$$
$$\text{where } c = \frac{6E_b^2 I_b E_p d_{zx} t_b (t_b + t_p)}{E_b^2 t_b^4 + 4E_b E_p t_b^3 t_p + 6E_b E_p t_b^2 t_p^2 + 4E_b E_p t_b t_p^3 + E_p^2 t_p^4} \tag{A.11}$$

The moment is proportional to the voltage applied across the PZT. The constant c depends on the geometric parameters, the structural properties of the PZT, and the elastic modulus of the composite beam. The actuator model is extended to a scenario with multiple PZTs using step functions (see Fig. A.2b). The model with n PZTs on the beam surface is given in (A.12), where V_i is the voltage applied across the ith PZT, c_i is the proportionality constant corresponding to the structural and geometric properties of the ith PZT, and $u(x)$ is the step function with unit

amplitude. The lengths l_{1i} and l_{2i} are defined for ith PZT as shown in Fig. A.2b.

$$M_b = \sum_{i=1}^{n} c_i V_i \left(u(x - l_{1i}) - u(x - l_{2i})\right) \tag{A.12}$$

Equations (A.11) and (A.12) are used to compute voltage signals required to achieve attitude control from the control signal in the simulations and experiments.

A.2.4 Dynamics

The system dynamics include elastic bending of the composite beam, modeled using Euler-Bernoulli beam theory, and rotational motion, which is modeled as planar rigid-body spacecraft rotation. The dynamic model, prescribed by the coupled ODE-PDE system with fixed-root and free-end boundary conditions, uses a hybrid coordinate system integrating one rigid body with an infinite-dimensional system (also called as distributed parameter system). In deriving the model, we assume that the deflections due to bending are small and that the beam has no longitudinal vibration; the effect of ϵ_0 in (A.10) is assumed to be negligible. The state of the system corresponding to beam deflection is $\xi(x,t)$, described by a continuous function of spatial radial position x and time t.

The system Lagrangian L involves several quantities, including the spacecraft bus (cylinder) mass moment of inertia J_θ, rotational kinetic energy of the spacecraft body T_s (Eq. (A.13)), beam kinetic energy assuming asymmetric strain-actuation T_b (Eq. (A.14)), and elastic potential energy and axial stiffening of the beams due to centrifugal force from bus rotation U (Eq. (A.15)).

$$T_s = \frac{1}{2} J_\theta \dot{\theta}^2 \tag{A.13}$$

$$T_b = \int_0^\ell m_R \Big[\dot{\theta}^2(r^2 + x^2 + \xi^2) + \dot{\xi}^2 + 2r\dot{\xi}\dot{\theta} + 2r\dot{\theta}^2 x + 2\dot{\xi}\dot{\theta}x\Big] dx \tag{A.14}$$

$$U = \int_0^\ell \left[EI\left(\xi''\right)^2\right] dx + \int_0^\ell \left[P\left(\dot{\theta}, x\right)\left(\xi'\right)^2\right] dx \tag{A.15}$$

The Lagrangian of the ODE-PDE system is given as $L = T_s + T_b - U$. Note that $()' = \frac{\partial}{\partial x}()$, and $P\left(\dot{\theta}, x\right) = \int_{r+x}^{r+\ell} \left[m_R \dot{\theta}^2 s\right] ds = p(x)\dot{\theta}^2$ is the axial tension in the beam due to rotation, where m_R is expressed as a function of $s = r + x$. The definition of the composite beam total rigidity EI is given in Section A.2.2. The nonconservative work W_{nc} done due to the moment applied by strain actuation is given by:

$$W_{\text{nc}} = \int_0^\ell \frac{M_b(x,t)^2}{E_b I_b} dx \tag{A.16}$$

where $M_b(x,t)$ is the internal beam moment (see (A.11)) due to the strain produced by PZTs. The equations of motion are obtained by using the explicit generalization of Lagrange's equations for infinite-dimensional systems approach (see [180] for more details). The extended Hamilton's principle can be stated as follows, where t_0 and t_f are the initial and final time values, respectively:

$$\int_{t_0}^{t_f} (\delta L + \delta W_{\text{nc}})\, dt = 0 \tag{A.17}$$

Applying integration by parts to the expanded variations in terms of the state variables and using boundary conditions, we obtain the following equations of motion in terms of the Lagrangian:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0 \tag{A.18}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\xi}}\right) - \frac{\partial L}{\partial \xi} + \frac{\partial}{\partial x}\left(\frac{\partial L}{\partial \xi'}\right) - \frac{\partial^2}{\partial x^2}\left(\frac{\partial L}{\partial \xi''}\right) = \frac{\partial^2}{\partial x^2}\left(M_b(x,t)\right) \tag{A.19}$$

The structural damping in the system is modeled using the Kelvin-Voigt constant μ. The equations of motion after substituting the Lagrangian are:

$$\left(J_\theta + 2\int_0^\ell \left(m_R\left((x+r)^2+\xi^2\right) - p(x)\xi^{'2}\right)dx\right)\ddot{\theta} + 2\int_0^\ell m_R\left(r+x\right)\ddot{\xi}dx + \int_0^\ell 4m_R\xi\dot{\xi}\dot{\theta}dx - \int_0^\ell 4\dot{\theta}p(x)\xi'\dot{\xi}'dx = 0 \tag{A.20}$$

$$2m_R\left(r+x\right)\ddot{\theta} + 2m_R\ddot{\xi} - 2m_R\dot{\theta}^2\xi - 2\left(\dot{\theta}^2 p(x)\xi'\right)' + 2\left(EI\xi'' + \mu EI\dot{\xi}''\right)'' = 2(M_b(x,t))'' \tag{A.21}$$

The beam boundary conditions, due to the fixed root and free end, are given as:

$$\begin{gathered}\xi(x,t)|_{x=0} = \xi'(x,t)\Big|_{x=0} = 0, \\ (EI\xi'' + \mu EI\dot{\xi}'')\Big|_{x=\ell} = 0,\ (EI\xi'' + \mu EI\dot{\xi}'')'\Big|_{x=\ell} = 0\end{gathered} \tag{A.22}$$

The physical and structural properties of the bus and the solar array constrain the maximum slew angle that can be achieved by SASA. The linearized dynamics

of (A.20) and (A.21) are given as:

$$\int_0^\ell (J_\theta + 2m_R(x+r)^2)dx\ \ddot{\theta} + \int_0^\ell 2m_R(x+r)\ \ddot{\xi}\ dx = 0 \tag{A.23}$$

$$\int_0^\ell 2m_R(x+r)dx\ddot{\theta} + \int_0^\ell 2m_R\ddot{\xi}dx + \int_0^\ell 2\left(EI\xi''\right)''\ dx = \int_0^\ell 2(M_b(x,t))''dx \tag{A.24}$$

The linearized dynamics, (A.23) and (A.24), correspond to momentum equilibrium about the axis of rotation and force equilibrium of an element dx of the solar array, respectively. It is evident from these equations that the bus angular acceleration is bounded due to the bus inertia, beam inertia, beam mass distribution, and the shear force term $\int_0^\ell 2\left(EI\xi''\right)''\ dx$. The shear force $\int_0^\ell 2\left(EI\xi''\right)''\ dx$ across the solar array and the input strain actuation $\int_0^\ell 2(M_b(x,t))''dx$ are bounded due to the structural constraints on the beam. The bus angular rotation is bounded, because a finite definite integral of a bounded continuous function is bounded.

A.3 Nonlinear ODE-PDE Control Design of SASA

The equations of motion in (A.20) and (A.21) are re-written in Section A.3.1, followed by a matrix form that is typical for robot system dynamic models. Equations (A.20) and (A.21) are expressed in the standard Euler-Lagrangian matrix form in (A.25) by applying integration by parts to the term $\int_0^\ell \dot{\theta}p(x)\xi'\dot{\xi}'dx$ in (A.20).

A.3.1 Matrix Form of Euler-Lagrangian System

$$\int_0^\ell M_s(x)\begin{bmatrix}\ddot{\theta}\\ \ddot{\xi}\end{bmatrix}dx + \int_0^\ell C_s(x)\begin{bmatrix}\dot{\theta}\\ \dot{\xi}\end{bmatrix}dx + \begin{bmatrix}0\\ S_s\end{bmatrix} = \begin{bmatrix}0\\ 2\int_0^\ell (M_b(x,t))''dx\end{bmatrix} \tag{A.25}$$

$$\begin{aligned} M_s(x,t) &= \begin{bmatrix} m_{11}(\xi) & m_{12} \\ m_{12} & m_{22}\end{bmatrix} \\ &= \begin{bmatrix} \left(J_\theta/\ell + 2\left(m_R\left((x+r)^2+\xi^2\right) - p(x)\xi'^2\right)\right) & 2m_R\left(x+r\right) \\ 2m_R\left(x+r\right) & 2m_R \end{bmatrix} \end{aligned} \tag{A.26}$$

$$\begin{aligned} C_s(x,t) &= \begin{bmatrix} c_{11}(\xi,\dot{\xi}) & c_{12}(\xi,\dot{\theta}) \\ -c_{12}(\xi,\dot{\theta}) & 0\end{bmatrix} \\ &= \begin{bmatrix} 2\left(m_R\xi\dot{\xi} - p(x)\xi'\dot{\xi}'\right) & 2\left(m_R\xi\dot{\theta} + (p(x)\xi')'\dot{\theta}\right) \\ -2\left(m_R\xi\dot{\theta} + (p(x)\xi')'\dot{\theta}\right) & 0 \end{bmatrix} \end{aligned} \tag{A.27}$$

$$S_s(x,t) = 2\int_0^\ell \left(EI\xi'' + \mu EI\dot{\xi}''\right)'' dx \tag{A.28}$$

It is important to note that $c_{11} = \frac{\dot{m}_{11}}{2}$ and $\left(\dot{M}_s - 2C_s\right)$ is skew-symmetric. This skew-symmetric property is exploited in the stability proof of the main control law. Also, the boundary conditions in (A.22) still apply.

A.3.2 Preliminaries

Definition 2. (*Spatial $\mathcal{L}_2$-norm*) Given the spatial domain $[0,\ell]$ with $\ell > 0$, let the standard Hilbert space $\mathcal{Z} = \mathcal{L}_2([0,\ell],\mathbb{R}^n)$ be equipped with inner product $\langle z_1, z_2\rangle_{\mathcal{Z}} = \int_0^\ell z_1^T z_2 dx$ and induced norm $\|z\|_{\mathcal{Z}} = \sqrt{\int_0^\ell z^T z dx}$ for all spatial functions $z, z_1, z_2 : [0,\ell] \mapsto \mathbb{R}^n$ in $\mathcal{Z}$.

Definition 3. (*Temporal ℓ_p-norm*) Given the time signal $q(t) \in \mathbb{R}^n \ \forall\ t \geq 0$ and arbitrary $s \in [0,\infty)$, let $(q(t))_s$ be the truncation of $q(t)$ where $(q(t))_s = q(t)$ for $t \in [0,s]$, and $(q(t))_s = 0$ otherwise. Let the truncated ℓ_p signal norm be $\|(q)_s\|_{\ell_p} \triangleq (\int_0^s \|q(t)\|^p dt)^{\frac{1}{p}} < \infty$ for $p \in [1,\infty)$, and $\|(q)_s\|_{\ell_\infty} \triangleq \sup_{t\geq 0} \|(q(t))_s\| < \infty$ for $p = \infty$, where $\|\cdot\|$ is any vector norm.

