Advances in Constrained Spacecraft Relative Motion Planning

by

Gregory R. Frey

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2018

Doctoral Committee:

Associate Professor Anouck R. Girard, Co-Chair
Professor Ilya V. Kolmanovsky, Co-Chair
Doctor Frederick A. Leve, Air Force Office of Scientific Research
Doctor Christopher D. Petersen, Air Force Research Laboratory
Professor Aaron J. Ridley

Gregory R. Frey

gfrey@umich.edu

ORCID iD: 0000-0001-9410-9514

Dedicated to my family

ACKNOWLEDGMENTS

I would like to thank Col Martin France and the other members of the Department of Astronautics at the U.S. Air Force Academy for introducing me to academia and giving me the opportunity to further my education. I'm looking forward to rejoining you at USAFA in a few years.

In my time here at the University of Michigan, there have been many people who have provided guidance and support for which I am very grateful. Firstly, I would like to thank my advisors Professor Ilya Kolmanovsky and Professor Anouck Girard. Thank you for your willingness to spend extra time helping me to complete my studies within my compressed timeline. Needless to say, without your guidance, I would not have gotten far. To Professor Kolmanovsky, thank you for your patience and willingness to explain concepts and ideas to me multiple times whenever necessary. Your suggestions and insights have been invaluable in helping me to make steady progress. To Professor Girard, thank you for your attention to detail and for keeping me focused on important milestones and goals required for completion. Your guidance and support have greatly increased the quality of my work.

Secondly, I would like to thank the other members of my committee: Dr. Chris Petersen, Dr. Fred Leve and Professor Aaron Ridley. To Chris Petersen, thank you for helping me get started doing real research in my first year, and later for continuing to provide much needed guidance and support from AFRL. To Fred Leve, thank you for your consistent involvement in my work and for your many tips and suggestions that have helped my research to be more relevant to real-world problems. To Professor Ridley, thank you for taking time out of your busy schedule to review my work.

To my fellow students, especially Will Dunham, Dom Liao-McPherson, Richard Sutherland and A.J. Berning, thank you for your friendship and all of the help you've given me. From collaborating on course homework and working together to get through prelims in the first year to the many conversations and helpful suggestions about research, going through this process with all of you has been great.

Finally, I would like to thank my family. To my parents, Pat and Roger Frey, thank you for your confidence in me and your support from the beginning. To Angie, Ethan, Leah and Megan, thanks for everything. Having you with me makes every day better.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

CW Clohessy-Wiltshire

CGs Command Governors

CMGs Control Moment Gyroscopes

DARPA Defense Advanced Research Projects Agency

FF Formation Flying

ISS International Space Station

KKT Karush-Kuhn-Tucker

LHS Left Hand Side

LQR Linear Quadratic Regulator

LTI Linear Time Invariant

LG Local Gradient

MPC Model Predictive Control

MILP Mixed Integer Linear Programming

OOS On-Orbit Servicing

PG Parameter Governor

QCQP Quadratically Constrained Quadratic Program

RRT Rapidly exploring Random Trees

RWAs Reaction Wheel Assemblies

RGs Reference Governors

R2BEOM Restricted Two-Body Equations of Motion

RHS Right Hand Side

SPI Safe, Positively Invariant

SSG Scale Shift Governor

SE-SCP Spherical Expansion-Sequential Convex Programming

SRMP Spacecraft Relative Motion Planning

bSRMP Basic Spacecraft Relative Motion Planning

TSG Time Shift Governor

TH Tschauner-Hempel

UAVs Unmanned Aerial Vehicles

VCP Vertex Coloring Problem

mVCP Minimal Vertex Coloring Problem

ABSTRACT

This dissertation considers Spacecraft Relative Motion Planning (SRMP), where maneuvers are planned for one or more spacecraft to execute in close proximity to obstacles or to each other. The need for this type of maneuver planning has grown in recent years as the space environment becomes more cluttered, and the focus on space situational awareness increases. In SRMP, maneuvers must accommodate non-linear and non-convex constraints, be robust to disturbances, and be implementable on-board spacecraft with limited computational capabilities. Consequently, many standard optimization or path planning techniques cannot be directly applied to SRMP. In this dissertation, three novel SRMP techniques are developed and simulations are presented to illustrate the implementation of each method.

Firstly, an invariance-based SRMP technique is proposed. Maneuvers are planned to transition a spacecraft between specified natural motion trajectories, which require no control to follow, while avoiding obstacles and accommodating minimum and maximum actuation limits. The method is based on a graph search applied to a "virtual net" with nodes corresponding to natural motion trajectories. Adjacency rules in the virtual net are based on safe positively invariant tubes built around each natural motion trajectory. These rules guarantee safe transitions between adjacent natural motion trajectories, even when set-bounded disturbances are present. Procedures to construct the safe positively invariant tubes and the virtual net are developed. Methods to reduce calculations are proposed and shown to significantly reduce computation time, with tradeoffs related to maneuver planning flexibility.

Secondly, a SRMP technique is developed for the specific problem of satellite inspection. In this setting, an inspector spacecraft maneuvers to gather information about a target

spacecraft. An information collection model is developed and used to construct a rapidly computable analytical control law based on the local gradient of the information rate. This control law drives the inspector spacecraft on a path along which the rate of information collection is strictly increasing. To ensure constraint satisfaction, the local gradient control law is combined with a state feedback control law, and rules are developed to govern switches between the two controllers. The method is shown to be effective in generating trajectories to gather information about a specified target point while accommodating disturbances.

Finally, a control strategy is proposed to generate a formation containing an arbitrary number of vehicles. This strategy is based on an add-on predictive control mechanism known as a parameter governor. Parameter governors work by modifying parameters, such as gains or offsets, in a nominal closed-loop system to enforce constraints and improve performance. The parameter governor is first developed in a general setting, using generic non-linear system dynamics and an arbitrary formation design. Required calculations are minimized, and non-convex constraints are accommodated through use of a parameter update strategy based on graph colorability theory, and by limiting parameter values to a discrete set. A convergence analysis is presented, proving that under reasonable assumptions, the parameter governor is guaranteed to generate the desired formation. Two specific parameter governors, referred to as the Scale Shift Governor and Time Shift Governor, are proposed and applied to generate formations of spacecraft. These parameter governors enforce constraints by modifying either scale- or time-shifts applied to the target trajectory provided to each spacecraft in formation. Simulation case studies show the effectiveness of each method and demonstrate robustness to disturbances.

CHAPTER 1

Introduction

Spacecraft Relative Motion Planning (SRMP) is the design and execution of maneuvers relative to a nominal target. Depending on the mission application, this target could be another spacecraft, a nominal position along a trajectory, or the center-point of a formation. In this dissertation, novel methods are developed and demonstrated for two specific cases of SRMP.

The first case, referred to as Basic Spacecraft Relative Motion Planning (bSRMP), involves maneuvers planned for a single spacecraft to execute in close proximity to another non-maneuvering spacecraft, and possibly to additional obstacles, e.g., orbital debris. Typical mission applications for bSRMP include satellite inspection and rendezvous and docking. In satellite inspection applications, one spacecraft executes maneuvers to gather information about a non-maneuvering spacecraft, whereas in rendezvous and docking applications, one spacecraft executes maneuvers to approach and dock with another spacecraft. The second specific case of SRMP considered in this dissertation, referred to as Formation Flying (FF), involves maneuvers planned for multiple spacecraft to execute simultaneously to generate and maintain a specified formation, i.e., track and maintain desired separations and positions among the spacecraft. Typical mission applications for FF include interferometry and gravity mapping. In interferometry applications, data from sensors aboard multiple spacecraft arrayed in a precise formation are combined to provide increased data quality, e.g., better image resolution. In gravity mapping applications, measurements of the relative position between spacecraft over time are used to obtain detailed estimates of abnormalities in Earth's gravity field. In this dissertation, SRMP is used as a general term, referring to both bSRMP and FF.

In this chapter, a brief historical background of SRMP is presented, the specific SRMP problem addressed in this dissertation is defined, a brief survey of related literature is discussed, and the specific contributions of this work are stated.

1.1 Historical Background

bSRMP has been an important area of space operations and research since NASA established the Gemini program in the 1960s with a goal of developing techniques for rendezvous and docking with another space vehicle [1]. The Gemini program's success in demonstrating techniques for rendezvous and docking led to further development and use of these procedures in the Apollo program. Since the final moon landing in 1972, bSRMP has remained a key component of manned space missions including the on-orbit satellite repair and retrieval missions executed by the Space Shuttles [2], and crew change missions to space stations including Skylab, Mir, Tiangong-1, Tiangong-2 and the International Space Station (ISS).

bSRMP has become increasingly important for unmanned space missions as well. High-profile examples of unmanned missions utilizing bSRMP techniques include those performed by SpaceX's Dragon spacecraft and Orbital ATK's Cygnus spacecraft to rendezvous with the ISS. In addition to these ISS resupply missions, several other unmanned missions utilizing bSRMP have been flown to develop and demonstrate capabilities in two mission areas that are becoming increasingly important in today's space environment: Space Situational Awareness (SSA) and On-Orbit Servicing (OOS).

Developing and maintaining SSA, i.e., knowledge of spacecraft or other objects located near a spacecraft of interest, is desirable because this knowledge can be used to ensure safe operation of expensive satellites performing vital missions. The bSRMP portion of SSA missions involves development of maneuvers to detect and characterize objects operating near a spacecraft. A recent example of this type of mission is the Air Force Research Laboratory (AFRL) Automated Navigation and Guidance Experiment for Local Space (ANGELS) program which developed and demonstrated bSRMP techniques to improve methods to detect, track and characterize objects in a local space environment [3].

Developing OOS capabilities is desirable because these capabilities may provide the ability to extend the lifetime of spacecraft, reduce the cost of space operations, and/or remove dead or malfunctioning satellites from orbit. The bSRMP portion of OOS missions necessarily includes rendezvous and docking, but these maneuvers are typically preceded by an inspection phase that involves circumnavigations of the target satellite and/or additional maneuvers to provide high-resolution imagery of specific areas on the target spacecraft in order to diagnose malfunctions or develop repair plans.

Several missions have been flown in recent years to develop bSRMP methods for use in OOS applications. In 1997, the Engineering Test Satellite-VII (ETS-VII) satellite operated by the National Space Development Agency (NASDA) of Japan successfully performed the

first autonomous rendezvous and docking between unmanned spacecraft, in part to develop technologies for use in servicing spacecraft [4]. In a similar vein, the Orbital Express program executed by Defense Advanced Research Projects Agency (DARPA) demonstrated several bSRMP techniques including autonomous rendezvous with, and docking and close-range circumnavigation of, a non-maneuvering spacecraft [5]. Other programs initiated to develop capabilities for OOS and bSRMP include the joint Technology Satellite for demonstration and verification of Space systems (TECSAS) program involving Germany, Canada and Russia [6], Germany's subsequent Deutsche Orbitale Servicing Mission (DEOS) program [7], and DARPA's sequence of programs related to OOS including Spacecraft for the Universal Modification of Orbits / Front-end Robotics Enabling Near-Term Demonstration (SUMO/FREND) [8], Phoenix [9], F6 [10], and the present-day Remote Servicing of Geosynchronous Vehicles (RSGS) [11].

While bSRMP is useful for missions involving rendezvous and docking, SSA and OOS, the utility of FF stems from several advantages offered by multi-spacecraft mission architectures compared to single-spacecraft architectures. Firstly, by utilizing sensors onboard multiple spacecraft, FF allows for distributed and variable aperture sensing which can be a useful capability for remote sensing mission applications. Secondly, by distributing sensors and mission capabilities amongst spacecraft, FF missions can provide enhanced mission survivability as the failure of a single spacecraft may not cause complete mission failure. Finally, in certain cases, FF missions can yield cost savings as launching multiple small spacecraft can be less expensive than launching a single, larger vehicle. Because of these advantages, NASA, the U.S. Department of Defense, the ESA, and other agencies have shown interest in developing FF missions [12].

Examples of past missions involving FF include NASA's Gravity Recovery and Climate Experiment (GRACE) mission [13], which flew two spacecraft in close proximity to map Earth's gravity field, and the German TerraSAR-X/TanDEM-X (TSX/TDX) mission that utilized two spacecraft to form a spaceborne radar interferometer [14]. Missions involving more than two spacecraft include NASA's Time History of Events and Macroscale Interactions during Substorms (THEMIS) mission [15], which utilized five spacecraft to study the causes of substorms in Earth's magnetosphere, and the Aerospace Corporation's Aero-4 mission that utilized four spacecraft to study the use of differential drag for cubesat FF [16]. In addition to these historical examples, there are several FF flight missions currently in the design and planning stages, see, e.g., [17].

1.2 Motivation and Problem Statement

As the previous discussion illustrates, SRMP techniques have been used in space operations for several decades. In the future, the use of SRMP techniques will likely increase as the increasingly cluttered space environment leads to spacecraft more frequently operating near other objects, and as the focus on missions to provide capabilities such as SSA and OOS grows. To meet the requirements for these future missions, new and improved SRMP techniques are needed due to five complicating factors, some of which can impede the straightforward application of legacy SRMP techniques or motion planning methods used in other fields, e.g., robotics [18].

Firstly, because of launch costs and the lack of on-orbit refueling capabilities, fuel efficiency is an important consideration for SRMP. Secondly, SRMP problems typically include constraints on control variables that must be enforced. Typical control variable constraints include thruster saturation limits and minimum impulse bit limits. These constraints place upper- and lower-bounds on the control inputs available for maneuver planning, respectively. Thirdly, SRMP problems frequently involve state variable constraints based either on exclusion zones centered on obstacles, or on minimum separation distance limits between spacecraft. These constraints must be enforced to ensure no collisions with obstacles, e.g., orbital debris, or between spacecraft in formation.

Considering only these three factors, it is possible to pose SRMP problems as constrained optimization problems. However, because of the types of constraints considered, this optimization problem is typically non-linear and non-convex. While several methods and solvers exist to handle optimization problems with these types of constraints, e.g., the Gauss Pseudospectral Optimization Solver-II (GPOPS-II) [19], these methods can require long solution times, and require a user-input initial guess that may be difficult to generate. Additionally, both convergence and the quality of the solution is typically dependent on the initial guess [20].

A fourth factor that complicates the development of SRMP techniques is the presence of perturbations due to inexact dynamics modeling and control system performance. Because of these perturbations, planned maneuvers must not only satisfy constraints and accomplish mission objectives, they must also be robust to disturbances. This limits the utility of many optimization and motion planning methods that calculate maneuvers to be executed open-loop.

Finally, spacecraft typically have limited on-board computational capabilities. Because many future SRMP-based mission concepts call for autonomous on-board implementation, e.g., many of the FF missions discussed in [21], this limitation leads to the requirement that

SRMP methods be computationally fast and straightforward.

Motivated by these five complicating factors, the general problem considered in this dissertation is given as follows.

Problem Statement: Develop methods for SRMP that:

- Plan fuel efficient trajectories,
- Account for control constraints,
- Ensure no collisions with obstacles,
- Are robust to disturbances, and
- Use fast and straightforward calculations.

In the following sections, a review of literature related to this problem is presented, and the specific contributions of the work in this dissertation are stated.

1.3 Literature Review

Because SRMP has been an active area of research for decades, there is an extensive body of literature on the subject. Books by Alfriend, et al., [22], Wang et al., [23] and Fehse [24], provide overviews of dynamics models, perturbations, and maneuver design methods related to SRMP. Additionally, there are several relevant literature surveys that provide overviews of SRMP-related research over the years. For example, surveys on trajectory planning for rendezvous can be found in references [20] and [25], while a survey on the broader field of numerical trajectory optimization was given by Betts [26]. A survey and assessment of the many spacecraft relative motion dynamics models used in SRMP was carried out by Sullivan et al., [27] and reviews of the guidance and control methods developed for FF are presented in references [12], [28] and [29]. In Sections 1.3.1-1.3.3 below, subsets of the SRMP-related literature most relevant to the work in this dissertation are summarized, and areas for improvement are highlighted.

1.3.1 Optimization Techniques

As discussed in Section 1.2, one of the main factors complicating SRMP is the presence of obstacle avoidance requirements, resulting in non-linear and non-convex constraints.

Several methods have been proposed to solve problems including these types of constraints using optimization-based techniques.

Richards and How proposed a SRMP method based on Mixed Integer Linear Programming (MILP) [30]. In this method, obstacles are represented as polytopes, and linear constraints that include binary variables are appended to a fuel-optimizing linear program. The binary variables are used to activate and deactivate constraints as needed to ensure obstacle avoidance. The resulting MILP can then be solved using standard solvers, e.g., the CPLEX method [31]. Liu and Lu proposed a method to replace a single non-convex optimization problem with a sequence of convex optimization problems that can be efficiently solved with standard algorithms [32, 33]. In the original non-convex problem, obstacle avoidance constraints are written as concave state inequality constraints. These constraints are then linearized in order to make the problem convex. In [32], it is shown that the solution to the converged sequence of convex problems is equivalent to the solution of the original problem. The method is demonstrated through simulation with a bSRMP example of rendezvous.

In a FF setting, several optimization-based methods have been proposed to generate or reconfigure spacecraft formations while ensuring no collisions between spacecraft. Cetin et al. proposed a method in which a MILP is formulated and then solved via a sequence of linear programs [34]. Kim et al. developed a method to reconfigure a two-spacecraft formation using state-constrained optimal control techniques [35]. Other optimal control-based strategies for FF are described in Chapter 5 of reference [23].

While some of these methods can be used to recast non-convex problems into more readily solvable forms, they still require the solution to one or more optimization problems to obtain a single trajectory. This leads to computations that may not be fast or straightforward. Additionally, in many cases, the methods described above may not scale well to accommodate multiple obstacles or formations with a large number of spacecraft.

1.3.2 Sampling- and Graph Search-based Motion Planning

SRMP using sampling-based algorithms, such as Rapidly exploring Random Trees (RRT) [36], and graph search algorithms, such as Dijkstra's algorithm [37], is attractive because these algorithms can be simple and efficient. In sampling-based motion planning algorithms, a path from an initial state to a goal state is built using a graph, with nodes (vertices) corresponding to points in the state-space, generated by randomly selected samples in the obstacle-free space [38]. Nodes are added to the graph if it is possible to execute a collision-free trajectory from a previous node to the node currently being considered. Ma-

neuvers planned using sampling-based algorithms can sometimes be executed closed-loop. This is a desirable feature for SRMP applications because executing maneuvers closed-loop allows for the rejection of disturbances encountered throughout the maneuver.

While sampling-based motion planning algorithms have been applied in FF settings, e.g., [39], applications to bSRMP are most relevant to the work in this dissertation. RRT-based algorithms have been applied to bSRMP and demonstrated the ability to generate collision free trajectories for spacecraft maneuvering in cluttered environments [40,41]. Other sampling-based algorithms developed for or applied to bSRMP include the Fast Marching Trees (FMT*) algorithm, which was modified and applied by Starek et al. to develop safe trajectories for satellite rendezvous [42–44], and the Spherical Expansion-Sequential Convex Programming (SE-SCP) algorithm proposed by Baldini et al. [45]. This algorithm includes a spherical expansion-based sampling step to find a feasible path between start and goal positions that avoids obstacles. Then, this feasible path is "locally" optimized using SCP. Other related work, not specifically developed for SRMP, includes an LQR-Trees algorithm developed to exploit a set of trajectories, calculated using trajectory optimization algorithms, and stabilized using time-varying Linear Quadratic Regulator (LQR) controllers [46]. Finally, Singh et al. proposed a motion planning method involving invariant "funnels" formed around a set of open-loop maneuvers [47]. These funnels are used to piece together individual maneuvers to obtain an overall maneuver that avoids obstacles.

One drawback to using sampling-based motion planning techniques is that a separate graph/tree must generated for each new trajectory, even if the environment is unchanged. This leads to computations that may not be feasible to execute on-board a satellite. This drawback can be mitigated by using motion planning methods based on graph search.

In graph search based methods, all calculations required to build the entire graph are performed first. After the graph has been generated, shortest-path algorithms can be used to plan multiple trajectories using the same pre-computed graph at low computational cost (assuming the environment does not change). In certain cases, computations required to build the graph may be efficient and implementable on-board, while in other cases the graph may be generated off-line and uploaded to the spacecraft. In reference [48], a bSRMP method using graph-search was developed to plan paths that avoid obstacles while guaranteeing robustness to set-bounded disturbances. These paths were generated using Dijkstra's algorithm on a graph with nodes corresponding to forced equilibria. Adjacency in the graph was determined using safe (constraint admissible) positively invariant sets built around each node. By weighting the graph based on the control (or time) required to transfer between nodes, either fuel- or time-optimal paths (with respect to the graph) were planned. One drawback to this method is that, by using forced equilibria as nodes (which require fuel to

stay at), the resulting trajectories require constant control input to follow.

1.3.3 Predictive Control

In predictive control techniques, control inputs or other inputs to a nominal closed-loop system are selected based on predicted system behavior. One popular predictive control technique is Model Predictive Control (MPC) [49], in which a dynamics model is used to predict system behavior, and system inputs are selected based on the solution to an optimization problem formulated over a receding time horizon. Only inputs for the current time instant are applied, and subsequent inputs are determined by new solutions to the optimization problem at the next time steps. MPC is an attractive option for SRMP because it allows for handling multiple constraints, and can accommodate disturbances. Several MPC-based techniques have been proposed for both bSRMP applications, e.g., [50–52] and FF applications, e.g., [53–57]. While these references demonstrate that general MPC approaches are promising for SRMP, in general, if a) the dynamics model is non-linear, b) the optimization problem is non-convex, or c) the prediction horizon is large, then computations to obtain control inputs at each time-step can be cumbersome. Several MPC-based approaches have been developed to address these issues.

In an approach known as Parametrized MPC [58, 59], the set of admissible control values is reduced to a finite set through a parametrization defined by a set of parameters. Then, to determine control inputs at each time step, the objective function is minimized over the parameter set, rather than over the control inputs, thereby reducing the number of adjustable variables in the optimization. Such Parametrized MPC schemes are shown in [58] to have good performance for a variety of applications such as underactuated spacecraft attitude control and aircraft stabilization. Other MPC-based approaches developed to reduce calculations by reducing the number of adjustable variables include [60–62]. In [60], the control is parametrized using Laguerre functions. This reduces the number of variables needed to describe the control trajectory and therefore reduces the number of adjustable variables in the optimization problem. In [61], the control is first restricted to consist of a static-state feedback term plus an adjustable offset, and then this offset is parametrized within a subspace of reduced dimension using a Singular Value Decomposition. The method is shown to reduce computations while also ensuring recursive feasibility and robustness to bounded disturbances. Finally, in [62], a method to decompose a system (state and control vectors) in terms of basis vectors based on the system dynamics and cost function is proposed. This decomposition is used to formulate a reduced-order optimization problem to determine control inputs at each time step. The resulting controller is shown have performance

comparable to standard MPC approaches.

In addition to the MPC-based schemes mentioned above, another subset of predictive control most relevant to the work in this dissertation are Reference Governors (RGs) and Command Governors (CGs). RG and CG are *add-on* schemes used to enforce pointwise-in-time state and control constraints by modifying the reference command provided to well-designed (for small signals) closed-loop systems, when implementing the nominal reference command would lead to constraint violations [63]. The modified reference command is obtained by solving a receding horizon optimization problem at each time step. RGs and CGs enjoy two advantages over more general MPC controllers, related to the computations required. Firstly, because of their add-on nature, use of a RG or CG ensures that desirable properties of the inner-loop controller, possibly including simple calculations, are retained whenever constraints are satisfied. Secondly, because the RG or CG is restricted to making changes to only the reference provided to the inner-loop system, the optimization problem to be solved has a small number of adjustable variables and can be solved quickly using straightforward algorithms (as long as the optimization problem is convex). As a result of these two properties, the computations required to implement RG or CG control schemes are frequently fast and straightforward, and thus their use in SRMP settings is logical.

While RGs and CGs have been applied in bSRMP settings, e.g., [64–66], applications to distributed systems in general, and FF specifically, are most relevant to the work in this dissertation. Garone, Tedesco et al. have proposed several implementations of CG for distributed systems [67–71]. In reference [67], a feed-forward CG was proposed to control a multi-agent system with coupled constraints. In this approach, the frequency and magnitude of changes to the reference commands were restricted. These restrictions, along with assumed stability properties of the inner-loop controller, allowed each agent to make independent modifications to their reference commands, without explicit state measurements from other agents, while guaranteeing constraint satisfaction for the aggregate system. In [68], a sequential update strategy was proposed in which a single agent updated their reference command at each time step while all other agents held their commands fixed. In [69, 70], a new command update strategy was proposed where agents were grouped into "turns" defined based on coupling between the agents. At each time step, all agents within a specified turn update their commands while all other agents hold their commands fixed. These command update strategies allow for simpler and faster calculations compared to a centralized approach where all agents make updates simultaneously. Finally, in reference [71], a distributed CG is proposed that accounts for collision avoidance constraints by formulating and solving a MILP at each time-instant. These distributed CG approaches are promising in that they enable the distribution of computations over time and between

agents. However, with the exception of [71], they have demonstrated only limited ability to handle non-convex collision avoidance constraints important to FF applications. Additionally, all of the aforementioned distributed CG schemes are limited to linear systems.

In a FF setting, Bacconi et al. proposed a CG-based strategy to control both the positions and attitudes of spacecraft in formation [72]. Their approach enforced control constraints and formation accuracy constraints, but relied on pre-computed (offline) trajectories for each spacecraft to follow in order to ensure no collisions between spacecraft.

The Parameter Governor (PG) is another add-on predictive controller similar to the RG and CG [73]. PGs work by modifying parameters, such as gains or offsets, in nominal closed loop control schemes in order to enforce constraints and improve performance. Parameters are updated based on a solution of a low-dimensional optimization problem, where the parameters are the adjustable variables. In this respect, PGs are also related to Parametrized MPC and other MPC-based control techniques discussed above. For PGs, in some cases, the values of adjustable variables (parameters) can be confined to a finite set of small cardinality so that the solution can be determined by direct search, i.e., running a small number of simulations over the range of possible parameter values. For this reason, PGs are an attractive option for SRMP. To the author's knowledge, the work developed in this dissertation represents the first application of PGs to SRMP, and specifically to FF.

1.4 Contributions and Outline

1.4.1 Contributions

The SRMP methods developed in this dissertation provide a means to partially overcome some of the difficulties discussed in Section 1.3. Specifically, the bSRMP and FF methods developed here are shown to be capable of accommodating non-linear and non-convex constraints to ensure collision avoidance, and use straightforward and fast computations that may be implementable onboard spacecraft. Robustness to disturbances is either theoretically guaranteed and/or demonstrated through simulation. The specific contributions of this dissertation are listed below.

Contributions related to bSRMP:

1. Development of an invariance-based satellite relative motion planning scheme. This method:

    - Plans maneuvers to transition a spacecraft between specified Natural Motion

Trajectories (NMTs), which require no control to follow (or minimal control with disturbances), utilizing a series of intermediate NMTs and fuel efficient transfers between them,

- Determines maneuvers via graph search, yielding fast computations,
- Accommodates minimum and maximum actuation limits and guarantees avoidance of prescribed exclusion zones (obstacles),
- Develops trajectories that are closed-loop followed,
- Accounts for set-bounded disturbances in the construction of the virtual net; thereby guaranteeing robustness to disturbances.

2. Development of an information collection model and an analytical control law for use in satellite inspection applications. This work:

   - Generates maneuvers to obtain information about a target spacecraft,
   - Utilizes a rapidly computable analytical control law that drives the spacecraft toward the optimal location for information collection,
   - Is shown to accommodate thruster saturation limits and collision avoidance constraints through construction of the control law and through simulation,
   - Demonstrates robustness to disturbances through simulation.

These contributions stem from the work described in Chapters 3 and 4, respectively.

Contributions related to FF:

3. Development of a general PG-based control scheme to generate and maintain formations containing an arbitrary number of agents (spacecraft). This method:

   - Is developed using generic non-linear dynamics and an arbitrary formation design,
   - Guarantees convergence to the desired formation under a specified set of assumptions,
   - Enforces non-linear and non-convex constraints on state and control variables,
   - Distributes required calculations over time and among agents,
   - Develops trajectories that are closed-loop followed,
   - Can be applied to spacecraft FF or other non-spacecraft formation control applications.

4. Development of two specific PG-based control schemes to generate and maintain spacecraft formations. This work:

    - Illustrates two possible applications of the general framework described above,
    - Generates formation of spacecraft travelling along NMTs at desired phasing,
    - Enforces thruster saturation limits and separation distance constraints between spacecraft,
    - Generates maneuvers that are fuel efficient and closed-loop followed,
    - Demonstrates robustness to disturbances through simulation.

These contributions stem from the work described in Chapter 5.

1.4.2 Outline

To begin the dissertation, in Chapter 2, topics, models and concepts common to the work described in Chapters 3-5 are presented. These include a derivation of the satellite relative motion dynamics model and basic state-feedback control law used in simulations. Additionally, a description of the various types of NMTs, i.e., trajectories that can be followed with no control (fuel) use (or minimal control if disturbances are considered), that are exploited in Chapters 3 and 5 is included.

In Chapter 3, an invariance-based relative motion planning scheme is developed. This method plans maneuvers using a graph search on a "virtual net" (directed graph) with nodes corresponding to NMTs. Adjacency and connection information in the virtual net is determined by conditions defined in terms of Safe, Positively Invariant (SPI) tubes built around each NMT. These conditions guarantee that transitions from one NMT to an adjacent NMT can be completed without constraint violation, even when set-bounded disturbances are present. Procedures to construct the SPI tubes and resulting virtual net are developed. Simulations demonstrate that the method can be used to plan and execute maneuvers that accommodate upper and lower bounds on control inputs and avoid specified exclusion zones (obstacles). While the computations required to plan each maneuver after the virtual net has been formed are minimal, the nominal calculations required to form the virtual net are extensive and may not be feasible for on-board implementation. Several methods to reduce the required computations are presented and shown to reduce computation time significantly with minor tradeoffs related to maneuver planning flexibility.

In Chapter 4, two tools are developed for use in satellite inspection applications. Firstly, an information collection model is developed that expresses the rate of information collection as a function of both distance- and angle-to-target. This model is then used to develop

a rapidly computable analytical control law that drives the inspector spacecraft on a path along which the rate of information collection is strictly increasing. This control law, combined with a simple state feedback control law, is shown through simulations to be capable of executing maneuvers to obtain information about a target spacecraft while enforcing thruster saturation limits and avoiding an exclusion zone centered on the target spacecraft.

In Chapter 5, a general PG-based control scheme is presented to generate and maintain multi-agent formations. The dynamics model and nominal inner-loop controller are described, and the basic implementation of a PG-based controller is introduced. Properties of the system controller, formation reference trajectories, and PG cost function needed to ensure convergence are clearly stated. To reduce the computations required in implementation, a "turn-based" parameter update method based on graph colorability theory is described. A convergence analysis is presented showing that the PG can be used to generate and maintain the desired formation if certain reasonable assumptions hold. Two specific PGs, the Scale Shift Governor (SSG) and Time Shift Governor (TSG), are then introduced as FF-specific applications of the general PG-based control scheme. The SSG and TSG work by modifying, for each spacecraft in formation, a scale factor which enlarges or shrinks the reference trajectory, or a time-shift along a specified reference trajectory, respectively. Simulations are presented to illustrate the implementation of both the SSG and TSG, and demonstrate robustness to additive input disturbances. Finally, concluding remarks and a discussion of future research directions are presented in Chapter 6.

Relevant publications pertaining to the bSRMP techniques discussed in this dissertation can be found in references [74–77], whereas publications pertaining to the FF techniques discussed in this dissertation can be found in references [78, 79].

1.5 Notation

Standard notations are used throughout the dissertation. The set of real numbers is denoted by $\mathbb{R}$ and the set of integers is denoted by $\mathbb{Z}$. Subsets of these sets are denoted by a subscript, e.g., the set of non-negative real numbers is denoted by $\mathbb{R}_{>0}$ and the set of integers between 0 and T is denoted by $\mathbb{Z}_{[0,T]}$. A symmetric positive-(semi)definite matrix Φ is denoted by $\Phi \succ (\succeq) 0$. The p-norm of a vector v is denoted by $\|v\|_p$. The symbol k is used to denote the discrete-time instant, while t is used to denote continuous-time. Other notations are defined throughout the dissertation immediately following their first use. In sections or chapters utilizing significant non-standard notations, these notations are summarized in either tables or in stand-alone sections for easy reference.

CHAPTER 2

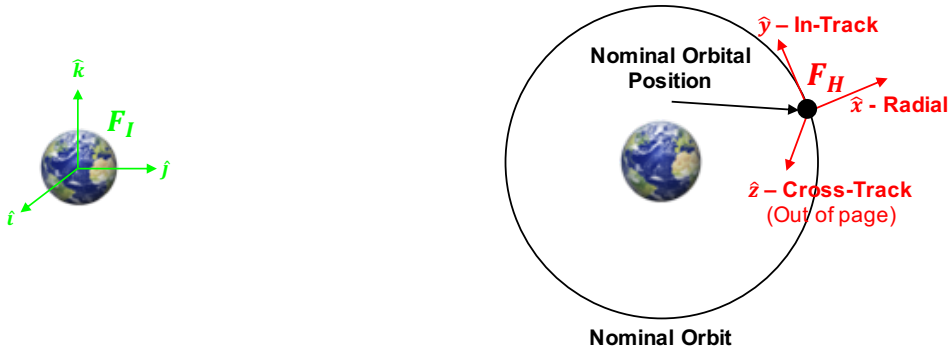
Preliminaries

2.1 Relative Motion Dynamics

This section contains a brief derivation of the equations of relative motion used in this dissertation. These equations express the position and velocity of a spacecraft maneuvering relative to a nominal target, i.e., position along a nominal orbit. Firstly, non-linear equations of relative motion are derived. Then, these equations are linearized, yielding the set of Linear Time Invariant (LTI) equations that are used extensively in Chapters 3-5.

2.1.1 Reference Frames

The derivation for the equations of relative motion in the following subsections utilizes two reference frames: an earth-centered-inertial (ECI) frame (denoted by F_I), and a local-vertical-local-horizontal frame commonly known as Hill's frame (denoted by F_H), see Figure 2.1. The axes for each of these reference frames are defined in Table 2.1. The axes for



(a) The ECI frame

(b) Hill's frame

Figure 2.1: Reference frames

Hill's frame are based on a nominal orbit, and a nominal position (target) along this orbit. Note that as this nominal target moves along the orbit, Hill's frame rotates, with respect to the ECI frame, about the $\hat{z}$ axis.

Table 2.1: Reference frame axes for ECI frame and Hill's frame

Axes for ECI frame	
$\hat{i}$	Vernal equinox direction
$\hat{k}$	Earth's axis of rotation
$\hat{j}$	Completes right-handed coordinate frame
Axes for Hill's frame	
$\hat{x}$	Along line from center of Earth to a nominal orbital position (radial)
$\hat{z}$	Along nominal orbit angular momentum vector (cross-track)
$\hat{y}$	Completes right-handed coordinate frame (in-track)

2.1.2 Non-linear Equations of Relative Motion

In this section, non-linear equations of motion are developed for a maneuvering spacecraft, denoted by M, relative to the nominal orbital position, denoted by T, in Hill's frame. The position vector of M with respect to T is given by

$$r_{M/T} = r_{M/O} - r_{T/O}, \tag{2.1}$$

where O denotes the center of the Earth, see Figure 2.2.

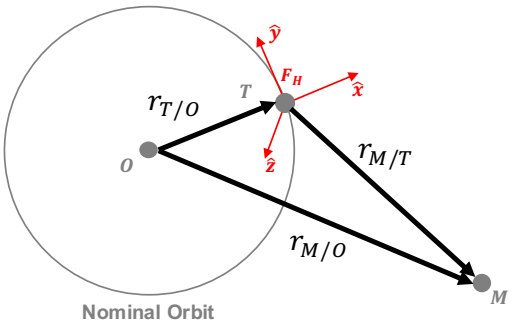


Figure 2.2: Basic set-up for derivation of equations of motion

Taking time-derivatives of (2.1) with respect to F_I yields

$$\overset{I}{\ddot{r}}_{M/T} = \overset{I}{\ddot{r}}_{M/O} - \overset{I}{\ddot{r}}_{T/O}. \tag{2.2}$$

The two terms on the Right Hand Side (RHS) of (2.2) are simply the inertial equations of motion for the maneuvering spacecraft and the nominal orbital position, respectively, and are given by the Restricted Two-Body Equations of Motion (R2BEOM) (see Chapter 2 in [22]). Substituting the R2BEOM into (2.2) results in

$$\overset{I\cdot\cdot}{r}_{M/T} = -\mu\frac{r_{M/O}}{\|r_{M/O}\|_2^3} + \mu\frac{r_{T/O}}{\|r_{T/O}\|_2^3}, \tag{2.3}$$

where μ is the gravitational parameter for Earth.

The expressions on the Left Hand Side (LHS) and RHS of (2.3) are now resolved in Hill's frame. First, consider the LHS. Using the transport theorem, the LHS of (2.3) is expressed as

$$\overset{I\cdot\cdot}{r}_{M/T} = \overset{H\cdot\cdot}{r}_{M/T} + 2\omega_{H/I} \times \overset{H\cdot}{r}_{M/T} + \overset{H\cdot}{\omega}_{H/I} \times r_{M/T} + \omega_{H/I} \times (\omega_{H/I} \times r_{M/T}), \tag{2.4}$$

where $\omega_{H/I}$ is the angular velocity of Hill's frame with respect to the ECI frame, and the superscripts $^{H\cdot}$ and $^{H\cdot\cdot}$ denote first and second time-derivaties with respect to F_H, respectively. Equation (2.4) is resolved in Hill's frame using the following notation,

$$\begin{gathered} r_{M/T}\big|_H = x\hat{x} + y\hat{y} + z\hat{z} = [x,\ \ y,\ \ z]^T, \\ \omega_{H/I}\big|_H = [0,\ \ 0,\ \ \theta]^T, \end{gathered} \tag{2.5}$$

resulting in

$$\overset{I\cdot\cdot}{r}_{M/T}\Big|_H = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + 2\begin{bmatrix} 0 \\ 0 \\ \theta \end{bmatrix} \times \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta} \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \theta \end{bmatrix} \times \left\{ \begin{bmatrix} 0 \\ 0 \\ \theta \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} \right\}. \tag{2.6}$$

By carrying out the cross products and combining terms, (2.6) simplifies to

$$\overset{I\cdot\cdot}{r}_{M/T}\Big|_H = \begin{bmatrix} \ddot{x} - 2\theta\dot{y} - \dot{\theta}y - \theta^2 x \\ \ddot{y} + 2\theta\dot{x} + \dot{\theta}x - \theta^2 y \\ \ddot{z} \end{bmatrix}. \tag{2.7}$$

Next, consider the RHS of (2.3):

$$-\mu\frac{r_{M/O}}{\|r_{M/O}\|_2^3} + \mu\frac{r_{T/O}}{\|r_{T/O}\|_2^3}. \tag{2.8}$$

The vectors $r_{T/O}$ and $r_{M/O}$, resolved in Hill's frame, are given by

$$r_{T/O}\big|_H = \left[\|r_{T/O}\|_2,\ \ 0,\ \ 0\right]^T, \tag{2.9}$$

and

$$r_{M/O}\big|_H = \left[\|r_{T/O}\|_2 + x,\ \ y,\ \ z\right]^T, \tag{2.10}$$

respectively. Substituting (2.9) and (2.10) into (2.8), and using the abbreviated notation $r_T = \|r_{T/O}\|_2$ yields

$$\left[-\mu\frac{r_{M/O}}{\|r_{M/O}\|_2^3} + \mu\frac{r_{T/O}}{\|r_{T/O}\|_2^3}\right]\Big|_H = \begin{bmatrix} -\mu\frac{r_T+x}{[(r_T+x)^2+y^2+z^2]^{3/2}} + \frac{\mu}{r_T^2} \\ -\mu\frac{y}{[(r_T+x)^2+y^2+z^2]^{3/2}} \\ -\mu\frac{z}{[(r_T+x)^2+y^2+z^2]^{3/2}} \end{bmatrix}. \tag{2.11}$$

Setting (2.7) equal to (2.11) component-wise, and solving for the acceleration terms yields the following non-linear equations of motion for the maneuvering spacecraft relative to the nominal orbital position, expressed in Hill's frame:

$$\begin{aligned} \ddot{x} &= -\tfrac{\mu(r_T+x)}{[(r_T+x)^2+y^2+z^2]^{3/2}} + \tfrac{\mu}{r_T^2} + 2\theta\dot{y} + \dot{\theta}y + \theta^2 x, \\ \ddot{y} &= -\tfrac{\mu y}{[(r_T+x)^2+y^2+z^2]^{3/2}} - 2\theta\dot{x} - \dot{\theta}x + \theta^2 y, \\ \ddot{z} &= -\tfrac{\mu z}{[(r_T+x)^2+y^2+z^2]^{3/2}}. \end{aligned} \tag{2.12}$$

If there are external disturbances (such as unmodeled dynamics), denoted by $w = [w^1,\ \ w^2,\ \ w^3]^T$, or control forces, denoted by $u = [u^1,\ \ u^2,\ \ u^3]^T$, these can be added to the RHS of (2.12) to yield

$$\begin{aligned} \ddot{x} &= -\tfrac{\mu(r_T+x)}{[(r_T+x)^2+y^2+z^2]^{3/2}} + \tfrac{\mu}{r_T^2} + 2\theta\dot{y} + \dot{\theta}y + \theta^2 x + w^1 + u^1, \\ \ddot{y} &= -\tfrac{\mu y}{[(r_T+x)^2+y^2+z^2]^{3/2}} - 2\theta\dot{x} - \dot{\theta}x + \theta^2 y + w^2 + u^2, \\ \ddot{z} &= -\tfrac{\mu z}{[(r_T+x)^2+y^2+z^2]^{3/2}} + w^3 + u^3. \end{aligned} \tag{2.13}$$

2.1.3 Linearized Equations of Relative Motion

In most SRMP scenarios, the maneuvering spacecraft is in close proximity to the nominal target, i.e., $\|r_{M/T}\|_2 \ll r_T$. In this case, the non-linear Equations (2.12) can be linearized about the nominal target, i.e., the origin $x = y = z = \dot{z} = \dot{y} = \dot{z} = 0$, yielding the

following linear equations of relative motion:

$$\begin{aligned} \ddot{x} &= \left(\tfrac{2\mu}{r_T^3} + \theta^2\right)x + \dot{\theta}y + 2\theta\dot{y}, \\ \ddot{y} &= -\dot{\theta}x + \left(\theta^2 - \tfrac{\mu}{r_T^3}\right)y - 2\theta\dot{x}, \\ \ddot{z} &= -\tfrac{-\mu}{r_T^3}z. \end{aligned} \tag{2.14}$$

In general, these equations are time-varying. However, simplifications to this general case can be made if the nominal target is along either an elliptical or circular orbit. If the nominal target orbit is elliptical, then (2.14) are time-periodic, with a period equal to that of the nominal orbit. In this case, (2.14) can be reformulated by scaling the position coordinates by r_T, and changing the independent variable from time to the true anomaly along the nominal orbit. These operations result in a set of linear time-periodic equations know as the Tschauner-Hempel (TH) equations [80]. Further simplifications are possible if the nominal target is assumed to be along a circular orbit.[1] In this case, (2.14) are time-invariant.

2.1.3.1 The Clohessy-Wiltshire Equations

When the nominal target orbit is circular, the angular velocity θ in (2.14) is constant, and is equal to the mean-motion, ω, of the orbit, i.e., $\theta = \omega$, $\dot{\theta} = 0$ and $\omega = \sqrt{\mu/r_T^3}$. In this case, Equations (2.14) simplify to a set of LTI equations given by

$$\begin{aligned} \ddot{x} &= 3\omega^2 x + 2\omega\dot{y}, \\ \ddot{y} &= -2\omega\dot{x}, \\ \ddot{z} &= -\omega^2 z. \end{aligned} \tag{2.15}$$

The LTI equations of relative motion given by (2.15) were originally developed for satellite rendezvous applications in the 1960s, and are commonly referred to as the Clohessy-Wiltshire (CW) equations [82]. If there are control inputs, $u = [u^1,\ \ u^2,\ \ u^3]^T$, these can be added to the RHS of (2.15), yielding

$$\begin{aligned} \ddot{x} &= 3\omega^2 x + 2\omega\dot{y} + u^1, \\ \ddot{y} &= -2\omega\dot{x} + u^2, \\ \ddot{z} &= -\omega^2 z + u^3. \end{aligned} \tag{2.16}$$

[1] The assumption of a nominal circular orbit is reasonable for many SRMP applications because the majority (79%) of satellite orbits are nearly circular, i.e., have an eccentricity of less than 0.025 [81].

By defining a state vector X as

$$X = [x,\ \ y,\ \ z,\ \ \dot{x},\ \ \dot{y},\ \ \dot{z}]^T, \tag{2.17}$$

where x, y, and z, are the relative position coordinates of the maneuvering spacecraft in Hill's frame, and $\dot{x}$, $\dot{y}$, and $\dot{z}$ are components of the relative velocity vector, (2.16) can be written in continuous-time state-space form,

$$\dot{X}(t) = A_c X(t) + B_c u(t), \tag{2.18}$$

where the dynamics matrix A_c is given by

$$A_c = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3\omega^2 & 0 & 0 & 0 & 2\omega & 0 \\ 0 & 0 & 0 & -2\omega & 0 & 0 \\ 0 & 0 & -\omega^2 & 0 & 0 & 0 \end{bmatrix}, \tag{2.19}$$

$\omega = \sqrt{\mu/r_T^3}$ is the mean motion of the nominal circular orbit, and the matrix B_c can take different forms depending on the nature of the control inputs. Note that the dynamics matrix A_c has eigenvalues given by $\lambda = \pm\omega j,\ \ \pm\,\omega j,\ 0,\ 0$. Therefore, natural (unforced) motion in the CW equations has both periodic and divergent modes.

Assuming a sampling period of ΔT seconds, (2.18) can be written in discrete-time,

$$X(k+1) = AX(k) + Bu(k), \tag{2.20}$$

where

$$A = \exp(A_c \Delta T), \tag{2.21}$$

and, again, depending on the form of the control input, the discrete-time input matrix B has different representations. In this dissertation, two types of control inputs are considered. Firstly, when the control inputs correspond to continuous thrust forces, the discrete-time input matrix is given by

$$B = B_f = \int_0^{\Delta T} \exp\left[A_c(\Delta T - \tau)\right] d\tau \begin{bmatrix} 0_{3\times 3} \\ \frac{1}{m} I_{3\times 3} \end{bmatrix}, \tag{2.22}$$

where m is the mass of the maneuvering spacecraft (assumed to be constant), $0_{3\times 3}$ is the

3×3 matrix of all zeros and $I_{3\times3}$ is the 3×3 identity matrix. Secondly, when the control vector corresponds to instantaneous velocity change (ΔV), the discrete-time input matrix is given by

$$B = B_v = A \begin{bmatrix} 0_{3\times3} \\ I_{3\times3} \end{bmatrix}. \tag{2.23}$$

Additive disturbances, denoted $w(k) = [w^1(k),\ \ w^2(k),\ \ w^3(k)]^T$, can be included in the model (2.20) by adding them to the RHS, yielding

$$X(k+1) = AX(k) + Bu(k) + B_w w(k), \tag{2.24}$$

where the disturbance input matrix, B_w, is appropriately selected based on the nature of the disturbance. For example, if the disturbances are forces, then $B_w = B_f$, where B_f is defined in (2.22). The discrete-time LTI equations of relative motion without disturbances, i.e., (2.20), and with disturbances, i.e., (2.24) are used throughout Chapters 3-5 of this dissertation.

2.2 Natural Motion Trajectories

A Natural Motion Trajectory (NMT) is defined as a solution to (2.20) with $u = 0$, or a solution to (2.24) with $u = w = 0$. As mentioned in Section 2.1.3.1, natural motions in the CW equations exhibit both periodic and divergent modes. Therefore, depending on the initial condition, X_0, many different types of NMTs can be obtained. Specifically, if the initial condition $X_0 = [x_0,\ y_0,\ z_0,\ \dot{x}_0,\ \dot{y}_0,\ \dot{z}_0]^T$ is chosen to satisfy

$$\dot{y}_0 = -2\omega x_0, \tag{2.25}$$

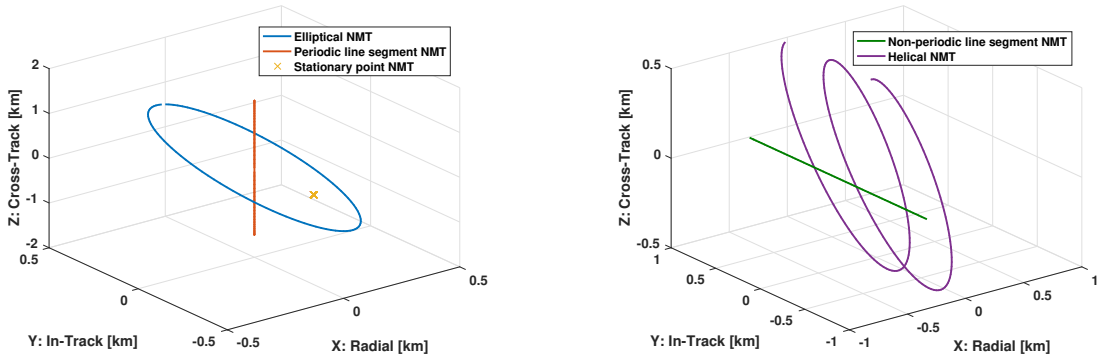
then the resulting NMT will be periodic (closed) [83]. For example, with appropriate initial conditions selected, elliptical trajectories can be obtained.[2] With other types of initial conditions chosen, the resulting NMT will be non-periodic (open), e.g., a straight-line segment.

NMTs are useful in a SRMP setting for a number of reasons. Firstly, when a satellite is traveling along a NMT, control must only be used to counteract disturbances, resulting in minimal fuel use. Secondly, if obstacles are located in known regions, placing spacecraft along NMTs which avoid these regions guarantees no collisions with obstacles. Additionally, closed NMTs are particularly useful for SRMP since spacecraft travelling along these

[2]The shape of an NMT here refers to the shape traced out when the NMT is plotted in Hill's frame.

NMTs are able to circumnavigate or remain near the target indefinitely while using minimal fuel.

Because of these advantages, several types of NMTs are used in the SRMP methods discussed in Chapters 3-5, including closed NMTs such as ellipses, periodic line segments and stationary points, and open NMTs such as helices and non-periodic line segments. Figure 2.3 shows examples of these types of NMTs. Methods to generate initial conditions for these types of NMTs are available in the literature (see, e.g., [84,85]). For completeness, one such method is summarized in Appendix A.



(a) Closed NMT examples

(b) Open NMT examples

Figure 2.3: Examples of different types of NMTs plotted in Hill's frame

2.3 Nominal Control law

The nominal control law used to obtain $u(k)$ in (2.20) and (2.24) is a static-state feedback control law given by

$$u(k) = K(X(k) - X_d(k)), \tag{2.26}$$

where K is a gain matrix and $X_d(k)$ is the controller reference point. Note that this reference point may be static, or time-varying, e.g., it may move along an NMT or other trajectory. If the controller gain matrix K is selected such that the matrix $\bar{A} = A + BK$ is Schur, i.e., all eigenvalues are in the interior of the unit disk on the complex plane, the spacecraft will asymptotically approach the controller reference if there are no disturbances. If bounded disturbances are present, the spacecraft will instead approach a neighborhood around the controller reference point, where the size of this neighborhood depends on the size of the disturbances [86].

Use of this type of control law in SRMP applications is beneficial for two reasons. Firstly, the calculations required to obtain control inputs are minimal: only a simple matrix multiplication is required, given the gain matrix K, a state measurement and a controller reference point. Secondly, the state feedback nature of the control law naturally accommodates sufficiently small disturbances, as disturbances encountered at a given time-instant can be compensated for with the control input at subsequent time instances.

The specific controller gain matrix K used in this work is obtained via the solution to the discrete-time LQR problem. In the discrete-time LQR problem, for an LTI system

$$X(k+1) = AX(k) + Bu(k), \tag{2.27}$$

a control law $u(k)$ is sought to minimize a quadratic cost function given by

$$J(X_0) = \sum_{k=0}^{\infty} \tfrac{1}{2} X(k)^T Q_{LQ} X(k) + \tfrac{1}{2} u(k)^T R_{LQ} u(k), \tag{2.28}$$

for any initial condition X_0, where $Q_{LQ} = Q_{LQ}^T \succeq 0$ and $R_{LQ} = R_{LQ}^T \succ 0$ are weighting matrices penalizing state deviation from the origin and control usage, respectively.

In addition to the benefits of using a general stabilizing state-feedback control law discussed above, the use of the LQR control law provides the additional benefit of being able to tune the weighting matrices Q_{LQ} and R_{LQ} in (2.28) to provide either faster response, by placing larger weights on state deviation, or more economical control use by placing larger weights on control use.

The discrete-time LQR problem can be solved using Bellman's principle for optimality [87]. Bellman's equation is given by

$$V(X(k)) = \min_{u(k)} \left[\tfrac{1}{2} X(k)^T Q_{LQ} X(k) + \tfrac{1}{2} u(k)^T R_{LQ} u(k) + V(AX(k) + Bu(k)) \right]. \tag{2.29}$$

We seek a $V(X(k))$ and $u^*(x(k))$ that satisfy (2.29). Consider a potential solution for $V(x(k))$ of the form

$$V(x(k)) = \tfrac{1}{2} X(k)^T P X(k),\ P = P^T. \tag{2.30}$$

Substituting (2.30) into (2.29) yields

$$\begin{aligned} \tfrac{1}{2} X(k)^T P X(k) = \min_{u(k)} \quad & \left[\tfrac{1}{2} X(k)^T Q_{LQ} X(k) + \tfrac{1}{2} u(k)^T R_{LQ} u(k) \right. \\ & \left. + \tfrac{1}{2} (AX(k) + Bu(k))^T P (AX(k) + Bu(k)) \right]. \end{aligned} \tag{2.31}$$

To obtain u^*, the derivative of the RHS of (2.31) with respect to u is given by

$$\frac{\partial}{\partial u(k)} = u(k)^T R_{LQ} + X(k)^T APB + u(k)^T B^T PB, \tag{2.32}$$

and setting this expression equal to 0 and solving for u yields

$$u^*(k) = -(R_{LQ} + B^T PB)^{-1} B^T PAX(k). \tag{2.33}$$

Hence the value for the RHS of (2.31) is given by

$$\begin{aligned} &\tfrac{1}{2}\left[X(k)^T Q_{LQ} X(k) + u(k)^T R_{LQ} u(k) + (AX(k) + Bu(k))^T P(AX(k) + Bu(k))\right]\Big|_{u^*(k)} \\ &= \tfrac{1}{2} X(k)^T Q_{LQ} X(k) - \tfrac{1}{2} X(k)^T A^T PB(R_{LQ} + B^T PB)^{-1} B^T PAX(k) \\ &\quad + \tfrac{1}{2} X(k)^T A^T PAX(k), \end{aligned} \tag{2.34}$$

and (2.31) becomes

$$\tfrac{1}{2} X(k)^T PX(k) = \tfrac{1}{2} X(k)^T \left[Q_{LQ} - A^T PB(R_{LQ} + B^T PB)^{-1} B^T PA + A^T PA\right] X(k). \tag{2.35}$$

Equation (2.35) is satisfied by any matrix P that satisfies

$$P = Q_{LQ} - A^T PB(R_{LQ} + B^T PB)^{-1} B^T PA + A^T PA, \tag{2.36}$$

which is known as the Discrete Algebraic Riccatti Equation (DARE) [88]. Hence, if a solution $P = P^T$ to the DARE exists (see Remark 2.1), then, Bellman's equation (2.29) is satisfied with $V(X(k)) = \frac{1}{2} X(k)^T PX(k)$, and the optimal control policy is given by

$$u(k) = KX(k), \tag{2.37}$$

where the matrix K is a constant matrix given by

$$K = -(R_{LQ} + B^T PB)^{-1} B^T P\bar{A}. \tag{2.38}$$

Remark 2.1. *If the pair (A, B) is stabilizable, $Q_{LQ} = C^T C$, and (A, C) is observable, then there exists a unique, positive definite solution to DARE (2.36), and the resulting control law given by (2.37) and (2.38) is asymptotically stabilizing, i.e., $A + BK$ is Schur, see Lemma 26.2 in [89]. In implementations, the solution to the DARE, P, and the associated state feedback gain matrix K can be obtained via the "dlqr" function in MATLAB.*

CHAPTER 3

Invariance-Based Spacecraft Relative Motion Planning

In this chapter, a novel method for bSRMP is developed to plan maneuvers that transition a spacecraft between specified NMTs while satisfying constraints. The method is based on a graph search applied to a "virtual net" composed of periodic (closed) and non-periodic (open) NMTs. Adjacency (connectivity) between NMTs in the virtual net is determined by conditions based on SPI tubes built around each NMT. For background information on invariance, SPI sets, and their use, see, e.g., [90–94]. The bSRMP method described here is based on a framework developed in [48] to plan maneuvers on a virtual net composed of forced equilibria. The use of NMTs has several advantages compared with the forced equilibria considered in [48]. Firstly, traveling along NMTs in steady state is possible with zero fuel consumption (or minimal fuel consumption with disturbances). In contrast, zero fuel consumption is only achieved for forced equilibria along the in-track (y) axis in the CW equations. Secondly, the use of NMTs expands the set of trajectories available to compose the overall maneuver from, while ensuring that the resulting maneuvers are fuel efficient. Specifically, when the maneuver consists of NMTs and transfers connecting them, fuel is consumed only during the transfers and to compensate for disturbances.

This chapter is organized as follows. Section 3.1 describes the spacecraft dynamics model, the disturbance model and constraints, and the types of NMTs considered. Section 3.2 describes SPI tubes, and Section 3.3 provides procedures that can be used to construct SPI tubes for NMTs. Section 3.4 describes the virtual net used for maneuver planning, and provides rules used to define adjacency in the virtual net, and calculations used to check these rules. The calculations developed in Section 3.4 are non-conservative, i.e., they correctly identify each pair of adjacent nodes in the virtual net, but they are slow. In Section 3.6, faster calculations are developed to determine adjacency in the virtual net with minor trade-offs related to maneuver planning flexibility and the fuel-efficiency of planned maneuvers. Simulation results are presented in Sections 3.5 and 3.7.

3.1 Modeling

In this chapter, spacecraft motion is modeled using the LTI discrete-time equations of relative motion (2.24), repeated here for clarity:

$$X(k+1) = AX(k) + Bu(k) + Bw(k). \tag{3.1}$$

In (3.1), the matrix A is given by (2.21) and both the control input $u(k)$ and disturbance $w(k)$ correspond to continuous force, hence the matrix B is given by (2.22). The disturbance vector $w(k)$ is assumed to belong to a compact set $\mathcal{W}$, i.e., $w(k) \in \mathcal{W} \subset \mathbb{R}^3$.

3.1.1 Natural Motion Trajectories

A periodic (closed) NMT, starting from initial condition X_0, is defined as a finite set of state vectors:

$$\mathcal{N}(X_0, k_{max}) = \mathcal{N} = \left\{ X_n(k) \mid X_n(0) = X_0,\ X_n(k+1) = AX_n(k),\ k \in \mathbb{Z}_{[0,k_{max}]} \right\}, \tag{3.2}$$

where the parameter k_{max} is chosen such that the set $\mathcal{N}$ completely defines the NMT. Specifically, the discrete-time update period, ΔT, is chosen such that $\frac{\tau}{\Delta T} \in \mathbb{Z}_{>0}$, where τ is the period of the nominal circular orbit, and

$$k_{max} = \frac{\tau}{\Delta T} - 1. \tag{3.3}$$

Note that the value for k_{max} depends only on the period of the nominal circular orbit and the discrete-time update period. Consequently, k_{max} is constant for all closed NMTs. With this choice of k_{max}, the sequence of state vectors repeats for $k > k_{max}$, i.e., $X_n(k_{max}+1) = X_n(0)$, $X_n(k_{max}+2) = X_n(1)$, etc. For closed NMTs, in general, $X_n(k) = X_n(\tilde{k})$, where

$$\tilde{k} = \text{mod}(k,\ k_{max}+1), \tag{3.4}$$

and the modulo function $\text{mod}(x, y)$ returns the remainder after division of x by y. In the subsequent developments, any $k > k_{max}$ along a closed NMT is taken to be the equivalent index $\tilde{k} \in \mathbb{Z}_{[0,\ k_{max}]}$ given by (3.4).

A portion of a non-periodic (open) NMT, starting from initial condition X_0, is also

defined as a finite set of state vectors:

$$\bar{\mathcal{N}}(X_0, k_{max}^{\bar{\mathcal{N}}}) = \bar{\mathcal{N}} = \left\{ X_n(k) \mid X_n(0) = X_0,\ X_n(k+1) = AX_n(k),\ k \in \mathbb{Z}_{[0,k_{max}^{\bar{\mathcal{N}}}]} \right\}, \tag{3.5}$$

where the initial condition X_0 and the parameter $k_{max}^{\bar{\mathcal{N}}}$ are chosen such that the segment of the open NMT defined by (3.5) lies in the region of interest for the present mission. Note that for open NMTs, $k_{max}^{\bar{\mathcal{N}}}$ may take different values for different NMTs.

3.1.2 Spacecraft Control Law and Closed Loop Dynamics

The nominal feedback law that guides the spacecraft to a given NMT is

$$u(k) = K(X(k) - X_n(k+\delta)), \tag{3.6}$$

where K is an LQR gain matrix, described in Section 2.3, for which the matrix $\bar{A} = A + BK$ is Schur, $X(k)$ is the current spacecraft state, $X_n(k+\delta) \in \mathcal{N}$(or $\in \bar{\mathcal{N}}$) is a time-varying reference along the NMT, and $\delta \in \mathbb{Z}$ is a shift which defines the controller reference point at the first time-instant the controller is switched to the specified NMT as the target.

Combining (3.1) and (3.6), the closed loop dynamics are given by

$$X(k+1) = \bar{A}X(k) - BKX_n(k+\delta) + Bw(k). \tag{3.7}$$

Definition 3.1. [Reachable set for the closed-loop system (3.7)] *The set of all states reachable at time instant $\bar{k}$ for the closed-loop system (3.7), starting from the set $\mathcal{X}_0$ at time instant 0, and with initial controller reference point $X_n(0+\delta) = X_{n0}$, under all possible disturbances $w(k) \in \mathcal{W},\ k \in [0, \bar{k}-1]$, is denoted as*

$$\mathcal{R}_X(\bar{k},\ \mathcal{X}_0,\ X_{n0},\ \mathcal{W}). \tag{3.8}$$

Defining the state error as $e(k,\delta) = X(k) - X_n(k+\delta)$, the error system dynamics are given by

$$e(k+1,\delta) = \bar{A}e(k,\delta) + Bw(k). \tag{3.9}$$

Definition 3.2. [Reachable set for the error system (3.9)] *The set of all error states reachable at time instant $\bar{k}$ for the error system (3.9), starting from the set ϵ_0 at time instant 0, under all possible disturbances $w(k) \in \mathcal{W},\ k \in [0, \bar{k}-1]$, is denoted as*

$$\mathcal{R}_e(\bar{k},\ \epsilon_0,\ \mathcal{W}). \tag{3.10}$$

In the following, the notation for the state error is occasionally simplified by omitting δ, and using a superscript $^+$ to denote the 1-step ahead error, i.e.,

$$e^+ = \bar{A}e + Bw. \tag{3.11}$$

3.1.3 Constraints

Three constraints are considered. Firstly, the thrust is limited by an upper bound in each direction, i.e.,

$$\|u(k)\|_\infty - u_{max} \leq 0, \tag{3.12}$$

where $\|\cdot\|_\infty$ denotes the infinity-norm, and u_{max} is the maximum allowable thrust in each direction. Secondly, the satellite must avoid one or more exclusion zones, modeled as ellipsoidal sets centered at specified points $s \in \mathbb{R}^3$. The ith exclusion zone is defined as

$$\mathcal{O}_i(s_i, S_i) = \{X \mid (\Phi X - s_i)^T S_i (\Phi X - s_i) \leq 1\}, \tag{3.13}$$

where $S_i \succ 0$ is a matrix reflecting the shape characteristics of the obstacle, including any uncertainty in its position, and the matrix $\Phi = [I_{3\times3}\ 0_{3\times3}]$ isolates the position components from the state vector, see Remark 3.1. The constraints on the spacecraft's position based on l exclusion zones are given by $X(k) \notin \mathcal{O}_i(s_i, S_i),\ i = 1, 2, \cdots, l$, which is equivalent to the inequality constraints

$$1 - (\Phi X(k) - s_i)^T S_i (\Phi X(k) - s_i) \leq 0,\ i = 1, 2, ..., l \tag{3.14}$$

The constraints (3.12) and (3.14) are enforced by incorporating them into the definition of safe tubes described in Sections 3.2.1 and 3.2.2.

A third constraint is also considered in which the thrust magnitude is limited by a lower bound in each direction, i.e.,

$$u^i(k) \in [\{0\} \cup \{u^i \mid |u^i| \geq u_{min}\}],\ i = 1, 2, 3, \tag{3.15}$$

where $u^i(k)$ is the ith component of the control vector and u_{min} is the minimum realizable thrust. This non-convex constraint is similar to minimum impulse bit constraints, which are common to operational spacecraft and result from hardware restrictions limiting the shortest amount of time a thruster can be active. The minimum control constraint (3.15) is incorporated in this work by treating the minimum thrust limit as a disturbance, as described in the next subsection, and including these disturbances in the generation of SPI

tubes described in Section 3.2.3. This approach ensures that all maneuvers planned using the proposed methodology avoid obstacles, even if control inputs violating (3.15) are commanded, but cannot be implemented. See Remark 3.2.

Remark 3.1. *Using ellipsoidal sets to model exclusion zones is beneficial in two respects. Firstly, if the exact obstacle location is unknown, position estimation algorithms, e.g., Kalman filters, can be used to obtain ellipsoidal bounds for the obstacle. Secondly, the use of ellipsoidal sets simplifies some of the calculations used to generate SPI tubes and form the virtual net, described in Sections 3.2-3.4. Note also that if a given exclusion zone is defined in more general terms, e.g., using polynomial barrier functions, these exclusion zones can always be over-bound with an ellipsoidal set in the form of (3.13).*

Remark 3.2. *The choice to accommodate the non-convex minimum thrust constraint using disturbances was made to simplify the required calculations. Because the minimum thrust limit is typically relatively small, this approach is not overly conservative. Other approaches to handling this type of constraint have been developed, e.g., [95]. Comparisons between these different approaches is a topic for future research.*

3.1.4 Disturbance Set

As previously noted, the disturbance vector $w(k)$ in (3.1) belongs to a compact set $\mathcal{W} \subset \mathbb{R}^3$. In this subsection, the form of the disturbances considered and the structure of the set $\mathcal{W}$ are defined in detail. The disturbance vector $w(k)$ consists of two parts:

$$w(k) = w_p(k) + w_u(k). \tag{3.16}$$

In (3.16), the term $w_p(k)$ is randomly assigned from the set

$$\mathcal{W}_p = \{w_p \mid \|w_p\|_\infty \leq \epsilon\}, \tag{3.17}$$

and may account for orbital perturbations, thruster alignment errors, etc. The term $w_u(k)$ in (3.16) directly accounts for the minimum thrust constraint (3.15),

$$\begin{gathered} w_u(k) = [w_u^1(k),\ \ w_u^2(k),\ \ w_u^3(k)]^T, \\ w_u^i(k) = \begin{cases} -u^i(k) & \text{if } |u^i(k)| \leq u_{min}, \\ 0 & \text{otherwise} \end{cases} \quad i = 1, 2, 3. \end{gathered} \tag{3.18}$$

Hence, $w_u(k) \in W_u$, where

$$\mathcal{W}_u = \{w_u \mid \|w_u\|_\infty \leq u_{min}\}, \tag{3.19}$$

and the total disturbance vector $w(k)$ belongs to the set

$$\mathcal{W} = \mathcal{W}_p \oplus \mathcal{W}_u = \{w \mid \|w\|_\infty \leq \epsilon + u_{min}\}. \tag{3.20}$$

Note that the set $\mathcal{W}$ in (3.20) is a polytope, and is the convex hull of eight vertices, i.e., $\mathcal{W} = \text{conv}\{w^1,\ w^2,\ \cdots,\ w^{n_w}\}$ where $n_w = 8$.

3.2 Safe Positively Invariant Tubes for NMTs

In this section, a SPI tube is defined for a NMT. In this context, "safe" (constraint admissible) implies that constraints (3.12) and (3.14) are satisfied point-wise within the tube. "Positively invariant" implies that if the spacecraft state is within the tube at a given time instant, and the spacecraft motion is given by (3.7), then it will remain within the tube, under all possible disturbances, for all future time instants for a closed NMT or, for an open NMT, until the controller reference point is set to the final state vector along the NMT, i.e., $X_n(k_{max}^{\bar{\mathcal{N}}})$. The SPI tube is formed by first generating safe ellipsoidal sets about each state vector along the NMT. The union of these safe sets forms a safe tube around the NMT. Then, adjustments to the safe tube are made such that the tube remains safe, but also enjoys the property of positive invariance. Figure 3.1 summarizes the major steps required to generate a SPI tube for a NMT, and provides references to where each step is described in the upcoming subsections. For brevity, the notation $\mathcal{N}$ is used to denote an NMT. However, all developments in this section also apply to open NMTs, $\bar{\mathcal{N}}$, unless otherwise noted.

3.2.1 Safe Sets

An ellipsoidal set centered at $X_n(k) \in \mathcal{N}$ with scale factor $\rho_k \geq 0$ is defined as

$$\mathcal{E}_{k,\mathcal{N}} = \{X \mid (X - X_n(k))^T P (X - X_n(k)) \leq \rho_k\}, \tag{3.21}$$

where the shape matrix $P = P^T > 0$ is chosen to satisfy the discrete-time Lyapunov inequality,

$$(A + BK)^T P (A + BK) - P < 0. \tag{3.22}$$

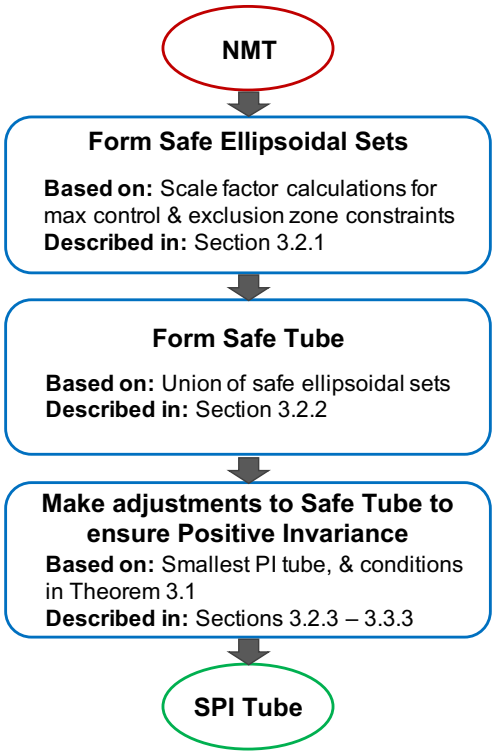


Figure 3.1: Major steps involved in generating a SPI tube for a NMT

The set $\mathcal{E}^s_{k,\mathcal{N}}$ defined by (3.21) with $\rho_k = \rho^s_k$, i.e.,

$$\mathcal{E}^s_{k,\mathcal{N}} = \{X \mid (X - X_n(k))^T P (X - X_n(k)) \leq \rho^s_k\}, \tag{3.23}$$

is safe if the scale factor ρ^s_k is set to the largest possible value such that:

a) The control constraint (3.12) is satisfied point-wise within the set with $\delta = 0$, i.e., $\|u(k)\|_\infty = \|K(X - X_n(k))\|_\infty \leq u_{max}$ for all $X \in \mathcal{E}^s_{k,\mathcal{N}}$, and

b) The exclusion zone constraints (3.14) for $i = 1, 2, \cdots, l$ are satisfied for all $X \in \mathcal{E}^s_{k,\mathcal{N}}$.

The scale factors ρ^s_k, satisfying (a) and (b) above, are determined by first calculating the maximum possible scale factor for which item (a) holds, denoted by ρ^u, and next separately calculating the maximum scale factor for which item (b) holds for each exclusion zone, denoted by $\rho^{\mathcal{O}_i}_k$. Then, ρ^s_k is determined by

$$\rho^s_k = \min\{\rho^u,\ \rho^{\mathcal{O}_i}_k,\ i = 1, 2, \cdots, l\}. \tag{3.24}$$

Note that the parameter ρ^u is independent of $X_n(k)$, i.e., the state vector along an

NMT (which requires 0 control to follow) corresponding to the center of the ellipsoidal set. Hence, ρ^u is constant for all $X_n(k)$ and all NMTs, and must only be calculated once. The parameter $\rho_k^{\mathcal{O}_i}$ is dependent on $X_n(k)$, therefore a separate $\rho_k^{\mathcal{O}_i}$ must be calculated for each $X_n(k)$ along each NMT. Calculations for ρ^u and $\rho_k^{\mathcal{O}_i}$ are based on methods developed in [48], and are included in Appendix B.

3.2.2 Safe Tubes

Using the set of safe scale factors for an NMT, ρ_k^s, $k \in \mathbb{Z}_{[0,k_{max}]}$, a safe tube centered on the NMT $\mathcal{N}$ is defined by

$$\mathcal{T}_{\mathcal{N}}^s = \bigcup_{k \in \mathbb{Z}_{[0,k_{max}]}} \mathcal{E}_{k,\mathcal{N}}^s. \tag{3.25}$$

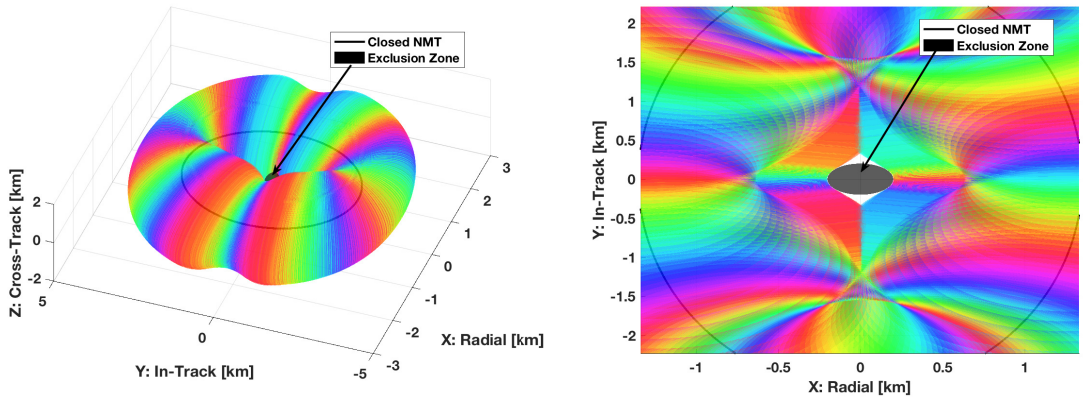
This tube is safe in the sense that for all $X \in \mathcal{T}_{\mathcal{N}}^s$, the exclusion zone constraints (3.14) are satisfied and there exists $\delta \in \mathbb{Z}$ such that the control constraint (3.12) is satisfied, i.e., there exists $X_n(k) \in \mathcal{N}$ that can be used as the controller reference point and (3.12) is satisfied.

Figure 3.2 shows four views of the projection of the six-dimensional tube $\mathcal{T}_{\mathcal{N}}^s$ onto the position space for an example closed NMT. The tube was formed considering the control constraint (3.12) and a single exclusion zone centered at the origin. In Figure 3.2, different colors correspond to different ellipsoidal sets $\mathcal{E}_{k,\mathcal{N}}^s \subset \mathcal{T}_{\mathcal{N}}^s$. Note that, as illustrated in Figure 3.2b, the boundary of the safe tube does not always intersect the boundary of the exclusion zone. Ellipsoidal sets $\mathcal{E}_{k,\mathcal{N}}^s \subset \mathcal{T}_{\mathcal{N}}^s$ whose boundaries *do not* intersect the exclusion zone have a scale factor equal to ρ^u, i.e., their size is limited by the control constraint (3.12). Ellipsoidal sets $\mathcal{E}_{k,\mathcal{N}}^s \subset \mathcal{T}_{\mathcal{N}}^s$ whose boundaries *do* intersect the exclusion zone have a scale factor equal to $\rho_k^{\mathcal{O}_i}$, i.e., their size is limited by the exclusion zone constraint (3.14).

If the spacecraft initial state $X(0) \in \mathcal{T}_{\mathcal{N}}^s$, then, with a suitable choice of δ, constraints are guaranteed to be satisfied at that instant. However, there is no guarantee that constraints will be satisfied for $k > 0$. This deficiency is addressed in the next subsection.