Definition 4. (*Spatiotemporal norms*) Given $q(x,t) \in \mathbb{R}^n \ \forall\ x \in [0,\ell]$ and $t \geq 0$, let $\mathcal{Q} \triangleq \ell_p(\mathbb{R}^+, \mathcal{L}_2([0,\ell],\mathbb{R}^n))$ denote the Hilbert space with induced spatial $\mathcal{L}_2$ norm $\|q(t)\|_{\mathcal{L}_2} \triangleq \sqrt{\int_0^\ell q(x,t)^T q(x,t) dx}$ and truncated ℓ_p signal norm $\|q\|_{\mathcal{Q}} \triangleq \|(\|q(t)\|_{\mathcal{L}_2})_s\|_{\ell_p}$.

In this paper, there are two generalized coordinates θ and ξ such that $q(x,t) \triangleq [\theta(t), \xi(x,t)]^T \in \mathbb{R}^2$ where $\theta : \mathbb{R}^+ \mapsto \mathbb{R}$ and $\xi : [0,\ell] \times \mathbb{R}^+ \mapsto \mathbb{R}$.

A.3.3 Properties of the Underactuated Hybrid System

Kinetic Symmetry

The dynamics in (A.25) possesses kinetic symmetry [200] with respect to the spacecraft attitude θ, as the inertia matrix $M_s(x)$ per unit length is independent of θ. The kinetic symmetry with respect to θ in the absence of gravitational effects leads to symmetry in mechanics, satisfying (A.29).

$$\frac{\partial K}{\partial \theta} = \frac{\partial L}{\partial \theta} = 0 \tag{A.29}$$

$$z_1 = \frac{\partial L}{\partial \dot{\theta}} = \int_0^\ell (m_{11}\dot{\theta} + m_{12}\dot{\xi}) dx, \qquad \dot{z}_1 = \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} = \frac{\partial L}{\partial \theta} = 0 \tag{A.30}$$

The above is true because the corresponding Lagrangian L is independent of θ. Note that the first generalized angular momentum for the variable θ, which is denoted by z_1, is given in (A.30).

Positive Definiteness and Uniform Boundedness of the Inertia Matrix

The positive definiteness and uniform bounds of the inertia matrix are used in the Lyapunov analysis of the closed-loop system to prove exponential stability. The inertia matrix here, unlike rigid-body robot dynamics, has axial stiffening component $-p(x)\xi'^2$ in the m_{11} term due to the centrifugal force on the solar array, which can make the inertia matrix non-positive definite. The effect of axial stiffing on the multibody dynamics is studied in [182, 214]. Using the Sylvester's criterion, we obtain the inequalities presented in (A.31) for the positive definiteness of the M_s matrix.

$$\begin{aligned} J_\theta/\ell + 2\left(m_R\left((x+r)^2+\xi^2\right) - p(x)\xi'^2\right) > 0 \\ 2m_R J_\theta/\ell + 4m_R^2\xi^2 - 4m_R p(x)\xi'^2 > 0 \end{aligned} \tag{A.31}$$

These inequalities are always satisfied due to the small deflection assumption of the beam. Assuming constant m_R, the inequalities in the (A.31) can be simplified to (A.32).

$$2\xi^2 - \xi'^2((\ell-x)(\ell+x+2r)) > -\frac{J_\theta}{\ell m_R} \tag{A.32}$$

The maximum of $((\ell-x)(\ell+x+2r))$ occurs at $x=0$. Equation (A.32) is satisfied if $2\xi^2 - \xi'^2(\ell^2+2\ell r) > -\frac{J_\theta}{\ell m_R}$. This is used to define a domain of permissible displacement and strain envelope as $2\xi^2 + \frac{J_\theta}{\ell m_R} > \xi'^2(\ell^2+2\ell r)$. Let the bounds of the inertia matrix be given by using the spatial $\mathcal{L}_2$ norm as follows:

$$0 < \sigma_1\|q(t)\|_{\mathcal{L}_2}^2 \leq \int_0^\ell q^T M_s(x,t) q dx \leq \sigma_2\|q(t)\|_{\mathcal{L}_2}^2 \text{ ; where } q(t) = (\theta,\xi)^T \tag{A.33}$$

The bounds are not uniform because they depend on the state ξ, if $\xi \to \infty$ both σ_1 and σ_2 grow unbounded. In the configuration space of the system, $\xi(x)$ at any time t is a function of the moment applied at that location and $\xi \leq \xi_{\max}$, where $\xi_{\max}$ is the maximum strain that can be achieved before violating structural constraints of the beam. For the Euler-Bernoulli beam considered, using the strain-moment relationship for symmetric bending $\xi_{\max} = \frac{hM_{\max}}{EI}$, where the thickness of the beam is $2h$, $M_{\max}$ is the maximum moment that can be applied at a point, E is the Young's modulus and I is the moment of inertia respectively. Therefore, we can always find σ_1 and σ_2, which are uniform bounds on the inertia matrix M_s.

A.3.4 Nonlinear ODE-PDE Control Law

The control objective is to ensure that the bus rotation angle $\theta(t) \in C^2([0,\infty))$ tracks a desired attitude trajectory $\theta_d(t) \in C^2([0,\infty))$. The nonlinear control law is developed using the properties discussed in Section A.3.3. To track a desired bus rotation trajectory θ_d, a reference rotation velocity signal, as shown in (A.34), is computed by shifting the rotation velocity with the position error term in which λ_θ is a positive value. The reference rotation acceleration is obtained by taking the time derivative of the (A.34):

$$\dot{\theta}_r(t) = \dot{\theta}_d(t) - \lambda_\theta(\theta - \theta_d(t)) \tag{A.34}$$

$$\ddot{\theta}_r(t) = \ddot{\theta}_d(t) - \lambda_\theta(\dot{\theta} - \dot{\theta}_d(t)) \tag{A.35}$$

The reference beam deflection signal ξ_r is computed by integrating (A.36). The reference signal ξ_r quantifies the θ dynamics for the reference signal θ_r that is augmented with the error feedback term $K_\theta\left(\dot{\theta} - \dot{\theta}_r\right)$, where K_θ is a positive feedback gain.

$$m_{12}\ddot{\xi}_r + m_{11}\ddot{\theta}_r + c_{11}\dot{\theta}_r + c_{12}\dot{\xi}_r = K_\theta(\dot{\theta} - \dot{\theta}_r) \tag{A.36}$$

Using (A.34), (A.35), and (A.36), and variables $s_\theta = \dot{\theta} - \dot{\theta}_r$ and $s_\xi = \dot{\xi} - \dot{\xi}_r$, the control law is defined in (A.37), where $\tau = (M_b(x,t))''$.

$$\tau = m_{12}\ddot{\theta}_r + m_{22}\ddot{\xi}_r - c_{12}\dot{\theta}_r + 2\left(\delta EI\xi'' + \mu EI\delta\dot{\xi}''\right)'' - K_\xi s_\xi \tag{A.37}$$

The closed-loop system with this controller is given in (A.39), where δ is a constant $\in [0,1]$, K_ξ is a positive feedback gain and $q_s(x,t) = \left[\begin{smallmatrix} s_\theta \\ s_\xi \end{smallmatrix}\right]$.

$$K_s = \begin{bmatrix} K_\theta & 0 \\ 0 & K_\xi \end{bmatrix}; \quad D(\delta, \xi'', \dot{\xi}'') = \begin{bmatrix} 0 \\ (1-\delta)\left(2EI\xi'' + \mu 2EI\dot{\xi}''\right)'' \end{bmatrix} \tag{A.38}$$

$$\int_0^\ell \left(M_s\dot{q}_s + C_s q_s + D(\delta, \xi'', \dot{\xi}'') + K_s q_s\right) dx = 0 \tag{A.39}$$

Notice that the parameter δ defines an auxiliary control input in the following sense. If $\delta = 1$, the beam bending stiffness term ($S_s(x,t)$ in (A.28)) is completely nulled (e.g., via feed-forward control), and if $\delta = 0$, the beam bending dynamics are natural (uncompensated). For $\delta = 1$, exponential stability of the closed-loop system is given in Theorem 9, and the robust stability against a bounded disturbance term is discussed in Theorem 10.

Theorem 9. *The closed-loop system, given in (A.40), with the control law (A.34–A.37) and $\delta = 1$, is globally exponentially stable in the sense of spatial $\mathcal{L}_2$ norm, defined in Definition 2, assuming the feedback gains $K_\xi, K_\delta > 0$ and the inertia matrix M_s is positive definite by (A.31) and uniformly bounded as in (A.33).*

$$\int_0^\ell \left(M_s\dot{q}_s + C_s q_s + K_s q_s\right) dx = 0 \tag{A.40}$$

Proof. The inertia matrix M_s, due to the properties discussed in Sections A.3.1 and A.3.3, is used to compute a spatial norm (A.41) for a Lyapunov-like stability analysis [193]:

$$V_L = \frac{1}{2}\int_0^\ell q_s^T M_s(x,t) q_s dx \tag{A.41}$$

The time-derivative of V_L is given as follows:

$$\dot{V}_L = \int_0^\ell q_s^T M_s \dot{q}_s dx + \frac{1}{2}\int_0^\ell q_s^T \dot{M}_s q_s dx; \text{ where } \dot{M}_s = \left[\begin{smallmatrix} \dot{m}_{11}(\xi) & 0 \\ 0 & 0 \end{smallmatrix}\right] \tag{A.42}$$

Using $\frac{\dot{m}_{11}(\xi)}{2} = c_{11}$, and adding and subtracting $c_{12}(\xi,\dot{\theta})s_\xi s_\theta$, it can be simplified to (A.43).

$$\dot{V}_L = -\int_0^\ell q_s^T K_s q_s \leq -\frac{2K}{\sigma_2} V_L \tag{A.43}$$

In the above inequality, we used the uniform bounds on the inertia matrix as given in (A.33), and $K = \min\{K_\theta, K_\xi\} > 0$. Also, note that the skew-symmetric property of $\dot{M}_s - 2C_s$ is used. Using the Comparison Lemma [158], and the bounds on the inertia matrix, we obtain:

$$\begin{aligned} V_L(t) &\leq \exp\left(-\frac{2K}{\sigma_2}t\right) V_L(0) \\ \left\| \left[\begin{smallmatrix} s_\theta(t) \\ s_\xi(x,t) \end{smallmatrix}\right] \right\|_{\mathcal{L}_2}^2 &\leq \frac{\sigma_2}{\sigma_1} \exp\left(-\frac{2K}{\sigma_2}t\right) \left\| \left[\begin{smallmatrix} s_\theta(0) \\ s_\xi(x,0) \end{smallmatrix}\right] \right\|_{\mathcal{L}_2}^2 \\ \left\| \left[\begin{smallmatrix} s_\theta(t) \\ s_\xi(x,t) \end{smallmatrix}\right] \right\|_{\mathcal{L}_2} &\leq \sqrt{\frac{\sigma_2}{\sigma_1}} \exp\left(-\frac{K}{\sigma_2}t\right) \left\| \left[\begin{smallmatrix} s_\theta(0) \\ s_\xi(x,0) \end{smallmatrix}\right] \right\|_{\mathcal{L}_2} \end{aligned} \tag{A.44}$$