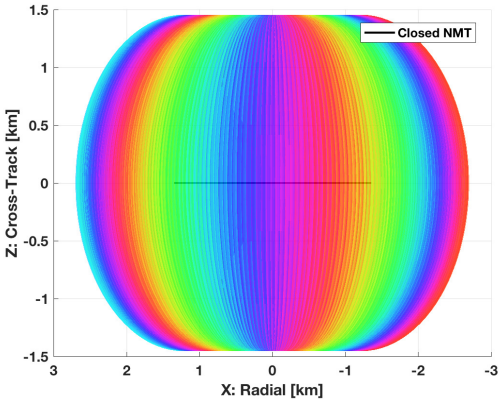
3.2.3 Safe, Positively Invariant Tubes

A SPI tube, $\mathcal{T}_{\mathcal{N}}$, is formed by generating new scale factors, ρ_k, from ρ_k^s such that the property of positive invariance holds and $\rho_k \leq \rho_k^s$ for all $k \in \mathbb{Z}_{[0,k_{max}]}$ (or $k \in \mathbb{Z}_{[0,k_{max}^{\mathcal{N}}]}$). Hence, $\mathcal{T}_{\mathcal{N}} \subseteq \mathcal{T}_{\mathcal{N}}^s$ and the safety of the tube is maintained. Because constraints are guaranteed to be satisfied only if the spacecraft state vector is within the safe ellipsoidal set corresponding to the current controller reference point, the following definitions for positive invariance are used for closed and open NMTs:

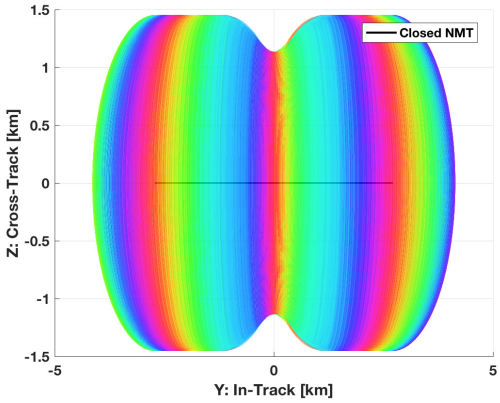


(a)

(b)



(c)



(d)

Figure 3.2: Visualization of the safe tube $\mathcal{T}_{\mathcal{N}}^{s}$ projected onto the position space, $\mathbb{R}^3$

Definition 3.3. *Given a closed NMT $\mathcal{N}$, a tube*

$$\mathcal{T}_{\mathcal{N}} = \bigcup_{k \in \mathbb{Z}_{[0,k_{max}]}} \mathcal{E}_{k,\mathcal{N}} \tag{3.26}$$

is positively invariant with respect to the closed loop dynamics given by (3.7) if there exists a $\delta \in \mathbb{Z}$ such that

$$\begin{gathered} X(k_1) \in \mathcal{E}_{k_1+\delta,\mathcal{N}} \implies X(k_2) \in \mathcal{E}_{k_2+\delta,\mathcal{N}} \\ \text{for all } k_2 \geq k_1,\ k_1 \in \mathbb{Z}_{\geq 0},\ k_2 \in \mathbb{Z}_{\geq 0}, \text{ and all } w(k) \in \mathcal{W}, k \in \mathbb{Z}_{[k_1,k_2]}. \end{gathered} \tag{3.27}$$

Definition 3.4. *Given an open NMT $\bar{\mathcal{N}}$, a tube*

$$\mathcal{T}_{\bar{\mathcal{N}}} = \bigcup_{k \in [0,k_{max}^{\bar{\mathcal{N}}}]} \mathcal{E}_{k,\bar{\mathcal{N}}} \tag{3.28}$$

is positively invariant with respect to the closed loop dynamics given by (3.7) if there exists a $\delta \in \mathbb{Z}$ such that

$$\begin{gathered} X(k_1) \in \mathcal{E}_{k_1+\delta,\bar{\mathcal{N}}} \implies X(k_2) \in \mathcal{E}_{k_2+\delta,\bar{\mathcal{N}}} \\ \text{for all } k_2 \geq k_1,\ k_1+\delta \in \mathbb{Z}_{[0,k_{max}^{\bar{\mathcal{N}}}]},\ k_2+\delta \in \mathbb{Z}_{[0,k_{max}^{\bar{\mathcal{N}}}]}, \text{ and all } w(k) \in \mathcal{W},\ k \in \mathbb{Z}_{[k_1,k_2]}. \end{gathered} \tag{3.29}$$

Remark 3.3. *The difference between the definitions for positive invariance given for closed NMTs (Definition 3.3) and open NMTs (Definition 3.4) is that, in Definition 3.3, the condition (3.27) must hold for all $k_1,\ k_2 \in \mathbb{Z}_{\geq 0}$, i.e., if the spacecraft state vector is within the SPI tube for a closed NMT at any time instant, then it will stay within the tube for all future time instants. In contrast to this, for open NMTs, the condition (3.29) must only hold for $k_1+\delta \in \mathbb{Z}_{[0,k_{max}^{\bar{\mathcal{N}}}]}$, and $k_2+\delta \in \mathbb{Z}_{[0,k_{max}^{\bar{\mathcal{N}}}]}$, i.e., if the spacecraft state vector is within the SPI tube for an open NMT at any time instant, then it will stay within the tube only until the controller reference is set to the last state vector along the open NMT, $X_n(k_{max}^{\bar{\mathcal{N}}})$.*

Remark 3.4. *The Definitions (3.3) and (3.4) for positive invariance can also be written in terms of the state error, defined in (3.9). Note that $X(k) \in \mathcal{E}_{k+\delta,\mathcal{N}}$ implies that $e(k,\delta)^T P e(k,\delta) \leq \rho_{k+\delta}$. Hence, line 1 of (3.27) can be re-written in terms of the error as*

$$e(k_1,\delta)^T P e(k_1,\delta) \leq \rho_{k_1+\delta} \implies e(k_2,\delta)^T P e(k_2,\delta) \leq \rho_{k_2+\delta}, \tag{3.30}$$

and similar for line 1 of (3.29).

3.3 Safe Positively Invariant Tube Construction

In this section, procedures are given to construct SPI tubes for closed and open NMTs. Specifically, Procedures 3.2 and 3.3 in Section 3.3.3 can be used to modify a set of safe scale factors, ρ_k^s, corresponding to a safe tube, to obtain a new set of scale factors, ρ_k, that correspond to a tube that remains safe, but is also positively invariant. These procedures are based on the smallest PI tube, defined in Section 3.3.1, and conditions for positive invariance developed in Section 3.3.2.

3.3.1 Smallest Constant Cross-Section Positively Invariant Tube

Consider the error system (3.11), repeated here for clarity:

$$e^+ = \bar{A}e + Bw. \tag{3.31}$$

Because the matrix $\bar{A}$ is Schur, the error system is Input-to-State Stable (ISS) and the function

$$V(e) = e^T P e, \tag{3.32}$$

where $P = P^T \succ 0$ satisfies the discrete-time Lyapunov inequality (3.22), is an ISS Lyapunov function [86]. In general terms, the ISS property means that the error trajectory corresponding to a sequence of bounded disturbances will remain bounded, and that errors eventually will become small if the disturbances are small, regardless of the initial condition. This implies that, since the disturbance set $\mathcal{W}$ is compact, the error trajectory will converge to and remain within some bounded positively-invariant set (as long as the controller reference point remains along a single NMT). Such positively-invariant sets can be characterized using sub-level sets of the ISS Lyapunov function (3.32).

Consider the following sub-level set of the ISS Lyapunov function (3.32):

$$\Omega(\rho) = \{e \mid e^T P e \leq \rho\}. \tag{3.33}$$

A sub-level set $\Omega(\rho)$ is positively invariant if

$$e \in \Omega(\rho) \implies e^+ \in \Omega(\rho) \text{ for all } w \in \mathcal{W}. \tag{3.34}$$

The minimally sized sub-level set, $\Omega(\rho_{min})$ for which (3.34) holds can be determined as

follows [48, 90]. Define

$$F(e, w, \rho) = \frac{1}{\rho} e^{+T} P e^{+} = \frac{1}{\rho} (\bar{A}e + Bw)^T P (\bar{A}e + Bw). \tag{3.35}$$

Note that (3.34) holds if

$$\max_{e \in \Omega(\rho),\ w \in \mathcal{W}} \quad F(e, w, \rho) \quad = F^*(\rho) \leq 1, \tag{3.36}$$

and therefore, ρ_{min} corresponds to the value for which $F^*(\rho_{min}) = 1$.

Because the disturbance set $\mathcal{W}$ is the convex hull of n_w vertices, the solution to (3.36) can be obtained by considering only $w \in \{w^1,\ w^2, \cdots, w^{n_w}\}$, and therefore replacing the single optimization problem in (3.36) with n_w simpler optimization problems given by

$$\begin{aligned} F_i^*(\rho^i) = \quad & \max_{e} \quad \frac{1}{\rho^i} (\bar{A}e + Bw^i)^T P (\bar{A}e + Bw^i) \\ & \text{subject to} \quad e^T P e \leq \rho^i, \end{aligned} \tag{3.37}$$

$i = 1, 2, \cdots, n_w$, that can be solved using, e.g., MATLAB's *fmincon* function. The value for ρ_{min} is then found using the following iterative procedure:

Procedure 3.1 [Calculation of ρ_{min}]

Inputs: Disturbance set vertices w^i, $i = 1, 2, \cdots, n_w$,
Initial guess for ρ^i,
Shape matrix P,
Convergence tolerance $\epsilon > 0$.

Output: ρ_{min}

1. For each $i = 1, 2, \cdots, n_w$
   (a) Calculate $F^*(\rho^i)$
   (b) If $F^*(\rho^i) < 1 - \epsilon$, decrease the value for ρ^i and go to step (1a),
   (c) If $F^*(\rho^i) > 1 + \epsilon$, increase the value for ρ^i and go to step (1a),
   (d) If $|F^*(\rho^i) - 1| \leq \epsilon$, set $\rho^{i*} = \rho^i$.
2. Set $\rho_{min} = \max\{\rho^{i*},\ i = 1, 2, \cdots, n_w\}$.

End Procedure.

Remark 3.5. *In steps (1b) and (1c) of Procedure 3.1, adjustments to the current value ρ^i can be made via bisections. Note that the calculation of ρ_{min} is independent of the location*

of any obstacles and depends only on the controller gain matrix K (which determines the matrix $\bar{A}$ in the error dynamics) and the disturbance set $\mathcal{W}$. Therefore, the calculation of ρ_{min} can be accomplished offline and uploaded to the satellite. Other methods to calculate ρ_{min} are also available. One of these is discussed later in Remark 3.14.

Remark 3.6. *The parameter ρ^u, i.e., the maximum safe scale factor based on the maximum control constraint (3.12), places an upper-bound on the value for ρ_{min}. Specifically, if ρ_{min} calculated using Procedure 3.1 is larger than ρ^u, then this implies that no minimally sized invariant set exists since the control system is incapable of counteracting the disturbances. For implementation, because disturbances are typically relatively small compared to the maximum control limit, $\rho_{min} \ll \rho^u$.*

As described above, the parameter ρ_{min} corresponds to the smallest ellipsoidal set $e^T P e \leq \rho$ that is positively invariant for the error trajectory. This implies that

$$e(k_1, \delta)^T P e(k_1, \delta) \leq \rho_{min} \implies e(k_2, \delta)^T P e(k_2, \delta) \leq \rho_{min} \text{ for all } k_2 \geq k_1. \tag{3.38}$$

In terms of the state trajectory, the parameter ρ_{min} corresponds to the smallest tube, centered on an NMT, formed as the union of ellipsoidal sets with constant scale factor, i.e., $\rho_k = \rho_{min}$ for $k \in \mathbb{Z}_{[0,k_{max}]}$, that is positively invariant by Definition 3.3 (or Definition 3.4 for open NMTs). This can be seen by writing (3.38) in terms of the state trajectory as

$$\begin{aligned} &(X(k_1) - X_n(k_1 + \delta))^T P (X(k_1) - X_n(k_1 + \delta)) \leq \rho_{min} \implies \\ &(X(k_2) - X_n(k_2 + \delta))^T P (X(k_2) - X_n(k_2 + \delta)) \leq \rho_{min} \text{ for all } k_2 \geq k_1, \end{aligned} \tag{3.39}$$

or, more compactly as

$$X(k_1) \in \mathcal{E}^{\rho_{min}}_{k_1+\delta,\mathcal{N}} \implies X(k_2) \in \mathcal{E}^{\rho_{min}}_{k_2+\delta,\mathcal{N}} \text{ for all } k_2 \geq k_1, \tag{3.40}$$

where

$$\mathcal{E}^{\rho_{min}}_{k,\mathcal{N}} = \{X \mid (X - X_n(k))^T P (X - X_n(k)) \leq \rho_{min}\}. \tag{3.41}$$

Figure 3.3 shows the tube

$$\mathcal{T}^{\rho_{min}}_{\mathcal{N}} = \bigcup_{k \in [0,k_{max}]} \mathcal{E}^{\rho_{min}}_{k,\mathcal{N}} \tag{3.42}$$

projected onto the position space for an example elliptical NMT.

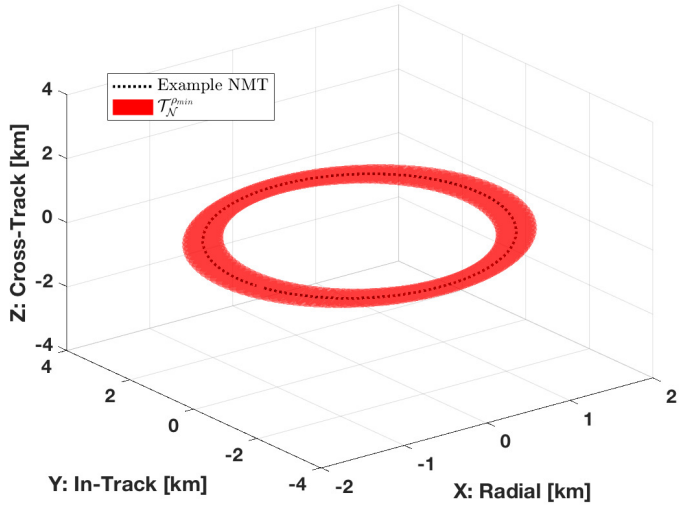


Figure 3.3: Example of the smallest positively-invariant tube, $\mathcal{T}_{\mathcal{N}}^{\rho_{min}}$

3.3.2 Conditions for Positive Invariance

Given the Definitions 3.3 and 3.4 for positive-invariance, it is natural to specify a condition on ellipsoidal set scale factors by looking forward in time, i.e., bounding ρ_{k+1}, the ellipsoidal set size for state vector $X_n(k+1)$, given a value for ρ_k, the ellipsoidal set size for state vector $X_n(k)$, (see, e.g., Lemma 1 in [96], or Lemma 1 in [74]). However, applying such a condition to modify the safe scale factors ρ_k^s may require increasing the value of certain ρ_k^s, resulting in a tube that is positively invariant, but not safe. To ensure that any adjustments to ρ_k^s are downward (and therefore the tube remains safe), a condition for positive invariance is given by looking backward in time, i.e., bounding the ellipsoidal set size at time instant k, (ρ_k), given the ellipsoidal set size at time instant $k+1$, (ρ_{k+1}):

Theorem 3.1. [Conditions for Positive Invariance]
The tube $\mathcal{T}_{\mathcal{N}}$ is positively-invariant if

$$\rho_k \leq \rho_{k+1} + d(\rho_{k+1}) \quad \text{for all} \quad k \in \mathbb{Z}_{\geq 0}, \tag{3.43}$$

where

$$\begin{aligned} d(\rho_{k+1}) = \quad & \min_{e^+,\, w \in \mathcal{W}} \quad e^T P e - e^{+T} P e^+ \\ & \textit{subject to} \quad e^{+T} P e^+ = \rho_{k+1}, \\ & \qquad\qquad\;\; e = \bar{A}^{-1}(e^+ - Bw), \end{aligned} \tag{3.44}$$

and

$$\rho_{k+1} \geq \rho_{r0} \text{ for all } k \in \mathbb{Z}_{\geq 0}, \tag{3.45}$$

where

$$\rho_{r0} = \max_{w \in \mathcal{W}} w^T B^T P B w. \tag{3.46}$$

The following remarks provide a description of some of the parameters defined in the conditions of Theorem 3.1.

Remark 3.7. *The parameter $d(\rho_{k+1})$ in (3.43) and (3.44) defines the largest allowable increase from a given ρ_{k+1} to the preceding ρ_k.*

Remark 3.8. *The parameter ρ_{r0} in (3.45) and (3.46) corresponds to the scale factor for the smallest ellipsoidal set, with shape matrix P, that contains $\mathcal{R}_e(1, 0, \mathcal{W})$, i.e., the 1-step forward reachable set for the error system starting from $e = 0$. Thus, if (3.45) does not hold, it is impossible for the tube to be positively invariant because even if the spacecraft started directly on the NMT at time-instant k, i.e., $e(k) = 0$, it would not be guaranteed to be within the tube at time-instant $k + 1$.*

Proof. Define the following sets:

$$\begin{aligned} D_1^+ = &\ \{e^+ \mid e^{+T} P e^+ = \rho_{k+1}\}, \\ D_1 = &\ \{e \mid \exists\, e^+ \in D_1^+, \text{ and } \exists\, w \in W, \text{ such that } e = \bar{A}^{-1}(e^+ - Bw)\}, \\ D_2 = &\ \{e \mid e^T P e \leq \rho_k\}, \\ D_2^+ = &\ \{e^+ \mid \exists\, e \in D_2, \text{ and } \exists\, w \in W, \text{ such that } e^+ = \bar{A} e + Bw\}, \end{aligned} \tag{3.47}$$

and consider the following notation:

$$e_1^+ \in D_1^+,\ e_1 \in D_1,\ e_2 \in D_2, \text{ and } e_2^+ \in D_2^+. \tag{3.48}$$

Remark 3.9. *The set D_1^+ is the set of all state error vectors on the boundary of an ellipsoidal set with scale factor ρ_{k+1}. The set D_1 is the 1-step backward reachable set of the error dynamics, from the set D_1^+, i.e., $D_1 = \mathcal{R}_e(-1, D_1^+, \mathcal{W})$. The set D_2 is the set of all error vectors included in an ellipsoidal set with scale factor ρ_k, and the set D_2^+ is the 1-step forward reachable set of the error dynamics from the set D_2, i.e., $D_2^+ = \mathcal{R}_e(1, D_2, \mathcal{W})$.*

The proof is completed in three main steps. These main steps are summarized here, and the detailed arguments are provided below. As an initial step, the requirement (3.27) in Definition 3.4 for positive invariance is rewritten in terms of the sets and notation defined

in (3.47) and (3.48). This rewritten requirement for positive invariance is given by

$$e_2^{+T} P e_2^+ \leq e_1^{+T} P e_1^+. \tag{3.49}$$

The requirement (3.49) is then shown to hold using two steps. Firstly, conditions (3.43) and (3.44) from Theorem 3.1 are used to derive a relationship between vectors $e_2 \in D_2$ and $e_1 \in D_1$, showing that any vector e_2 can be written as a "shortened" version of a vector e_1, i.e., for any vector $e_2 \in D_2$, there exists an $e_1 \in D_1$ such that

$$e_2 = \lambda e_1, \ \lambda \in [0, 1]. \tag{3.50}$$

Finally, using arguments based on the linear error system and condition (3.45) in Theorem 3.1, it is shown that (3.50) implies (3.49), and therefore the tube is positively invariant. The detailed arguments based on these major steps are as follows.

Without loss of generality, in (3.27), let $\delta = 0$ and $k_2 = k_1 + 1$. Suppose $X(k) \in \mathcal{E}_{k,\mathcal{N}}$, i.e., $X(k) - X_n(k) = e_2 \in D_2$. For positive invariance to hold, i.e., condition (3.27), it must hold that $X(k+1) \in \mathcal{E}_{k+1,\mathcal{N}}$, or, equivalently, that

$$e_2^{+T} P e_2^+ \leq \rho_{k+1}. \tag{3.51}$$

Noting that $e_1^{+T} P e_1^+ = \rho_{k+1}$, requirement (3.51) is re-written as

$$e_2^{+T} P e_2^+ \leq e_1^{+T} P e_1^+. \tag{3.52}$$

Hence, if (3.52) holds, the tube is positively invariant.

Next, we show that (3.52) holds if (3.43), (3.44) and (3.45) hold. Firstly, (3.43) and (3.44) are used to derive a relationship between $e_2 \in D_2$ and $e_1 \in D_1$. Then, this relationship is used, along with (3.45), to show that (3.52) holds.

Suppose (3.43) and (3.44) hold. From (3.44) it follows that

$$d(\rho_{k+1}) \leq e_1^T P e_1 - \rho_{k+1}, \tag{3.53}$$

which is equivalently written as

$$e_1^T P e_1 \geq \rho_{k+1} + d(\rho_{k+1}), \tag{3.54}$$

and, combining (3.54) with (3.43) yields

$$e_1^T P e_1 \geq \rho_{k+1} + d(\rho_{k+1}) \geq \rho_k. \tag{3.55}$$

From (3.47) and (3.48), $e_2^T P e_2 \leq \rho_k$. This, along with (3.55) yields the following relationship between e_1 and e_2:

$$e_2^T P e_2 \leq e_1^T P e_1. \tag{3.56}$$

Therefore, for any vector $e_2 \in D_2$, there exists an $e_1 \in D_1$ such that e_2 can be written as

$$e_2 = \lambda e_1, \ \lambda \in [0, 1]. \tag{3.57}$$

Using the error dynamics $e_{(\cdot)}^+ = \bar{A} e_{(\cdot)} + Bw$, and (3.57), e_2^+ is re-written in terms of e_1^+, i.e.,

$$\begin{aligned} e_2^+ &= \bar{A} e_2 + Bw \\ &= \lambda \bar{A} e_1 + Bw \\ &= \lambda \bar{A}(\bar{A}^{-1} e_1^+ - \bar{A}^{-1} Bw) + Bw, \\ &= \lambda e_1^+ + (1 - \lambda) Bw. \end{aligned} \tag{3.58}$$

Substituting (3.58), and the relationship $e_1^{+T} P e_1^+ = \rho_{k+1}$ into the requirement for positive invariance given by (3.52) and rearranging terms yields

$$(\lambda^2 - 1)\rho_{k+1} - 2\lambda(\lambda - 1) w^T B^T P e_1^+ + (\lambda - 1)^2 w^T B^T P B w \leq 0. \tag{3.59}$$

Note that if $\lambda = 1$, (3.59), and therefore (3.52), are trivially satisfied.

Consider now the case where $\lambda \in [0, 1)$. Each side of (3.59) is divided by $(\lambda - 1) < 0$ and terms are rearranged to yield

$$(\lambda + 1)\rho_{k+1} + (\lambda - 1) w^T B^T P B w \geq 2\lambda w^T B^T P e_1^+. \tag{3.60}$$

Consider the following proposition, used to establish an upper-bound for the RHS of (3.60):

Proposition 3.1. *The following holds:*

$$\max_{e_1^+ \in D_1^+} \ w^T B^T P e_1^+ = \sqrt{\rho_{k+1}} \sqrt{w^T B^T P B w}$$

Proof. See Appendix E.1.

Based on Proposition 3.1, to show that (3.60) holds, it is sufficient to show that the

following expression holds:

$$(\lambda+1)\rho_{k+1} + (\lambda-1)w^T B^T P B w \geq 2\lambda\sqrt{\rho_{k+1}}\sqrt{w^T B^T P B w}. \tag{3.61}$$

To show (3.61) holds, suppose condition (3.45) holds. This requirement can be written as

$$\rho_{k+1} = w^T B^T P B w + \nu, \tag{3.62}$$

where $\nu \geq 0$. By substituting (3.62) into (3.61) and manipulating terms, (3.61) becomes

$$2\lambda w^T B^T P B w + \nu(\lambda+1) \geq 2\lambda\sqrt{(w^T B^T P B w)^2 + \nu w^T B^T P B w}. \tag{3.63}$$

Note that all terms in (3.63) are non-negative. Squaring both sides of (3.63), and cancelling terms results in

$$4\lambda\nu w^T B^T P B w + \nu^2(\lambda+1)^2 \geq 0. \tag{3.64}$$

Because $\nu \geq 0$, $\lambda \in [0,1)$, and $P = P^T \succ 0$, it is clear that (3.64) holds. Therefore, the tube $\mathcal{T}_{\mathcal{N}}$ is positively invariant. □

Remark 3.10. *For open NMTs, the conditions for positive invariance and the proof of Theorem 3.1 also hold. The only change is that conditions (3.43) and (3.45) must hold only for all $k \in \mathbb{Z}_{[0,k^{\mathcal{N}}_{max}-1]}$ rather than for all $k \in \mathbb{Z}_{\geq 0}$.*

3.3.3 Procedures for SPI Tube Construction

SPI tubes are constructed by applying the conditions of Theorem 3.1 to modify the safe scale factors ρ^s_k and obtain a set of scale factors ρ_k that yield a tube that is still safe, but also positively invariant. Below, procedures are given to generate these scale factors for both closed NMTs and open NMTs.

3.3.3.1 Procedure for closed NMTs

The procedure for closed NMTs (Procedure 3.2 presented below) is based on the condition for positive invariance in Theorem 3.1, and also on the parameter ρ_{min} defined in Section 3.3.1. Because it is impossible for the spacecraft to reach any ellipsoidal set with $\rho < \rho_{min}$ from a larger ellipsoidal set, if $\rho_{k+1} < \rho_{min}$, then $d(\rho_{k+1})$ defined in (3.44) is negative. This intuitive result is confirmed through simulations. Hence, for a closed NMT, if any $\rho^s_k < \rho_{min}$, then no SPI tube exists since it is not possible for (3.43) to hold for all $k \in \mathbb{Z}_{\geq 0}$ due to the periodicity, i.e., k and $k + (k_{max}+1)$ correspond to the same state vector along

the NMT. This fact is reflected in step 1 of Procedure 3.2 below.

Procedure 3.2 [Calculation of SPI scale factors for closed NMTs]

Inputs: Set of safe scale factors, ρ_k^s, $k \in [0, k_{max}]$,
Scale factor for minimally sized positively invariant set, ρ_{min},
Disturbance set $\mathcal{W}$.

Output: Set of SPI scale factors ρ_k, $k \in [0, k_{max}]$.

1. If $\rho_k^s < \rho_{min}$ for any $k \in [0, k_{max}]$, set $\rho_k = 0$ for all $k \in [0, k_{max}]$ and end procedure.

2. Start at k such that
$$k + 1 = \underset{k \in [0, k_{max}]}{\text{argmin}} \ (\rho_k^s) \tag{3.65}$$
and set $\rho_{k+1} = \rho_{k+1}^s$. Note that if multiple k satisfy (3.65), any such k can be chosen as the starting location.

3. Determine $d(\rho_{k+1})$ and set
$$\rho_k = \begin{cases} \rho_k^s \text{ if } \rho_k^s \leq \rho_{k+1} + d(\rho_{k+1}), \\ \rho_{k+1} + d(\rho_{k+1}) \text{ otherwise.} \end{cases}$$

4. Increment $k = k - 1$ and repeat step 3. When $k = -1$, set $k = k_{max}$ and continue until returning to the starting index.

End Procedure.

Remark 3.11. *If there exists a $\rho_k^s < \rho_{min}$ then travel along the NMT is not safe. In this case, the NMT could be removed the virtual net constructed Section 3.4. Alternatively, setting all $\rho_k = 0$ as in step 1 above, guarantees that such an NMT will not be included in any maneuvers planned using the virtual net. Hence, as long as the spacecraft does not start on an unsafe NMT, safety is guaranteed. For completeness, an additional step could be included after step 1 to account for the condition (3.45), stating "if $\rho_k^s < \rho_{r0}$ for any $k \in [0, k_{max}]$, set $\rho_k = 0$ for all $k \in [0, k_{max}]$ and end procedure." In practice, this additional step is not needed since simulations show $\rho_{min} > \rho_{r0}$.*

Figure 3.4 illustrates the implementation of Procedure 3.2 for an example closed NMT. The dashed blue line shows the safe scale factors, and the solid red line denotes the set of SPI scale factors determined using Procedure 3.2. The upper-limit on the scale factors due to the control constraint (3.12) and the minimum scale factor for positive invariance, ρ_{min}, for this example are shown as dashed red and magenta lines, respectively.

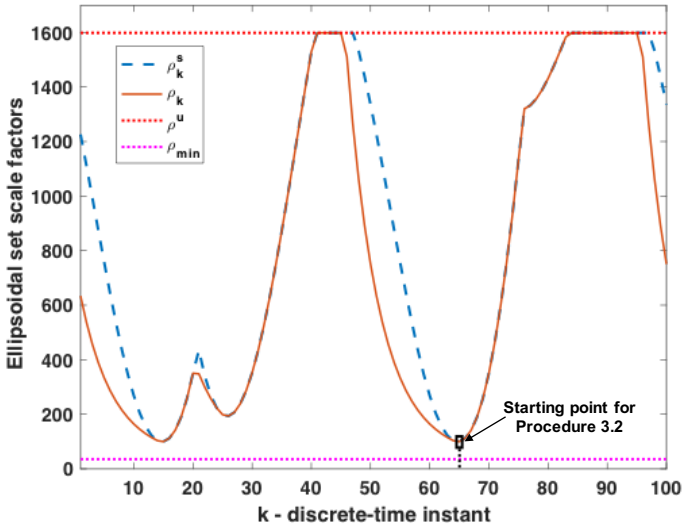


Figure 3.4: Illustration showing use of Procedure 3.2 to generate SPI scale factors ρ_k from safe scale factors ρ_k^s for an example closed NMT

3.3.3.2 Procedure for open NMTs

The procedure to generate SPI scale factors for open NMTs differs from Procedure 3.2 above for closed NMTs for two reasons. Firstly, because the NMT is open, the condition (3.43) must only hold for $k \in [0,\ k_{max}^{\bar{\mathcal{N}}} - 1]$. Secondly, it is possible to form an SPI tube for an open NMT with some $\rho_k < \rho_{min}$. This second reason is because a spacecraft travelling along an open NMT only visits each state vector along the NMT once. Therefore, unlike in the closed NMT case, there is no requirement that a spacecraft can return to the ellipsoidal set for a given state vector along an open NMT.

Procedure 3.3 [Calculation of SPI scale factors for open NMTs]

Inputs: Set of safe scale factors, ρ_k^s, $k \in \mathbb{Z}_{[0,k_{max}^{\bar{\mathcal{N}}}]}$,
Scale factor ρ_{r0} defined in (3.46),
Disturbance set $\mathcal{W}$.

Output: Set of SPI scale factors ρ_k, $k \in \mathbb{Z}_{[k_{min},k_{max}^{\bar{\mathcal{N}}}]}$, $0 \leq k_{min} \leq k_{max}^{\bar{\mathcal{N}}}$.

1. Set $k_{min} = 0$.

2. Define a set $\mathcal{K}$ containing the parameter $k_{max}^{\bar{\mathcal{N}}}$ and the indices of strict local minima of ρ_k^s:

$$\mathcal{K} = \{k_{max}^{\bar{\mathcal{N}}},\ k \in \mathbb{Z}_{[1,k_{max}^{\bar{\mathcal{N}}}-1]} \mid \rho_k^s < \rho_{k-1}^s \text{ and } \rho_k^s < \rho_{k+1}^s\}.$$

3. Form a vector v from the set $\mathcal{K}$ as follows:

   (a) Initialize an index variable $c = 1$,

   (b) $v^c = \max\{\underset{k\in\mathcal{K}}{\text{argmin}}(\rho_k^s)\}$,

   (c) If $v^c = k_{max}^{\bar{\mathcal{N}}}$, go to step 4,

   (d) Increment $c = c + 1$,

   (e) Set $v^c = \max\{\underset{k\in\mathcal{K},k>v^{c-1}}{\text{argmin}}(\rho_k^s)\}$, go to step 3c,

4. Determine ρ_k from ρ_k^s as follows:

   (a) Initialize an index variable $j = 1$,

   (b) Set $k + 1 = v^j$, and set $\rho_{k+1} = \rho_{k+1}^s$,

   (c) If $\rho_{k+1} < \rho_{r0}$, set $k_{min} = k + 1$, set $\rho_{\bar{k}} = 0$ for all $\bar{k} < k + 1$ and go to step 4g.

   (d) Determine $d(\rho_{k+1})$ and set

$$\rho_k = \begin{cases} \rho_k = \rho_k^s \text{ if } \rho_k^s \leq \rho_{k+1} + d(\rho_{k+1}), \\ \rho_k = \rho_{k+1} + d(\rho_{k+1}) \text{ otherwise} \end{cases}$$

   (e) If $\rho_k < \rho_{r0}$, set $k_{min} = k$, set $\rho_{\bar{k}} = 0$ for all $\bar{k} < k$ and go to step 4g.

   (f) Increment $k = k - 1$, if $k > -1$, go to step 4d,

   (g) if $j < c$, increment $j = j + 1$, go to step 4b,

   (h) if $j = c$, end procedure.

End Procedure.

Remark 3.12. *Note that superscripts appended to the vector v denote the element of the vector, i.e., v^1 is the first element of v. Procedure 3.3 results in a set of SPI scale factors for the portion of the NMT from $X_n(k_{min})$ to $X_n(k_{max}^{\bar{\mathcal{N}}})$. Steps 2 and 3 in Procedure 3.3 determine the starting points from which adjustments to ρ_k^s are made. Specifically, the first starting point, v^1, corresponds to the index of the smallest strict local minima of ρ_k^s (or $k_{max}^{\bar{\mathcal{N}}}$), the second starting point, v^2, corresponds to the index of the smallest strict local minima larger than v^1 (or $k_{max}^{\bar{\mathcal{N}}}$), etc. Step 4 in Procedure 3.3 makes adjustments to the safe scale factors such that conditions (3.43) and (3.45) hold. Specifically, step 4d is used to enforce condition (3.43) while steps 4c and 4e account for condition (3.45) by defining a parameter k_{min} and setting all scale factors ρ_k for $k < k_{min}$ equal to 0 if (3.45) is found to be violated. In this case, travel along the NMT in question is not safe for state vector*

corresponding to $k \in \mathbb{Z}_{[0,k_{min}]}$. Setting all of these scale factors equal to zero ensures that this portion of the NMT will not be included in any planned maneuvers.

Figure 3.5 illustrates the implementation of Procedure 3.3 for two example open NMTs. In Figure 3.5a, an example NMT is shown for which k_{min} is determined to be 0. Therefore, Procedure 3.3 yields a set of SPI scale factors, ρ_k, for $k \in \mathbb{Z}_{[0,k_{max}^{\bar{N}}]}$. In Figure 3.5b, an example NMT is shown for which $k_{min} = 7$. Therefore, for this NMT, Procedure 3.3 yields a set of SPI scale factors, ρ_k, for $k \in \mathbb{Z}_{[7,k_{max}^{\bar{N}}]}$.

3.3.3.3 Calculation of $d(\rho_{k+1})$

Implementation of Procedures 3.2 and 3.3 require calculation of the parameter $d(\rho_{k+1})$ for multiple values of ρ_{k+1}. The calculation of $d(\rho_{k+1})$ can be accomplished quickly using straightforward calculations as follows. Firstly, because the disturbance set $\mathcal{W}$ is a bounded polyhedron, the calculation of $d(\rho_{k+1})$ can be accomplished by considering only the vertices of the disturbance set, i.e., taking $w = w^i,\ i = 1, 2, \cdots, n_w$, and therefore replacing the optimization problem in (3.44) with n_w optimization problems given by

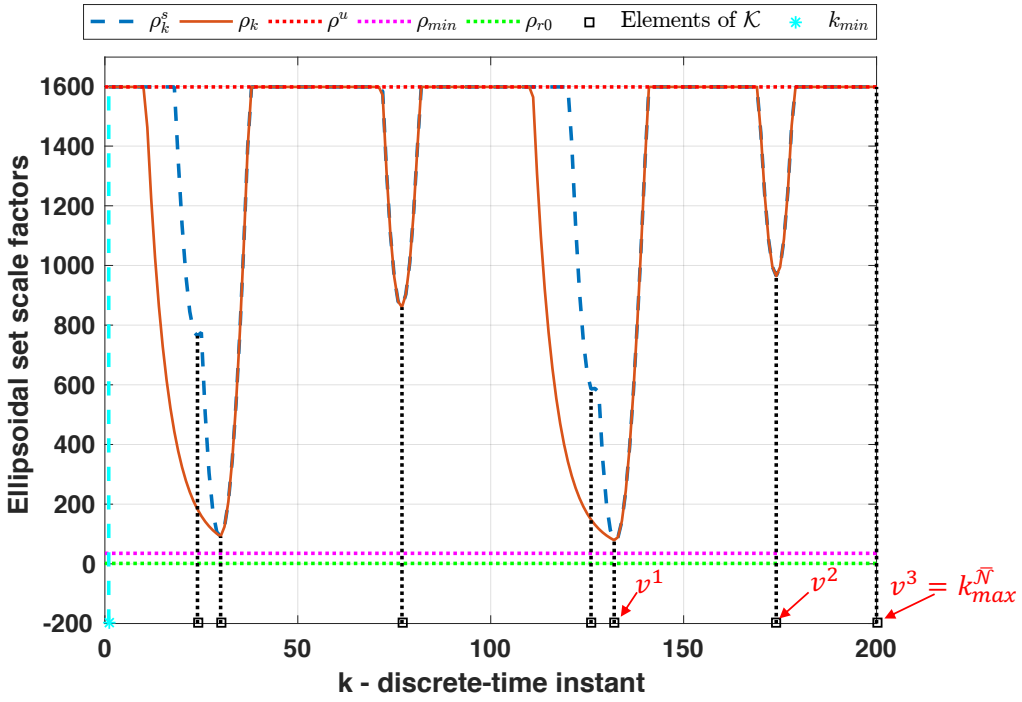
$$\begin{aligned} d^i(\rho_{k+1}) = \quad & \min_{e^+} \quad e^T P e - e^{+T} P e^+ \qquad i = 1, 2, \cdots, n_w, \\ & \text{subject to} \quad e^{+T} P e^+ = \rho_{k+1}, \\ & \qquad\qquad\ e = \bar{A}^{-1}(e^+ - B w^i). \end{aligned} \tag{3.66}$$

The solution to each of the n_w Quadratically Constrained Quadratic Program (QCQP) in (3.66) can be obtained by reducing the problem to a scalar root finding problem using the method described in [97]. Details are included in Appendix C.1. After each $d^i(\rho_{k+1})$ is obtained, then

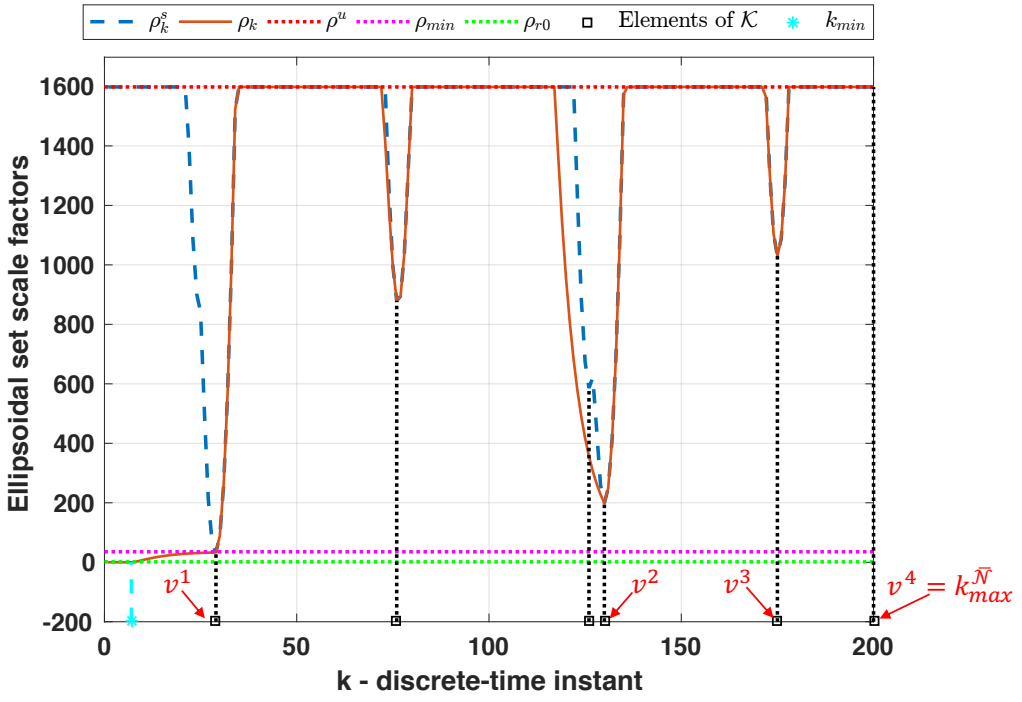
$$d(\rho_{k+1}) = \min_{i=1,2,\cdots,n_w} d^i(\rho_{k+1}). \tag{3.67}$$

Remark 3.13. *It is also possible to pre-calculate $d(\rho_{k+1})$ for multiple values of ρ_{k+1} using the the method described above, and then fit a polynomial to this data. For on-line implementation, rather than obtaining $d(\rho_{k+1})$ through solutions to (3.66), an approximation for $d(\rho_{k+1})$ can be obtained using the curve-fit. Using this approximation can significantly reduce the time require to generate SPI tubes, with certain trade-offs. This is discussed further in Section 3.7.*

Remark 3.14. *The parameter ρ_{min} can also be obtained through the calculation of $d(\rho_{k+1})$. Specifically, ρ_{min} corresponds to the value for which $d(\rho_{min}) = 0$. Hence, ρ_{min} can be obtained by calculating $d(\rho)$ for multiple values of ρ and setting ρ_{min} equal to the value for*



(a) An example with all $\rho_k > \rho_{min} > \rho_{r0}$, $k_{min} = 0$



(b) An example with some $\rho_k < \rho_{r0} < \rho_{min}$, $k_{min} = 7$

Figure 3.5: Illustrations showing use of Procedure 3.3 to generate SPI scale factors ρ_k from safe scale factors ρ_k^s for two example open NMTs

which $d(\rho_{min}) = 0$. Simulations confirm that calculating ρ_{min} using this method produces the same value as the method described in Procedure 3.1, in less time.