Therefore, the system in (A.40) converges exponentially to the manifold $s_\theta = 0$ and $s_\xi = 0$ for a positive definite gain matrix in the sense of $\mathcal{L}_2$, $\int_0^\ell (\dot{\theta} - \dot{\theta}_r)^2 dx \to 0$ and $\int_0^\ell (\dot{\xi} - \dot{\xi}_r)^2 dx \to 0$, implying $\dot{\theta} \to \dot{\theta}_r$ and $\dot{\xi} \to \dot{\xi}_r$. To prove $\theta \to \theta_d$, we need to subtract $\dot{\theta}$ from both sides of the (A.34) to obtain $\dot{\theta}_r(t) - \dot{\theta} = \dot{\theta}_d(t) - \dot{\theta} - \lambda_\theta(\theta - \theta_d(t))$, and use the fact that $\left(\dot{\theta} - \dot{\theta}_r\right) \to 0$ to form the first-order ODE, $\dot{\theta} - \dot{\theta}_d(t) = -\lambda_\theta(\theta - \theta_d(t))$. This equation becomes $\dot{e} = -\lambda e$ with $e = \theta - \theta_d(t)$, which implies $e(t) = e(0)\exp[-\lambda_\theta t]$ for any initial tracking error condition $e(0) \in \mathbb{R}$. Thus $\theta \to \theta_d$ holds globally with an exponential rate of convergence $\lambda_\theta > 0$. □

If $\delta \neq 1$, the beam dynamics term $D(\delta, \xi'', \dot{\xi}'')$ is regarded as a bounded disturbance at the input of the closed-loop system in (A.40), where $\delta \in [0, 1)$. The closed-loop dynamics in (A.39) can be written as the following equation:

$$\int_0^\ell \left(M_s \dot{q}_s + C_s q_s + K_s q_s\right) dx = -\int_0^\ell D(\delta, \xi'', \dot{\xi}'') dx \tag{A.45}$$

In order to satisfy the structural constraints and the Euler-Bernoulli beam assumption, the term $D(\delta, \xi'', \dot{\xi}'')$ must be bounded in the context of the Definitions 2-4. This situation can be generalized to a case when the closed-loop system in (A.40) has a bounded perturbation $\int_0^\ell d(x,t)dx$ at the input, as defined in Theorem 10. The stability of the closed-loop system in (A.45) is essential for tracking a trajectory when $\delta \neq 1$ with a bounded tracking error. In the Theorem 10, we prove the finite-gain ℓ_p stability of the closed-loop system in (A.40) in the sense of spatial $\mathcal{L}_2$ norm by approximating $-\int_0^\ell D(\delta, \xi'', \dot{\xi}'')dx$ in the (A.45) as a general bounded disturbance term $\int_0^\ell d(x,t)dx$ as an the input to the closed-loop system in (A.40).

Theorem 10. *The closed-loop system, given in (A.40), with a bounded perturbation at the input of the form* $\int_0^\ell d(x,t)dx = \int_0^\ell \left(\begin{bmatrix} \delta_\theta \\ \delta_\xi \end{bmatrix} + \begin{bmatrix} \gamma_\theta & 0 \\ 0 & \gamma_\xi \end{bmatrix} \begin{bmatrix} s_\theta \\ s_\xi \end{bmatrix} \right) dx$ *exponentially converges to the error ball,* $\lim_{t\to\infty} \left\| \begin{bmatrix} s_\theta(t) \\ s_\xi(x,t) \end{bmatrix} \right\|_{\mathcal{L}_2} \to \frac{\sigma_2 \gamma_\infty}{\sigma_1 K}$*, assuming that the inertia matrix* M_s *is positive definite by (A.31) and uniformly bounded by (A.33). Also, it is assumed that the nonvanishing perturbation term at the origin,* $\delta_{\mathcal{L}_2}(t) = \left\| \begin{bmatrix} \delta_\theta \\ \delta_\xi \end{bmatrix} \right\|_{\mathcal{L}_2}$ *satisfies the bound* $\sup_{t\geq 0} |\delta_{\mathcal{L}_2}(t)| \leq \gamma_\infty$*. Note that the functions* $\gamma_\theta, \gamma_\xi : \mathbb{R} \to \mathbb{R}$ *are nonnegative and continuous* $\forall\ t \geq 0$ *and* γ_∞ *is a positive constant. Furthermore, (A.40) is finite-gain* ℓ_p *stable* $\forall\ p \in [1, \infty]$ *in the sense of spatial* $\mathcal{L}_2$ *norm, given that the* ℓ_p *time norm of* $\delta_{\mathcal{L}_2}(t)$ *is bounded, as in* $\|(\delta_{\mathcal{L}_2}(t))_s\|_{\ell_p} \leq \gamma_p$*, where* γ_p *is a positive constant.*

Proof. We proceed to take the V_L as in (A.41), $V_L = \frac{1}{2}\int_0^\ell q_s^T M_s(x,t) q_s dx$. Using the skew-symmetric property of $\dot{M}_s - 2C_s$, its time-derivative simplifies to the following (A.46).

$$\dot{V}_L \leq -\frac{2}{\sigma_2} \min\{K_\theta, K_\xi\} V_L + \frac{2}{\sigma_1} \max\{\gamma_\theta, \gamma_\xi\} V_L + \int_0^\ell (s_\theta \delta_\theta + s_\xi \delta_\xi) dx \tag{A.46}$$

Let $K \triangleq \left(\min\{K_\theta, K_\xi\} - \frac{\sigma_2}{\sigma_1} \max\{\gamma_\theta, \gamma_\xi\} \right)$. For sufficiently large $K_\theta, K_\xi > 0$,

we can prove $K > 0, \forall\, t \geq 0$. Using the Cauchy-Schwarz inequality,

$$\dot{V}_L \leq -\frac{2K}{\sigma_2} V_L + \left\| \begin{bmatrix} s_\theta(t) \\ s_\xi(x,t) \end{bmatrix} \right\|_{\mathcal{L}_2} \left\| \begin{bmatrix} \delta_\theta \\ \delta_\xi \end{bmatrix} \right\|_{\mathcal{L}_2} \tag{A.47}$$

$$\leq -\frac{2K}{\sigma_2} V_L + \sqrt{\frac{2V_L}{\sigma_1}} \delta_{\mathcal{L}_2}(t) \tag{A.48}$$

Using the transformation $W = \sqrt{V_L}$, followed by application of the Comparison Lemma [158] and the uniform bounds on the inertia matrix (A.33), we get the following inequality in terms of spatial $\mathcal{L}_2$ norm of states.

$$\left\| \begin{bmatrix} s_\theta(t) \\ s_\xi(x,t) \end{bmatrix} \right\|_{\mathcal{L}_2} \leq \sqrt{\frac{\sigma_2}{\sigma_1}} \exp\left(-\frac{Kt}{\sigma_2}\right) \left\| \begin{bmatrix} s_\theta(0) \\ s_\xi(x,0) \end{bmatrix} \right\|_{\mathcal{L}_2} + \frac{1}{\sigma_1} \int_0^t \left(\delta_{\mathcal{L}_2}(\tau) \exp\left(-\frac{K(t-\tau)}{\sigma_2}\right) \right) d\tau \tag{A.49}$$

Given that $\sup_{\tau \geq 0} |\delta_{\mathcal{L}_2}(\tau)| \leq \gamma_\infty$, moving it to outside the integral and expanding the integral, we get the following inequality.

$$\left\| \begin{bmatrix} s_\theta(t) \\ s_\xi(x,t) \end{bmatrix} \right\|_{\mathcal{L}_2} \leq \sqrt{\frac{\sigma_2}{\sigma_1}} \exp\left(-\frac{Kt}{\sigma_2}\right) \left\| \begin{bmatrix} s_\theta(0) \\ s_\xi(x,0) \end{bmatrix} \right\|_{\mathcal{L}_2} + \frac{\sigma_2 \gamma_\infty}{\sigma_1 K} \left(1 - \exp\left(-\frac{Kt}{\sigma_2}\right)\right) \tag{A.50}$$

By taking a limit $t \to \infty$ on both sides of (A.50), we get the bound, $\lim_{t\to\infty} \left\| \begin{bmatrix} s_\theta(t) \\ s_\xi(x,t) \end{bmatrix} \right\|_{\mathcal{L}_2} \to \frac{\sigma_2 \gamma_\infty}{\sigma_1 K}$, on the spatial $\mathcal{L}_2$ norm of the states. Taking ℓ_p norm on both sides of the (A.49), with $q_s(x,t) = [s_\theta, s_\xi]$, we get the following inequality.

$$\|q_s\|_{\mathcal{Q}} \leq \sqrt{\frac{\sigma_2}{\sigma_1}} \|q_s(x,0)\|_{\mathcal{L}_2} \left\| \exp\left(-\frac{Kt}{\sigma_2}\right) \right\|_{\ell_p} + \frac{1}{\sigma_1} \left\| \exp\left(-\frac{Kt}{\sigma_2}\right) \right\|_{\ell_1} \|\delta_{\mathcal{L}_2}(t)\|_{\ell_p} \tag{A.51}$$

$$\leq \sqrt{\frac{\sigma_2}{\sigma_1}} \|q_s(x,0)\|_{\mathcal{L}_2} \zeta + \frac{\sigma_2}{K\sigma_1} \|\delta_{\mathcal{L}_2}(t)\|_{\ell_p} \tag{A.52}$$

$$\leq \sqrt{\frac{\sigma_2}{\sigma_1}} \|q_s(x,0)\|_{\mathcal{L}_2} \zeta + \frac{\sigma_2 \gamma_p}{\sigma_1 K} \tag{A.53}$$

In (A.52), $\zeta = 1$ if $p = \infty$ and $\zeta = \frac{1}{(Kp/\sigma_2)^{\frac{1}{p}}}$ if $p \in [1, \infty)$. Thus, the closed-loop system with bounded disturbance is shown to be finite-gain time-signal ℓ_p norm stable with ℓ_p and $\mathcal{L}_2$ norms defined in Definitions 2-4 using Theorem 5.1 in [158]. □

Remark. Theorem 9 offers global exponential stability in the spatial $\mathcal{L}_2$ norm sense under the implicit assumption that the underlying Euler-Bernoulli beam model remains valid *globally*; i.e., for all bending states with arbitrarily large strain energy.

In practice, overloading physical beams causes plastic deformations which violate elastic beam model assumptions. To mitigate this issue, an auxiliary momentum management control system [175] is often needed to reduce strain levels before the onset of plasticity is allowed to occur.

In the control law in (A.37), not canceling the beam dynamics completely is a convenient way to add bounded disturbance at the input during simulations and experiments. The dynamics and the control laws discussed so far are all in PDE form. In the following section, we will discuss the implementation of the closed-loop system for simulations and experimental validation using finite-dimensional approximations of the system dynamics. The values of γ_ξ and γ_θ that can be overcome with a high gain depends on the actuator saturation limits.

A.4 Control Implementation and Simulation Results

A.4.1 Galerkin Formulation

The ODE-PDE model is discretized to obtain an ODE model using the Galerkin method [177, 180] for numerical simulations and experiments. The beam deflection is approximated as $\xi(x,t) = \phi^T(x)\eta(t)$, where $\phi(x)$ and $\eta(t)$ are continuously differentiable functions with respect to x, t respectively. $\phi(x)$ are Galerkin functions. These functions comprise an approximation of the basis for the space of possible deflections of the flexible beam. The discretization is implemented by minimizing the weighted residual of the $\xi(x,t)$ dynamics (see (A.54)).

$$\begin{aligned}&\int_0^\ell \phi\left(m_R\left(r+x\right)\ddot{\theta} + m_R\ddot{\xi} - m_R\dot{\theta}^2\xi\right)dx \\ &-\int_0^\ell \phi\left(\left(\dot{\theta}^2 p(x)\xi^{'}\right)^{'} + \left(EI\xi^{''} + \mu EI\dot{\xi}^{''}\right)^{''} - M_b^{''}(x,t)\right)dx = 0\end{aligned} \tag{A.54}$$

The Galerkin functions $\phi_j(x)$, where $j \in \{1,2,3,4\}$, in (A.55) are taken from [180], and are chosen to satisfy boundary conditions. We use four Galerkin functions $\phi = [\phi_1(x), \phi_2(x), \phi_3(x), \phi_4(x)]^T$, which describe dynamics sufficiently for tracking a sinusoidal attitude trajectory and slewing shown here, as the frequency of the trajectory chosen is well below the first mode of the composite beam.

$$\phi_j(x) = 1 - \cos\left(\frac{j\pi x}{\ell}\right) + \frac{1}{2}(-1)^{j+1}\left(\frac{j\pi x}{\ell}\right)^2 \tag{A.55}$$

A.4.2 Dynamics in Galerkin Form

The equations of motion in Galerkin form are given in (A.56). The matrices $[A]$, $[B]$, and $[C]$ are used to define the mass matrix $[M_g]$ and the gyroscopic cross coupling terms in matrix $[C_g]$. The matrix $[E]$ corresponds to beam stiffness. The matrices in (A.59) are referred to as Galerkin matrices in this paper.

$$[\boldsymbol{M_g}]\begin{bmatrix}\ddot{\theta}\\ \ddot{\boldsymbol{\eta}}\end{bmatrix}+[\boldsymbol{C_g}]\begin{bmatrix}\dot{\theta}\\ \dot{\boldsymbol{\eta}}\end{bmatrix}+\begin{bmatrix}0\\ 2[\boldsymbol{E}](\boldsymbol{\eta}+\mu\dot{\boldsymbol{\eta}})\end{bmatrix}=\begin{bmatrix}0\\ \int_0^\ell 2\boldsymbol{\phi}M_b^{''}dx\end{bmatrix} \tag{A.56}$$

$$\begin{aligned}[\boldsymbol{M_g}] &= \begin{bmatrix} m_{11_g} & m_{12_g}\\ m_{21_g} & m_{22_g}\end{bmatrix}\\ &= \begin{bmatrix}\left(J_\theta+2\int_0^\ell m_R\left(x+r\right)^2dx+2\boldsymbol{\eta}^T\left([\boldsymbol{A}]-[\boldsymbol{B}]\right)\boldsymbol{\eta}\right) & 2[\boldsymbol{C}]\\ 2[\boldsymbol{C}]^T & 2[\boldsymbol{A}]\end{bmatrix}\end{aligned} \tag{A.57}$$

$$[\boldsymbol{C_g}]=\begin{bmatrix}c_{11_g} & c_{12_g}\\ c_{21_g} & c_{22_g}\end{bmatrix}=\begin{bmatrix}2\dot{\boldsymbol{\eta}}^T\left([\boldsymbol{A}]-[\boldsymbol{B}]\right)\boldsymbol{\eta} & 2\boldsymbol{\eta}^T\left([\boldsymbol{A}]-[\boldsymbol{B}]\right)\dot{\theta}\\ -2([\boldsymbol{A}]-[\boldsymbol{B}])\boldsymbol{\eta}\dot{\theta} & 0\end{bmatrix} \tag{A.58}$$

$$\begin{aligned}[\boldsymbol{A}]&=\int_0^\ell m_R\boldsymbol{\phi}\boldsymbol{\phi}^Tdx,\quad [\boldsymbol{B}]=\int_0^\ell p\boldsymbol{\phi}^{'}\boldsymbol{\phi}^{'T}dx,\\ [\boldsymbol{C}]&=\int_0^\ell m_R(x+r)\boldsymbol{\phi}^Tdx,\quad [\boldsymbol{E}]=\int_0^\ell\boldsymbol{\phi}(EI\boldsymbol{\phi}^{''T})''dx\end{aligned} \tag{A.59}$$

A.4.3 Computation of Voltage Signal from Control Signal

The voltage signal, which is the input to the PZT actuator, is computed from the control signal in Galerkin form $\int_0^\ell 2\boldsymbol{\phi}M_b^{''}dx$ during experiments as follows. For n discrete PZT actuators bonded onto the solar array (see Fig. A.2b), the moment produced can be modeled as described in (A.12), which is used in the following:

$$\tau=\int_0^\ell\phi M_b^{''}dx \tag{A.60}$$

$$=\int_0^\ell\phi\sum_{i=1}^n c_iV_i(t)\left(u(x-l_{1i})-u(x-l_{2i})\right)^{''}dx \tag{A.61}$$

$$=\sum_{i=1}^n c_iV_i(t)\int_0^\ell\phi\left(u(x-l_{1i})-u(x-l_{2i})\right)^{''}dx \tag{A.62}$$

where the function u is a step function, whose derivative is the Dirac delta function δ_d. The term $\int_0^\ell\phi\left(u(x-l_{1i})-u(x-l_{2i})\right)^{''}dx$ can be simplified to $\int_0^\ell-\phi^{'}\left(\delta_d(x-l_{1i})-\delta_d(x-l_{2i})\right)dx$ using integration by parts. Using the shifting

property of the impulse function, we get (A.63):

$$\tau = \sum_{i=1}^{n} c_i V_i(t) \left(\phi^{'}\left(l_{2i}\right) - \phi^{'}\left(l_{1i}\right) \right) \tag{A.63}$$

With two PZT actuators on the beam, the voltage signal is computed using (A.63) as follows:

$$\mathbf{\Phi} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \tau, \text{ where } \mathbf{\Phi} = \begin{bmatrix} \phi_1^{'}\left(l_{21}\right) - \phi_1^{'}\left(l_{11}\right) & \phi_1^{'}\left(l_{22}\right) - \phi_1^{'}\left(l_{12}\right) \\ \phi_2^{'}\left(l_{21}\right) - \phi_2^{'}\left(l_{11}\right) & \phi_2^{'}\left(l_{22}\right) - \phi_2^{'}\left(l_{12}\right) \\ \phi_3^{'}\left(l_{21}\right) - \phi_3^{'}\left(l_{11}\right) & \phi_3^{'}\left(l_{22}\right) - \phi_3^{'}\left(l_{12}\right) \\ \phi_4^{'}\left(l_{21}\right) - \phi_4^{'}\left(l_{11}\right) & \phi_4^{'}\left(l_{22}\right) - \phi_4^{'}\left(l_{12}\right) \end{bmatrix}, \tag{A.64}$$

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \left(\mathbf{\Phi}^T \mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^T \tau \tag{A.65}$$

Equation (A.65) is the optimal least squares solution to (A.64), assuming all the columns of $\mathbf{\Phi}$ are independent. The model can be simplified to the case with one PZT by removing the second column of the Galerkin function dependent matrix in (A.64).

A.4.4 Nonlinear Controller in Galerkin Form

The reference signal ξ_r can be approximated using Galerkin functions as $\xi_r = \phi^T(x)\eta_r(t)$ assuming it satisfies the boundary conditions on ξ. The control law discussed in Section A.3.4 can be expressed in Galerkin form by minimizing the weighted residual.

$$\begin{gathered} s_\theta = \dot{\theta} - \dot{\theta}_r, \;\; s_\xi = \phi^T\left(\dot{\eta} - \dot{\eta}_r\right) \\ m_{12_g}\ddot{\eta}_r + c_{12_g}\dot{\eta}_r = K_\theta(\dot{\theta} - \dot{\theta}_r)\ell - m_{11_g}\ddot{\theta}_r - c_{11_g}\dot{\theta}_r \end{gathered} \tag{A.66}$$

$$\begin{aligned} \tau = \int_0^\ell 2\boldsymbol{\phi} M_b^{''} dx = m_{21_g}\ddot{\theta}_r + m_{22_g}\ddot{\eta}_r + c_{21_g}\dot{\theta}_r + 2\delta[E](\eta + \mu\dot{\eta}) \\ - K_\xi \int_0^\ell \boldsymbol{\phi}\boldsymbol{\phi}^T dx \left(\dot{\eta} - \dot{\eta}_r\right) \end{aligned} \tag{A.67}$$

The control effort given in (A.67) is used to compute the voltage using (A.65).

A.4.5 Simulation Results

In this section, we apply the proposed control law to a cylinder with flexible composite beams as shown in Fig. A.3. The model corresponds to the experimental

setup described in Section A.5, whose nominal design parameters are given in Table A.1. We simulate numerically the closed-loop system for controlled rotation of the cylinder by applying a voltage to the piezoelectric actuators on the beam. Here we track a sinusoidal signal of amplitude 0.001 radians and frequency 0.02 Hz, and slew from a zero initial attitude to 0.001 radians for different values of δ. We present results for two cases: 1) Configuration 1: with only 'PZT LEFT 1' and 'PZT RIGHT 1' active (see Fig. A.3), and 2) Configuration 2: Distributed configuration, with all four PZT actuators active. The numerical implementation of the closed-loop discretized system in ODE form is shown in Fig. A.4.

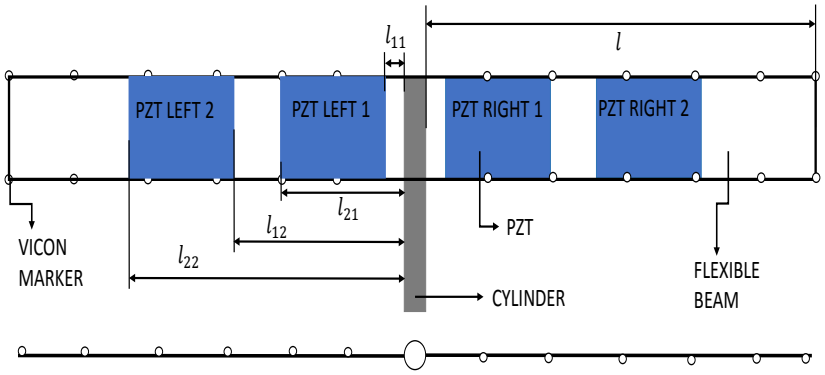


Figure A.3: Front and top view of the experimental setup.