3.4 Virtual Net for Safe Maneuver Planning

Using the SPI tubes generated in Section 3.3, a "virtual net" is formed. Using this virtual net, safe maneuvers are planned to transition a spacecraft from an initial NMT to a desired final NMT. The virtual net is formed from a given a set of NMTs,

$$\mathcal{M} = \{\mathcal{N}_1,\ \mathcal{N}_2, \cdots, \mathcal{N}_{l_c},\ \bar{\mathcal{N}}_1,\ \bar{\mathcal{N}}_2 \cdots, \bar{\mathcal{N}}_{l_o}\}, \tag{3.68}$$

where $\mathcal{N}_1$ through $\mathcal{N}_{l_c}$ are closed and $\bar{\mathcal{N}}_1$ through $\bar{\mathcal{N}}_{l_o}$ are open. This virtual net is a directed graph, $\mathcal{G}(\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ is a set containing all nodes (vertices) in the virtual net, and $\mathcal{A} \subset \mathcal{V} \times \mathcal{V}$ is the edge set containing pairs of adjacent nodes.

The node set $\mathcal{V}$ contains two types of nodes:

1. Each closed NMT is a single node. These nodes are denoted by $\mathcal{N}_i \in \mathcal{V}$, $i = 1, 2, \cdots, l_c$.

2. Each state vector along each open NMT is a single node. The node for state vector $X_{ni}(k) \in \bar{\mathcal{N}}_i$ is denoted by $(\bar{\mathcal{N}}_i,\ k) \in \mathcal{V}$, where $i = 1, \cdots, l_o$ and $k \in \mathbb{Z}_{[0, k_{max}^{\bar{\mathcal{N}}_i}]}$.

Note that a single node is sufficient for each closed NMT due to the periodicity, i.e., if the spacecraft is able to reach a single state vector along a closed NMT, it is able to reach all state vectors along the NMT given sufficient time. However, for open NMTs, this property does not hold. Therefore, a separate node is required to represent each state vector along open NMTs. Each node in the virtual net is assigned a number between 1 and p, where p is the total number of nodes in the virtual net,

$$p = l_c + \sum_{i=1}^{l_o} \left[k_{max}^{\bar{\mathcal{N}}_i} + 1 \right]. \tag{3.69}$$

The edge set $\mathcal{A} \subset \mathcal{V} \times \mathcal{V}$ contains all pairs of adjacent nodes in the virtual net, i.e., for nodes $n \in \mathbb{Z}_{[1,p]}$ and $m \in \mathbb{Z}_{[1,p]}$, if node n is adjacent to node m, then $(n, m) \in \mathcal{A}$. The edge set $\mathcal{A}$ is determined by adjacency rules and calculations described in the following subsections.

The remainder of this section is organized as follows. Adjacency definitions are presented in Section 3.4.1, and calculations used to check whether a given pair of nodes are

adjacent are described in Section 3.4.2. Section 3.4.3 defines and describes the calculation of an adjacency matrix and connection array used to represent the virtual net and plan safe transfers between nodes. Specifically, the adjacency matrix is used to determine if a safe transfer is possible, and the connection array provides the starting location and initial controller reference point used to actually execute the transfer. After the adjacency matrix and connection array have been calculated, efficient graph search algorithms, such as Dijkstra's algorithm, [37], are used to generate safe maneuvers connecting specified nodes in the virtual net via a sequence of adjacent nodes. The generation of these maneuvers is described in Section 3.4.4.

3.4.1 Virtual Net Adjacency: Definitions

Two nodes in the virtual net are adjacent if it is possible to execute a safe transfer from one node to the other. In a virtual net consisting of both closed and open NMTs, there are four types of adjacency to consider:

Adjacency Type (A1): Closed NMT node to closed NMT node,
Adjacency Type (A2): Open NMT node to closed NMT node,
Adjacency Type (A3): Closed NMT node to open NMT node,
Adjacency Type (A4): Open NMT node to open NMT node.

Definitions for each type of adjacency are provided below. Each type of adjacency involves an "origin" NMT, from a which transfer is initiated, and a "destination" NMT, at which the transfer completes. In the definitions and calculations below, a subscript i is used to denote the origin NMT, and a subscript j is used to denoted the destination NMT. Furthermore, discrete-time indices with a subscript i, i.e., k_i, denote the discrete-time indices for state vectors along the origin NMT ($X_{ni}(k_i)$). Discrete-time indices with a subscript j, i.e., k_j and $\hat{k}_j$, correspond to state vectors $X_{nj}(k_j)$ and $X_{nj}(\hat{k}_j)$ along the destination NMT. Key notation and parameters used in the adjacency definitions and calculations below are summarized in Table 3.1.

The adjacency definitions for types (A1) and (A2), i.e., adjacency to closed NMTs, are based on two observations. Firstly, a safe transfer to a desired closed NMT $\mathcal{N}_j$ can be generated from the current state vector $X(k)$ by setting the controller reference to $X_{nj}(k+\delta)$ along the desired NMT for any $X_{nj}(k+\delta)$ that satisfies $X(k) \in \mathcal{E}_{k+\delta,\mathcal{N}_j}$. Secondly, due to disturbances, the spacecraft state vector is only guaranteed to converge to the set $\mathcal{E}^{\rho_{min}}_{k,\mathcal{N}_j}$ [defined in (3.41)] about any $X_{nj}(k)$ along an NMT. The definitions for adjacency

Table 3.1: Notation used for adjacency definitions and calculations

Symbol	Description
$\mathcal{N}_i$ or $\bar{\mathcal{N}}_i$	Origin NMT
k_i and $X_{ni}(k_i)$	Discrete-time index and state vector along the origin NMT at which a potential transfer begins
$\mathcal{N}_j$ or $\bar{\mathcal{N}}_j$	Destination NMT
$\hat{k}_j$ and $X_{nj}(\hat{k}_j)$	Discrete-time index and state vector along the destination NMT used as the initial controller reference point for a potential transfer
k_j and $X_{nj}(k_j)$	Discrete-time index and state vector along the destination NMT corresponding to the destination node for adjacency types (A3) and (A4)
$\bar{k}$	Final discrete-time instant along a potential transfer trajectory, starting at $k = 0$.

types (A1) and (A2) are give below. Figure 3.6 illustrates the parameters used in these definitions.

Definition 3.5. [Adjacency Type (A1)] *Node $\mathcal{N}_i$ is adjacent to node $\mathcal{N}_j$ if there exists $k_i,\ \hat{k}_j \in \mathbb{Z}_{[0,\ k_{max}]}$, and $\alpha > 0$ such that*

$$\mathcal{E}_{k_i,\mathcal{N}_i}^{\rho_{min}+\alpha} \subset \mathcal{E}_{\hat{k}_j,\mathcal{N}_j}, \tag{3.70}$$

where

$$\mathcal{E}_{k_i,\mathcal{N}_i}^{\rho_{min}+\alpha} = \{X \mid (X - X_{ni}(k_i))^T P (X - X_{ni}(k_i)) \leq \rho_{min} + \alpha\}. \tag{3.71}$$

Definition 3.6. [Adjacency Type (A2)] *Node $(\bar{\mathcal{N}}_i, k_i)$ is adjacent to node $\mathcal{N}_j$ if there exists $\hat{k}_j \in \mathbb{Z}_{[0,\ k_{max}]}$ and $\alpha > 0$ such that*

$$\mathcal{E}_{k_i,\bar{\mathcal{N}}_i}^{\rho_{min}+\alpha} \subset \mathcal{E}_{\hat{k}_j,\mathcal{N}_j}. \tag{3.72}$$

Remark 3.15. *The requirement (3.70) could be replaced with*

$$\mathcal{E}_{k_i,\mathcal{N}_i}^{\rho_{min}} \subset \text{int}\big(\mathcal{E}_{\hat{k}_j,\mathcal{N}_j}\big), \tag{3.73}$$

where int$(\cdot)$ *denotes the interior of a set. If (3.73) holds, this implies that there exists an $\alpha > 0$ such that (3.70) holds. A similar replacement could be used for requirement (3.72). The choice to use requirements (3.70) and (3.72) in the adjacency definitions was made*

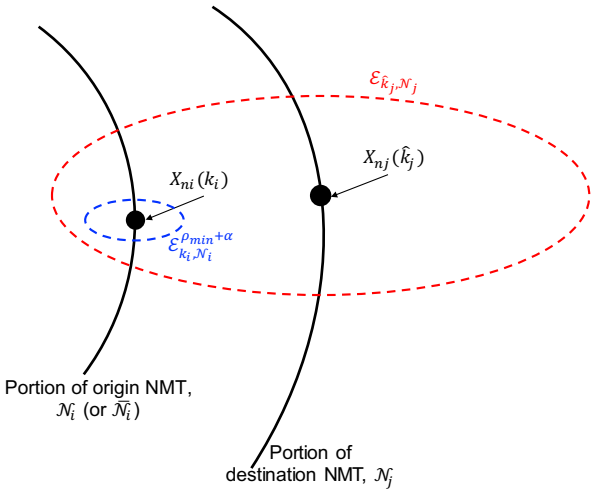


Figure 3.6: Illustration of the parameters used in definitions for adjacency to closed NMT nodes

because a) this simplifies the adjacency calculations described in Section 3.4.2, and b) this facilitates the development of switching criteria used to plan maneuvers in the virtual net. For implementation, a single small $\alpha > 0$ is chosen and used for all calculations. In general, choosing α to be larger will result in a virtual net with fewer pairs of adjacent NMTs, limiting the manuevers that can be planned between NMTs. However, larger α may allow for the generation of shorter (in terms of time) trajectories between given NMTs.

In Definitions 3.5 and 3.6, the requirements (3.70) and (3.72) ensure that, from all state vectors near $X_{ni}(k_i)$ along the origin NMT, the state vector $X_{nj}(\hat{k}_j)$ can be selected as the controller reference point, and constraints (3.12) and (3.14) will be satisfied at the current time-instant. Furthermore, because the tube $\mathcal{T}_{\mathcal{N}_j}$ is SPI, these constraints are guaranteed to be satisfied for all future time-instants, as long as the controller reference points remain along $\mathcal{N}_j$. Finally, because the NMT $\mathcal{N}_j$ is closed, the spacecraft state vector is guaranteed to converge "close to" any state vector along $\mathcal{N}_j$ at some future time-instant. Specifically, it is guaranteed that for each $k_j \in [0, k_{max}]$ and for any $\alpha > 0$, there exists a k such that the spacecraft state vector $X(k)$ satisfies $X(k) \in \mathcal{E}^{\rho_{min}+\alpha}_{k_j,\mathcal{N}_j}$. However, if the destination NMT is open, this property does not hold. Because of this, for adjacency types (A3) and (A4) where the destination NMT is open, an additional requirement is included in the definitions for adjacency provided below. Figure 3.7 illustrates the parameters used in Definitions 3.7 and 3.8.

Definition 3.7. [Adjacency Type (A3)] *Node $\mathcal{N}_i$ is adjacent to node $(\bar{\mathcal{N}}_j, k_j)$ if there exists*

$k_i \in \mathbb{Z}_{[0,\, k_{max}]}$, $\hat{k}_j \in \mathbb{Z}_{[0,\, k_{max}^{\bar{\mathcal{N}}_j}]}$, $\bar{k} \in \mathbb{Z}_{\geq 0}$ *and* $\alpha > 0$ *such that*

$$\mathcal{E}_{k_i,\mathcal{N}_i}^{\rho_{min}+\alpha} \subset \mathcal{E}_{\hat{k}_j,\bar{\mathcal{N}}_j}, \tag{3.74}$$

and

$$\mathcal{R}_X(\bar{k},\ \mathcal{E}_{k_i,\mathcal{N}_i}^{\rho_{min}+\alpha},\ X_{nj}(\hat{k}_j),\ \mathcal{W}) \subset \mathcal{E}_{k_j,\bar{\mathcal{N}}_j}^{\rho_{min}+\alpha}, \tag{3.75}$$

where $\mathcal{R}_X(\bar{k},\ \mathcal{E}_{k_i,\mathcal{N}_i}^{\rho_{min}+\alpha},\ X_{nj}(\hat{k}_j),\ \mathcal{W})$ *is the reachable set at time-instant* $\bar{k}$ *for the closed-loop system (3.7), starting from the set* $\mathcal{E}_{k_i,\mathcal{N}_i}^{\rho_{min}+\alpha}$*, and with initial controller reference point* $X_{nj}(\hat{k}_j)$*, introduced in Definition 3.1.*

Definition 3.8. [Adjacency Type (A4)] *Node* $(\bar{\mathcal{N}}_i, k_i)$ *is adjacent to node* $(\bar{\mathcal{N}}_j, k_j)$ *if there exists* $\hat{k}_j \in \mathbb{Z}_{[0,\, k_{max}^{\bar{\mathcal{N}}_j}]}$, $\bar{k} \in \mathbb{Z}_{\geq 0}$ *and* $\alpha > 0$ *such that*

$$\mathcal{E}_{k_i,\bar{\mathcal{N}}_i}^{\rho_{min}+\alpha} \subset \mathcal{E}_{\hat{k}_j,\bar{\mathcal{N}}_j}, \tag{3.76}$$

and

$$\mathcal{R}_X(\bar{k},\ \mathcal{E}_{k_i,\bar{\mathcal{N}}_i}^{\rho_{min}+\alpha},\ X_{nj}(\hat{k}_j),\ \mathcal{W}) \subset \mathcal{E}_{k_j,\bar{\mathcal{N}}_j}^{\rho_{min}+\alpha}. \tag{3.77}$$

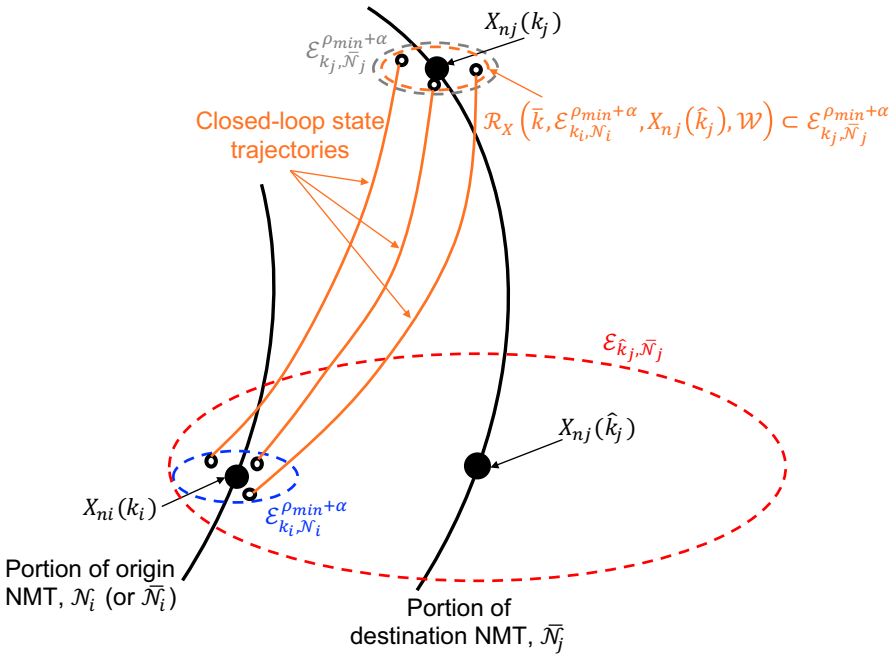


Figure 3.7: Illustration of the parameters used in definitions for adjacency to open NMT nodes

In Definitions 3.7 and 3.8, the requirements (3.75) and (3.77) ensure that the closed-

loop transfer trajectories, starting from any state vector near the origin node will reach (or closely approach) the destination node at some future time instant.

3.4.2 Virtual Net Adjacency: Calculations

In this section, calculations are developed to determine if the adjacency requirements given in Definitions 3.5-3.8 hold. Note that, in the adjacency definitions, there are two types of requirements. The first type of requirement, represented by (3.70), (3.72), (3.74) and (3.76) in Definitions 3.5-3.8, respectively, determine whether it is feasible to select a state vector along the destination NMT as the controller reference point when the spacecraft state is near the origin node. Hence, these requirements are referred to as "controller reference point" adjacency requirements. The second type of requirement, represented by (3.75) and (3.77) in Definitions 3.7 and 3.8, respectively, determine whether closed-loop transfer trajectories starting near the origin node will approach the destination node along an open NMT at some future time-instant. Hence, these requirements are referred to as "transfer trajectory" adjacency requirements.

3.4.2.1 Controller Reference Point Adjacency Requirement Calculations

The controller reference point adjacency requirement for adjacency type (A1) is given by (3.70), repeated here for clarity:

$$\mathcal{E}_{k_i,\mathcal{N}_i}^{\rho_{min}+\alpha} \subset \mathcal{E}_{\hat{k}_j,\mathcal{N}_j}. \tag{3.78}$$

In this section, calculations are developed to check if (3.78) holds for a given $\mathcal{N}_i$, k_i and $\mathcal{N}_j$, $\hat{k}_j$. Note that the controller reference point adjacency requirements for adjacency types (A2)-(A4), given by (3.72), (3.74) and (3.76), respectively, are identical to (3.78), with the exception that the origin NMT and/or the destination NMT are open, rather than closed. Therefore, the calculations described in this section can be used to check the controller reference point requirements for all adjacency types (A1)-(A4).

To determine whether (3.78) holds, the following procedure is executed:

Procedure 3.4 [Check for controller reference point adjacency requirement]

Inputs: Discrete-time index, and state vector, along the origin NMT: k_i, and $X_{ni}(k_i)$,
Discrete-time index, state vector, and SPI scale factor along destination NMT:
$\hat{k}_j$, $X_{nj}(\hat{k}_j)$ and $\rho_{\hat{k}_j}$,
Adjacency parameter $\alpha > 0$,

Scale factor for smallest SPI set, ρ_{min}.

Output: Determination of whether the controller reference point adjacency requirement holds.

1. Solve the following optimization problem:

$$\begin{aligned} F(k_i, \hat{k}_j) = \quad & \max_X \ (X - X_{nj}(\hat{k}_j))^T P (X - X_{nj}(\hat{k}_j)), \\ \text{subject to} \quad & (X - X_{ni}(k_i))^T P (X - X_{ni}(k_i)) = \rho_{min} + \alpha, \end{aligned} \tag{3.79}$$

2. If $F(k_i, \hat{k}_j) \leq \rho_{\hat{k}_j}$, then the controller reference point adjacency requirement holds,

3. If $F(k_i, \hat{k}_j) > \rho_{\hat{k}_j}$, then the controller reference point adjacency requirement does not hold.

End Procedure.

The value of $F(k_i, \hat{k}_j)$, i.e., the solution to the optimization problem in (3.79), corresponds to the scale factor for the largest ellipsoidal set, with shape matrix P and centered at $X_{nj}(\hat{k}_j)$, that shares exactly one point of intersection with $\mathcal{E}_{k_i,\mathcal{N}_i}^{\rho_{min}+\alpha}$. Calculation of $F(k_i, \hat{k}_j)$ is fast and straightforward. Because the quadratic objective and constraint functions in (3.79) share the same shape matrix P, the solution to (3.79) can be obtained by solving a scalar quadratic equation, as described next. Figure 3.8 illustrates the parameters used in the following discussion.

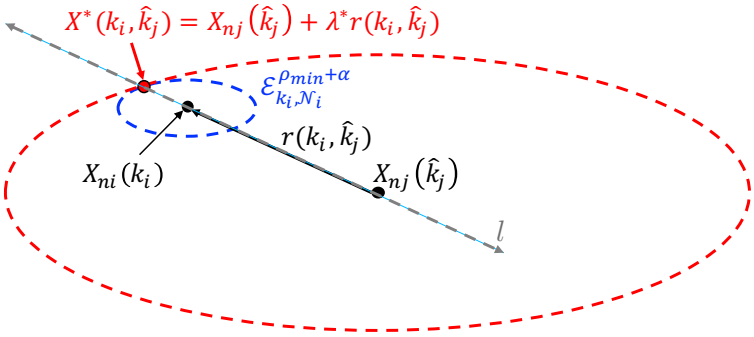


Figure 3.8: Illustration of the parameters used in controller reference point adjacency requirement calculations

Define $r(k_i, \hat{k}_j)$ as the vector from $X_{nj}(\hat{k}_j)$ to $X_{ni}(k_i)$, i.e.,

$$r(k_i, \hat{k}_j) = X_{ni}(k_i) - X_{nj}(\hat{k}_j). \tag{3.80}$$

Then, the line connecting $X_{nj}(\hat{k}_j)$ to $X_{ni}(k_i)$ is given by $l = X_{nj}(\hat{k}_j) + \lambda r(k_i, \hat{k}_j)$, $\lambda \in \mathbb{R}$. The maximizer to (3.79), denoted by $X^*(k_i, \hat{k}_j)$, corresponds to the state vector along this line, farthest from $X_{nj}(\hat{k}_j)$, that also lies on the boundary of $\mathcal{E}_{k_i,\bar{\mathcal{N}}_i}^{\rho_{min}+\alpha}$. The maximizer $X^*(k_i, \hat{k}_j)$ can be found by determining the largest value (in terms of absolute value) for λ, denoted λ^*, that satisfies

$$(X - X_{ni}(k_i))^T P (X - X_{ni}(k_i)) - (\rho_{min} + \alpha) = 0, \tag{3.81}$$

where

$$X = X_{nj}(\hat{k}_j) + \lambda r(k_i, \hat{k}_j). \tag{3.82}$$

To determine λ^*, (3.82) is substituted into (3.81), and the resulting expression is simplified to yield

$$\lambda^2 - 2\lambda + \left(1 - \frac{\rho_{min} + \alpha}{r(k_i, \hat{k}_j)^T P r(k_i, \hat{k}_j)}\right) = 0. \tag{3.83}$$

As (3.83) is a scalar quadratic equation in λ, the two roots can be obtained using the quadratic formula, i.e.,

$$\lambda = 1 \pm \sqrt{\frac{\rho_{min} + \alpha}{r(k_i, \hat{k}_j)^T P r(k_i, \hat{k}_j)}}, \tag{3.84}$$

and therefore,

$$\lambda^* = 1 + \sqrt{\frac{\rho_{min} + \alpha}{r(k_i, \hat{k}_j)^T P r(k_i, \hat{k}_j)}}. \tag{3.85}$$

Then, the maximizer to (3.79) is given by

$$X^*(k_i, \hat{k}_j) = X_{nj}(\hat{k}_j) + \lambda^* r(k_i, \hat{k}_j), \tag{3.86}$$

and the solution to (3.79) is given by

$$F(k_i, \hat{k}_j) = (X^*(k_i, \hat{k}_j) - X_{nj}(\hat{k}_j))^T P (X^*(k_i, \hat{k}_j) - X_{nj}(\hat{k}_j)). \tag{3.87}$$

3.4.2.2 Transfer Trajectory Adjacency Requirement Calculations

The transfer trajectory adjacency requirement for adjacency type (A3) is given by (3.75), repeated here for clarity:

$$\mathcal{R}_X(\bar{k},\ \mathcal{E}_{k_i,\bar{\mathcal{N}}_i}^{\rho_{min}+\alpha},\ X_{nj}(\hat{k}_j),\ \mathcal{W}) \subset \mathcal{E}_{k_j,\bar{\mathcal{N}}_j}^{\rho_{min}+\alpha}. \tag{3.88}$$

In this section, calculations are developed to check whether (3.88) holds for given $\mathcal{N}_i$, k_i, $\bar{\mathcal{N}}_j$, $\hat{k}_j$ and $\bar{k}$. Note that the transfer trajectory adjacency requirement for adjacency type (A4), given by (3.77), is identical to (3.88) with the exception that the origin NMT is open, rather than closed. Therefore, the calculations described in this section can be used to check the transfer trajectory adjacency requirements for adjacency types (A3) and (A4).

Recall that the set $\mathcal{R}_X(\bar{k},\ \mathcal{E}_{k_i,\mathcal{N}_i}^{\rho_{min}+\alpha},\ X_{nj}(\hat{k}_j),\ \mathcal{W})$ is the set of all states reachable at time instant $\bar{k}$, starting from the set $\mathcal{E}_{k_i,\mathcal{N}_i}^{\rho_{min}+\alpha}$ with initial controller reference point $X_{nj}(\hat{k}_j)$ under all possible disturbances, $w(k) \in \mathcal{W}$. To check requirement (3.88) directly, this set must be calculated. However, while efficient methods exist to calculate reachable sets when both the initial set and the disturbance set are ellipsoidal, e.g., [98], or when they are both polyhedral, e.g., [99–101], calculation of the reachable set $\mathcal{R}_X(\bar{k},\ \mathcal{E}_{k_i,\mathcal{N}_i}^{\rho_{min}+\alpha},\ X_{nj}(\hat{k}_j),\ \mathcal{W})$ is not straightforward because the initial set $\mathcal{E}_{k_i,\mathcal{N}_i}^{\rho_{min}+\alpha}$ is ellipsoidal, while the disturbance set $\mathcal{W}$ is polyhedral. Therefore, instead of directly calculating $\mathcal{R}_X(\bar{k},\ \mathcal{E}_{k_i,\mathcal{N}_i}^{\rho_{min}+\alpha},\ X_{nj}(\hat{k}_j),\ \mathcal{W})$, a different approach is taken in which (3.88) is checked by formulating a single non-convex optimization problem, and then solving this problem by breaking it into multiple simpler QCQPs, each of which can be solved quickly.

The non-convex optimization problem is formulated through the following observations and steps:

1. Note that the requirement (3.88) can be re-stated as

$$\mathcal{R}_X(\bar{k},\ Y,\ X_{nj}(\hat{k}_j),\ \mathcal{W}) \subset \mathcal{E}_{k_j,\bar{\mathcal{N}}_j}^{\rho_{min}+\alpha} \text{ for all } Y \in \mathcal{E}_{k_i,\bar{\mathcal{N}}_i}^{\rho_{min}+\alpha}. \tag{3.89}$$

2. Based on the closed-loop dynamics (3.7), for a given $Y \in \mathcal{E}_{k_i,\bar{\mathcal{N}}_i}^{\rho_{min}+\alpha}$, the set $\mathcal{R}_X(\bar{k},\ Y,\ X_{nj}(\hat{k}_j),\ \mathcal{W})$ is given by

$$\begin{aligned}\mathcal{R}_X(\bar{k},\ Y,\ X_{nj}(\hat{k}_j),\ \mathcal{W}) = \ & \{X(\bar{k}) \ :\ \exists\, w(k) \in \mathcal{W}\ \forall\ k \in [0,\ \bar{k}-1] \\ & \text{such that } X(\bar{k}) = \bar{A}^{\bar{k}}Y - M(\bar{k})X_{nj}(\hat{k}_j) \\ & +[\bar{A}^{\bar{k}-1}Bw(0) + \bar{A}^{\bar{k}-2}Bw(1) + \cdots + \bar{A}^{0}Bw(\bar{k}-1)]\},\end{aligned} \tag{3.90}$$

where the matrix $M(\bar{k}) \in \mathbb{R}^{6\times 6}$ is given by

$$M(\bar{k}) = \bar{A}^{\bar{k}-1}BKA^0 + \bar{A}^{\bar{k}-2}BKA^1 + \cdots + \bar{A}^{0}BKA^{\bar{k}-1}. \tag{3.91}$$

3. Next, note that, for a given $Y \in \mathcal{E}^{\rho_{min}+\alpha}_{k_i,\bar{\mathcal{N}}_i}$, (3.89) is satisfied if

$$\begin{aligned}(X(\bar{k}) - X_{nj}(k_j))^T P (X(\bar{k}) - X_{nj}(k_j)) \leq \rho_{min} + \alpha \\ \text{for all } X(\bar{k}) \in \mathcal{R}_X(\bar{k},\ Y,\ X_{nj}(\hat{k}_j),\ \mathcal{W}).\end{aligned} \tag{3.92}$$

4. The vector $(X(\bar{k}) - X_{nj}(k_j))$ in (3.92) can be written in terms of the error, i.e.,

$$\begin{aligned}(X(\bar{k}) - X_{nj}(k_j)) &= X(\bar{k}) - X_{nj}(\bar{k} + \hat{k}_j) + X_{nj}(\bar{k} + \hat{k}_j) - X_{nj}(k_j) \\ &= e(\bar{k}, \hat{k}_j) + g(\bar{k}, \hat{k}_j, k_j),\end{aligned} \tag{3.93}$$

where

$$g(\bar{k}, \hat{k}_j, k_j) = X_{nj}(\bar{k} + \hat{k}_j) - X_{nj}(k_j), \tag{3.94}$$

and, from the definition of the state error, $e(k, \delta) = X(k) - X_n(k + \delta)$, and using the error dynamics, (3.9), $e(\bar{k}, \hat{k}_j)$ is given by

$$\begin{aligned}e(\bar{k}, \hat{k}_j) = \ & \bar{A}^{\bar{k}} e(0, \hat{k}_j) + \bar{A}^{\bar{k}-1} B w(0) + \bar{A}^{\bar{k}-2} B w(1) + \cdots + \bar{A}^0 B w(\bar{k} - 1), \\ & w(k) \in \mathcal{W} \text{ for all } k \in [0,\ \bar{k} - 1],\end{aligned} \tag{3.95}$$

where $e(0, \hat{k}_j) = Y - X_{nj}(\hat{k}_j)$.

5. Define the tail of (3.95) containing the disturbance terms as a vector f, i.e.,

$$f = \bar{A}^{\bar{k}-1} B w(0) + \bar{A}^{\bar{k}-2} B w(1) + \cdots + \bar{A}^0 B w(\bar{k} - 1),\ w(k) \in \mathcal{W} \text{ for } k \in [0,\ \bar{k} - 1], \tag{3.96}$$

and note that $f \in \mathcal{R}_e(\bar{k},\ 0,\ \mathcal{W})$, i.e., f is in the reachable set for the error system at time instant $\bar{k}$ starting from $e(0) = 0$.

6. Based on (3)-(5), for a given $Y \in \mathcal{E}^{\rho_{min}+\alpha}_{k_i,\bar{\mathcal{N}}_i}$, the first set inclusion in (3.89) is satisfied if

$$\left[\bar{A}^{\bar{k}} e(0, \hat{k}_j) + g(\bar{k}, \hat{k}_j, k_j) + f\right]^T P \left[\bar{A}^{\bar{k}} e(0, \hat{k}_j) + g(\bar{k}, \hat{k}_j, k_j) + f\right] \leq \rho_{min} + \alpha, \tag{3.97}$$

for all $f \in \mathcal{R}_e(\bar{k},\ 0,\ \mathcal{W})$, where $e(0, \hat{k}_j) = Y - X_{nj}(\hat{k}_j)$ and
$g(\bar{k}, \hat{k}_j, k_j) = X_{nj}(\bar{k} + \hat{k}_j) - X_{nj}(k_j)$.

7. Finally, note that $Y \in \mathcal{E}^{\rho_{min}+\alpha}_{k_i,\bar{\mathcal{N}}_i} \iff (Y - X_{ni}(k_i))^T P (Y - X_{ni}(k_i)) \leq \rho_{min} + \alpha$.

Therefore, considering items (1)-(7) above, the transfer trajectory adjacency require-

ment (3.88) can be checked by solving the following non-convex optimization problem:

$$\begin{aligned} F(\bar{k}, k_i, \hat{k}_j) = \quad & \max_{Y,f} \left[\bar{A}^{\bar{k}} e(0, \hat{k}_j) + g(\bar{k}, \hat{k}_j, k_j) + f\right]^T P \left[\bar{A}^{\bar{k}} e(0, \hat{k}_j) + g(\bar{k}, \hat{k}_j, k_j) + f\right], \\ \text{subject to} \quad & (Y - X_{ni}(k_i))^T P (Y - X_{ni}(k_i)) \leq \rho_{min} + \alpha, \\ & f \in \mathcal{R}_e(\bar{k},\ 0,\ \mathcal{W}), \end{aligned} \tag{3.98}$$

where $e(0, \hat{k}_j) = Y - X_{nj}(\hat{k}_j)$ and $g(\bar{k}, \hat{k}_j, k_j) = X_{nj}(\bar{k} + \hat{k}_j) - X_{nj}(k_j)$. If $F(\bar{k}, k_i, \hat{k}_j) \leq \rho_{min} + \alpha$, then the requirement (3.88) holds.

The optimization problem given by (3.98) can be solved quickly by breaking it into several simpler QCQPs. Firstly, note that the set $\mathcal{R}_e(\bar{k},\ 0,\ \mathcal{W})$ can be easily determined. Because the set $\mathcal{W}$ is a polytope and the error system (3.9) is linear, $\mathcal{R}_e(\bar{k},\ 0,\ \mathcal{W})$ is also a polytope. Furthermore, the vertices of $\mathcal{R}_e(\bar{k},\ 0,\ \mathcal{W})$ are obtained by mapping each vertex of $\mathcal{W}$ forward over $\bar{k}$ time-instants through (3.9), i.e., the vertices of $\mathcal{R}_e(\bar{k},\ 0,\ \mathcal{W})$ are given by

$$f^l = (\bar{A}^{\bar{k}-1} + \bar{A}^{\bar{k}-2} + \cdots + \bar{A}^0) B w^l,\ l = 1, 2, \cdots, n_w, \tag{3.99}$$

where w^l are the vertices of $\mathcal{W}$. Secondly, because $\mathcal{R}_e(\bar{k},\ 0,\ \mathcal{W})$ is a polytope, the solution to (3.98) can be obtained by considering only $f \in \{f^1,\ f^2,\ \cdots, f^{n_w}\}$, and therefore, replacing the single optimization problem in (3.98) with n_w simpler optimization problems given by

$$\begin{aligned} F^l(\bar{k}, k_i, \hat{k}_j) = \quad & \max_{Y} \left[\bar{A}^{\bar{k}} e(0, \hat{k}_j) + g(\bar{k}, \hat{k}_j, k_j) + f^l\right]^T P \left[\bar{A}^{\bar{k}} e(0, \hat{k}_j) + g(\bar{k}, \hat{k}_j, k_j) + f^l\right], \\ \text{subject to} \quad & (Y - X_{ni}(k_i))^T P (Y - X_{ni}(k_i)) \leq \rho_{min} + \alpha, \end{aligned} \tag{3.100}$$

where $l = 1, 2, \cdots, n_w$, $e(0, \hat{k}_j) = Y - X_{nj}(\hat{k}_j)$ and $g(\bar{k}, \hat{k}_j, k_j) = X_{nj}(\bar{k} + \hat{k}_j) - X_{nj}(k_j)$. The optimization problems in (3.100) can be simplified further due to the following proposition:

Proposition 3.2. *Let $D \subset \mathbb{R}^n$ be a compact, convex set with a non-empty interior, and let $f(X) :\ D \to \mathbb{R}$ be convex. If $f(X) \leq a$ when $X \in \partial D$, i.e., X is in the boundary of D, then $f(X) \leq a$ for all $X \in D$.*

Proof. Suppose $X \in \text{Int}(D)$. Then, X can be expressed as

$$X = \lambda X' + (1 - \lambda) X'', \tag{3.101}$$

where $X' \in \partial D,\ X'' \in \partial D$ and $\lambda \in [0, 1]$ can be found by drawing any line through X and finding the points where that line intersects the boundary of D. Then, because $f(X)$ is

convex, and $f(X') \leq a$, $f(X'') \leq a$,

$$f(X) \leq \lambda f(X') + (1-\lambda) f(X'') \leq \lambda a + (1-\lambda)a = a. \tag{3.102}$$

□

Note that in (3.100), the objective function is convex since $P = P^T \succ 0$, and the maximization is done over the compact, convex set $\{Y \ : \ (Y - X_{ni}(k_i))^T P (Y - X_{ni}(k_i)) \leq \rho_{min} + \alpha\}$, with a non-empty interior. Therefore, by Proposition 3.2, the inequality constraint in (3.100) is replaced with an equality constraint, and (3.100) becomes

$$\begin{aligned} F^l(\bar{k}, k_i, \hat{k}_j) = \ & \max_{Y} \ \left[\bar{A}^{\bar{k}} e(0, \hat{k}_j) + g(\bar{k}, \hat{k}_j, k_j) + f^l\right]^T P \left[\bar{A}^{\bar{k}} e(0, \hat{k}_j) + g(\bar{k}, \hat{k}_j, k_j) + f^l\right], \\ \text{subject to} \ & (Y - X_{ni}(k_i))^T P (Y - X_{ni}(k_i)) = \rho_{min} + \alpha. \end{aligned} \tag{3.103}$$

After solving (3.103) for each $F^l(\bar{k}, k_i, \hat{k}_j)$, the solution to the original optimization problem (3.98) is given by $F(\bar{k}, k_i, \hat{k}_j) = \max\{F^l(\bar{k}, k_i, \hat{k}_j)\}$.

The procedure to determine whether the transfer trajectory adjacency requirement holds is as follows:

Procedure 3.5 [Check for transfer trajectory adjacency requirements]

Inputs: Discrete-time index, and state vector, along the origin NMT: k_i, and $X_{ni}(k_i)$,
Discrete-time index, state vector, and SPI scale factor along destination NMT: $\hat{k}_j$, $X_{nj}(\hat{k}_j)$ and $\rho_{\hat{k}_j}$,
Destination node information: k_j and $X_{nj}(k_j)$,
Number of discrete-time instants along transfer trajectory: $\bar{k}$,
Adjacency parameter $\alpha > 0$,
Scale factor for smallest SPI set, ρ_{min},
Vertices of $\mathcal{R}_e(\bar{k},\ 0,\ \mathcal{W})$: f^l, $l = 1, 2, \cdots, n_w$.

Output: Determination of whether the transfer trajectory adjacency requirement holds.

1. Solve the following optimization problems:

$$\begin{aligned} F^l(\bar{k}, k_i, \hat{k}_j) = \ & \max_{Y} \ \left[\bar{A}^{\bar{k}} e(0, \hat{k}_j) + g + f^l\right]^T P \left[\bar{A}^{\bar{k}} e(0, \hat{k}_j) + g + f^l\right], \\ \text{subject to} \ & (Y - X_{ni}(k_i))^T P (Y - X_{ni}(k_i)) = \rho_{min} + \alpha, \end{aligned} \tag{3.104}$$

$l = 1, 2, \cdots, n_w$, where

$$e(0, \hat{k}_j) = Y - X_{nj}(\hat{k}_j),$$
$$g = g(\bar{k}, \hat{k}_j, k_j) = X_{nj}(\bar{k} + \hat{k}_j) - X_{nj}(k_j),$$
$$f^l = (\bar{A}^{\bar{k}-1} + \bar{A}^{\bar{k}-2} + \cdots + \bar{A}^0)Bw^l.$$

2. Set $F(\bar{k}, k_i, \hat{k}_j) = \max\{F^l(\bar{k}, k_i, \hat{k}_j)\}$.

3. If $F(\bar{k}, k_i, \hat{k}_j) \leq \rho_{min} + \alpha$, then the transfer trajectory adjacency requirement holds.

4. If $F(\bar{k}, k_i, \hat{k}_j) > \rho_{min} + \alpha$ then the transfer trajectory adjacency requirement does not hold.

End Procedure.

Remark 3.16. *The solution to each QCQP in (3.104) can be obtained quickly by reducing the problem to a scalar root finding problem. Details are included in Appendix C.2. Note that using this solution method, numerical issues are observed when $\bar{k} + \hat{k}_j \gg k_j$. These issues, however, are not encountered in the calculation of the Virtual Net when using Procedures 3.7 and 3.10 described below. Simulations also show that it may be possible to solve (3.104) either numerically, e.g., using MATLAB's fmincon function, or by using the S-Procedure to reformulate the problem as a Semi-Definite Program (SDP) [102]. The method described in Appendix C.2 was chosen for implementation because this method was shown to be the fastest.*

3.4.3 Virtual Net Matrix Representation

The virtual net is represented by a weighted adjacency matrix, Π, and a connection array, Ξ. For a virtual net containing p nodes, the adjacency matrix $\Pi \in \mathbb{R}^{p \times p}$. For $n \in \mathbb{Z}_{[1,p]}$, $m \in \mathbb{Z}_{[1,p]}$, if node n is not adjacent to node m, then the corresponding adjacency matrix element is set to $+\infty$, i.e., $\Pi(n, m) = +\infty$. If node n is adjacent to node m, then the corresponding adjacency matrix element is set to $J \geq 0$, where J represents the cost of transition from node n to node m.

Because fuel efficiency is an important consideration for SRMP, the cost of transition is taken to be the total control used along the transfer trajectory connecting node the two nodes. The calculation of this cost of transition depends on the type of adjacency for the two nodes under consideration, i.e., adjacency type (A1)-(A4), and is calculated using Procedure 3.6.

Procedure 3.6 [Calculation of transition cost between adjacent nodes]

Inputs: Discrete-time index and state vector along origin NMT: k_i and $X_{ni}(k_i)$
Discrete-time index and state vector along destination NMT: $\hat{k}_j$ and $X_{nj}(\hat{k}_j)$
Adjacency parameter: $\alpha > 0$,
Scale factor for smallest SPI set: ρ_{min}.
Adjacency type for the two nodes under consideration: (A1), (A2), (A3) or (A4)
If adjacency type is (A3) or (A4): Destination node, $X_n(k_j)$

Output: Cost of transition between nodes, J.

1. If the adjacency type is (A1) or (A2), i.e., the destination nodes is a closed NMT, do the following:

   (a) Calculate a transfer trajectory from $k = 0$ to $k = \bar{k}$ using

   $$\begin{aligned} X(k+1) &= AX(k) + Bu(k) + Bw(k), \\ u(k) &= K(X(k) - X_{nj}(k + \hat{k}_j)), \\ w(k) &\in \mathcal{W}, \end{aligned} \tag{3.105}$$

   with $X(0) = X_{ni}(k_i)$, and $\bar{k}$ is determined as the first discrete-time instant for which

   $$X(\bar{k}) \in \mathcal{E}^{\rho_{min}+\alpha}_{\bar{k}+\hat{k}_j, \mathcal{N}_j}. \tag{3.106}$$

   (b) Calculate the cost of transition as

   $$J = \sum_{k=0}^{\bar{k}} \|u(k) + w_u(k)\|_1. \tag{3.107}$$

2. If the adjacency type is (A3) or (A4), i.e., the destination nodes is along an open NMT, do the following:

   (a) Calculate a transfer trajectory from $k = 0$ to $k = \bar{k}$ using

   $$\begin{aligned} X(k+1) &= AX(k) + Bu(k) + Bw(k), \\ u(k) &= K(X(k) - X_{nj}(k + \hat{k}_j)), \\ w(k) &\in \mathcal{W}, \end{aligned} \tag{3.108}$$

   with $X(0) = X_{ni}(k_i)$, and $\bar{k}$ is determined as the first discrete-time instant for which

   $$X(\bar{k}) \in \mathcal{E}^{\rho_{min}+\alpha}_{k_j, \bar{\mathcal{N}}_j}. \tag{3.109}$$

(b) Calculate the cost of transition as

$$J = \sum_{k=0}^{\bar{k}} \|u(k) + w_u(k)\|_1 \tag{3.110}$$

End Procedure.

Remark 3.17. *The difference in the calculation of J for adjacency types (A1) and (A2), (Step 1 of Procedure 3.6), and the calculation of J for adjacency types (A3) and (A4), (Step 2), is in when the transfer trajectory is terminated, i.e., how $\bar{k}$ is determined. For adjacency types (A1) and (A2), the destination node is a closed NMT. Hence, $\bar{k}$ is determined in (3.106) to be the first time-instant for which the spacecraft state is "close to" the current controller reference point along the destination NMT (node), i.e, $X_{nj}(\bar{k} + \hat{k}_j)$. For adjacency types (A3) and (A4), the destination node is a state vector along an open NMT. Hence, $\bar{k}$ is determined in (3.109) as the first time-instant for which the spacecraft state is "close to" the unique state vector that corresponds to the destination node, i.e., $X_{nj}(k_j)$.*

Remark 3.18. *In (3.105) and (3.108), the disturbance terms $w(k)$ are determined as described in Section 3.1.4, i.e., $w(k) = w_p(k) + w_u(k)$, where $w_p(k)$ is randomly assigned from a specified set, and $w_u(k)$ accounts for the minimum control constraint by cancelling a commanded control input if the input violates the constraint. The cost J in (3.107) and (3.110) is calculated as a summation of $\|u(k) + w_u(k)\|_1$ because this quantity represents the implemented (and not simply commanded) control inputs.*

While the adjacency matrix Π stores the cost of transition between adjacent nodes as described above, the connection array Ξ stores information used to generate the lowest-cost transfer trajectory between each pair of adjacent nodes. Specifically, if node n is adjacent to node m, then the corresponding element of the connection array contains the discrete-time index for the transfer location along the origin NMT (k_i) and the discrete-time index for the initial controller reference point along the destination NMT that yield the lowest cost transfer $(\hat{k}_j)$, i.e., $\Xi(i, j) = [k_i, \hat{k}_j]$.

The weighted adjacency matrix and the connection array are generated using the following procedure. This procedure is based on the adjacency calculations described in Procedures 3.4 and 3.5, and the calculation of transfer costs given in Procedure 3.6:

Procedure 3.7: [Calculation of adjacency matrix and connection array]

Inputs: Set of $l_c + l_o$ NMTs to be included in the virtual net: $\mathcal{M}$,
Set of p nodes to be included in the virtual net,

Final discrete-time index for all NMTs to be included in the virtual net: k_{max}, and $k_{max}^{\bar{\mathcal{N}}_j}$, $j \in \mathbb{Z}_{[1,l_o]}$,
Initial adjacency matrix with $\Pi(n,m) = +\infty$ for all $n,\ m \in \mathbb{Z}_{[1,p]}$,
Initial connection array with $\Xi(n,m) = [+\infty, +\infty]$ for all $n,\ m \in \mathbb{Z}_{[1,p]}$.

Output: Final adjacency matrix, Π, and connection array, Ξ.

For all pairs (n,m), where $n \in \mathbb{Z}_{[1,p]}$ is the origin node $m \in \mathbb{Z}_{[1,p]}$ is the destination node, do the following:

1. Determine the adjacency type, (A1)-(A4), for nodes n and m.
2. Extract parameters i and j, where $i \in \mathbb{Z}_{[1,l_c+l_o]}$ corresponds to the NMT for the origin node n, and $j \in \mathbb{Z}_{[1,l_c+l_o]}$ corresponds to the NMT for the destination node m.
3. If adjacency type is (A1): For each pair $(k_i,\ \hat{k}_j)$, where $k_i \in \mathbb{Z}_{[0,k_{max}]}$ and $\hat{k}_j \in \mathbb{Z}_{[0,k_{max}]}$, do the following:
   (a) Set $J = +\infty$.
   (b) Determine whether controller reference point adjacency requirement (3.70) holds using Procedure 3.4.
   (c) If requirement (3.70) does not hold, continue to the next pair $(k_i, \hat{k}_j)$.
   (d) If requirement (3.70) does hold, the nodes are adjacent. Calculate a transition cost between nodes, J_n using Procedure 3.6. If $J_n < J$:

   $$\begin{aligned} &\text{set } \Pi(n,m) = J_n, \\ &\text{set } \Xi(n,m) = [k_i, \hat{k}_j], \\ &\text{set } J = J_n. \end{aligned}$$

   Continue to the next pair $(k_i,\ \hat{k}_j)$.
4. If adjacency type is (A2): For each pair $(k_i,\ \hat{k}_j)$, where k_i is a fixed value between 0 and $k_{max}^{\bar{\mathcal{N}}_i}$ corresponding to the origin node, and $\hat{k}_j \in \mathbb{Z}_{[0,k_{max}]}$, do the following:
   (a) Set $J = +\infty$.
   (b) Determine whether controller reference point adjacency requirement (3.72) holds using Procedure 3.4.
   (c) If requirement (3.72) does not hold, continue to the next pair $(k_i, \hat{k}_j)$.

(d) If requirement (3.72) does hold, the nodes are adjacent. Calculate a transition cost between nodes, J_n using Procedure 3.6. If $J_n < J$:

$$\begin{aligned}&\text{set } \Pi(n,m) = J_n,\\ &\text{set } \Xi(n,m) = [k_i, \hat{k}_j],\\ &\text{set } J = J_n.\end{aligned}$$

Continue to the next pair $(k_i,\ \hat{k}_j)$.

5. If adjacency type is (A3): For each pair $(k_i,\ \hat{k}_j)$ where $k_i \in \mathbb{Z}_{[0,k_{max}]}$ and $\hat{k}_j \in \mathbb{Z}_{[0,k_{max}^{\bar{\mathcal{N}}_j}]}$, do the following:
   (a) Set $J = +\infty$.
   (b) Determine whether controller reference point adjacency requirement (3.74) holds using Procedure 3.4.
   (c) If requirement (3.74) does not hold, continue to the next pair $(k_i, \hat{k}_j)$.
   (d) If requirement (3.74) does hold, for each $\bar{k} \in \mathbb{Z}_{[1,k_{max}^{\bar{\mathcal{N}}_j} - \hat{k}_j]}$ do the following:
      i. Determine whether transfer trajectory adjacency requirement (3.75) holds using Procedure 3.5.
      ii. If requirement (3.75) does not hold, continue to the next $\bar{k}$.
      iii. If requirement (3.75) does hold, the nodes are adjacent. Calculate a transition cost between nodes, J_n using Procedure 3.6. If $J_n < J$:

$$\begin{aligned}&\text{set } \Pi(n,m) = J_n,\\ &\text{set } \Xi(n,m) = [k_i, \hat{k}_j],\\ &\text{set } J = J_n.\end{aligned}$$

      Continue to the next $\bar{k}$.
   (e) Continue to the next pair $(k_i,\ \hat{k}_j)$.

6. If adjacency type is (A4): For each pair $(k_i,\ \hat{k}_j)$, where k_i is a fixed value between 0 and $k_{max}^{\bar{\mathcal{N}}_i}$ corresponding to the origin node and $\hat{k}_j \in \mathbb{Z}_{[0,k_{max}^{\bar{\mathcal{N}}_j}]}$, do the following:
   (a) Set $J = +\infty$.
   (b) Determine whether controller reference point adjacency requirement (3.76) holds using Procedure 3.4.
   (c) If requirement (3.76) does not hold, continue to the next pair $(k_i, \hat{k}_j)$.

(d) If requirement (3.76) does hold, for each $\bar{k} \in \mathbb{Z}_{[1,k_{max}^{\bar{N}_j} - \hat{k}_j]}$ do the following:

i. Determine if transfer trajectory adjacency requirement (3.77) holds using Procedure 3.5.

ii. If requirement (3.77) does not hold, continue to the next $\bar{k}$.

iii. If requirement (3.77) does hold, the nodes are adjacent. Calculate a transition cost between nodes, J_n using Procedure 3.6. If $J_n < J$:

$$\begin{aligned} &\text{set } \Pi(n,m) = J_n, \\ &\text{set } \Xi(n,m) = [k_i, \hat{k}_j], \\ &\text{set } J = J_n. \end{aligned}$$

Continue to the next $\bar{k}$.

(e) Continue to the next pair $(k_i,\ \hat{k}_j)$.

End Procedure.

Remark 3.19. *The following remarks provide a description of certain steps in Procedure 3.7.*

- *For adjacency type (A1), step (3a) sets a placeholder value for the cost of transition. In step (3b), a determination is made as to whether the current nodes are adjacent by the $k_i,\ \hat{k}_j$ under consideration. In step (3d), if the current nodes are adjacent, a cost of transition is calculated using $k_i,\ \hat{k}_j$. If this cost of transition is lower than any previously calculated transition costs, then this cost is stored in the adjacency matrix and the current indices $k_i,\ \hat{k}_j$ are stored in the connection array. A similar discussion applies for adjacency type (A2).*

- *For adjacency type (A3), step (5a) sets a placeholder value for the cost of transition. In step (5b) and (5di), a determination is made as to whether the current nodes are adjacent by the $k_i,\ \hat{k}_j$ under consideration. Note that in step (5d), the values of $\bar{k}$ considered are restricted such that controller reference points remain along the open destination NMT for the entire transfer trajectory, i.e., when $\bar{k} = k_{max}^{\bar{N}_j} - \hat{k}_j$, the controller reference point will be the final state vector along the open NMT, $X_{nj}(k_{max}^{\bar{N}_j})$. In step (5diii), if the current nodes are adjacent, a cost of transition is calculated using $k_i,\ \hat{k}_j$. If this cost of transition is lower than any previously calculated transition costs, then this cost is stored in the adjacency matrix and the current indices $k_i,\ \hat{k}_j$ are stored in the connection array. A similar discussion applies for adjacency type (A4).*

In implementation of Procedure 3.7, several simplifications and assumptions can be made, if desired, to reduce calculations for adjacency types (A3)-(A4), i.e., cases where the destination node is along an open NMT.

1. If origin node n and destination node m are both along the same given open NMT, and $k_i \leq k_j$, i.e., the destination node is past the origin node, then node n is adjacent to node m with zero cost of transition.

2. If it has been determined that an origin node n is adjacent to a destination node m along a given open NMT by state vectors $X_{ni}(k_i)$ and $X_{nj}(\hat{k}_j)$, and with cost of transition J, then node n is also adjacent to any destination nodes $\bar{m}$ that are past node m along the same open NMT, by the same state vectors $X_{ni}(k_i)$ and $X_{nj}(\hat{k}_j)$, and with the same cost of transition J.

3. In steps (5) and (6), the range of $\hat{k}_j$ to consider can be restricted to $\hat{k}_j \in [1, \min\{k_j + \mu, k_{max}^{\bar{N}_j}\}]$, where μ is a small integer. Using this restriction, only initial controller reference points that precede, or are shortly after the destination node along the destination NMT are considered. In steps (5d) and (6d), the range of $\bar{k}$ can be restricted to $\bar{k} \in [1, \min\{k_j + \mu - \hat{k}_j, k_{max}^{\bar{N}_j}\}]$. Using this restriction, only transfer trajectories that terminate when the controller reference point is near the destination node are considered. These simplifications are reasonable because the satellite state-vector asymptotically approaches a neighborhood around the current controller set-point. Therefore, the satellite will generally be closest to the destination node at a time instant when the controller reference point is near the destination node.

Note that items (1) and (2) above are non-conservative in that applying these simplifications will not result in a loss of adjacency in the virtual net. Applying item (3) may result in a slight loss of adjacency, but results in a significant decrease in computation time and also avoids numerical issues in checking the transfer trajectory adjacency requirement, described in Remark 3.16. Finally, note that, in item (2), the actual cost of transition to each node $\bar{m}$ along the destination NMT would be slightly higher than the cost of transition to the preceding node. Similarly, in item (1) the actual cost of transition to nodes along the same NMT would be non-zero due to disturbances. The choice to set the cost of transition to the same value for all nodes in item (1) and to zero for item (2) is made to reduce and simplify calculations.

3.4.4 Maneuver Generation Using the Virtual Net

After the adjacency matrix and connection array have been calculated, safe maneuvers can be planned between specified starting and ending nodes in the virtual net. Using the adjacency matrix, Π, Dijkstra's algorithm is applied to generate a sequence of nodes that connects given starting and ending nodes. Then, using the connection array Ξ, a safe maneuver is planned to transition the spacecraft from the starting node to the ending node. This maneuver is planned by switching the controller reference to the next initial reference point in the sequence once the spacecraft reaches a small neighborhood of each transfer location. Specifically, for a spacecraft travelling toward node n before transferring to node m, the connection array element is given by $\Xi(n,m) = [k_i, \hat{k}_j]$. The controller reference point is switched to $X_{nj}(\hat{k}_j)$ at the first time instant k when the spacecraft state vector satisfies

$$X(k) \in \mathcal{E}^{\rho_{min}+\alpha}_{k_i,\mathcal{N}_i(\text{or}\bar{\mathcal{N}}_i)}. \tag{3.111}$$

Remark 3.20. *The switching criteria given by (3.111) is consistent with the adjacency definitions developed earlier. Specifically, if the controller reference point is switched based on (3.111), then the controller reference point adjacency requirements ensure that constraints will remain satisfied. Furthermore, the periodicity of closed NMTs, and the transfer trajectory adjacency requirements for connections to open NMTs, ensure that for any transfer between NMTs, (3.111) will be satisfied for some k, i.e., the spacecraft will always be able to switch to the next NMT in the sequence.*

3.5 Simulation Examples

In this section, simulations are presented to illustrate the implementation of the invariance-based SRMP method developed in Sections 3.1-3.4. Table 3.2 lists parameters used in simulations. The state-feedback gain matrix K for the controller (3.6) is an LQR gain matrix corresponding to the control and state weighting matrices given by $R_{LQ} = 2 \times 10^7 I_{3\times 3}$ and $Q_{LQ} = 100\text{diag}(1,\ 1,\ 1,\ 1\times 10^5,\ 1\times 10^5,\ 1\times 10^5)$. The shape matrix P for the ellipsoidal sets is chosen to be the solution to the discrete-time Riccatti equation, described in Section 2.3. For trajectory generation, Dijkstra's algorithm is implemented using the MATLAB BGL toolbox [103]. Ellipsoidal sets are plotted using the MATLAB ellipsoidal toolbox [104].

The virtual net is formed from a set of 100 NMTs, including 84 closed NMTs (54 elliptical NMTs centered at the origin, 15 periodic line segment NMTs, and 15 stationary point NMTs), and 16 open NMTs (4 non-periodic line segment NMTs and 12 helical NMTs).

Table 3.2: Parameters used in simulations

Parameter	**Symbol**	**Value**
Spacecraft mass	m	140 kg
Nominal orbital radius	R_0	7728.137 km
Mean motion	ω	0.001027 rad/s
Discrete-time update period	ΔT	61.16 s
Closed NMT Max discrete-time index	k_{max}	99
Maximum control limit	u_{max}	0.005 kgkm/s^2
Minimum control limit	u_{min}	0.0001 kgkm/s^2
Random disturbance norm bound	ϵ	0.0001 kgkm/s^2
Centers of exclusion zones	$s_1,\ s_2$	$[0\ \pm 1\ 0]^T$ km
	s_3	$[1.5\ 0\ 1]^T$ km
Exclusion zone shape matrices	$S_i, i = 1, 2, 3$	$\frac{1}{0.2^2} I_{3\times 3}$
Param. for adjacency/switching	α	0.1

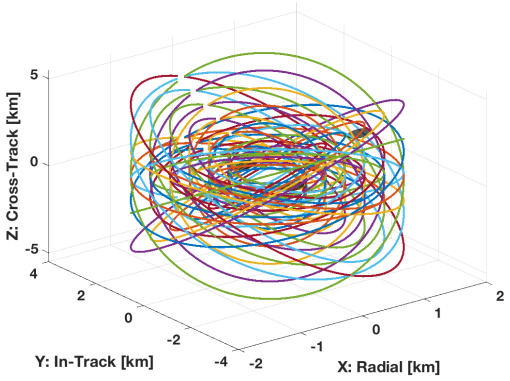
These NMTs are chosen to be evenly spaced within a box of $3.5 \times 8 \times 10$ km in the x, y, and z directions, respectively, centered at the origin. This virtual net contains $2,882$ nodes, including 84 nodes corresponding to closed NMTs and $2,798$ nodes corresponding to state vectors along open NMTs. Figure 3.9 shows the set of NMTs used to form the virtual net, along with the three exclusion zones considered.

The control-cost for a simulated maneuver between specified initial and final NMTs is calculated as

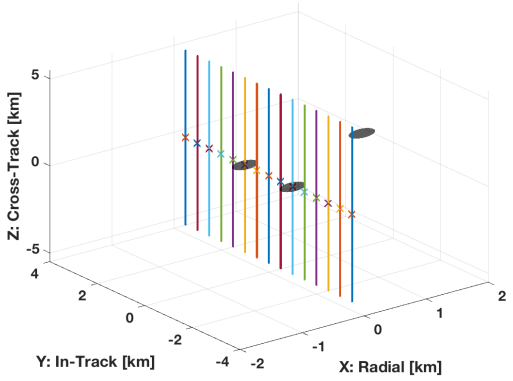
$$J = \Delta T \sum_{k=0}^{k_f} \|u(k)\|_1, \tag{3.112}$$

where k_f is the final discrete-time instant for the maneuver. Because of the random nature of the disturbances, the control-cost varies with each simulation. Therefore, the control costs reported below are the average control-costs obtained over 20 simulated maneuvers between specified initial and final NMTs.

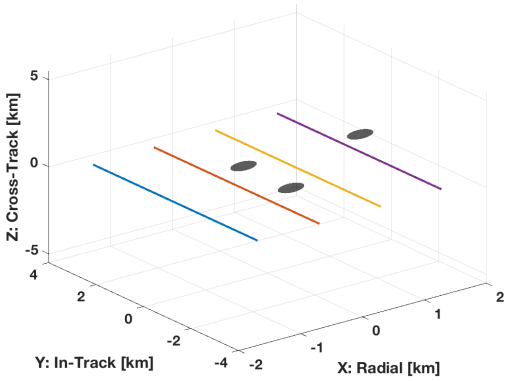
The first example maneuver considered is one planned to transition a spacecraft between two elliptical NMTs. Figure 3.10a shows the trajectory for this maneuver (plotted in pink). The spacecraft travels to four intermediate elliptical NMTs before reaching the final NMT. The average control-cost for this maneuver is $1,814$ N·s. Figure 3.10b shows that the maximum thrust constraint (3.12) and the exclusion zone constraints (3.14) are enforced throughout the maneuver (recall that constraints are satisfied if they are ≤ 0). Figure 3.10d shows commanded and actual thrust in the x, y and z directions, verifying that the minimum thrust constraint (3.15) is accommodated throughout the maneuver, i.e.,



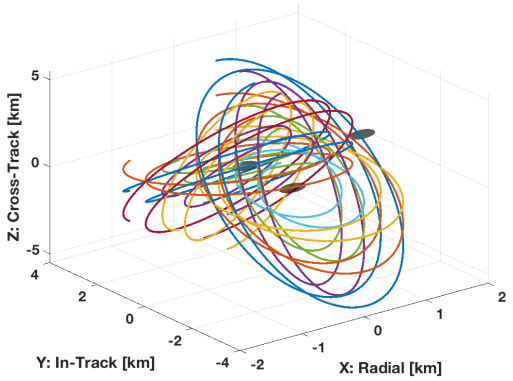
(a) Elliptical NMTs (closed)



(b) Periodic line segment and stationary point NMTs (closed)



(c) Non-periodic line segment NMTs (open)



(d) Helical NMTs (open)

Figure 3.9: NMTs included in the virtual net; Grey ellipsoids represent exclusion zones

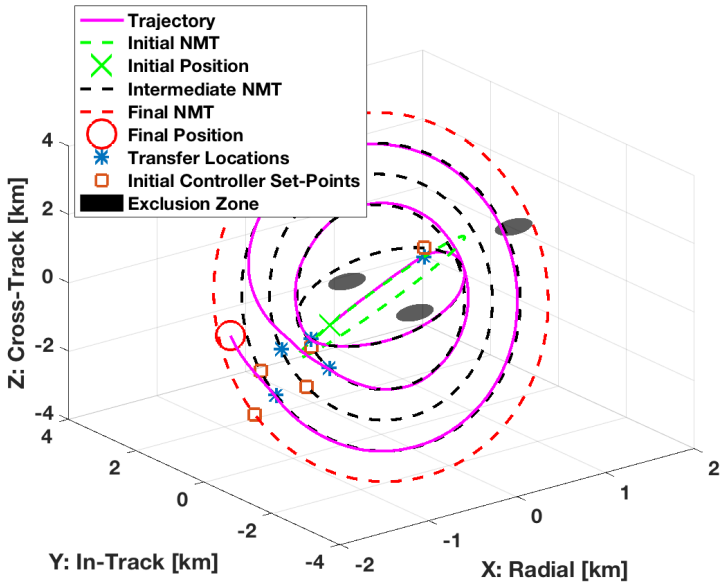
only commanded thrust values satisfying (3.15) are implemented. Finally, to verify that the spacecraft remains within the SPI tubes throughout the maneuver, the parameter

$$y_{el}(k) = e(k)^T P e(k) - \rho_{k+\delta} \tag{3.113}$$

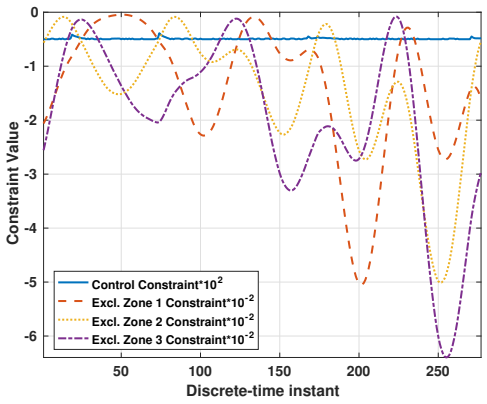
is plotted in Figure 3.10c, where $\rho_{k+\delta}$ is the SPI ellipsoidal set scale factor for the current controller reference point. Note that if $y_{el}(k) \leq 0$, then the spacecraft state is within the SPI tube for the NMT corresponding to the current controller reference point.

In the example maneuver shown in Figure 3.10, only closed NMTs are included in the transfer. A second example maneuver between two elliptical NMTs is shown in Figure 3.11. In this example, the spacecraft travels to four intermediate NMTs, including three elliptical NMTs and one open line-segment NMT, before reaching the final NMT. The average control cost for this maneuver is $1,212$ N·s, whereas a maneuver connecting the same initial and final NMTs utilizing only closed intermediate NMTs has an average control cost of $1,545$ N·s. Hence, in certain cases, the additional maneuver planning flexibility provided by including open NMTs in the virtual net results in maneuvers requiring less control (fuel) use.

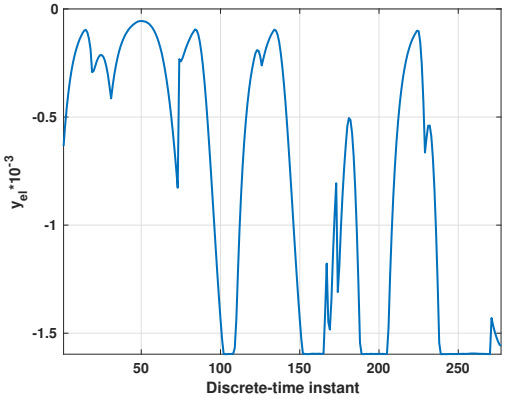
A final example maneuver is shown in Figure 3.12. In this maneuver, the spacecraft transitions from a stationary point NMT to a closed line-segment NMT by travelling to five intermediate NMTs, including two elliptical NMTs, one open line segment NMT and two open helical NMTs. In this example, utilizing open NMTs also results in fuel savings: the average control cost for the maneuver in Figure 3.12 is $2,247$ N·s while the average cost for a maneuver using only closed NMTs is $3,798$ N·s.



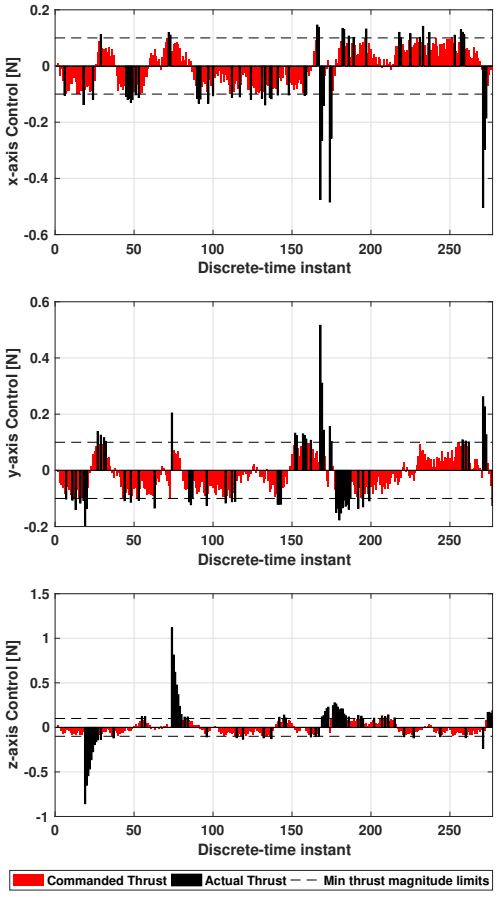
(a) Trajectory



(b) Constraints

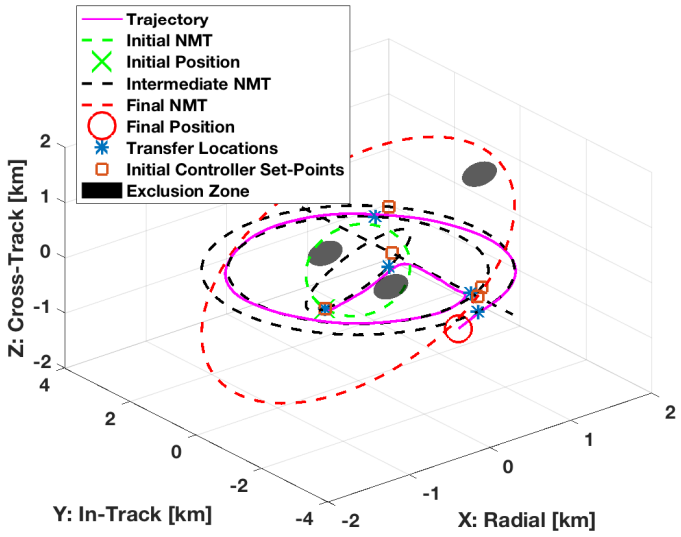


(c) SPI tube verification

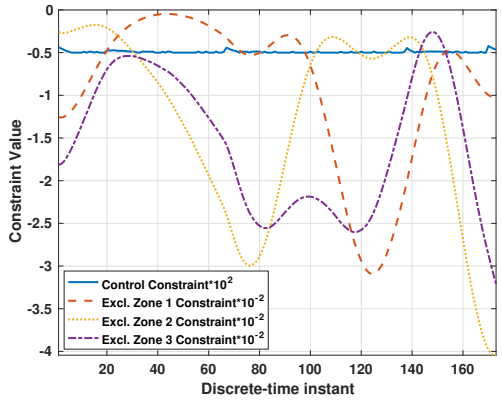


(d) Commanded and actual ahrust

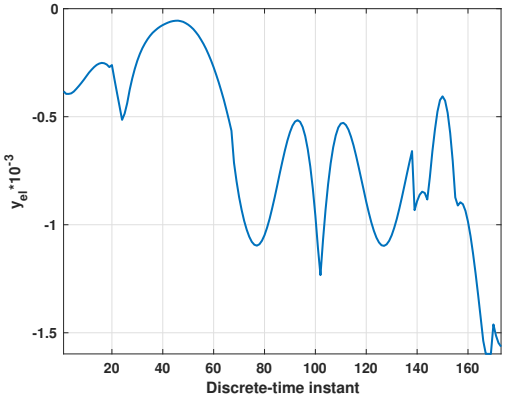
Figure 3.10: Example 1: maneuver connecting initial and final elliptical NMTs



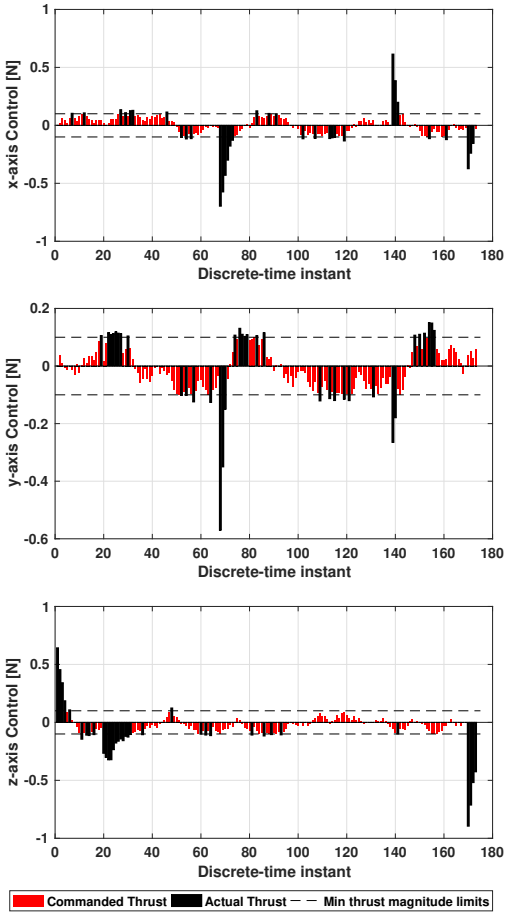
(a) Trajectory



(b) Constraints

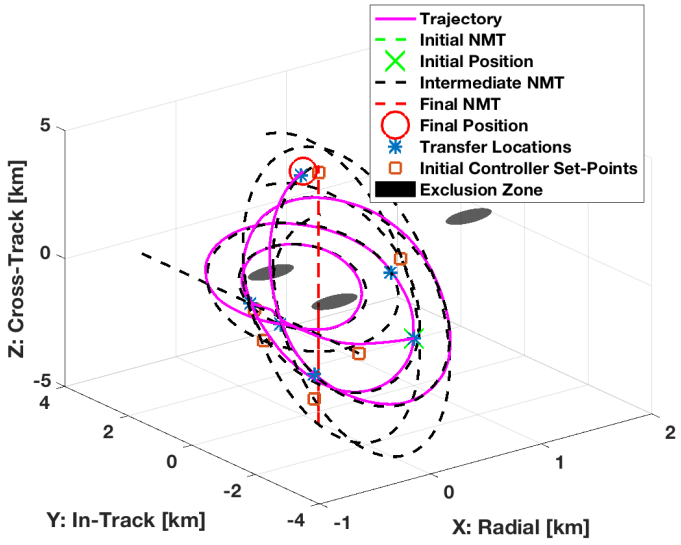


(c) SPI tube verification

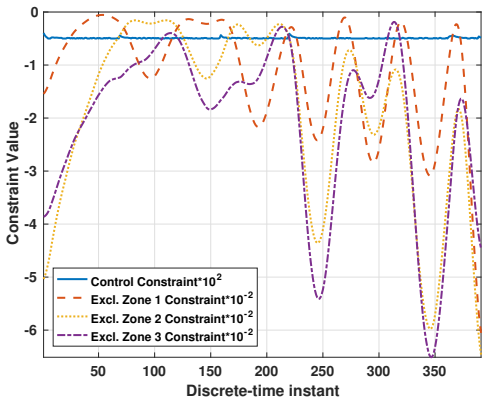


(d) Commanded and actual thrust

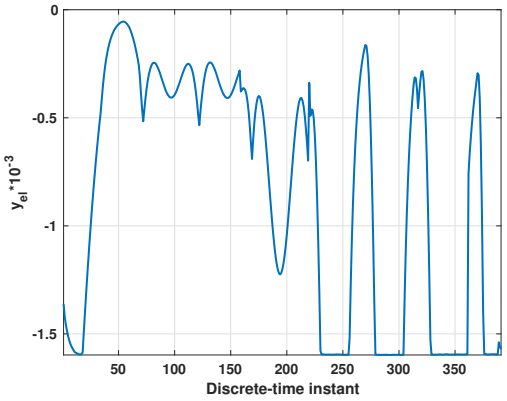
Figure 3.11: Example 2: maneuver connecting initial and final elliptical NMTs



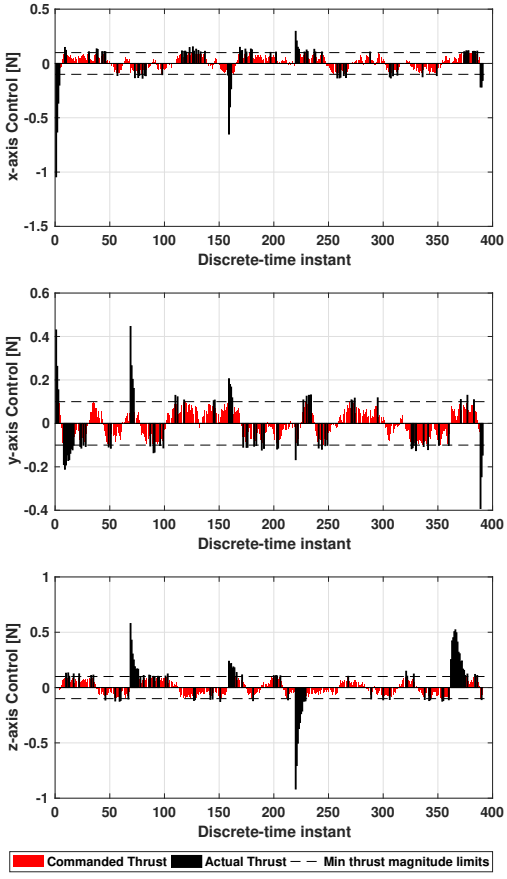
(a) Trajectory



(b) Constraints



(c) SPI tube verification



(d) Commanded and actual thrust

Figure 3.12: Example 3: maneuver connecting a stationary point NMT to a closed line segment NMT

3.6 Fast Calculations for Virtual Net Generation

Procedure 3.7 given in Section 3.4.3 to calculate the virtual net adjacency matrix and connection array is non-conservative in that it will correctly identify each pair of adjacent nodes, i.e., Procedure 3.7 results in an adjacency matrix such that, for all $n\ , m \in [1, p]$, $\Pi(n, m) < \infty$ if and only if node n is adjacent to node m. This property follows from the fact that Procedure 3.7 determines the adjacency matrix by performing exhaustive grid-searches over all potential state vectors $X_{ni}(k_i)$ and $X_{nj}(\hat{k}_j)$, to check the controller reference point adjacency requirement, and over all potential transfer trajectory durations, $\bar{k}$, to check the transfer trajectory adjacency requirements for each pair of nodes. While this property is desirable, implementing Procedure 3.7 is inefficient as a large number of checks must be carried out to determine adjacency for each pair of nodes. For example, if nodes n and m correspond to closed NMTs, then determining the adjacency of nodes n and m requires $(k_{max} + 1)^2$ checks.