Table A.1: Physical, geometrical and structural parameters of bus, array, and PZT.

ℓ	29.7×10^{-2} m	E_b	68.9 GPa	E_p	66 GPa
t_b	0.45×10^{-3} m	ρ_b	2738 kg/m^3	ρ_p	7800 kg/m^3
w	0.036 m	μ	10^{-4}	t_p	0.48×10^{-3} m
d_{31}	190×10^{-12} m/V	ℓ_{11}	1.1×10^{-2}m	ℓ_{21}	8.144×10^{-2} m
ℓ_{12}	8.614×10^{-2} m	ℓ_{22}	15.858×10^{-2} m	r	0.6×10^{-2}
J_θ	1.0759×10^{-6} m				

The Galerkin matrices are computed using (A.59) for the configuration shown in Fig. A.3. Using the Galerkin matrices, the closed-loop system defined by the ODE given in (A.56)-(A.59) and (A.66)-(A.67) can be integrated using Euler's method. The Galerkin matrices are computed offline for the nominal design parameters. The closed-loop system is simulated using Simulink with the Galerkin matrices. The trajectory tracking and slewing is achieved for $\delta = \{0, 0.5, 1\}$ with the gain values specified in the figures.

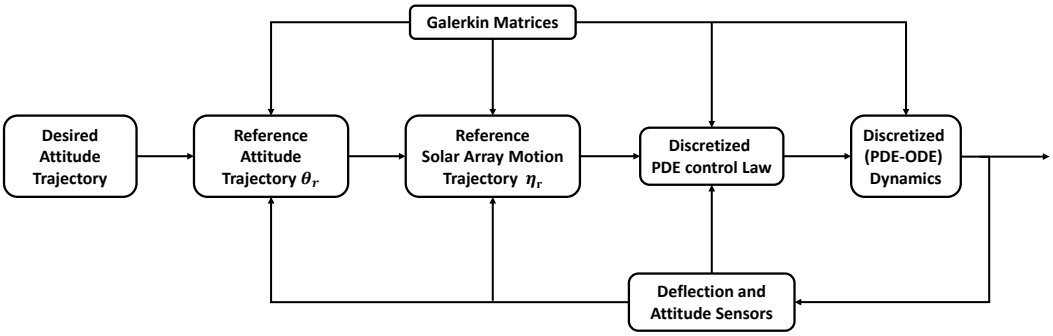
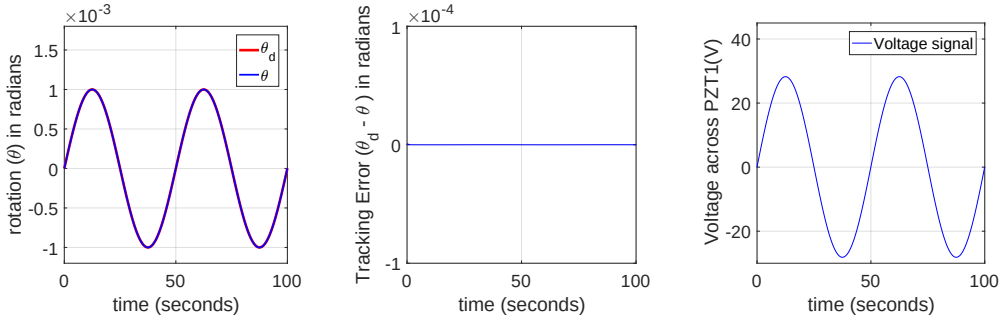


Figure A.4: Closed-loop ODE system as implemented in Simulink for simulations.

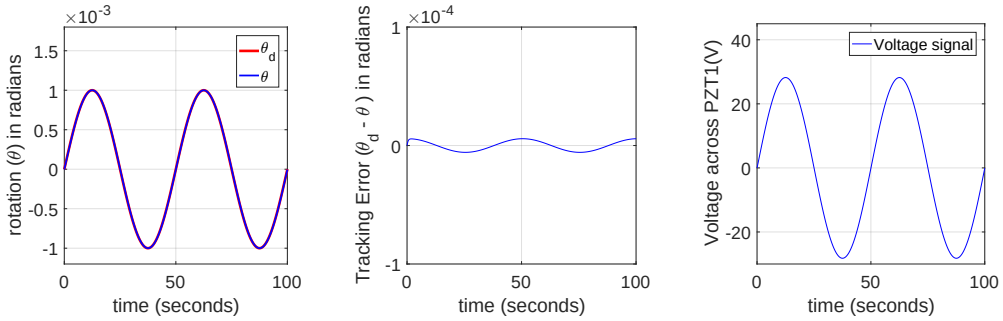
For configuration 1, we present results for $\delta = \{1, 0.5, 0\}$; see Figs. A.5–A.7 for trajectory tracking, and Figs. A.8–A.10 for slewing. If the closed-loop simulation is performed with the same gain values for the three δ values, it was observed that the tracking error is inversely proportional to the δ value in the range $[0, 1]$. For $\delta = 1$, trajectory tracking is achieved exponentially fast. The voltage signals were computed from the control signal offline using (A.65). Note that the maximum voltage required to do the tracking and slewing is well within ± 200 V, which is the saturation limit of the PZT actuators used in experiments. It is observed that controller 1 predominantly uses the first free vibration mode of the beam to achieve tracking. For slewing results, the time required to reach steady state is increased for smaller δ. Figures A.11–A.14 show tracking and slewing in configuration 2. Based on these simulation results, experiments are performed for values of δ for which control effort is within ± 200 V.



(a) Trajectory tracking (b) Tracking error (c) Control effort

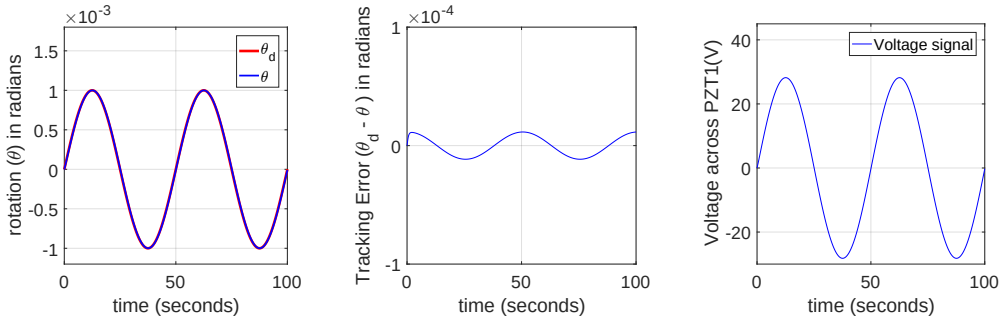
d

Figure A.5: Trajectory tracking (simulation) for $\delta = 1$ with gains $\lambda_\theta = 3$, and $K_\theta = 0.5$, $K_\xi = 0.5$.



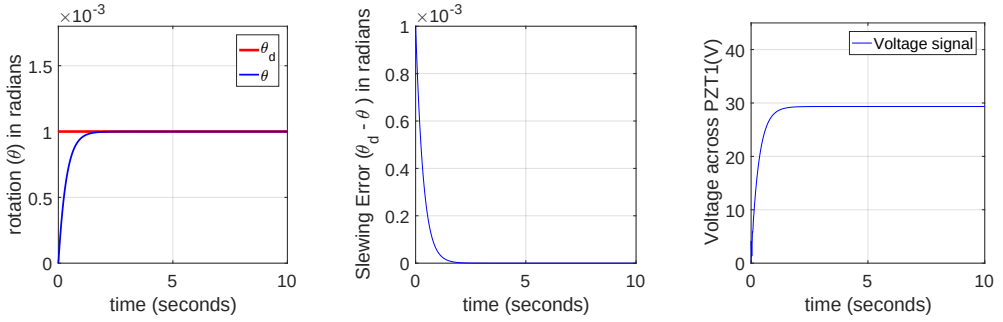
(a) Trajectory tracking (b) Tracking error (c) Control effort

Figure A.6: Trajectory tracking (simulation) for $\delta = 0.5$ with gains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$.



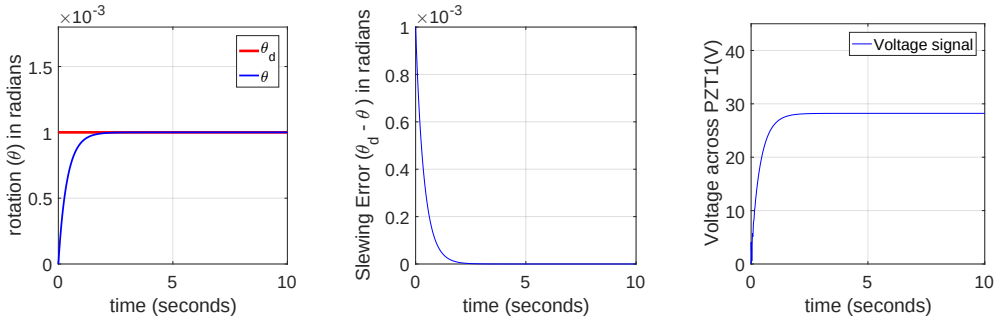
(a) Trajectory tracking (b) Tracking error (c) Control effort

Figure A.7: Trajectory tracking (simulation) for $\delta = 0$ with gains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$.



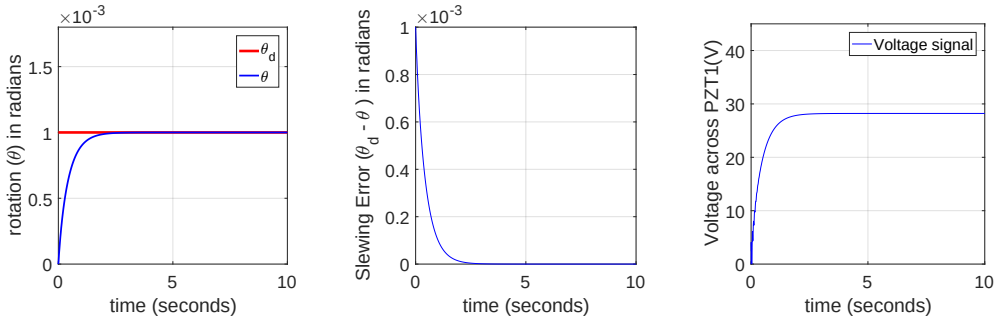
(a) Slewing (b) Slewing error (c) Control effort

Figure A.8: Slewing (simulation) for $\delta = 1$ with gains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$.



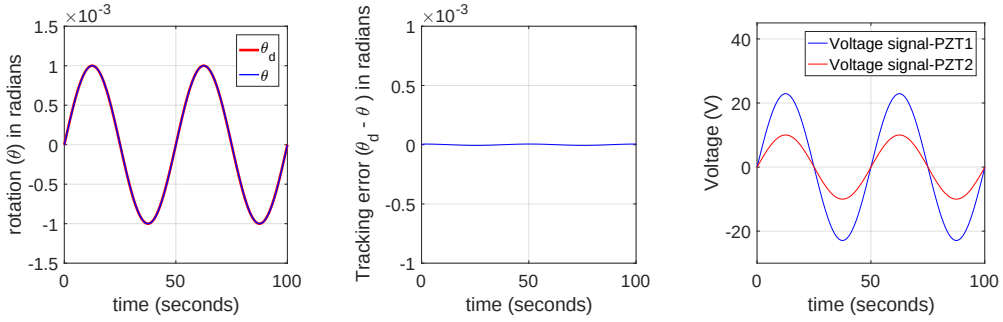
(a) Slewing (b) Slewing error (c) Control effort

Figure A.9: Slewing (simulation) for $\delta = 0.5$ with gains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$.