In this section, a more efficient procedure is developed to generate an adjacency matrix and connection array. This efficiency is achieved through faster adjacency calculations that work by checking the controller reference point requirement without an exhaustive grid search. Furthermore, a single value for $\bar{k}$ is chosen and used to check the transfer trajectory adjacency requirement for adjacency types (A3) and (A4). The faster adjacency calculations are developed in Sections 3.6.1 and 3.6.2. Then, a revised procedure to generate the virtual net adjacency matrix and connection array is given in Section 3.6.3. Simulations show that this revised procedure can be used to calculate an adjacency matrix and connection array much faster than Procedure 3.7, with minor trade-offs related to the number of pairs of adjacent nodes in the resulting virtual net.

3.6.1 Fast Calculations for Controller Reference Point Adjacency Requirements

Recall the controller reference point adjacency requirement for adjacency type (A1), given by (3.70) and repeated here for clarity:

$$\mathcal{E}_{k_i,\mathcal{N}_i}^{\rho_{min}+\alpha} \subset \mathcal{E}_{\hat{k}_j,\mathcal{N}_j}. \tag{3.114}$$

All calculations developed in this section to check requirement (3.114) can also be applied to check the controller reference point adjacency requirements for adjacency types (A2)-(A4) with minor notational adjustments. These minor changes are described in Remark 3.21.

In Procedure 3.4, condition (3.114) was checked, for a given k_i and $\hat{k}_j$, by calculating

$$\begin{aligned} F(k_i, \hat{k}_j) = \quad & \max_X \ (X - X_{nj}(\hat{k}_j))^T P (X - X_{nj}(\hat{k}_j)), \\ & \text{subject to } (X - X_{ni}(k_i))^T P (X - X_{ni}(k_i)) = \rho_{min} + \alpha, \end{aligned} \tag{3.115}$$

or, equivalently by

$$F(k_i, \hat{k}_j) = (X^*(k_i, \hat{k}_j) - X_{nj}(\hat{k}_j))^T P (X^*(k_i, \hat{k}_j) - X_{nj}(k_j)), \tag{3.116}$$

where $X^*(k_i, \hat{k}_j)$ is the maximizer to (3.115), calculated using the methods given in Section 3.6.1. Then, if

$$F(k_i, \hat{k}_j) \leq \rho_{\hat{k}_j}, \tag{3.117}$$

the controller reference point adjacency requirement holds.

Rather than checking (3.117) for each possible pair $(k_i, \hat{k}_j)$, adjacency can be checked more efficiently based on two facts. Firstly, trivially, note that if (3.117) holds, then

$$F(k_i, \hat{k}_j) - \rho_{\hat{k}_j} \leq 0. \tag{3.118}$$

Secondly, the controller reference point adjacency requirement (3.114) holds for a given pair of nodes if (3.118) holds for *any* possible pair $(k_i, \hat{k}_j)$. Hence, the controller reference point adjacency requirement holds for a given pair of nodes if

$$G = \min_{k_i, \hat{k}_j} \left[(X^*(k_i, \hat{k}_j) - X_{nj}(\hat{k}_j))^T P (X^*(k_i, \hat{k}_j) - X_{nj}(k_j)) - \rho_{\hat{k}_j} \right] \leq 0, \tag{3.119}$$

where the minimization is carried out over the appropriate ranges of k_i and $\hat{k}_j$, e.g., if the origin NMT is closed, $k_i \in \mathbb{Z}_{[0, k_{max}]}$, and if the destination NMT is open, $\hat{k}_j \in \mathbb{Z}_{[0, \bar{N}_j k_{max}]}$.

Because k_i and $\hat{k}_j$ are discrete variables, the solution to (3.119) can be obtained by exhaustive grid search, similar to the methods used in Procedure 3.7. To facilitate faster calculations, problem (3.119) is reformulated in continuous-time by replacing the discrete-time variables k_i and $\hat{k}_j$ with continuous variables, t_i and $\hat{t}_j$:

$$G_c = \min_{t_i, \hat{t}_j} \left[(X^*(t_i, \hat{t}_j) - X_{nj}(\hat{t}_j))^T P (X^*(t_i, \hat{t}_j) - X_{nj}(t_j)) - \rho_{\hat{t}_j} \right] \leq 0, \tag{3.120}$$

where t_i and $\hat{t}_j$ are related to k_i and $\hat{k}_j$ through the discrete-time update period, ΔT, i.e.,

$t_i = \Delta T k_i$, and $\hat{t}_j = \Delta T \hat{k}_j$, and the vectors $X_{nj}(\hat{t}_j)$ and $X^*(t_i, \hat{t}_j)$ are defined as follows:

$$X_{nj}(\hat{t}_j) = \text{exp}(A_c \hat{t}_j) X_{0j}, \tag{3.121}$$

$$X_{ni}(t_i) = \text{exp}(A_c t_i) X_{0i}, \tag{3.122}$$

where $X_{0(\cdot)}$ is the initial condition for NMT $\mathcal{N}_{(\cdot)}$, and

$$X^*(t_i, \hat{t}_j) = X_{nj}(\hat{t}_j) + \lambda^* r(t_i, \hat{t}_j), \tag{3.123}$$

where

$$r(t_i, \hat{t}_j) = X_{ni}(t_i) - X_{nj}(\hat{t}_j), \tag{3.124}$$

and

$$\lambda^* = 1 + \sqrt{\frac{\rho_{min} + \alpha}{r(t_i, \hat{t}_j)^T P r(t_i, \hat{t}_j)}}. \tag{3.125}$$

Note that the expressions for X^*, r and λ^* in (3.123)-(3.125) are equivalent to the expressions (3.80), (3.85-3.86) developed in Section 3.6.1, with the discrete-variables k_i and $\hat{k}_j$ replaced with the continuous variables t_i and $\hat{t}_j$.

The advantage of using (3.120), rather than (3.119) directly, to check the controller reference point adjacency requirement is that, if, for each $\hat{t}_j$, the scale factor $\rho_{\hat{t}_j}$ is approximated by a quadratic curve-fit through points in the discrete set of scale factors $\rho_{\hat{k}_j}$, then the objective function $[\cdot]$ in (3.120) is twice continuously differentiable in t_i and $\hat{t}_j$, and therefore local minima can be obtained via a Newton-type iteration. This iterative technique to obtaining local minima is described in Appendix D. While this Newton-type iteration is not guaranteed to find the global minimum, simulations show that if the iteration is started from a few different initial conditions $(t_i, \hat{t}_j)$, the global minimum G_c in (3.120) can be obtained in most cases.

After the global minimum G_c in (3.120) is obtained, the requirement (3.119) can be checked by converting the continuous time variables (t_i and $\hat{t}_j$) corresponding to the global minimum G_c to the nearest discrete-time instants, i.e., $k_i = \text{round}(t_i/\Delta T)$ and $\hat{k}_j = \text{round}(\hat{t}_j, \Delta T)$, where the function round$(\cdot)$ rounds the argument to the nearest integer, and checking (3.119) using these values.

The preceding discussion is summarized in Procedure 3.8, which is used to check the controller reference point requirements for a given pair of nodes corresponding to origin NMT $\mathcal{N}_i$ and destination NMT $\mathcal{N}_j$.

Procedure 3.8 [Fast check for controller reference point adjacency requirement]

Inputs: Origin NMT information: $\mathcal{N}_i$, and $k_{max}^{\mathcal{N}_i}$,
Destination NMT information: $\mathcal{N}_j$, $k_{max}^{\mathcal{N}_j}$, and ρ_k, $k \in \mathbb{Z}_{[0,k_{max}]}$,
Discrete-time update period: ΔT,
Adjacency parameter: α,
Scale factor for smallest SPI set: ρ_{min}.

Outputs: Determination of whether the controller reference point adjacency requirement holds,
If the controller reference point adjacency requirement does hold: parameters k_i^*, $\hat{k}_j^*$.

1. Calculate the continuous-time "length" of each NMT:

$$L_i = k_{max}^{\mathcal{N}_i}\Delta T, \quad L_j = k_{max}^{\mathcal{N}_j}\Delta T, \tag{3.126}$$

2. Specify three initial guesses for $(t_i, \hat{t}_j)$:

$$(t_i, \hat{t}_j)_1 = (0,\ 0), \quad (t_i, \hat{t}_j)_2 = \left(\tfrac{L_i}{3}, \tfrac{L_j}{3}\right), \quad (t_i, \hat{t}_j)_3 = \left(\tfrac{2L_i}{3}, \tfrac{2L_j}{3}\right) \tag{3.127}$$

3. Starting from each initial guess $(t_i, \hat{t}_j)_l$, $l = 1, 2, 3$, calculate a local minimum

$$G_c^l = \min_{t_i, \hat{t}_j} \left[(X^*(t_i, \hat{t}_j) - X_{nj}(\hat{t}_j))^T P (X^*(t_i, \hat{t}_j) - X_{nj}(t_j)) - \rho_{\hat{t}_j} \right], \tag{3.128}$$

using the Newton-type iteration described in Appendix D.

4. Define

$$l^* = \underset{l=1,2,3}{\text{argmin}} \{G_c^l\}, \tag{3.129}$$

and

$$(t_i^*, \hat{t}_j^*) = (t_i, \hat{t}_j)_{l^*}, \tag{3.130}$$

and calculate

$$k_i^* = \text{round}(t_i^*/\Delta T), \quad \hat{k}_j^* = \text{round}(\hat{t}_j^*/\Delta T). \tag{3.131}$$

5. Calculate

$$G = (X^*(k_i^*, \hat{k}_j^*) - X_{nj}(\hat{k}_j^*))^T P (X^*(k_i^*, \hat{k}_j^*) - X_{nj}(k_j^*)) - \rho_{\hat{k}_j^*}, \tag{3.132}$$

where $X^*(k_i^*, \hat{k}_j^*)$ is given by (3.86).

6. If $G \leq 0$, then the controller reference point adjacency requirement holds.

7. If $G > 0$, then the controller reference point adjacency requirement does not hold.

End Procedure.

Remark 3.21. *The following remarks provide a description of certain steps in Procedure 3.8, and a description of notational changes needed for different adjacency types, (A1)-(A4).*

- *In step 2, the three initial guesses used to initiate the Newton-type iteration were selected because they yielded good performance in simulations, i.e., in most cases, these three initial guesses were sufficient to obtain the global minimum of the objective function. Different and/or more/less initial guesses may be selected if desired. Choosing more initial guesses may improve accuracy at the cost of computation time.*

- *Steps 4 and 5 are included to ensure no "false positives." In certain cases, the minimum G_c calculated using continuous-time variables may be non-negative, while the minimum G calculated using discrete-time variables is positive.*

- *Regarding notation, if the origin NMT is closed, i.e., adjacency types (A1) and (A3), then $k_{max}^{\mathcal{N}_i} = k_{max}$. If the origin NMT is open, i.e., adjacency types (A2) and (A4), then $\mathcal{N}_i$ is replaced with $\bar{\mathcal{N}}_i$. Equivalent notational adjustments are made depending on whether the destination NMT is closed or open.*

3.6.2 Fast Calculations for Transfer Trajectory Adjacency Requirements

Recall that the transfer trajectory adjacency requirement for adjacency type (A3) is given by (3.75), repeated here for clarity,

$$\mathcal{R}_X(\bar{k},\ \mathcal{E}_{k_i,\mathcal{N}_i}^{\rho_{min}+\alpha},\ X_{nj}(\hat{k}_j),\ \mathcal{W}) \subset \mathcal{E}_{k_j,\bar{\mathcal{N}}_j}^{\rho_{min}+\alpha}. \tag{3.133}$$

Given values for k_i, $\hat{k}_j$, and $\bar{k}$, requirement (3.133) can be checked using Procedure 3.5. To reduce the number of calculations required to form the adjacency matrix and connection array, rather than checking requirement (3.133) for each possible pair $(k_i, \hat{k}_j)$ and each possible value for $\bar{k}$ (as is done using Procedure 3.7), requirement (3.133) is only checked once, for a single set of values $\bar{k}$, k_i and $\hat{k}_j$. These values are selected in steps 1-3 of Procedure 3.9 below.

Procedure 3.9: [Fast check for transfer trajectory adjacency requirement]

Inputs: Origin NMT information: $\mathcal{N}_i$ and k_{max}, or $\bar{\mathcal{N}}_i$ and $k_{max}^{\bar{\mathcal{N}}_i}$
Destination NMT information: $\bar{\mathcal{N}}_j$, $k_{max}^{\bar{\mathcal{N}}_j}$, and ρ_k, $k \in \mathbb{Z}_{[0,k_{max}^{\bar{\mathcal{N}}_j}]}$
Destination node information: k_j and $X_{nj}(k_j)$,
Output parameters from Procedure 3.8: k_i^*, $\hat{k}_j^*$,
Adjacency parameter: α,
Scale factor for smallest SPI set: ρ_{min}.

Outputs: Determination of whether the transfer trajectory adjacency requirement holds,
If the transfer trajectory adjacency requirement does hold:
parameters k_i^*, $\hat{k}_j^{min}$.

1. Given k_i^*, use a backtracking line search starting from $\hat{k}_j = \hat{k}_j^*$ to find the smallest value of $\hat{k}_j$ for which $G \leq 0$, where G is given by (3.132). Denote this minimal $\hat{k}_j$ as $\hat{k}_j^{min}$.

2. If $\hat{k}_j^{min} > k_j$, conclude that the transfer trajectory adjacency requirement does not hold and end procedure.

3. Set
$$\bar{k} = k_j - \hat{k}_j^{min}, \tag{3.134}$$

4. Using Procedure 3.5 with inputs of $k_i = k_i^*$, $X_{ni}(k_i) = X_{ni}(k_i^*)$, $\hat{k}_j = \hat{k}_j^{min}$, $X_{nj}(\hat{k}_j) = X_{nj}(\hat{k}_j^{min})$, $\rho_{\hat{k}_j} = \rho_{\hat{k}_j^{min}}$, k_j, $X_{nj}(k_j)$, and $\bar{k} = k_j - \hat{k}_j^{min}$, determine whether the transfer trajectory adjacency requirement holds.

End Procedure.

Remark 3.22. *The following remarks provide a description of certain steps in Procedure 3.9.*

- *In step 1, the calculation of $\hat{k}_j^{min}$ estimates the "earliest" state vector along the destination NMT for which the controller reference point adjacency requirement holds. This state vector is the "earliest" state vector that can be used as the initial controller reference point for a transfer. The calculation of $\hat{k}_j^{min}$ is beneficial because, since the closed-loop system is ISS, the longer the controller reference point remains along the destination NMT, the closer the spacecraft will converge to the tube defined by ρ_{min}. Therefore, choosing the smallest possible $\hat{k}_j$ as the initial controller reference point for a transfer maximizes the chances that the transfer trajectory adjacency requirement will hold.*

- *In step 2, if $\hat{k}_j^{min} > k_j$, then it is impossible to choose any state vector that precedes the destination node along the destination NMT as the initial controller reference point for the transfer. In this case, it is unlikely that the transfer trajectory will approach the destination node (although exceptions do occur, e.g., if $X_{ni}(k_i)$ is very close to $X_{nj}(k_j)$).*

- *In step 3, the implicit assumption is made that if the transfer trajectory requirement holds for any $\bar{k}$, then it will hold at $\bar{k} = k_j - \hat{k}_j^{min}$. This assumption is reasonable; it states that the closed-loop state trajectory will be "closest" to the destination node $X_{nj}(k_j)$ at the time instant when the controller reference point is also $X_{nj}(k_j)$. Note that with the choice of $\bar{k} = k_j - \hat{k}_j^{min}$, the parameter $g(\bar{k}, \hat{k}_j^{min}, k_j)$ in Step 1 of Procedure 3.5 is equal to 0, slightly simplifying the calculation of $F^l(\bar{k}, k_i, \hat{k}_j^{min})$.*

3.6.3 Fast Calculation of Adjacency Matrix and Connection Array

Based on the fast adjacency calculations in Procedures 3.8 and 3.9, the revised procedure to generate the virtual net adjacency matrix and connection array is given as follows:

Procedure 3.10: [Fast Calculation of adjacency matrix and connection array]

Inputs: Set of $l_c + l_o$ NMTs to be included in the virtual net: $\mathcal{M}$.
Set of p nodes to be included in the virtual net.
Final discrete-time index for all NMTs to be included in the virtual net: k_{max}, and $k_{max}^{\mathcal{N}_j}$, $j \in [1, l_o]$,
Initial adjacency matrix with $\Pi(n, m) = +\infty$ for all $n,\ m \in \mathbb{Z}_{[1,p]}$.
Initial connection array with $\Xi(n, m) = [+\infty, +\infty]$ for all $n,\ m \in \mathbb{Z}_{[1,p]}$.

Output: Final adjacency matrix, Π, and connection array, Ξ.

For all pairs (n, m), where $n \in \mathbb{Z}_{[1,p]}$ is the origin node and $m \in \mathbb{Z}_{[1,p]}$ is the destination node, do the following:

1. Determine the adjacency type, (A1)-(A4), for nodes n and m.

2. Extract parameters i and j, where $i \in \mathbb{Z}_{[1,l_c+l_o]}$ corresponds to the NMT for the origin node, n, and $j \in \mathbb{Z}_{[1,l_c+l_o]}$ corresponds to the NMT for the destination node m.

3. If adjacency type is (A1), do the following:

   (a) Determine whether controller reference point adjacency requirement (3.70) holds using Procedure 3.8.

(b) If requirement (3.70) does hold, the nodes are adjacent. Calculate a transition cost between nodes, J using Procedure 3.6 and inputs of k_i^*, $X_{ni}(k_i^*)$, $\hat{k}_j^*$ and $X_{nj}(\hat{k}_j^*)$, and:

$$\text{set } \Pi(n,m) = J,$$
$$\text{set } \Xi(n,m) = [k_i, \hat{k}_j^*].$$

4. If adjacency type is (A2), do the following:

   (a) Determine whether controller reference point adjacency requirement (3.72) holds using Procedure 3.8.

   (b) If requirement (3.72) does hold, the nodes are adjacent. Calculate a transition cost between nodes, J using Procedure 3.6 and inputs of k_i^*, $X_{ni}(k_i^*)$, $\hat{k}_j^*$ and $X_{nj}(\hat{k}_j^*)$, and:

$$\text{set } \Pi(n,m) = J,$$
$$\text{set } \Xi(n,m) = [k_i, \hat{k}_j^*].$$

5. If adjacency type is (A3), do the following:

   (a) Determine whether controller reference point adjacency requirement (3.74) holds using Procedure 3.8.

   (b) If requirement (3.74) does hold, determine whether the transfer trajectory adjacency requirement (3.75) holds using Procedure 3.9,

      i. If requirement (3.75) does hold, the nodes are adjacent. Calculate a transition cost between nodes, J using Procedure 3.6 and inputs of k_i^*, $X_{ni}(k_i^*)$, $\hat{k}_j^{min}$, $X_{nj}(\hat{k}_j^{min})$ and $X_{nj}(k_j)$, and:

$$\text{set } \Pi(n,m) = J,$$
$$\text{set } \Xi(n,m) = [k_i, \hat{k}_j^{min}].$$

6. If adjacency type is (A4) do the following:

   (a) Determine whether controller reference point adjacency requirement (3.76) holds using Procedure 3.8.

   (b) If requirement (3.76) does hold, determine whether the transfer trajectory adjacency requirement (3.77) holds using Procedure 3.9,

      i. If requirement (3.77) does hold, the nodes are adjacent. Calculate a transition cost between nodes, J using Procedure 3.6 and inputs of k_i^*, $X_{ni}(k_i^*)$,

$\hat{k}_j^*$, $X_{nj}(\hat{k}_j^{min})$ and $X_{nj}(k_j)$, and:

$$\text{set } \Pi(n,m) = J,$$
$$\text{set } \Xi(n,m) = [k_i, \hat{k}_j^{min}].$$

End Procedure.

Remark 3.23. *Comparing Procedure 3.10 to Procedure 3.7, note that in Procedure 3.10, all of the loops over indices k_i, $\hat{k}_j$ and $\bar{k}$ have been eliminated through the use of the fast adjacency calculations of Procedures 3.8 and 3.9. Simulations show that eliminating these loops results in a significant reduction in the amount of time required to generate the adjacency matrix and connection array. These simulations are detailed in Section 3.7, along with additional comparison information between the virtual nets generated using Procedures 3.7 and 3.10.*

In the implementation of Procedure 3.10, additional assumptions and simplifications can be made to further reduce calculations for adjacency types (A3)-(A4), i.e., cases where the destination nodes is along an open NMT. Some of these assumptions and simplifications are as follows:

1. If origin node n and destination node m are both along the same given open NMT, and $k_i \leq k_j$, i.e., the destination node is past the origin node, then node n is adjacent to node m with zero cost of transition.

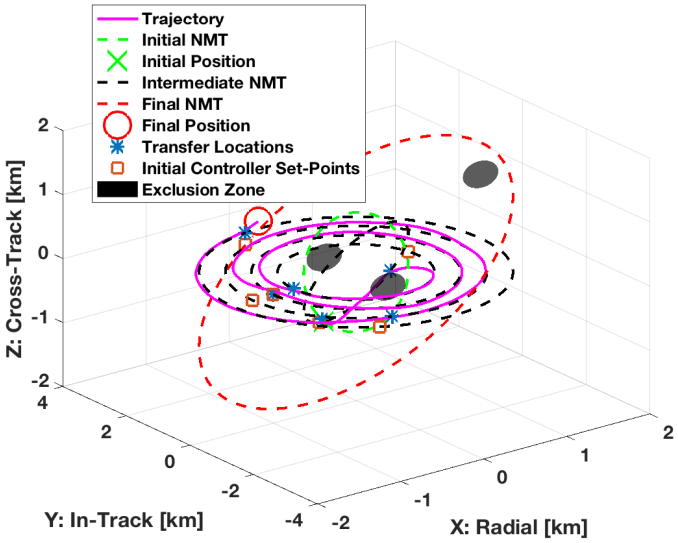
2. If it has been determined that the controller reference point requirement *is not* satisfied for an origin node n, and a destination node m along a given open NMT, then node n is not adjacent to any node $\bar{m}$ along the same open NMT.

3. If it has been determined that the controller reference point adjacency requirement *is* satisfied for an origin node n, and a destination node m along a given open NMT, by state vectors $X_{ni}(k_i)$ and $X_{nj}(\hat{k}_j)$, then the controller reference point adjacency requirement is also satisfied by the same state vectors $X_{ni}(k_i)$ and $X_{nj}(\hat{k}_j)$ for node n and any destination node $\bar{m}$ along the same open NMT.

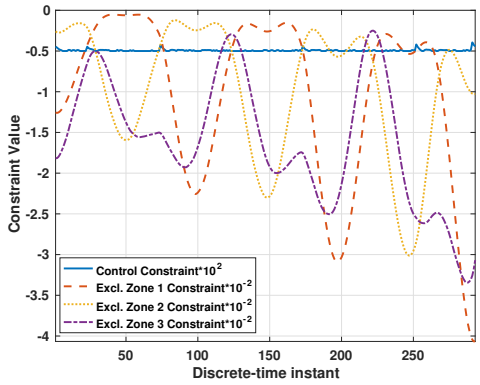
4. If it has been determined that an origin node n is adjacent to a destination node m along a given open NMT by state vectors $X_{ni}(k_i)$ and $X_{nj}(\hat{k}_j)$, and with cost of transition J, then node n is also adjacent to any destination nodes $\bar{m}$ that are past node m along the same open NMT, by the same state vectors $X_{ni}(k_i)$ and $X_{nj}(\hat{k}_j)$, and with the same cost of transition J.

3.6.4 Simulation Example

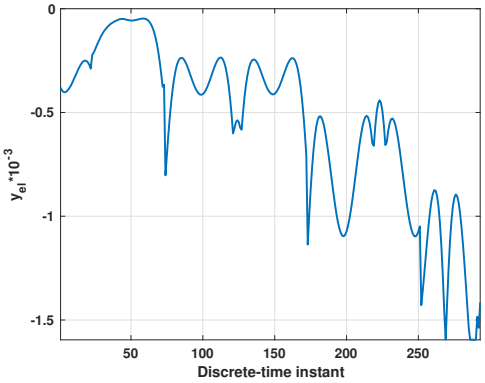
Figure 3.13 shows an example maneuver planned between two elliptical NMTs on a virtual net generated using the fast calculations of Procedure 3.10. The simulation set-up, i.e., simulation parameters and NMTs included in the virtual net, is the same as that considered in Section 3.5. In 3.13a, the spacecraft travels to five intermediate elliptical NMTs before arriving at the final NMT. Constraint and SPI verification data for this maneuver is provided in Figures 3.13b-3.13d, showing that constraints are enforced and the spacecraft remains within the SPI tubes at all time-instants. The average control cost for this maneuver is $1,755$ N·s.



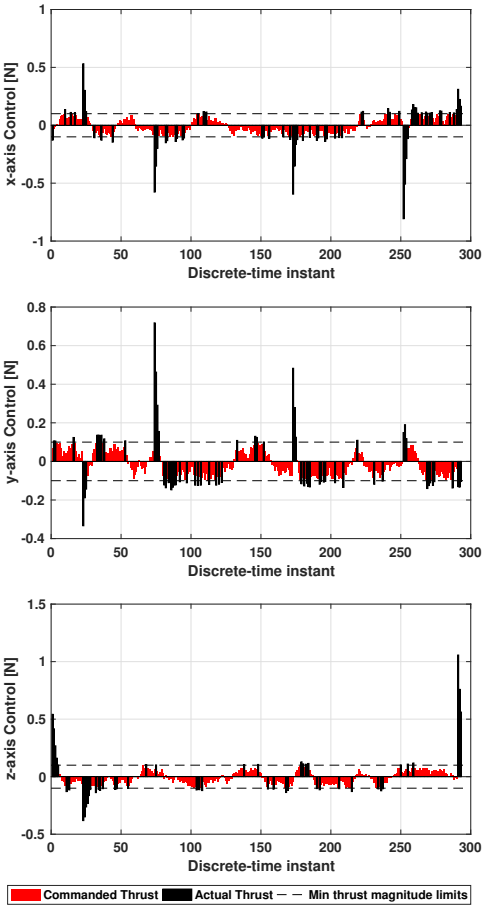
(a) Trajectory



(b) Constraints



(c) SPI tube verification



(d) Commanded and actual thrust

Figure 3.13: Maneuver connecting initial and final elliptical NMTs

3.7 Comparisons and Discussion

In this section, comparisons are made between the virtual nets calculated using the "baseline" method of Procedure 3.7 and the "fast" method of Procedure 3.10. In the comparison, both virtual nets are calculated using the simulation parameters and the set of 100 NMTs described in Section 3.5. Based on this comparison, the utility of including open NMTs in the virtual net is discussed, as well as possibilities for real-time implementation on-board a spacecraft.

3.7.1 Comparisons

Table 3.3 contains data comparing the computation time[1] required to generate the adjacency matrix and connection array. This data illustrates the utility of using the fast calculations of Procedure 3.10: the time required to calculate the adjacency matrix and connection array is reduced by nearly 97% compared to the baseline method. Table 3.4 contains data comparing the number of edges (pairs of adjacent nodes) in the corresponding virtual nets. This data illustrates the primary drawback of using the fast calculations of Procedure 3.10: the resulting virtual net has 14% fewer edges than the baseline case. This leads to less flexibility in maneuver planning.

Table 3.3: Comparison of computation time required to generate virtual nets using Procedure 3.7 and Procedure 3.10

	Baseline Method (Procedure 3.7)	**Fast Method (Procedure 3.10)**	
Adjacency Type	Computation Time	Computation Time	$\%$ Reduction
A1	562.3 sec	35.1 sec	93.8%
A2	189.3 sec	75.4 sec	60.1%
A3	$41,410.7$ sec	165.8 sec	99.6%
A4	$28,641.6$ sec	$1,938.9$ sec	93.2%
Totals	$70,803.9$ sec	$2,215.2$ sec	96.9%

3.7.2 Discussion

While the simulation examples in Section 3.5 demonstrated one benefit to including open NMTs in the virtual net, i.e., the ability to plan lower-cost maneuvers in certain cases,

[1] All computation times are for simulations run using MATLAB r2016a and a 2.8 GHz processor

Table 3.4: Comparison of number of edges included in the virtual nets calculated using Procedure 3.7 and Procedure 3.10

	Baseline Method (Procedure 3.7)	**Fast Method (Procedure 3.10)**	
Adjacency Type	♯ of Edges	♯ of Edges	% Reduction
A1	$1,754$	$1,749$	0.3%
A2	$19,058$	$18,249$	4.2%
A3	$59,549$	$45,533$	23.5%
A4	$840,524$	$726,425$	13.6%
Totals	$920,885$	$791,956$	14.0%

the data in Tables 3.3 and 3.4 illustrate several drawbacks to including open NMTs in the virtual net. Firstly, including open NMTs results in a severe increase in computation time (recall that adjacency type A1 involves only closed NMTs, while types A2-A4 involve open NMTs). This sharp increase is mainly due to the need to represent each state vector along open NMTs as a single node, leading to a much larger number of pairs of nodes to check for adjacency. In the virtual net considered here, the addition of 16 open NMTs results in an additional $2,798$ nodes. Secondly, in addition to the increased computation time, the large increase in the number of nodes resulting from including open NMTs leads to increased memory requirements for on-board storage of the virtual net data. Finally, if the fast calculations are used to reduce computation times, most of the edges lost (compared to the baseline case) involve open NMT nodes: of the $128,929$ edges lost, only 5 involve only closed NMTs.

For possible implementation of the invariance-based SRMP method proposed in this chapter, two options may be feasible. Firstly, a mix of on-line and off-line calculations could be considered. Specifically, all calculations to generate SPI tubes and form the adjacency matrix and connection array could be done offline, and the adjacency matrix, connection array and initial condition for each NMT could be uploaded to the spacecraft. Then, maneuvers between specified initial and final NMTs could be planned on-line using Dijkstra's algorithm and implemented using the method described in Section 3.4.4. Simulations show that individual maneuvers can be generated on the order of 0.1 sec. If this option is chosen, then it may be feasible to include open NMTs in the virtual net.

Secondly, if only periodic NMTs are included in the virtual net, it may be feasible to perform all calculations on-line. For the example set of 100 NMTs used here, generating SPI tubes using procedures described in Section 3.3 takes approximately 120 sec (see Remark 3.24), and calculation of the adjacency matrix and connection array using Procedure

3.10 takes approximately 35 seconds. Hence, it may be feasible to perform these on-board if the spacecraft was first maneuvered to an NMT known to be safe. The advantage of this second method, i.e., complete on-board implementation, is that the spacecraft could autonomously re-form the virtual net if additional obstacles are identified during operations.

Remark 3.24. *The time required to generate SPI tubes can be sped up dramatically by approximating* $d(\rho_{k+1})$ *with a curve-fit. This curve-fit could be generated offline and uploaded to the satellite. For the set of 100 NMTs used here, simulations show that doing this can reduce the time required to generate SPI tubes from approximately 120 seconds to less than 10 seconds. Using an approximation for* $d(\rho_{k+1})$ *has trade-offs, e.g., if the approximation is higher than the actual value, then tubes generated using the approximation may not be positively-invariant. Further investigation into the potential use of curve-fits to approximate* $d(\rho_{k+1})$ *is left to future work.*

CHAPTER 4

Relative Motion Planning for Satellite Inspection

In this chapter, two tools are developed for the specific bSRMP problem of satellite inspection. In this setting, sensors onboard an "inspector" spacecraft are used to collect information about a "target" spacecraft. Firstly, an information collection model is developed that expresses the rate of information collection as a function of both angle and distance to a specified target point on the target satellite. Then, this information collection model is used to develop an analytical control law based on the Local Gradient (LG) of the information collection rate.

The LG control method developed here is related to vehicle motion planning methods using "source seeking" or "gradient ascent/descent" techniques. In these methods, vehicles are guided to a location corresponding to the maximum (or minimum) of a field, based on the direction of the gradient of this field, where the direction of the gradient is obtained through sensor measurements. These methods have been studied for application both to single vehicles, e.g., [105] and to vehicle networks, e.g., [106, 107].

LG methods have also been applied to satellite attitude control problems involving Control Moment Gyroscopes (CMGs) and Reaction Wheel Assemblies (RWAs). In these methods, control inputs in the actuator null space are calculated based on the gradient of a suitably defined objective function. For CMGs, these methods have been applied to develop steering algorithms used to avoid singularities [108, 109], while for RWAs, LG methods have been applied to the problem of parameter identification [110].

The LG control law developed in this chapter is based on the gradient of the information rate. This control law can be used to drive the inspector spacecraft to the location corresponding to the maximum information collection rate, on a path along which the rate of information collection is strictly increasing.

This chapter is organized as follows. In Section 4.1, the dynamics model and constraints are described and the problem statement is defined, and in Section 4.2, the information collection model is introduced. In Section 4.3.1, a control law based on the LG

of the information collection rate is proposed, and closed-loop trajectories using this LG control law are analyzed. Based on this analysis, a state feedback controller is developed in Section 4.3.2, and a switching scheme between the LG and state-feedback controllers is proposed to promote constraint satisfaction. Simulations are presented in Section 4.4 to illustrate the implementation of the proposed methods, and, finally several options to improve performance are discussed in Section 4.5.

4.1 Modeling

4.1.1 Relative Motion Dynamics

The target spacecraft is assumed to be non-maneuvering, and operating on a known, nominal circular orbit. The inspector spacecraft is assumed to be maneuvering near the target spacecraft, hence the motion of the inspector spacecraft relative to the target spacecraft is modeled by the discrete-time CW equations derived in Section 2.1.3.1 and repeated here for clarity,

$$X(k+1) = AX(k) + Bu(k), \tag{4.1}$$

where the matrix A is given by (2.21), and control inputs u correspond to instantaneous velocity change, ΔV, hence, the matrix B is given by (2.23).

4.1.2 Constraints

Constraints on control and state variables are considered. Firstly, the inspector spacecraft's propulsion system has limited control authority, i.e.,

$$u(k) \in \mathbb{U} = \{u \mid \|u\|_\infty - u_{max} \leq 0\}, \tag{4.2}$$

where u_{max} is the maximum allowable ΔV in each direction. Secondly, the inspector spacecraft must stay out of a spherical exclusion zone, with radius r_{min}, centered on the target spacecraft, i.e., the origin of Hill's frame. This exclusion zone ensures no collisions between the inspector and target spacecraft, and also may account for considerations such as thruster plume impingement and any uncertainty in the position of the target spacecraft. The constraint on the inspector spacecraft state vector, based on this exclusion zone, is given by

$$X(k) \in \mathbb{X} = \{X \mid r_{min} - \|\Phi X\|_2 \leq 0\}, \tag{4.3}$$

where the matrix $\Phi = [I_{3\times3}\ 0_{3\times3}]$ isolates the position components from the state vector.

4.1.3 Problem Statement

The goal of the inspector spacecraft is to obtain information about a point $r_T \in \mathbb{R}^3$, located on the surface of the target spacecraft, while satisfying constraints. Associated with the target point r_T is the unit vector $\hat{n} \in \mathbb{R}^3$, which is normal to the surface at point r_T. Both r_T and $\hat{n}$ are assumed to be static (in Hill's frame) and known:

$$r_T = [x_T,\ \ y_T,\ \ z_T]^T\ ,\ \ \hat{n} = [n_x,\ \ n_y,\ \ n_z]^T\ . \tag{4.4}$$

It is assumed that information about the target point can only be obtained when there is an unobstructed line of sight from the sensor to the target point. Hence, the rate at which information is obtained is necessarily a function of both distance and angle to target. These functional dependencies are included in the information collection model developed in the next subsection.

4.2 Information Collection Model

The rate at which information is collected is assumed to be constant over the discrete-time update period, ΔT, and is determined by the inspector spacecraft position at the beginning of the update period. Hence, the information dynamics are modeled in discrete-time as

$$I(k+1) = I(k) + \dot{I}(X(k))\Delta T. \tag{4.5}$$

The information rate, $\dot{I}(X(k))$, is dependent on both distance-to-target and angle-to-target, i.e.,

$$\dot{I}(X(k)) = \dot{I}_d(X(k))\dot{I}_\phi(X(k)), \tag{4.6}$$

where the term $\dot{I}_d(X(k))$ describes the dependence on distance-to-target and the term $\dot{I}_\phi(X(k))$ describes the dependence on angle-to-target.

The dependence on distance-to-target is modeled with an expression based on the Shannon channel capacity equation [111], which gives the maximum rate at which data may be sent over a channel with zero losses. This approach is similar to approaches used in work related to path planning and information collection for Unmanned Aerial Vehicles (UAVs) [112–114]. The distance-to-target dependence is given by

$$\dot{I}_d(X(k)) = \alpha \log_2 \left(1 + \frac{\beta}{\|d(X(k))\|_2}\right), \tag{4.7}$$

where $\alpha > 0$ is a constant representing channel bandwidth, and the term

$$\frac{\beta}{\|d(X(k))\|_2} \tag{4.8}$$

represents the signal-to-noise ratio. In (4.8), $\beta > 0$ is a constant representing the target's visibility and $\|d(X(k))\|_2$ is the Euclidean distance from the inspector spacecraft to the target point, i.e.,

$$d(X(k)) = \left[x(k) - x_T,\ \ y(k) - y_T,\ \ z(k) - z_T\right]^T. \tag{4.9}$$

The inspector spacecraft's attitude control system is assumed to utilize a dedicated controller that is capable of keeping the sensor pointing directly at the target at all times. Hence, the sensor bore sight axis is represented by $d(X(k))$. Furthermore, it is assumed that due to considerations such as the target satellite's shape, and sensor limitations, useable information may only be obtained when the angle, ϕ, between $d(X(k))$ and $\hat{n}$ is below some prescribed value, ϕ_{max}, i.e., when the inspector spacecraft position vector is within an "information collection cone", defined by angle ϕ_{max}, with vertex at point r_T and a central axis of $\hat{n}$. More specifically, the information rate's dependence on angle-to-target is modeled by the following expression:

$$\dot{I}_\phi(X(k)) = \exp\left(\frac{-\phi(X(k))^2}{\frac{2}{9}\phi_{max}^2}\right), \tag{4.10}$$

where

$$\phi(X(k)) = \cos^{-1}\left(\frac{d(X(k)) \cdot \hat{n}}{\|d(X(k))\|_2}\right). \tag{4.11}$$

Note that (4.10) is a normal distribution-like function, with the standard deviation of $\frac{\phi_{max}}{3}$, scaled such that the maximum value is 1. Combining (4.7) and (4.10), the total rate at which the inspector satellite collects information about the target point is given by

$$\dot{I}(X(k)) = \dot{I}_d(X(k))\dot{I}_\phi(X(k)) = \alpha \log_2\left(1 + \frac{\beta}{\|d(X(k))\|_2}\right) \exp\left(\frac{-\phi(X(k))^2}{\frac{2}{9}\phi_{max}^2}\right). \tag{4.12}$$

Figure 4.1 shows the information rate plotted as a function of d and ϕ. For this example, the information collection parameters α and β are taken to be $\alpha = 1.0$ and $\beta = 0.01$, and the angle $\phi_{max} = 30^o$.

Note that the information rate (4.12) is undefined if the inspector spacecraft is exactly at the target point because, in this case, $\|d(X(k))\|_2 = 0$. In practice, this is not an issue

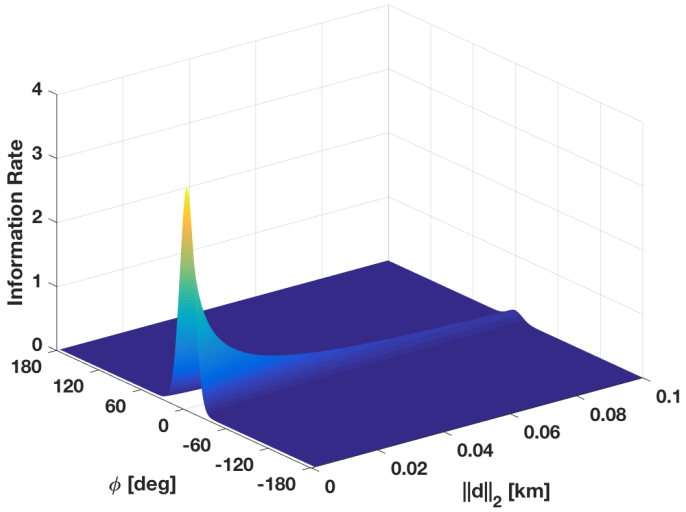


Figure 4.1: Information rate plotted as a function of d and ϕ

because the target point, which is assumed to be on the surface of the target satellite, lies within the exclusion zone.

Remark 4.1. *As mentioned previously, it is assumed that useable information can only be obtained when the satellite is within the cone defined by ϕ_{max}, r_T and $\hat{n}$. The inclusion of the angular dependence term $\dot{I}_\phi(X(k))$, defined in (4.10), results in $\dot{I}(X(k))$ taking "large" values only when $X(k)$ is within this cone, while outside of this cone, $\dot{I}(X(k))$ is nearly 0. This approach is used, rather than an alternative approach in which $\dot{I}(X(k))$ is set to exactly 0 outside of the cone, because this allows the gradient of $\dot{I}(X(k))$ to be defined almost everywhere, facilitating the development of the LG control law described in Section 4.3.1.*

4.3 Control Law for Safe Information Collection

In this section, a two-phase control law is developed to drive the inspection satellite on a path along which information about the target point can be obtained, while all constraints remain satisfied. Firstly, an analytical control law is developed based on the local gradient of the information collection rate, i.e.,

$$\frac{\partial \dot{I}}{\partial X} = \left[\frac{\partial \dot{I}}{\partial x}, \frac{\partial \dot{I}}{\partial y}, \frac{\partial \dot{I}}{\partial z}, \frac{\partial \dot{I}}{\partial \dot{x}}, \frac{\partial \dot{I}}{\partial \dot{y}}, \frac{\partial \dot{I}}{\partial \dot{z}}\right]. \tag{4.13}$$

where $\dot{I}$ is given by (4.12). This control law is used to drive the inspection satellite on a path along which the rate of information collection is strictly increasing. Secondly, a state-feedback controller is developed to guide the inspector spacecraft to a static reference point selected by a simple RG-like controller. This state-feedback controller is switched on as the inspector spacecraft approaches the target point, and ensures that constraints remain satisfied.

4.3.1 Local Gradient Control Law

In this section, a rapidly computable analytical control law is developed such that the rate of information collection using this control law is strictly increasing, i.e., the information collection rate satisfies $\dot{I}(X(k+1)) > \dot{I}(X(k))$, or, equivalently,

$$\dot{I}(X(k+1)) - \dot{I}(X(k)) > 0. \tag{4.14}$$

To obtain the control law satisfying (4.14), $\dot{I}(k+1)$ is expanded in a Taylor series about $X(k)$, i.e.,

$$\dot{I}(X(k+1)) = \dot{I}(X(k)) + \frac{\partial \dot{I}}{\partial X}\left[X(k+1) - X(k)\right] + \cdots . \tag{4.15}$$

For sufficiently small $[X(k+1) - X(k)]$, higher order terms may be neglected. Then, substituting (4.15) into the requirement (4.14) yields

$$\frac{\partial \dot{I}}{\partial X}\left[X(k+1) - X(k)\right] > 0, \tag{4.16}$$

and, substituting the CW dynamics (4.1) for $X(k+1)$ in (4.16) results in

$$\frac{\partial \dot{I}}{\partial X}(A - I_{6\times 6})X(k) + \frac{\partial \dot{I}}{\partial X}Bu(k) > 0. \tag{4.17}$$

Hence, a control $u(k)$ is sought such that (4.17) is satisfied.

Choosing

$$u(k) = \frac{u_{mag}}{\|\bar{u}\|_2}\bar{u}, \tag{4.18}$$

where u_{mag} is a positive parameter chosen by the mission designer[1] and

$$\bar{u} = \left[\frac{\partial \dot{I}}{\partial X} B\right]^T, \tag{4.19}$$

the second term on the LHS of (4.17) is guaranteed to be at least non-negative. To promote (4.17), and to improve the approximation in (4.15) by keeping $X(k+1) - X(k)$ small, an additional term is added to $u(k)$:

$$u(k) = \frac{u_{mag}}{\|\bar{u}\|_2}\bar{u} + u^*, \tag{4.20}$$

where u^* is given by

$$u^* = -B^\dagger(A - I_{6\times 6})X(k). \tag{4.21}$$

In (4.21), $B^\dagger$ is the pseudo-inverse of the matrix B, and is given by

$$B^\dagger = (B^T B)^{-1} B^T. \tag{4.22}$$

Hence, (4.21) minimizes

$$\|Bu^* + (A - I_{6\times 6})X(k)\|_2^2, \tag{4.23}$$

see Remark 4.2.

The local gradient control law is now given by

$$u(k) = \frac{u_{mag}}{\|\bar{u}\|_2}\bar{u} - B^\dagger(A - I_{6\times 6})X(k). \tag{4.24}$$

The physical result of this control law is that the inspection spacecraft traverses a path closely corresponding to the direction of the local gradient of $\dot{I}$ at each discrete-time instant. Therefore, along this path, the rate at which information is collected is increasing.

Remark 4.2. *An alternative approach is to define u^* in (4.20) as*

$$u^* = -\left[\frac{\partial \dot{I}}{\partial X} B\right]^\dagger \frac{\partial \dot{I}}{\partial X}(A - I_{6\times 6})X(k), \tag{4.25}$$

where

$$\left[\frac{\partial \dot{I}}{\partial X} B\right]^\dagger = \left[\frac{\partial \dot{I}}{\partial X} B\right]^T \left(\left[\frac{\partial \dot{I}}{\partial X} B\right]\left[\frac{\partial \dot{I}}{\partial X} B\right]^T\right)^{-1}, \tag{4.26}$$

[1] Selection of the parameter u_{mag} is discussed in Section 4.3.1.2

which gives the minimum 2-norm solution to

$$\frac{\partial \dot{I}}{\partial X} B u^* + \frac{\partial \dot{I}}{\partial X}(A - I_{6\times 6})X(k) = 0. \tag{4.27}$$

Simulations show that defining u^* as in (4.21) provides better performance. This is possibly a result of (4.21) facilitating better approximation accuracy for (4.15) by keeping $X(k+1) - X(k)$ small and reducing the influence of the dynamics.

4.3.1.1 Addressing Singularities in the Gradient

To ensure that the control given by (4.24) is always defined, singularities in the gradient $\frac{\partial \dot{I}}{\partial X}$ are addressed and patched as follows. The first component of the information rate gradient vector is given by

$$\frac{\partial \dot{I}}{\partial x} = \frac{9\phi \dot{I}\left[\frac{n_x}{\|d\|_2} - \frac{x - x_T}{\|d\|_2^2}\cos(\phi)\right]}{\phi_{max}^2 \sqrt{1 - \cos^2(\phi)}} - \frac{\alpha\beta \dot{I}_\phi (x - x_T)}{\log(2)\|d\|_2^3 \left(\frac{\beta}{\|d\|_2} + 1\right)}. \tag{4.28}$$

Note that the second and third components, $\frac{\partial \dot{I}}{\partial y}$ and $\frac{\partial \dot{I}}{\partial z}$, are given by (4.28) after substituting y, y_T, and n_y, or z, z_T and n_z for x, x_T, and n_x, respectively, and the final three components, $\frac{\partial \dot{I}}{\partial \dot{x}}$, $\frac{\partial \dot{I}}{\partial \dot{y}}$ and $\frac{\partial \dot{I}}{\partial \dot{z}}$, are all equal to 0.

Based on (4.28), the gradient is undefined (a) when $\|d\|_2 = 0$, and (b) when $\phi = 0,\ \pi$. As discussed in Section 4.2, case (a) when $\|d\|_2 = 0$ is not encountered in practice as the point where $\|d\|_2 = 0$ will lie within the exclusion zone. To address case (b), note that when $\phi = 0$, the inspector spacecraft is directly "above" the target point, i.e., along the normal vector to the target point, $\hat{n}$, and within the information collection cone. Conversely, when $\phi = \pi$, the inspector spacecraft is "below" the target point, i.e., along $-\hat{n}$ and outside of the information collection cone, see Figure 4.2. Hence, when ϕ has a value near 0 or π radians, the gradient is patched using the following rules:

1. When $\phi < \epsilon$, where the tolerance ϵ is a small positive value, the gradient is set to point directly at the target point, i.e.,

$$\frac{\partial \dot{I}}{\partial X} = \left[-d(X)^T,\ 0,\ 0,\ 0\right]. \tag{4.29}$$

2. When $\phi \geq \pi - \epsilon$, the gradient is set to point in a direction perpendicular to the current position vector, i.e.,

$$\frac{\partial \dot{I}}{\partial X} = \left[p(X)^T,\ 0,\ 0,\ 0\right], \tag{4.30}$$

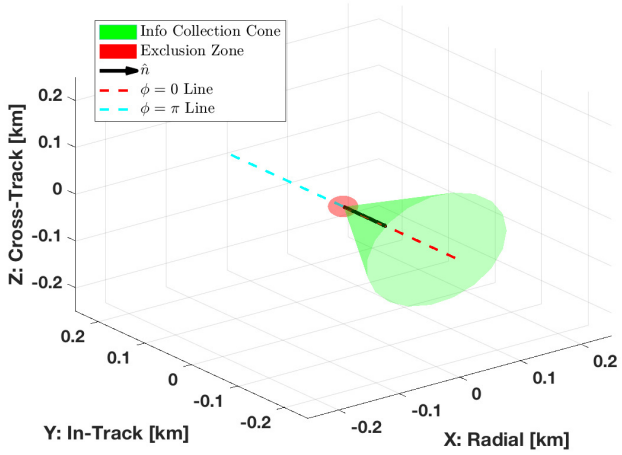


Figure 4.2: Depiction of lines defined by $\phi = 0$ and $\phi = \pi$ where the gradient is undefined

where $p(X) \in \mathbb{R}^3$ satisfies $p(X) \cdot [x,\ y,\ z]^T = 0$.

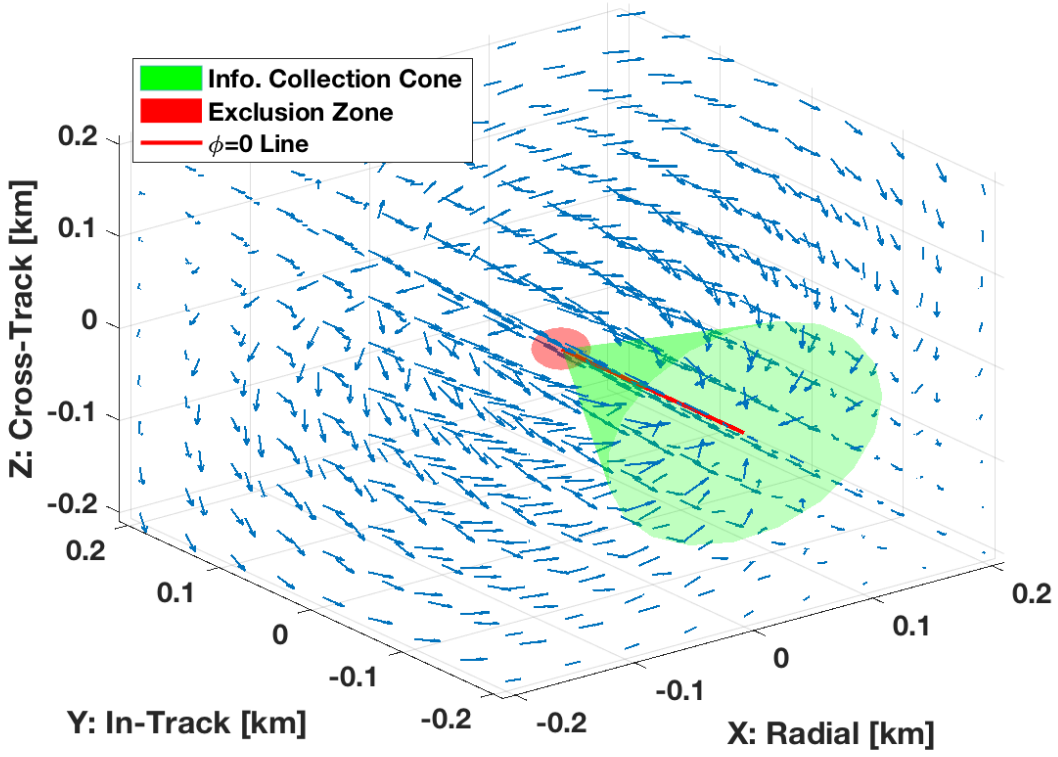
In simulations, setting $\epsilon = 1 \times 10^{-5}$ was shown to yield good performance.

4.3.1.2 Constraint Satisfaction

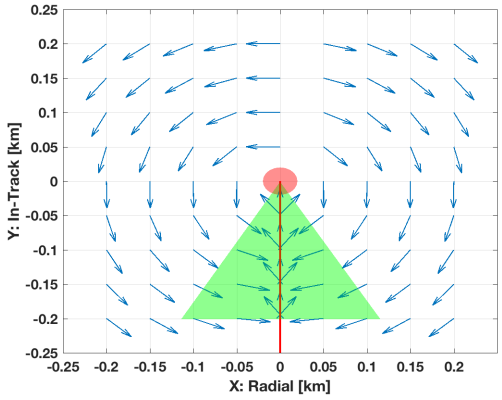
Using the LG control law (4.24) does not guarantee that the control and state constraints given by (4.2) and (4.3), respectively, will be satisfied. However, simulation results using this method show that a control of the form (4.24) with appropriate $u_{mag} < u_{max}$ selected is sufficient to satisfy both the requirement given by (4.17) and the control constraint given by (4.2). To obtain insight into the satisfaction of the state constraint (4.3), the expected behavior of the closed loop trajectories using (4.24) is now analyzed.

As noted above, the LG control law drives the inspector spacecraft in the direction of the gradient of the information rate, $\frac{\partial \dot{I}}{\partial X}$, at each discrete-time instant. Figure 4.3 includes three plots showing the direction of $\frac{\partial \dot{I}}{\partial X}$ for a sampling of points in the position space. Table 4.1 contains parameters used to produce the data in Figure 4.3.

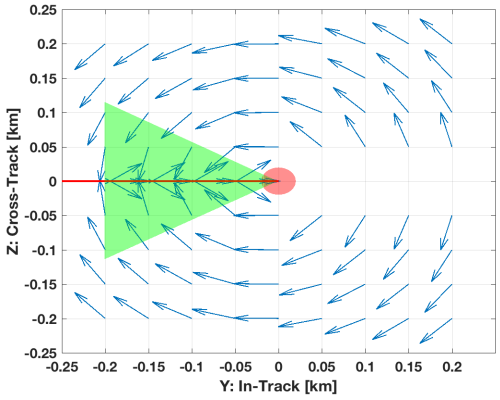
Note that everywhere away from the $\phi = 0$ line, i.e., everywhere where ϕ is large, a path following the direction of $\frac{\partial \dot{I}}{\partial X}$ will travel around the exclusion zone, and stay a nearly constant distance from the origin. This behavior is best illustrated in Figures 4.3b and 4.3c by the arrows outside of the information collection cone. Hence, a closed-loop trajectory is expected to trace a nearly circular arc around the exclusion zone, until it approaches the line defined by $\phi = 0$. This expected behavior is confirmed by simulations.



(a) 3-d view



(b) Slice through the plane $z = 0$



(c) Slice through the plane $x = 0$

Figure 4.3: Direction of $\frac{\partial \dot{I}}{\partial X}$ at sampled points in the position space

Table 4.1: Parameters used to calculate LG direction in Figure 4.3

Parameter	**Symbol**	**Value**
Target Point	r_T	$[0,\ -0.001,\ 0]^T$ km
Target Normal Vector	$\hat{n}$	$[0,\ -1,\ 0]^T$
Information Collection Cone Angle	ϕ_{max}	30 deg
Tolerance for Gradient Patching Rules	ϵ	1×10^{-5}

At points near the $\phi = 0$ line, the direction of the gradient begins to point toward the exclusion zone, as illustrated in Figures 4.3b and 4.3c by the arrows near the $\phi = 0$ line. Hence, as a closed-loop trajectory approaches the $\phi = 0$ line, it will dive toward the inspection target point at an increasingly steep angle, and, therefore, eventually enter the exclusion zone. To prevent this, and ensure that the constraint (4.3) remains satisfied, a state-feedback control law is switched on once the inspection spacecraft is near the exclusion zone and target point. Specifically, the control is switched to the state-feedback control law at the first time instant when both of the following requirements are met:

1. The inspection satellite is near the exclusion zone, i.e.,

$$r = \sqrt{x^2 + y^2 + z^2} \leq r_s, \tag{4.31}$$

   where $r_s > r_{min}$, and

2. The inspection satellite is near the $\phi = 0$ line, i.e.,

$$\phi \leq \phi_s, \tag{4.32}$$

   where ϕ_s is a small, positive value.

Remark 4.3. *The parameters r_s and ϕ_s used to determine when the LG control law is switched off, and the state feedback law is switched on, can be determined off-line through numerical experiments. Additional related discussion is included in Section 4.5.*

4.3.2 State-Feedback Control Law

At the first time instant when $r \leq r_s$ and $\phi \leq \phi_s$, the control law is switched to a state feedback law given by

$$u(k) = KX(k) + \Gamma X_d(k), \tag{4.33}$$

where $X_d(k)$ is the controller reference point given by

$$X_d(k) = \Phi^T \left[r_T + \delta(k)\hat{n} \right], \tag{4.34}$$

where $\delta(k) \in \mathbb{R}_{\geq 0}$ and $\Phi = [I_{3\times 3}\ 0_{3\times 3}]$. In (4.33), the matrix Γ is given by

$$\Gamma = \left[\Phi \left(I_{6\times 6} - \bar{A} \right)^{-1} B \right]^{-1} \Phi. \tag{4.35}$$

Note that the controller reference point $X_d(k)$ corresponds to a point at or directly above the information collection point r_T along the normal vector $\hat{n}$, with zero velocity. Furthermore, the closed loop trajectory under control law (4.33) will asymptotically approach the reference point $X_d(k)$ if $X_d(k)$ is held fixed.

To determine the controller reference point at each time instant k, the parameter $\delta(k) > 0$ in (4.34) is determined by considering the following optimization problem over a prediction horizon of T discrete-time steps:

$$\begin{aligned} \min_{\delta(k)} \quad & \delta(k) \\ \text{subject to} \quad & X(k+\sigma|k) \in \mathbb{X},\ \sigma \in \mathbb{Z}_{[0,T]}, \\ & u(k+\sigma|k) \in \mathbb{U},\ \sigma \in \mathbb{Z}_{[0,T-1]}, \\ & \delta(k+\sigma|k) = \delta(k),\ \sigma \in \mathbb{Z}_{[0,T-1]}, \\ & \delta(k) \geq 0, \\ & X(k|k) = X(k), \end{aligned} \tag{4.36}$$

where $X(k+\sigma|k)$ denotes the predicted state vector at time instant $k+\sigma$, when the prediction is made at time instant k, and $u(k+\sigma|k)$ and $\delta(k+\sigma|k)$ are similarly defined. The state and control predictions, $X(k+\sigma|k)$ and $u(k+\sigma|k)$, are made using (4.1), (4.33), and (4.34). Note that the optimization problem (4.36) is non-convex because the constraint set $\mathbb{X}$ is non-convex. To quickly obtain a reasonable approximation to the exact minimizer, a direct search is carried out over $\delta(k) \in \{0, \nu,\ 2\nu, \cdots\}$, for some small step size $\nu > 0$, and selecting $\delta(k)$ to be the smallest value corresponding to a feasible solution to (4.36). Several other options are available to determine a solution to (4.36), and to determine the controller reference point X_d; see Remark 4.4

Remark 4.4. *The optimization problem (4.36) can be made convex by replacing the non-convex state constraint (4.3) with a planar constraint. Specifically, the state-space can be bisected with the hyperplane that is tangent to the exclusion zone boundary at the point where the line segment, starting from r_T and proceeding in the direction of $\hat{n}$, intersects this boundary. Then, the inspector spacecraft state vector can be constrained to always be*

within the half-space defined by this hyperplane that does not contain the exclusion zone. Using this planar constraint, the problem (4.36) becomes a LP.

Another option to determine the controller reference point X_d is to obtain, through off-line simulations, a constant value for δ in (4.34) that is sufficiently large to ensure constraint satisfaction. Then, set $\delta(k) = \delta$ for all k.

4.3.3 Combined Control Law

The complete control law proposed for satellite inspection is as follows:

1. While $\phi > \phi_s$ or $r > r_s$, the LG controller is used:

$$u(k) = \frac{u_{mag}}{\|\bar{u}\|_2}\bar{u} - B^{\dagger}(A - I_{6\times 6})X(k), \text{ where } \bar{u} = \left[\frac{\partial \dot{I}}{\partial X}B\right]^T. \tag{4.37}$$

2. At the first time instant k when $\phi \leq \phi_s$, and $r \leq r_s$, the control law switches to the state feedback controller:

$$u(k) = KX(k) + \Gamma X_d(k), \text{ where } \Gamma = \left[\Phi\left(I_{6\times 6} - \bar{A}\right)^{-1}B\right]^{-1}\Phi, \tag{4.38}$$

   with reference point $X_d(k)$ determined at each discrete-time instant as the solution to (4.36).

4.4 Simulations

Simulation case studies are now considered to demonstrate the utility of the proposed approach. Parameters used in simulations are shown in Table 4.2.

Three simulations are run, starting from initial state vectors given by

$$\begin{aligned} X_1(0) &= [\ -0.1 \quad 0 \quad -0.1 \quad 0 \quad 0 \quad 0\ ]^T, \\ X_2(0) &= [\ 0 \quad -0.2 \quad 0 \quad 0 \quad 0 \quad 0\ ]^T, \\ X_3(0) &= [\ 0.1 \quad 0 \quad 0 \quad 0 \quad -0.2\omega \quad 0\ ]^T, \end{aligned} \tag{4.39}$$

respectively. Note that $X_1(0)$ is located on the opposite side of the exclusion zone from the information collection cone, i.e., it is along $-\hat{n}$, $X_2(0)$ corresponds to an unforced equilibrium point in the CW dynamics and $X_3(0)$ is along a closed NMT in the $x-y$ plane. Data from these simulations is shown in Figure 4.4. Trajectories are shown in Figure 4.4a, and a view showing the trajectory behavior near the exclusion zone is shown in Figure 4.4b.

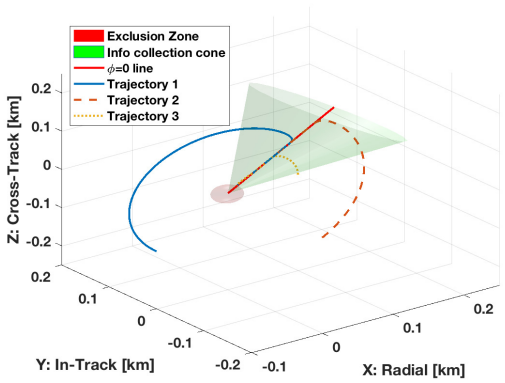
Table 4.2: Parameters used in simulations

Parameter	**Symbol**	**Value**
Nominal orbital radius	R_0	6828.137 km
Mean motion	ω	0.001112 rad/s
Discrete-time update period	ΔT	10 s
State-error weighting matrix used to obtain LQR gain matrix, K	Q_{LQ}	diag$(10,\ 10,\ 10,\ 0.01,\ 0.01,\ 0.01)$
Control weighting matrix used to obtain LQR gain matrix, K	R_{LQ}	$10^6 I_{3\times 3}$
Maximum control limit	u_{max}	0.001 km/s
Exclusion zone radius	r_{min}	0.02 km
LG control law scale factor	u_{mag}	0.0004
Target point	r_T	$[0.001,\ 0,\ 0.001]^T$ km
Target normal vector	$\hat{n}$	$[\frac{1}{\sqrt{2}},\ 0,\ \frac{1}{\sqrt{2}}]^T$
Info. collection cone angle	ϕ_{max}	$\frac{\pi}{6}$ rad (30 deg)
Info. collection parameter	α	1.0
Info. collection parameter	β	0.01
Tolerance for gradient patching rules	ϵ	1×10^{-5}
Control switching parameter	r_s	0.05 km
Control switching parameter	ϕ_s	5 deg
Prediction horizon used to determine state-feedback control reference point	T	100
Step size used to determine state-feedback control reference point	ν	0.001

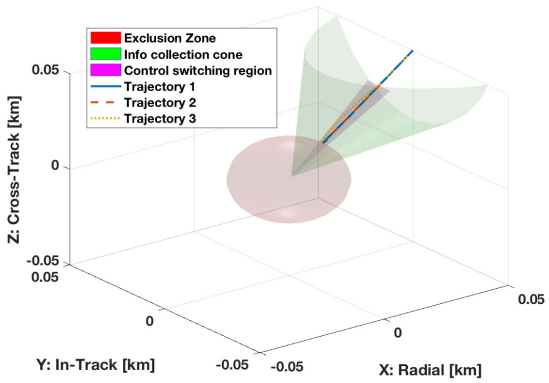
Note that the closed-loop trajectories starting from each of these initial conditions approach the $\phi = 0$ line using the LG control law given by (4.37), and then approximately follow this line until they reach the control switching region, shown in Figure 4.4b. At this point, the control switches to the state feedback law given by (4.38) and the trajectories approach, but do not enter, the exclusion zone.

The information collection rate at each discrete-time instant k along the trajectories is shown in Figure 4.4c. The time instants where the control law switches from the LG law to the state-feedback law are shown as dashed black lines. Note that the information rate is strictly increasing for all time instants from 0 to the switching time, as expected. The total information obtained along these trajectories, calculated using (4.5), is shown in Figure 4.4d. The control and state variable constraints, (4.2) and (4.3), are plotted in Figures 4.4e and 4.4f, respectively. These plots show that constraints are satisfied along the trajectories (constraints are satisfied if they are ≤ 0).

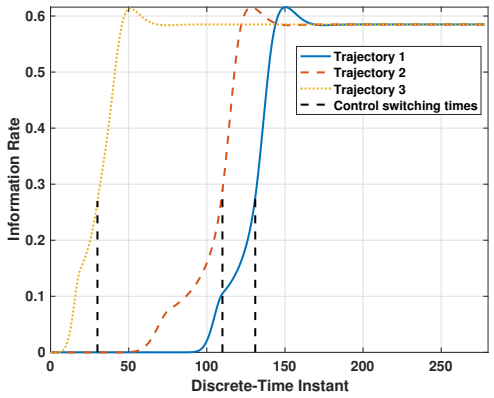
Remark 4.5. *The simulations presented here demonstrate that the proposed methodology is capable of generating trajectories useful for information collection. As currently implemented, these trajectories end with the inspector spacecraft "hovering" above the target point. Additional rules could be developed to safely transition the spacecraft to a closed NMT, or unforced equilibria, after a desired amount of information is collected. Development of such rules is left to future work.*



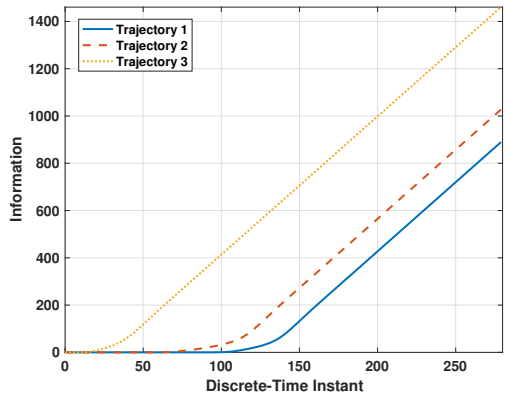
(a) Trajectories



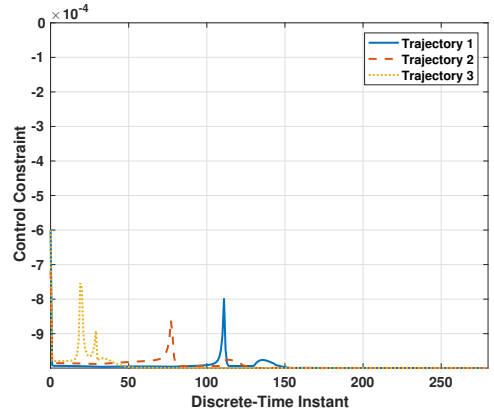
(b) Close-up view of trajectories



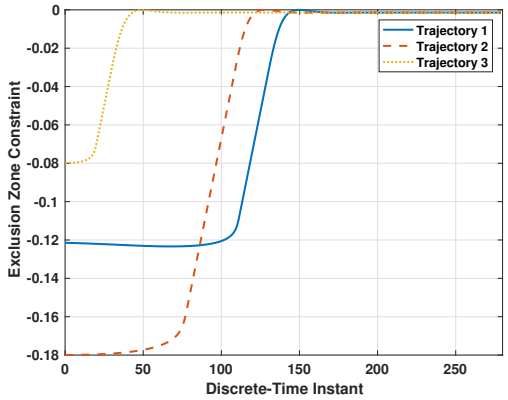
(c) Information rate



(d) Total information



(e) Control constraint



(f) Exclusion zone constraint

Figure 4.4: Example trajectories and associated data

To demonstrate that the methodology is robust to choice of initial conditions, simulations are run from 369 initial state vectors. Of these initial states, 342 have position components evenly spread throughout a cube of $0.4 \times 0.4 \times 0.4$ km, centered at the origin, and velocity components all set to 0. The remaining 27 initial states are along closed elliptical NMTs. The resulting trajectories are shown in Figure 4.5, and a close-up view of the portion of these trajectories near the exclusion zone is shown in Figure 4.6. In Figure 4.6, note that some of the trajectories exhibit a slight chatter back and forth along the $\phi = 0$ line before reaching the control switching region. In Section 4.5, several ways to address this undesirable behavior are discussed and compared.

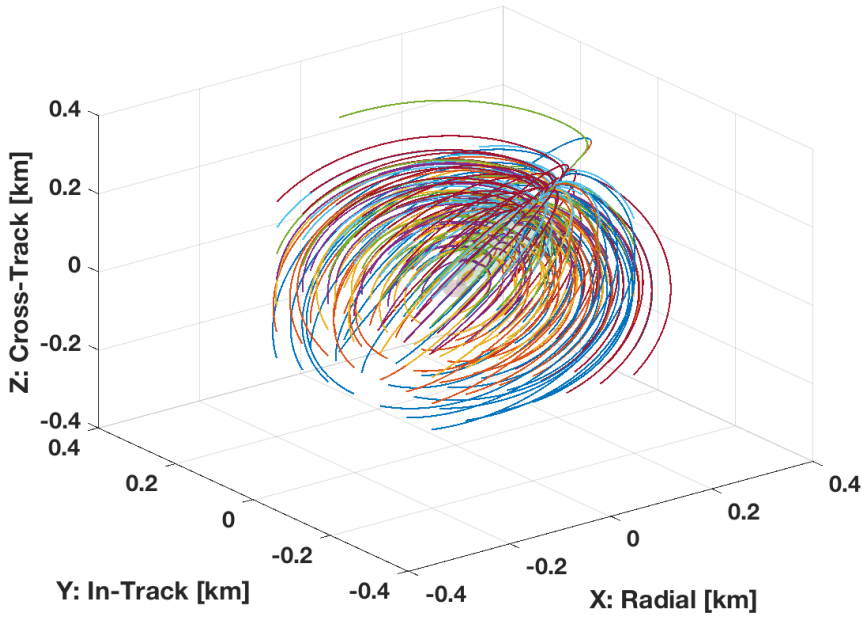


Figure 4.5: Closed-loop trajectories from a variety of initial conditions

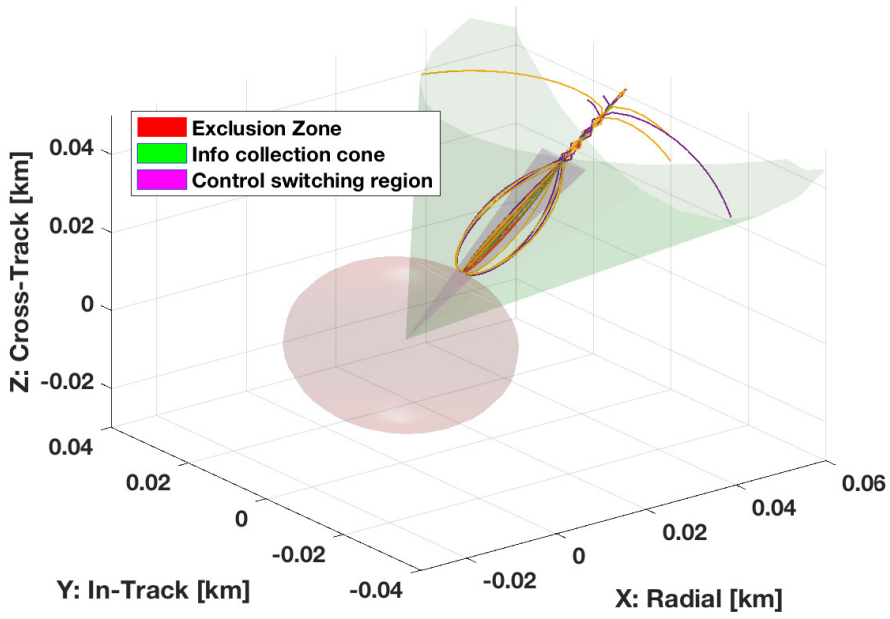


Figure 4.6: Close-up of closed-loop trajectories near exclusion zone

4.5 Discussion

The tendency of trajectories to chatter around the $\phi = 0$ line, illustrated in Figure 4.6, is a result of two factors. Firstly, the gradient of the information rate points toward the $\phi = 0$ line at all points near this line (see Figure 4.3). Secondly, the inspector spacecraft dynamics are modeled in discrete-time. Therefore, when near the $\phi = 0$ line, the satellite state can transition from one side of the $\phi = 0$ line to the other, and, hence, at the next time instant the LG control law pushes the satellite back in the opposite direction.