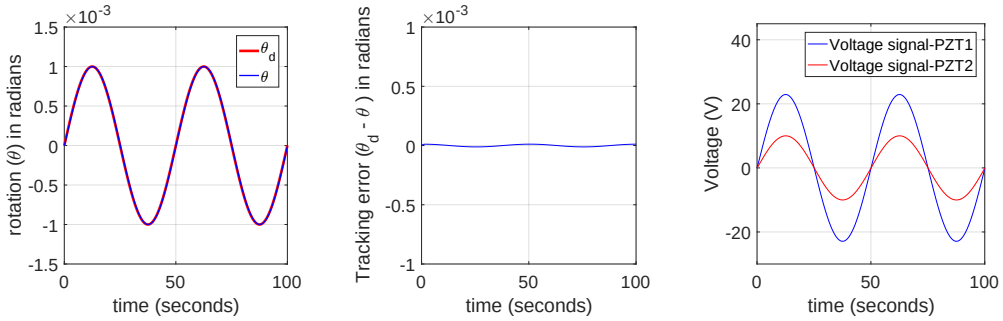
(a) Slewing (b) Slewing error (c) Control effort

Figure A.10: Slewing (simulation) for $\delta = 0$ with gains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$.



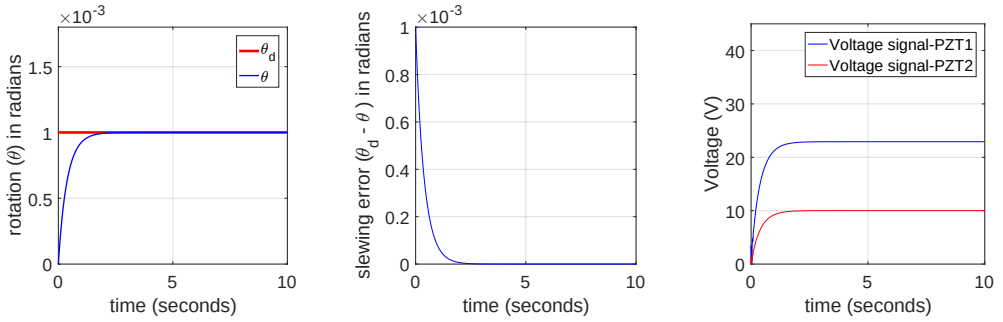
(a) Trajectory tracking (b) Tracking error (c) Control effort

Figure A.11: Trajectory tracking (simulation) for $\delta = 0.5$ in configuration 2 with gains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$.



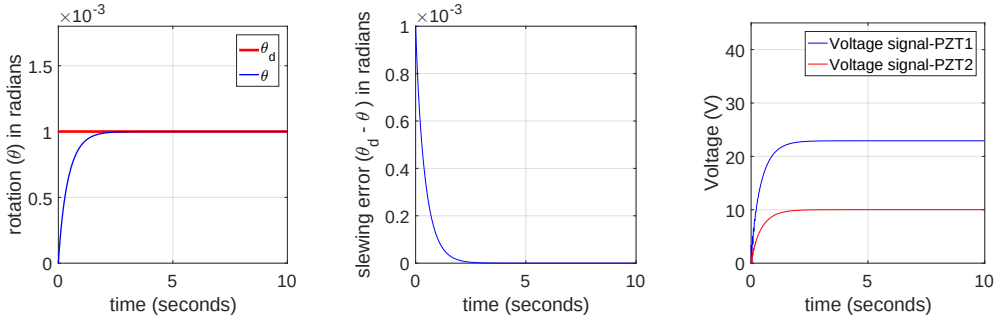
(a) Trajectory tracking (b) Tracking error (c) Control effort

Figure A.12: Trajectory tracking (simulation) for $\delta = 0$ in configuration 2 with gains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$.



(a) Slewing (b) Slewing error (c) Control effort

Figure A.13: Slewing (simulation) for $\delta = 0.5$ in configuration 2 with gains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$.



(a) Slewing (b) Slewing error (c) Control effort

Figure A.14: Slewing (simulation) for $\delta = 0$ in configuration 2 with hains $\lambda_\theta = 3$, $K_\theta = 0.5$, $K_\xi = 0.5$.

A.5 Experiments

A.5.1 Actuator Model Validation

The quasi-static actuator model in (A.11) is validated using a cantilever beam setup (see Fig. A.15). The system parameters of the setup are given are $\ell_{ba} = 24.4 \times 10^{-2}$ m, $\ell_{1pa} = 0.9 \times 10^{-2}$ m, $\ell_{pa} = 7.244 \times 10^{-2}$ m. The beam and PZT density are listed in Table A.1.

(a) Beam test setup

(b) Beam schematic

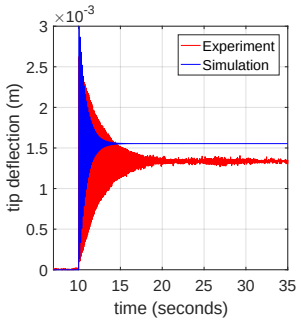
Figure A.15: Open-loop beam experimental setup.

The cantilever beam dynamics in PDE form is given in (A.68) (see [177]). The Galerkin method was used to discretize the PDE equation to form the ODE in (A.69).

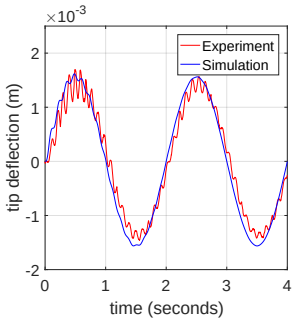
$$m_R\ddot{\xi} + \left(EI\xi'' + \mu EI\dot{\xi}''\right)'' - M_b''(x,t) = 0 \tag{A.68}$$

$$\left(\int_0^{\ell_{ab}} m_R\phi\phi^T dx\right)\ddot{\eta} + \left(\int_0^{\ell_{ab}} \phi\left(EI\phi''^T\right)'' dx\right)(\eta + \mu\dot{\eta}) - \int_0^{\ell_{ab}} \phi M_b''(x,t)dx = 0 \tag{A.69}$$

In Figs. A.16, we compare the open-loop response of the beam computed from simulations against results measured using the Vicon motion capture system during experiments at the tip of the beam.



(a) Response to step input



(b) Response to a sinusoidal input

Figure A.16: Open-loop experiment vs simulation, beam tip deflection.

For a sinusoidal input, the frequency of the simulated response matches experimental results. The amplitude of the response is not an exact match because 1) we assume there is perfect bonding between the beam and PZT, 2) the beam is fixed at the root using a 'C-clamp', which is not an ideal cantilever beam, and 3) the Vicon marker effect on the beam is not modeled. We proceed and use this model in the real time experiments as the difference between the simulation and experiments can be compensated with additional control effort (it can be considered as a bounded uncertainty at the input of the system (A.25)).

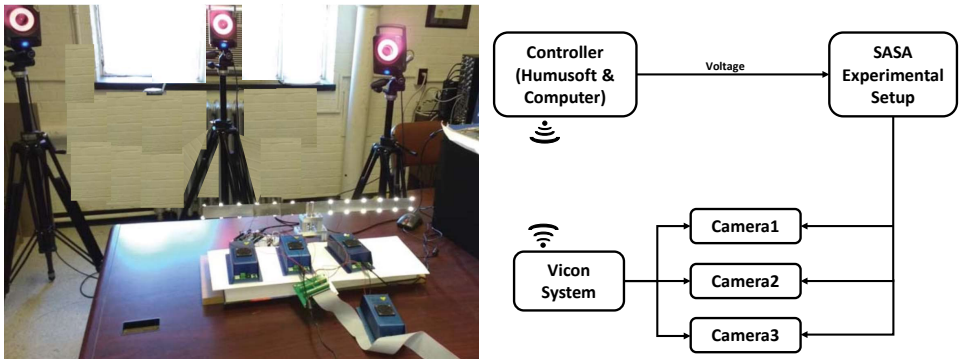
A.5.2 SASA Experimental Setup

The experimental setup is shown in Fig. A.17, which includes SASA setup, Vicon system, amplifiers, and the controller. The top and side view of the SASA setup are shown in Fig. A.18. The setup has a cylinder, a beam, and two plates. The properties of the beam and PZT used are given in Table A.1. The cylinder has a slot machined using wire cut Electrical Discharge Machining (EDM), through which beam is fitted to the cylinder. The ball bearings are secured to the plates using an interference fit and the cylinder is press fit into the inner bearing race. Two ball bearings are used to resist torques perpendicular to the axis of the cylinder, and to allow rotation about the cylinder axis. Here we used a T120-A4E-602 PZT sheet manufactured by Piezo Systems, Inc. with a maximum input voltage of ± 200 V.

Two PZTs are bonded on each side of the beam using superglue[1] (the above cantilever beam experiments used this bonding agent as well), for asymmetric actuation. A voltage signal is supplied to the PZT amplifiers using a Humusoft MF624 controller, which can produce up to ± 10 V. We use linear amplifiers manufactured by Piezo systems to amplify the voltage supplied by the controller before applying it across the PZTs. The real-time code for experiment was setup in Matlab using Simulink® Desktop real-time™ toolbox. For computation of derivatives and integrals we used the discrete time function blocks[2] available in Simulink®.

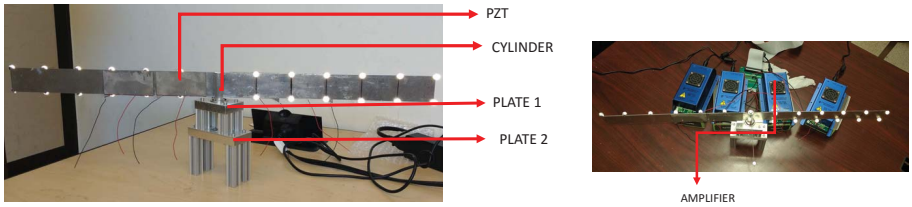
[1] Loctite® Super Glue

[2] http://www.mathworks.com/help/simulink/discrete.html



(a) SASA experimental setup (b) Schematic of SASA test setup

Figure A.17: SASA setup



(a) Side view (b) Top view with amplifiers

Figure A.18: SASA setup views

The properties in Table A.1 are estimated using physical properties of the aluminum beam, cylinder, and steel ball bearings. The experimental system is built to be symmetric so that measurements from a single beam can be used to compute control effort during the real-time experiments. To use the dynamics and control law derived, we transform the Vicon system world coordinate system, e_{a_1} and e_{a_2}, to an inertial coordinate system fixed to the center of the cylinder, e_{c_1} and e_{c_2}. Note that in Figs. A.3, A.17a, and A.19, there are 12 Vicon markers at 6 locations on each beam. The x and y coordinates of these Vicon markers are measured in e_{a_1} and e_{a_2}, and then transformed to coordinate system e_{c_1} and e_{c_2} to compute deflection at these locations. The deflection at the 6 Vicon marker locations is computed by solving an inverse kinematic problem given in (A.70), where θ is measured using the Vicon marker as shown in Fig. A.19. The flowchart in A.20 shows the flow of the real-time implementation. As seen in Fig. A.20, we use three Vicon cameras to measure the beam deflection and cylinder rotation. The Vicon system communicates with the controller (i.e., a desktop computer with the Humusoft controller) using a local wireless area network.

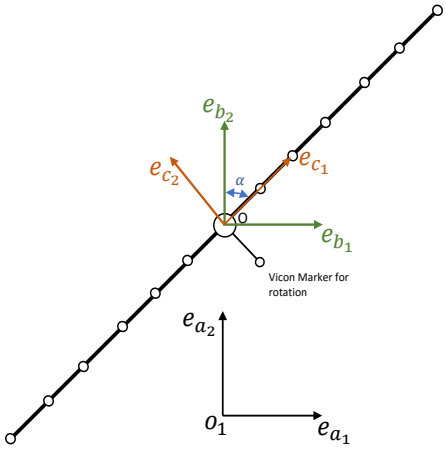


Figure A.19: Top view of the setup showing coordinate systems used in experiments.

$$\begin{bmatrix} r+x \\ \xi \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{A.70}$$

Let the deflection measured at the 6 Vicon marker locations (with distances from root $[\ell_{v_1}\ \ell_{v_2}\ \ell_{v_3}\ \ell_{v_4}\ \ell_{v_5}\ \ell_{v_6}]$) using the Vicon system be $\xi_{exp} = [\xi_1\ \xi_2\ \xi_3\ \xi_4\ \xi_5\ \xi_6]$. The measurements ξ_{exp} are used to compute η for the four Galerkin functions in (A.71). The derivatives $\dot{\theta}$ and $\dot{\eta}$ are estimated using discrete derivatives of the computed θ and η values from Vicon measurements.

$$\xi_{exp}^T = \begin{bmatrix} \phi_1(\ell_{v_1}) & \phi_2(\ell_{v_1}) & \phi_3(\ell_{v_1}) & \phi_4(\ell_{v_1}) \\ \phi_1(\ell_{v_2}) & \phi_2(\ell_{v_2}) & \phi_3(\ell_{v_2}) & \phi_4(\ell_{v_2}) \\ \phi_1(\ell_{v_3}) & \phi_2(\ell_{v_3}) & \phi_3(\ell_{v_3}) & \phi_4(\ell_{v_3}) \\ \phi_1(\ell_{v_4}) & \phi_2(\ell_{v_4}) & \phi_3(\ell_{v_4}) & \phi_4(\ell_{v_4}) \\ \phi_1(\ell_{v_5}) & \phi_2(\ell_{v_5}) & \phi_3(\ell_{v_5}) & \phi_4(\ell_{v_5}) \\ \phi_1(\ell_{v_6}) & \phi_2(\ell_{v_6}) & \phi_3(\ell_{v_6}) & \phi_4(\ell_{v_6}) \end{bmatrix} \eta \tag{A.71}$$

A.5.3 SASA Open-loop Simulation versus Experiments

For the open-loop results, we measure the bus rotation due to an applied voltage across 'PZT RIGHT 1' and 'PZT LEFT 1' on the beam. The estimation errors are seen clearly in Fig. A.21, which illustrates a comparison of open-loop simulation and experimental results. The response to a sinusoidal input is a good match. For the step input, simulations predict larger rotation. This difference is due to the estimation of system parameters and not modeling the ball bearing friction. In spite

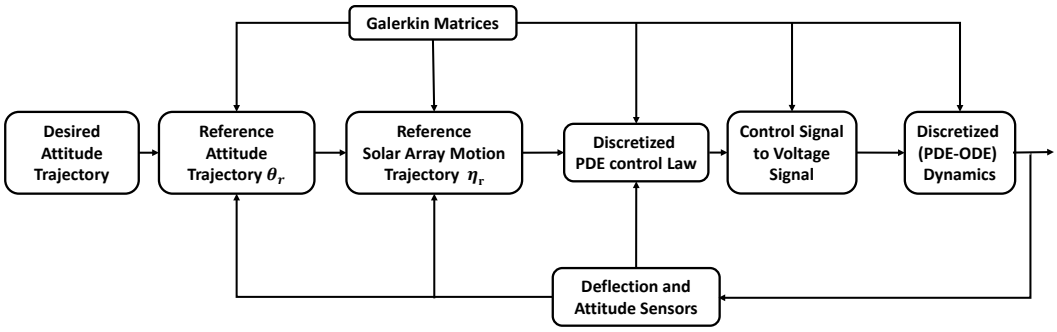
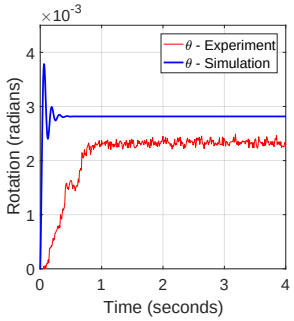
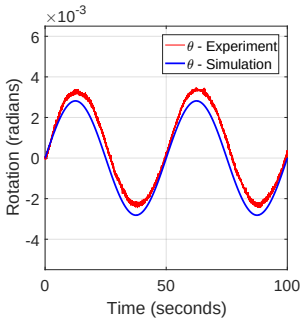


Figure A.20: Closed-loop ODE system as implemented in Simulink for experiments.

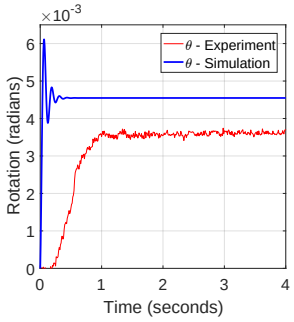
of the modeling errors, the closed-loop system achieves the control objectives (see Section. A.5.4 for details).



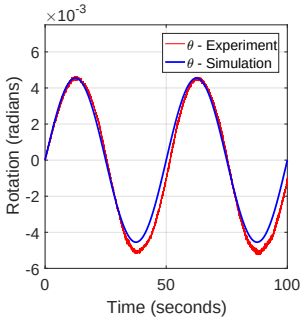
(a) Step response for an input of 100V



(c) Sinusoidal response for an input of amplitude 100V.



(b) Step response for an input of 160V



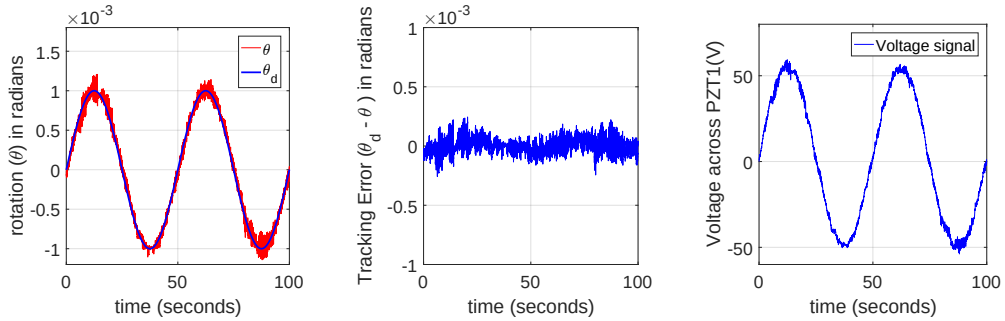
(d) Sinusoidal response for an input of amplitude 160V

Figure A.21: Open-loop experiment versus simulation, bus rotation.

A.5.4 Control Experiment Results

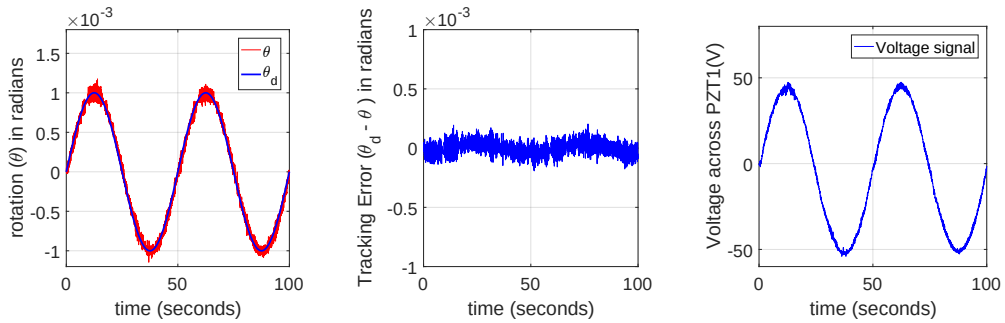
The experimental setup described above is used to test the control law. Based on the simulation results, we test the controller in configuration 1 for $\delta = \{1, 0.5, 0\}$

(Figs. A.22–A.29) and in configuration 2 for $\delta = \{0.5, 0\}$ (Figs. A.28–A.31). Similar to the simulation, we track a sinusoidal signal of amplitude 0.001 radians, with a frequency of 0.02 Hz, and slew to 0.001 radians from a zero initial angle. As mentioned earlier, the real-time code was implemented using Simulink Desktop real-time toolbox. For all the δ values listed earlier we achieve the control objectives. The measurements are made and control signal is applied to PZTs at the same frequency (100 Hz) during the experiment. During the closed-loop experiments, it was observed that the setup has a fundamental frequency around 50-60 Hz. In configuration 2, the PZTs on each side become more active as the δ value is reduced toward zero. Note that the controllers work notwithstanding the modeling errors and parametric uncertainties. The control effort computed by the control algorithm during experiments is at least 20 Volts more than predicted during simulations.



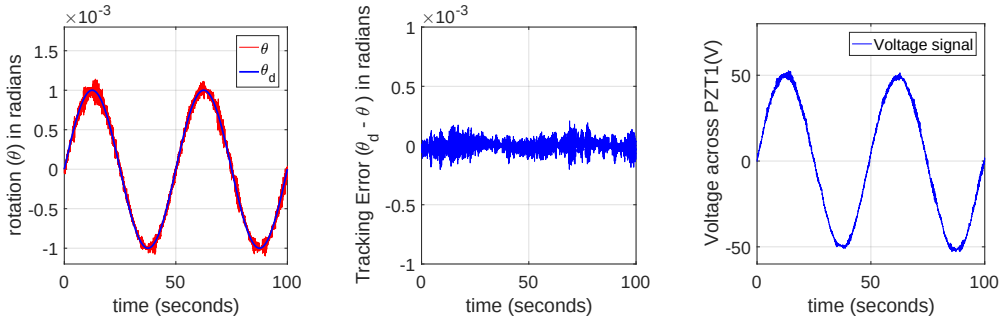
(a) Trajectory tracking (b) Tracking error (c) Control effort

Figure A.22: Trajectory tracking (experiment) for $\delta = 1$ with gains $\lambda_\theta = 1.5$, $K_\theta = 0.5$, $K_\xi = 0.5$.



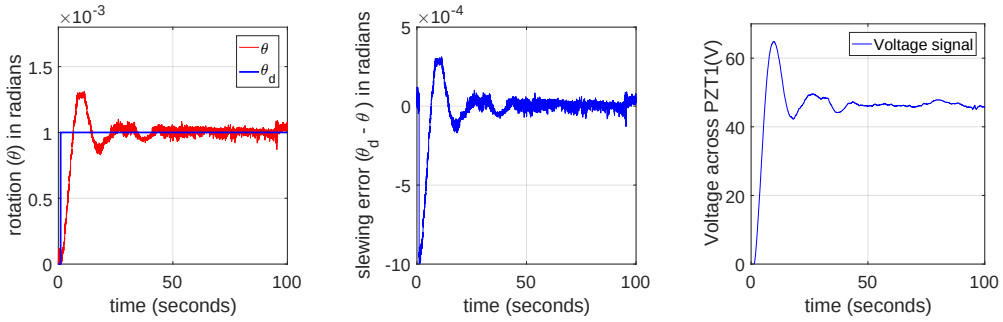
(a) Trajectory tracking (b) Tracking error (c) Control effort

Figure A.23: Trajectory tracking (experiment) for $\delta = 0.5$ with gains $\lambda_\theta = 1.5$, $K_\theta = 0.5$, $K_\xi = 0.5$.



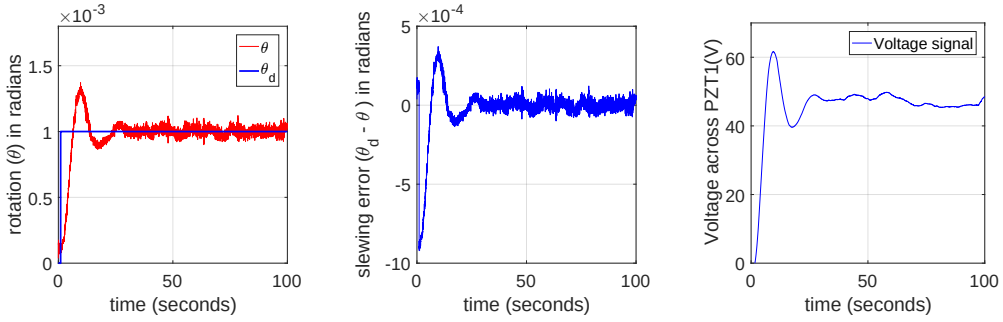
(a) Trajectory tracking (b) Tracking error (c) Control effort

Figure A.24: Trajectory tracking (experiment) for $\delta = 0$ with gains $\lambda_\theta = 2$, $K_\theta = 0.5$, $K_\xi = 0.5$.



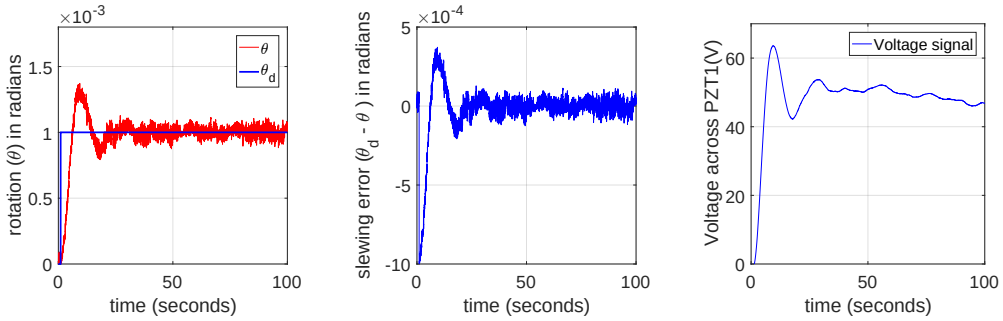
(a) Slewing (b) Slewing error (c) Control effort

Figure A.25: Slewing (experiment) for $\delta = 1$ with gains $\lambda_\theta = 1$, $K_\theta = 0.25$, $K_\xi = 0.5$.



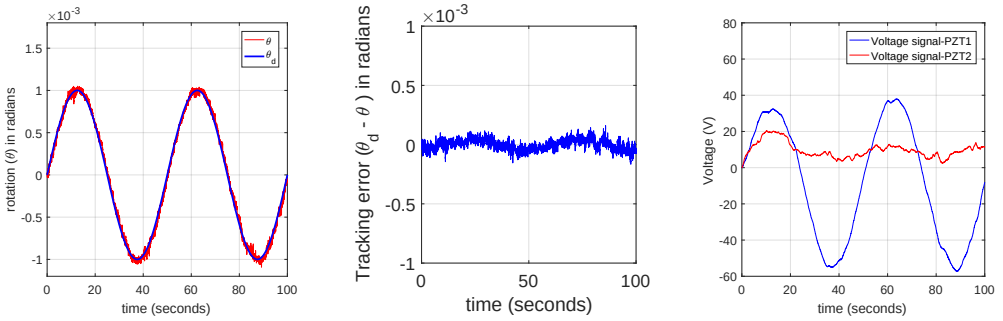
(a) Slewing (b) Slewing error (c) Control effort

Figure A.26: Slewing (experiment) for $\delta = 0.5$ with gains $\lambda_\theta = 1$, $K_\theta = 0.25$, $K_\xi = 0.5$.



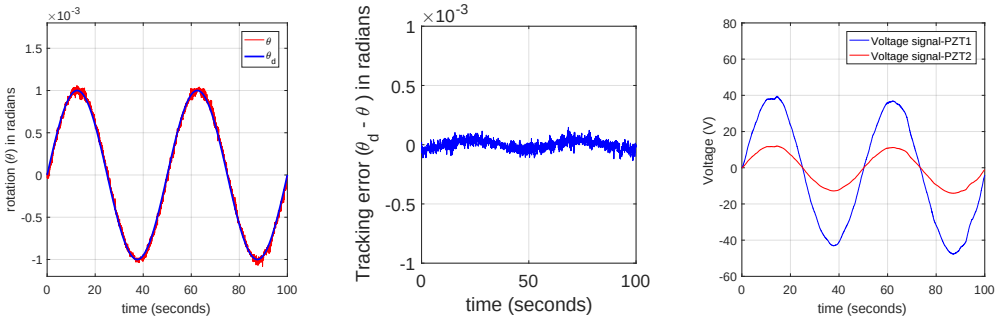
(a) Slewing (b) Slewing error (c) Control effort

Figure A.27: Slewing (experiment) for $\delta = 0$ with gains $\lambda_\theta = 1$, $K_\theta = 0.25$, $K_\xi = 0.5$.



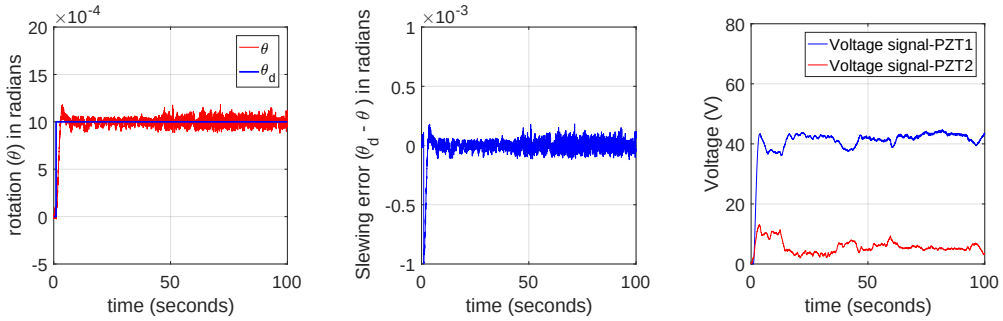
(a) Trajectory tracking (b) Tracking error (c) Control effort

Figure A.28: Trajectory tracking (experiment) for $\delta = 0.5$ in configuration 2 with gains $\lambda_\theta = 2$, $K_\theta = 0.5$, $K_\xi = 0.5$.



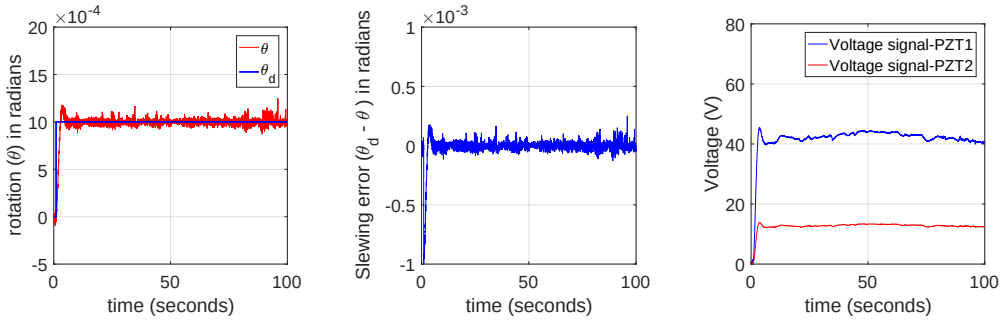
(a) Trajectory tracking (b) Tracking error (c) Control effort

Figure A.29: Trajectory tracking (experiment) for $\delta = 0$ in configuration 2 with gains $\lambda_\theta = 2$, $K_\theta = 0.5$, $K_\xi = 0.5$.



(a) Slewing (b) Slewing error (c) Control effort

Figure A.30: Slewing (experiment) for $\delta = 0.5$ in configuration 2 with gains $\lambda_\theta = 2$, $K_\theta = 0.5$, $K_\xi = 0.5$.



(a) Slewing (b) Slewing error (c) Control effort

Figure A.31: Slewing (experiment) for $\delta = 0$ in configuration 2 with gains $\lambda_\theta = 2$, $K_\theta = 0.5$, $K_\xi = 0.5$.

A.6 Conclusion

In this paper, we derived nonlinear equations of motion that describe the one-degree-of-freedom rotation of a spacecraft with Strain-Actuated Solar Arrays (SASA), and presented a novel control technology with a new Ordinary Differential Equations (ODE)-Partial Differential Equation model (PDE) control algorithm. The equations of motion form a nonlinear ODE-PDE system, with ODE describing bus rotation and PDE describing solar array flexible dynamics. The attitude control objective was achieved using the inertial coupling between the cylinder and the flexible appendage. The control law computes the beam dynamics required to achieve the desired attitude and actuates the flexible beam using the distributed actuation using an actuator to control mapping. The attitude tracking control law is designed in terms of a variable δ that scales with the stiffness term to simulate an uncanceled or partially canceled beam dynamics. The exponential stability of the closed-loop

system for $\delta = 1$ corresponding to full beam dynamics cancellation with a feed-forward term was proven. The closed-loop system was simulated for different δ values to verify trajectory tracking and slewing. To test the control algorithms in real-time, we developed an experimental setup. The setup includes a cylinder and a beam connected using an interference fit. Strain actuation in the beam was achieved using piezoelectric actuators (PZT). The quasi-static actuator model used for SASA experiments was validated by comparing tip deflections of a cantilever beam obtained from simulations against experimental measurements. We achieved controlled cylinder rotation using beam strain-actuation on the experimental setup, which validates the SASA control algorithm. Strain-actuation was extended to a distributed actuation configuration with multiple PZTs. This model was used to test the controllers in a distributed configuration for trajectory tracking and slewing.