Below, four options are proposed to reduce the prevalence and severity of this chatter. These options are based on either modifying parameters in the inspector spacecraft dynamics, or modifying parameters in the criteria used to switch from the LG control law to the state-feedback control law, originally described in Section 4.3.3. Figure 4.7 illustrates the results of using each option[2]. The trajectories plotted in Figures 4.7a-d are based on the same set of initial states used to generate the trajectories in Figures 4.5 and 4.6.

Option 1: Shorten the discrete-time update period, ΔT. This option results in the inspector spacecraft covering less distance in each discrete-time update period, thereby

[2]The discontinuous line-segments appearing in 4.7c-d are portions of trajectories. These trajectories only appear discontinuous due to the axis limits selected for these figures.

reducing chatter. Figure 4.7a shows trajectories with $\Delta T = 5$ sec (reduced from $\Delta T = 10$ sec in Figure 4.6).

Option 2: Reduce the value of the LG control scale factor, u_{mag} in (4.37). This option also results in the inspector spacecraft covering less distance in each discrete-time update period, thereby reducing chatter. Figure 4.7b shows trajectories with $u_{mag} = 0.0002$ (reduced from $u_{mag} = 0.0004$ in Figure 4.6).

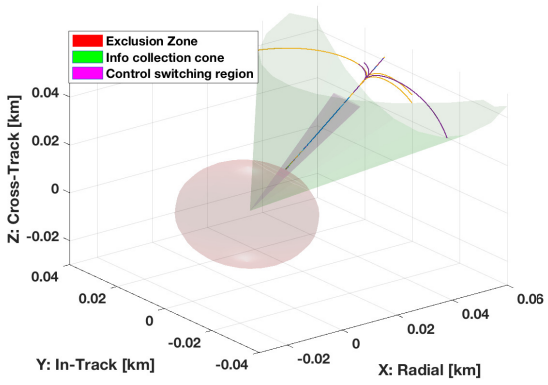
Option 3: Increase the value of r_s, used to determine when to switch from the LG control law to the state-feedback control law. In Figure 4.6, note that the chattering behavior occurs only when the inspector spacecraft is near the exclusion zone. Hence, because this option results in the state-feedback law being switched on when the inspector spacecraft is farther from the target point, the chatter is eliminated. Figure 4.7c shows trajectories with $r_s = 0.1$ km (increased from $r_s = 0.05$ km in Figure 4.6).

Option 4: Eliminate the parameter r_s from the control switching criteria, i.e., switch to the state-feedback control law at the first time-instant when $\phi \leq \phi_s$. This option also results in the state-feedback law being switching on when the inspector spacecraft is farther from the target point (in most cases), and therefore eliminates chatter. Figure 4.7d shows trajectories with switching determined based only on angle, i.e., when $\phi \leq \phi_s$.

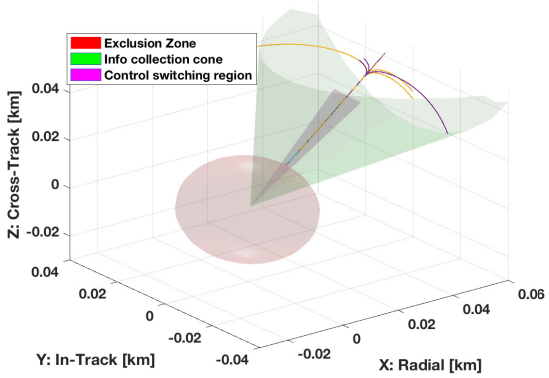
To compare the utility of Options 1-4 for satellite inspection purposes, simulations are run using each method from the 369 initial state vectors described in Section 4.4 and used to generate Figures 4.5-4.7. Each trajectory is propagated for a total of $3,000$ seconds, i.e., over discrete-time instants $k = 0, 1, \cdots, k_f = \frac{3{,}000}{\Delta T}$. For each trajectory, the total information obtained is calculated using (4.5) and the total control used along each trajectory is calculated as

$$u_{tot} = \sum_{k=0}^{k_f - 1} \|u(k)\|_1. \tag{4.40}$$

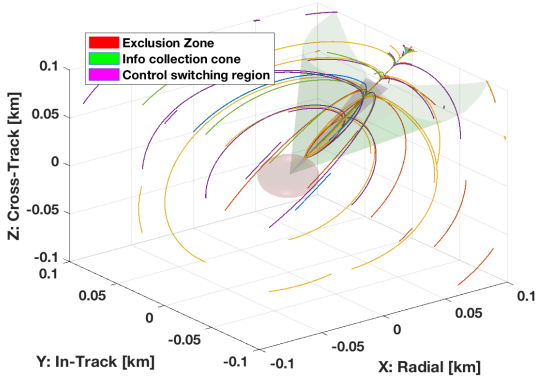
In Table 4.3, the number of initial states with trajectories corresponding to maximum and minimum information obtained, and maximum and minimum control used, is reported for each option. This analysis suggests that if control usage is an important consideration, Option 2 may be preferable. Conversely, if fast information collection is important, Option 4 may be the best solution. Further analysis to determine what values of ΔT, u_{mag}, r_s and ϕ_s result in the best performance for a given mission application is left to future work.



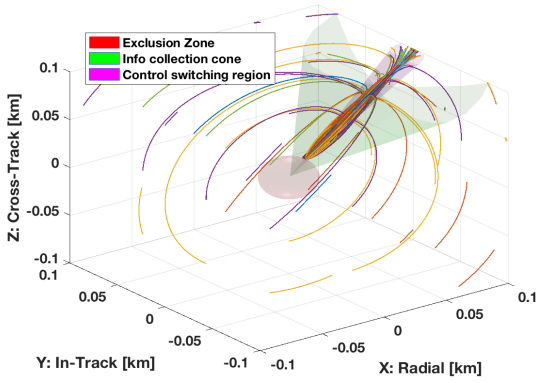
(a) Trajectories using Option 1 - reduced ΔT



(b) Trajectories using Option 2 - reduced u_{mag}



(c) Trajectories using Option 3 - increased r_s



(d) Trajectories using Option 4 - angle-only switching criteria

Figure 4.7: Simulations illustrating different options to reduce chatter

Table 4.3: Number of trajectories with Max/Min information collection and control used for Options 1-4

	Max Info	**Min Info**	**Max Control**	**Min Control**
Option 1	26	0	6	0
Option 2	0	369	0	369
Option 3	3	0	20	0
Option 4	340	0	343	0

Remark 4.6. *Note that Option 1 (reducing ΔT) may be infeasible due to computational considerations or sensor limitations, i.e., a certain amount of time may be required to obtain sensor readings needed to produce a state estimate.*

Remark 4.7. *Other options may also be useful in improving performance. For example, modifications to other parameters, such as the angular tolerance ϵ used to patch singularities in the gradient (described in Section 4.3.1.1), or the gain matrix K used in the state-feedback control law may also have an effect on control used and/or information obtained along trajectories. Additionally, other potential options to reduce chatter include defining the direction of the gradient near the $\phi = 0$ line using rules based on a hysteresis region [115], rather than the simple rules described in Section 4.3.1.1. Investigation of these options is left to future work.*

4.6 Robustness to Disturbances

To demonstrate that the proposed methodology is robust to sufficiently small disturbances, consider inspector spacecraft dynamics modeled by the discrete-time CW equation with additive disturbances, i.e.,

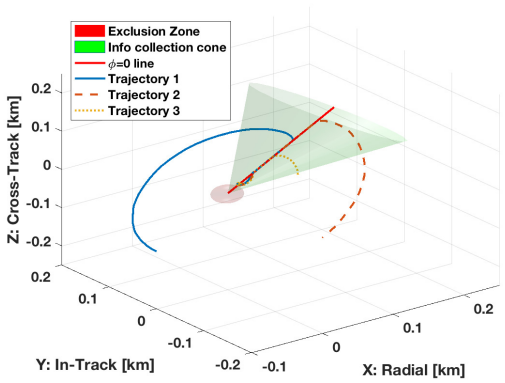
$$X(k+1) = AX(k) + Bu(k) + Bw(k), \tag{4.41}$$

where $w(k) \in \mathbb{R}^3$ is the disturbance vector randomly assigned from a uniform distribution over an infinity-norm ball centered at the origin with a radius of 0.025 m/sec. Simulations using these dynamics show that closed-loop trajectories using the LG control law are relatively unchanged. Hence, it is expected that constraints will remain satisfied when the LG control law is used, even with small disturbances are considered. Furthermore, as expected, when disturbances are included, closed loop trajectories using the state feedback control law converge to a small region around the controller reference point, rather than exactly to the reference point. Hence, to ensure that the exclusion zone constraint remains satisfied, the reference point must be chosen to be "sufficiently far" from the exclusion zone. Simulations imply that, for the simulation parameters considered here, requiring the controller reference point $X_d(k)$ in (4.34) to be at least three meters outside of the exclusion zone is sufficient to guarantee constraint satisfaction.

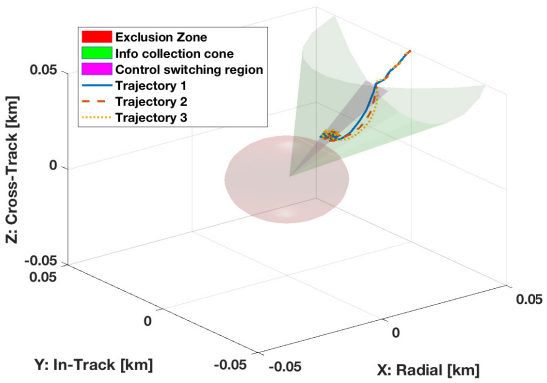
To illustrate the preceding points, three simulations are run from the initial states given by (4.39), using satellite dynamics given by (4.41), simulation parameters given in Table 4.2, and the nominal control switching rules described in Section 4.3.3. Simulation data, including trajectories, information collection data, and constraint data are shown in Figure

4.8. Note that constraints are satisfied along all three trajectories.

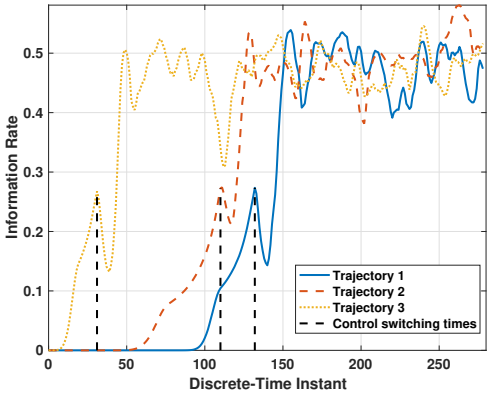
To identify the impact of disturbances, the data in Figure 4.8 are compared to Figure 4.4, which contains data for trajectories starting from the same initial states, but without disturbances. In Figures 4.8a and 4.4a, note that the beginning portions of the trajectories are nearly identical and in Figures 4.8c and 4.4c, note that the control switching times are similar. This illustrates that small disturbances have little impact on the closed loop trajectories when the LG control law is used. A comparison of 4.8c and 4.4c also shows that when disturbances are included, the information collection rate exhibits greater variability after the control switching time, and, for large k, the information collection rate with disturbances is lower than it is without disturbances. These differences are due to the disturbances pushing the inspector spacecraft away from the specified controller reference point, and due to this reference point being farther from the target point when disturbances are included.



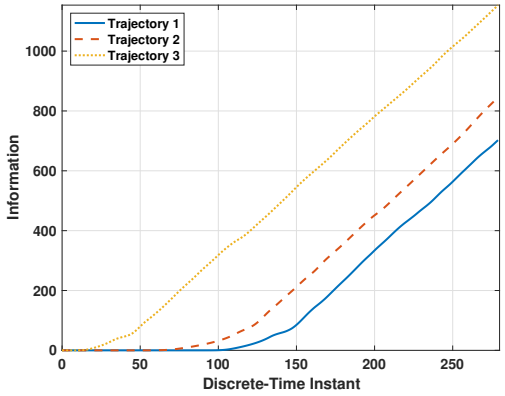
(a) Trajectories



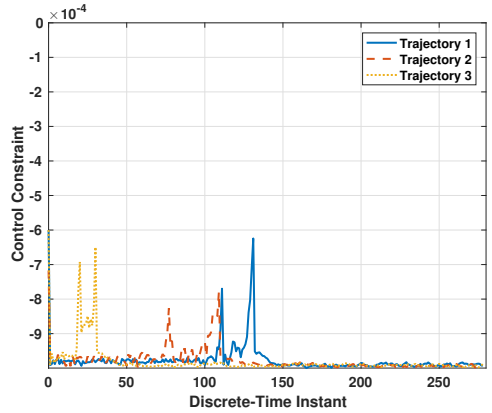
(b) Close-up view of trajectories



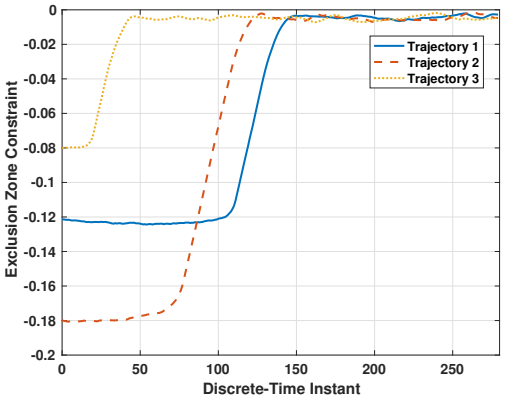
(c) Information rate



(d) Total information



(e) Control constraint



(f) Exclusion zone constraint

Figure 4.8: Data for trajectories including bounded disturbances

CHAPTER 5

Parameter Governors for Constrained Control of Multi-agent Formations

In this chapter, an add-on predictive controller known as a Parameter Governor (PG) is developed and applied to generate and maintain a formation of agents (vehicles). The PG is applied to a nominal inner-loop system, composed of an arbitrary number of agents, and modifies parameters, such as gains or offsets, to enforce point-wise in time constraints on state and control variables, and to improve system performance. PGs were first proposed in [73], and were demonstrated to be effective in enforcing constraints for a monolithic systems. To the author's knowledge, the work in this dissertation (and associated publications [78, 79]) represents the first application of PGs to formation control, or to control of distributed systems in general.

As discussed briefly in Chapter 1, as an add-on control scheme, the PG is similar to the RG and CG [63]. The PG enjoys several advantages compared to either the RG or CG. Firstly, the PG offers additional flexibility: while the RG and CG are limited to only making adjustments to the system reference, the PG can, depending on the types of parameters chosen, either adjust the system reference or make modifications to the inner-loop system dynamics through, e.g., adjustments to controller gains. Secondly, the PG developed here offers the benefit of fast and straightforward calculations even when constraints are non-convex. Specifically, the parameters can be confined to a finite set of small cardinality, and updated parameter values can be chosen by running a small number of simulations and choosing the values that minimize (or possibly simply decrease) a cost function subject to constraints. In this way, computations remain fast and tractable even if the optimization problem and/or the constraints are non-linear or non-convex. Figure 5.1 provides a general schematic of a PG applied to a nominal inner-loop system composed of q individual subsystems.

This chapter is organized as follows. Notation is summarized in Section 5.1, and in Section 5.2, the PG is introduced and analyzed in a general setting. Section 5.2.1 defines

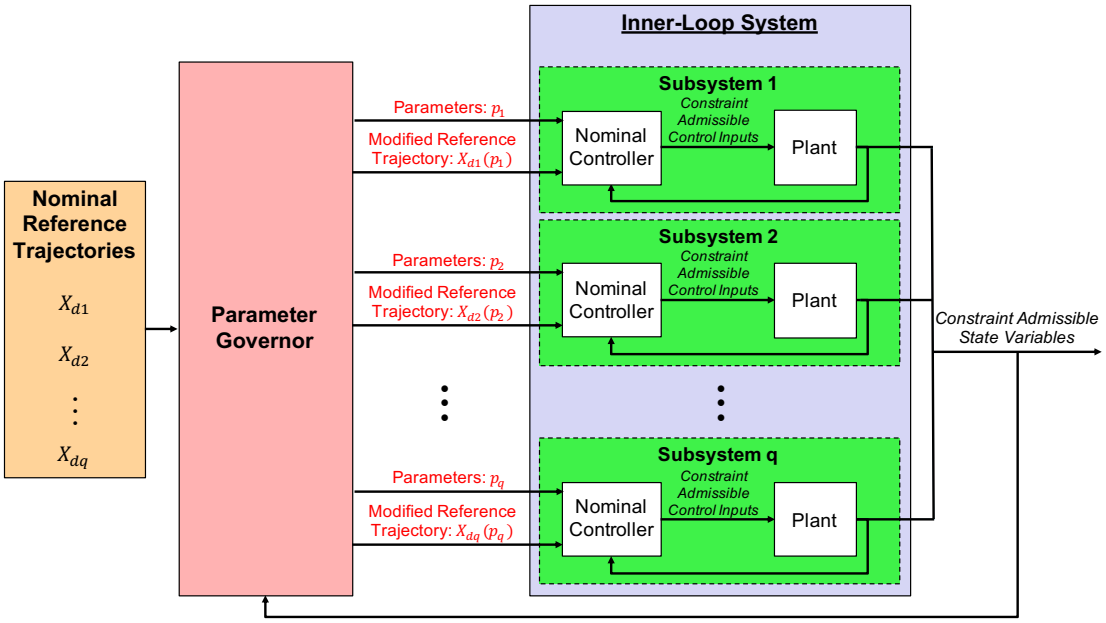


Figure 5.1: A general schematic of the parameter governor

the system dynamics and control law for each agent in the formation and Section 5.2.2 describes how the overall inner-loop system is composed from a collection of agents. The PG is defined in Section 5.2.3, including the set of admissible parameters, and the form of the cost function used to determine parameter updates. In Sections 5.2.1-5.2.3, required properties of the system dynamics, controller and cost function are clearly stated. Section 5.2.4, describes the method to obtain updated parameter values at each time instant, and a convergence analysis is presented in Section 5.2.5, showing that, under reasonable assumptions, the PG is guaranteed to generate and maintain the desired formation. The general framework for PGs developed in Section 5.2 can be applied to a variety of distributed systems, and is not limited to FF applications. In Section 5.3, two specific PGs, referred to as the SSG and TSG, respectively, are proposed and applied to the specific problem of FF. Simulation results demonstrate that each PG is capable of generating and maintaining formations of spacecraft while satisfying constraints, including non-convex constraints that ensure no collisions between spacecraft.

5.1 Notation

For convenience, a summary of the notation used in this chapter is as follows. The set of integers is $\mathbb{Z}$ and the set of real numbers is $\mathbb{R}$. Subsets of these sets are identified by a

subscript, e.g., $\mathbb{Z}_{[0,T]}$ denotes the set of integers between 0 and T, and $\mathbb{R}_{\geq 0}$ denotes the set of non-negative real numbers. A superscript b appended to a set Λ denotes the bth order Cartesian product of the set, i.e., $\Lambda^3 = \Lambda \times \Lambda \times \Lambda$. For a vector v, a subscript of an i, a j or an integer denotes a subsystem, e.g., v_i is the vector v for the ith subsystem. A superscript of an i, a j, or an integer appended to a vector denotes the ith element of the vector, e.g., v^i denotes the ith element of vector v. This notation is abused slightly in the case of a superscript 0. In this case v^0 denotes a nominal value for the vector v. The p-norm of a vector v is denoted $\|v\|_p$. A superscript appended to this notation denotes the vector norm raised to a power, i.e., $\|v\|_2^3 = \|v\|_2 \cdot \|v\|_2 \cdot \|v\|_2$. For a vector v and a square positive definite matrix Ξ, $\|v\|_\Xi^2 = v^T \Xi v$.

The symbol $k \in \mathbb{Z}$ denotes the discrete-time instant. The predicted value of a variable s, which is a function of variable r, at time instant $k+\sigma$ when the prediction is made at time instant k is denoted by $s(k+\sigma|k, s(k), r)$. The shorthand $s(k+\sigma|k)$ is occasionally used for brevity when there is no ambiguity. A normed unit ball is denoted by $\mathcal{B}$ and, finally, the modulo function mod(x, y) returns the remainder after division of x by y.

5.2 General Framework

5.2.1 Individual Agent Modeling

Consider a system composed of q subsystems (agents) and let $\mathbb{S} = \{1,\ 2,\ \cdots,\ q\}$. The state $X_i \in \mathbb{R}^l$ of each subsystem $i \in \mathbb{S}$ evolves in discrete-time according to

$$X_i(k+1) = f(k,\ X_i(k),\ u_i(k)), \tag{5.1}$$

where $u_i(k) \in \mathbb{R}^m$ is the control for the ith subsystem. The function f is identical for each subsystem, and is assumed to satisfy the following property:

P1) f is globally Lipschitz in k, X_i and u_i.

The control objective is to guide the subsystems to reference points, denoted by $X_{di}(k)$, along nominal reference trajectories, while satisfying constraints on both state and control variables. The nominal reference trajectory for the ith subsystem is defined by a nominal control input of $\bar{u}_{di}(k)$, and a specified initial condition, $\bar{X}_{di}^0$, and is given by

$$\bar{X}_{di}(k+1) = f(k, \bar{X}_{di}(k), \bar{u}_{di}(k)),\ \bar{X}_{di}(0) = \bar{X}_{di}^0. \tag{5.2}$$

The controller $u_i(k)$ in (5.1) contains both a feed-forward term, and a feedback term

designed to track the specified reference trajectory, i.e.,

$$u_i(k) = u_{di}(k, p_i(k)) + u_{fi}(p_i(k), X_i(k), X_{di}(k, p_i(k))), \tag{5.3}$$

where $p_i(k) \in \mathbb{P} \subset \mathbb{R}^n$ is a vector of parameters that are adjusted by a PG in order to enforce constraints and improve system performance. The PG, and the selection of $p_i(k)$, are described in detail in Sections 5.2.3 and 5.2.4, respectively. The feedback portion of the control law (5.3) is assumed to satisfy the following property:

P2) u_{fi} is continuous in its arguments and $u_{fi}(p_i(k), X_{di}(k, p_i(k)), X_{di}(k, p_i(k))) = 0$ for all $p_i(k) \in \mathbb{P}$.

Note that the latter property simply states that if the current subsystem state is equal to the current controller reference point, i.e., $X_i(k) = X_{di}(k, p_i(k))$, then the feedback control is 0.

In (5.3), the feed-forward term, $u_{di}(k, p_i(k))$, and the controller reference point, $X_{di}(k, p_i(k))$, in the feedback term are related to the nominal reference trajectory and control inputs by

$$u_{di}(k, p_i(k)) = \bar{u}_{di}(k + \theta_i, p_i(k)), \tag{5.4}$$

and

$$X_{di}(k, p_i(k)) = \bar{X}_{di}(k + \theta_i, p_i(k)), \tag{5.5}$$

respectively, where $\theta_i \in \mathbb{Z}$ is a constant time-shift specifying the desired phasing along the nominal reference trajectory. In (5.3)-(5.5), the additional argument $p_i(k)$ in u_{fi}, $\bar{u}_{di}$ and $\bar{X}_{di}$, respectively, is included to show that the parameter governor may either modify the nominal target provided to each subsystem or modify the action of the feedback portion of the control law.

It is assumed that the control $u_i(k)$ is stabilizing to the reference point when any adjustments made to the parameter do not change the reference trajectory, i.e., the following property holds

P3) $X_i(k + \sigma|k) \to X_{di}(k + \sigma, p_i(k + \sigma))$ as $\sigma \to \infty$
whenever $X_{di}(k + \sigma, p_i(k)) = X_{di}(k + \sigma, p_i(k + \sigma))$, $p_i(k + \sigma) \in \mathbb{P}$ for all $\sigma \in \mathbb{Z}_{\geq 0}$.

Note that line 2 of property (P3) holds if $p_i(k)$ is held fixed, i.e., if $p_i(k) = p_i(k + \sigma)$ for all $\sigma \in \mathbb{Z}_{\geq 0}$.

Based on (5.5) and (5.2), the controller reference point evolves according to

$$X_{di}(k + 1, p_i(k + 1)) = f(k, X_{di}(k, p_i(k + 1)), u_{di}(k, p_i(k + 1))). \tag{5.6}$$

5.2.2 Overall Inner-loop System

The overall inner-loop system is composed from the q subsystems and has state, control, parameter, and reference (controller reference point) vectors corresponding to

$$\begin{aligned} X(k) &= [X_1(k)^T,\ X_2(k)^T,\ \cdots,\ X_q(k)^T]^T, \\ u(k) &= [u_1(k)^T,\ u_2(k)^T,\ \cdots,\ u_q(k)^T]^T, \\ p(k) &= [p_1(k)^T,\ p_2(k)^T,\ \cdots,\ p_q(k)^T]^T, \\ X_d(k, p(k)) &= [X_{d1}(k, p_1(k))^T,\ X_{d2}(k, p_2(k))^T,\ \cdots,\ X_{dq}(k, p_q(k))^T]^T, \end{aligned} \tag{5.7}$$

respectively. Pointwise-in-time constraints are imposed on the state, control and parameter of the overall system as

$$X(k) \in \mathbb{X},\ u(k) \in \mathbb{U},\ p(k) \in \mathbb{P}^q, \tag{5.8}$$

where $\mathbb{X}$, $\mathbb{U}$ and $\mathbb{P}$ are compact sets. The sets $\mathbb{X}$ and $\mathbb{U}$ are defined in terms of inequalities, i.e,

$$\mathbb{X} = \{X \mid y^x_{ci}(X) \leq 0, i = 1, 2, \cdots, n_x\}, \tag{5.9}$$

and

$$\mathbb{U} = \{u \mid y^u_{ci}(u) \leq 0, i = 1, 2, \cdots, n_u\}, \tag{5.10}$$

where each constraint y^x_{ci} and y^u_{ci} may be a function of the overall system state or control, respectively, e.g., $y^x_{ci} = y^x_{ci}(X(k))$, a function of the state or control of two or more subsystems, e.g., $y^x_{ci} = y^x_{ci}(X_1(k), X_2(k))$, or a function of the state or control of a single subsystem, e.g., $y^x_{ci} = y^x_{ci}(X_1(k))$. The parameter set $\mathbb{P}$ is defined in more detail in the next subsection.

Because the nominal control law for each system is stabilizing to the reference point (due to property P3), this control law can be used to generate and maintain the desired formation without the addition of a PG, i.e., with no PG, the subsystems would converge to reference points corresponding to $\bar{X}_{di}(k + \theta_i)$ along their nominal reference trajectories. However, with this simple approach, there is no guarantee that the system response would satisfy constraints. The addition of a PG provides this guarantee of constraint satisfaction, while requiring minimal additional calculations at each discrete-time instant.

5.2.3 The Parameter Governor

The PG is added to the nominal inner-loop system, and selects $p(k)$ at each discrete-time instant in order to enforce constraints and improve overall system performance. Figure 5.2

illustrates the application of the PG to the nominal inner-loop system.

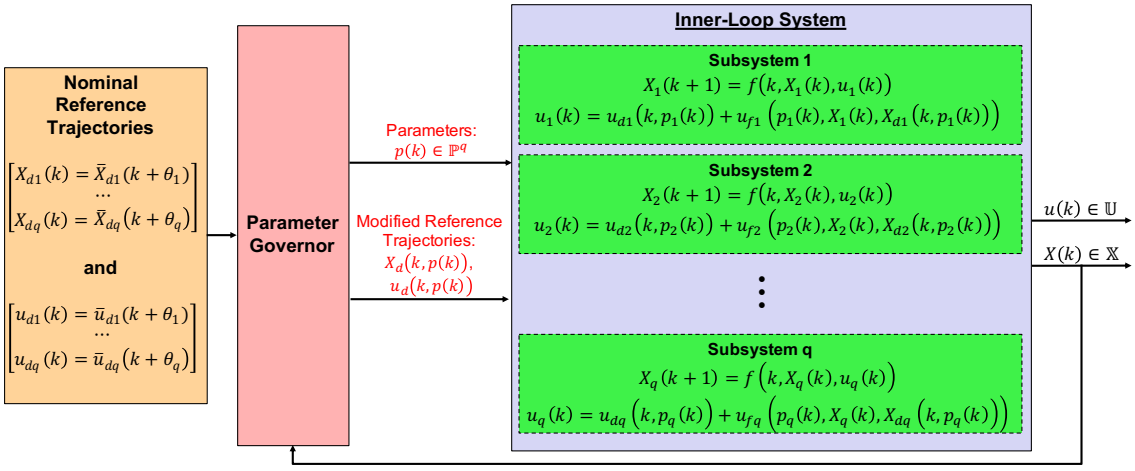


Figure 5.2: The parameter governor

The parameter vector $p_i(k)$ for each subsystem $i \in \mathbb{S}$ is selected from a prescribed set,

$$p_i(k) \in \mathbb{P} \subset \mathbb{R}^n. \tag{5.11}$$

The set $\mathbb{P}$ is defined as the Cartesian product of n finite sets, $\mathbb{P}_j$, $j = 1, 2, ..., n$, i.e.,

$$\mathbb{P} = \mathbb{P}_1 \times \mathbb{P}_2 \times \cdots \times \mathbb{P}_n, \tag{5.12}$$

where

$$\mathbb{P}_j = \{p^j_{min},\ p^j_{min} + \zeta^j,\ p^j_{min} + 2\zeta^j, \cdots, p^j_{min} + \alpha^j\zeta^j\} \subset \mathbb{R},\ j = 1, 2, \cdots, n, \tag{5.13}$$

and where $p^j_{min} \in \mathbb{R}$ is the minimum value for the jth element of the parameter vector, $\zeta^j \in \mathbb{R}_{>0}$ is the step size for the jth element of the parameter vector, and $\alpha^j \in \mathbb{Z}_{\geq 0}$ defines the number of elements in the set $\mathbb{P}_j$. Note that p^j_{min}, ζ^j and α^j are all fixed, chosen values (see Remark 5.1), and that the set $\mathbb{P}$ is compact, and has finite cardinality, i.e., has a finite number of elements. Note also that the parameter vector for the combined system belongs to the set $\mathbb{P}^q$, i.e., $p(k) \in \mathbb{P}^q$.

Remark 5.1. *Choosing a smaller value for p^j_{min}, a larger value for ζ^j or a larger value for α^j results in a larger range of parameter values. This may improve the performance of the PG by providing additional flexibility to enforce constraints. Choosing a smaller value for ζ^j facilitates satisfaction of Assumption (A5) used in the convergence analysis in Section*

5.2.5; hence, smaller ζ^j may enable the parameter governor to obtain convergence to the desired formation. However, smaller ζ^j values may also result in longer convergence times because, using the parameter update strategy described in Section 5.2.4, a smaller step-size results in more adjustments that must be made before reaching the desired parameter values.

Define the minimum and maximum parameter step sizes as

$$\zeta_{min} = \min\{\zeta^j,\ j = 1, 2, \cdots, n\}, \tag{5.14}$$

and

$$\zeta_{max} = \max\{\zeta^j,\ j = 1, 2, \cdots, n\}. \tag{5.15}$$

The parameter vector is selected at each discrete-time instant to minimize, or possibly simply to decrease (when feasible) the following cost function over a horizon of T steps:

$$J(k, p(k), X(k)) = W(p(k)) + \Omega(k, p(k), X(k)), \tag{5.16}$$

subject to the condition that constraints are satisfied with the parameter held constant over the prediction horizon, i.e., $p(k+\sigma|k) = p(k) \in \mathbb{P}^q$ and $X(k+\sigma|k) \in \mathbb{X},\ u(k+\sigma|k) \in \mathbb{U}$ for $\sigma = 0, 1, ..., T$. In (5.16), the term W is the *parameter cost* that depends only on the parameter value for the combined system. The term Ω is the *incremental cost* that may depend on the parameter, control and state of the combined system. The functions W and Ω can be flexibly defined by the user, subject to properties (P4)-(P7) below, based on the desired formation (system configuration), the types of parameters included in p, and desired performance characteristics.

The function $W(p(k)) : \mathbb{P}^q \to \mathbb{R}_{\geq 0}$ is constructed such that the following properties hold:

P4) $W(p)$ is continuous in p and bounded for $p \in \mathbb{P}^q$,

P5) If $W(p) = 0$, then p corresponds to the desired overall system configuration, and,

P6) Whenever $W(p) > 0$, there exists $\hat{p} \in \mathbb{P}^q$ such that

$$W(\hat{p}) \leq W(p) - \zeta^i, \tag{5.17}$$

where

$$\begin{aligned} p &= [p^1\ p^2 \cdots p^i \cdots p^{nq}]^T, \\ \hat{p} &= [p^1\ p^2 \cdots \hat{p}^i \cdots p^{nq}]^T, \\ \hat{p}^i &\in \{p^i + \zeta^i,\ p^i - \zeta^i\} \cap \mathbb{P}_i, \end{aligned} \tag{5.18}$$

ζ^i is the step size for parameter p^i, and $\mathbb{P}_i$ is the admissible set of values for parameter p^i. This property ensures that whenever $W(p(k)) > 0$, there always exists a single parameter that may be adjusted by a single step to obtain a decrease in W. See Remark 5.2.

Remark 5.2. *Note that if the current value for an element of the parameter vector is on the boundary of the admissible set, adjustments to this element can only be made in a single direction, e.g., if $p^i = p^i_{min}$, then the value $\hat{p}^i = p^i - \zeta^i$ is not feasible. This restriction is captured in line 3 of (5.18) by intersecting the set of possible values for $\hat{p}^i$ with the admissible set: $\hat{p}^i \in \{p^i + \zeta^i,\ p^i - \zeta^i\} \cap \mathbb{P}_i$.*

Designing a function $W(p)$ to satisfy properties (P4)-(P6) is straightforward. For example, suppose there exist desired values for each element of the parameter vector, denoted by p^i_d, $i = 1, 2, ..., nq$. Then, the function $W(p) = \sum_{i=1}^{nq} |p^i - p^i_d|$ satisfies properties (P4)-(P6).

The function $\Omega(k, p(k), X(k)) : \mathbb{Z} \times \mathbb{P}^q \times \mathbb{R}^{ql} \to \mathbb{R}_{\geq 0}$ in (5.16) contains terms that penalize deviation from the reference trajectory,

$$\begin{aligned} \Omega(k, p(k), X(k)) = \sum_{i=1}^{q} \sum_{\sigma=0}^{T} \quad & \{E_i[X_i(k+\sigma|k) - X_{di}(k+\sigma|k, p_i(k))] + \\ & U_i[u_i(k+\sigma|k) - u_{di}(k+\sigma|k, p_i(k))]\}, \end{aligned} \tag{5.19}$$

and is constructed to satisfy the following property:

P7) The functions E_i and U_i are continuous and strictly positive-definite, i.e.,

$$\begin{aligned} E_i(X_i - X_{di}) &= 0 \quad \text{if } X_i - X_{di} = 0, \\ E_i(X_i - X_{di}) &> 0 \quad \text{otherwise}, \end{aligned} \tag{5.20}$$

and

$$\begin{aligned} U_i(u_i - u_{di}) &= 0 \quad \text{if } u_i - u_{di} = 0, \\ U_i(u_i - u_{di}) &> 0 \quad \text{otherwise}. \end{aligned} \tag{5.21}$$

Based on property (P7), it follows that $\Omega = 0$ if and only if each subsystem has reached its specified location along the reference trajectory. Designing a function $\Omega(k, p(k), X(k))$ to satisfy property (P7) is straightforward. For example, Property (P7) can be satisfied by taking $E_i = \|X_i - X_{di}\|^2_\Phi$ and $U_i = \|u_i - u_{di}\|^2_\Theta$, where $\Phi \succ 0$ and $\Theta \succ 0$. Note that based

on the form of the cost (5.16), i.e., $J = W + \Omega$, and Properties (P5) and (P7), the desired formation is attained if $J = 0$.

5.2.4 Parameter Update Strategy

As mentioned previously, the PG updates the parameter vector p at each discrete-time instant to minimize, or possibly simply decrease, the cost J subject to constraints. There are several possibilities for how the updated parameter values can be obtained. In this section, a centralized update strategy is first proposed and its limitations are identified. Then, a turn-based strategy is introduced that can be used to obtain convergence to the desired formation at a lower computational cost.

5.2.4.1 Centralized Strategy

In a centralized parameter update strategy, the parameter vectors for each subsystem are updated concurrently at each time instant. Such a centralized scheme determines the parameter vector $p(k)$ by obtaining the solution to the following optimization problem at each discrete-time instant:

$$\begin{aligned} \min_{p(k)} \quad & J(k, p(k), X(k)) \\ \text{subject to} \quad & X(k+\sigma|k) \in \mathbb{X},\ \sigma \in \mathbb{Z}_{[0,T]}, \\ & u(k+\sigma|k) \in \mathbb{U},\ \sigma \in \mathbb{Z}_{[0,T-1]}, \\ & p(k+\sigma|k) = p(k),\ \sigma \in \mathbb{Z}_{[0,T]}, \\ & p(k) \in \mathbb{P}^q, \\ & X(k|k) = X(k). \end{aligned} \tag{5.22}$$

By finding the exact minimizer to (5.22) at each time-step, this centralized method may provide the fastest convergence and result in trajectories with the lowest overall cost. However, if the constraint sets $\mathbb{X}$ and/or $\mathbb{U}$ are non-convex, obtaining the exact minimizer may be complicated. Furthermore, if the number of subsystems and/or parameters is large, the number of adjustable variables in (5.22) becomes large, which also complicates computations. These difficulties may be partially overcome if the parameter set $\mathbb{P}$ is discrete, as in this case the solution to (5.22) can be obtained by running a finite number of simulations. However, again, if the number of subsystems and/or parameters is large, running the full set of simulations may be time consuming and therefore not possible in real-time.

Another limitation of the centralized scheme is that the calculations must either be done at a centralized location, or within a single subsystem, and then the solution ($p(k)$) must

be shared with all other subsystems at each time instant so that each subsystem can apply its new parameter values. This implies that the communication graph of the overall system must be connected, i.e., there must be a communication path from the centralized location to each subsystem, at each time instant.

To overcome these issues, while still maintaining desired convergence properties, a "turn-based" parameter update strategy is used, based on a similar strategy for distributed command governors introduced in [69, 70].

5.2.4.2 Turn-based strategy

In the turn-based strategy, rather than updating all subsystem parameters concurrently at each time instant, only a subset of subsystems update their parameters at a given time instant while all other subsystem parameters are held fixed. The turn-based strategy is based on the following definitions:

Definition 5.1 (Neighborhood)**.** *The neighborhood of the ith subsystem, denoted by $\mathbb{N}_i$, consists of all subsystems $j \in \mathbb{S}$ for which either of the following properties hold:*

a) the jth subsystem evolution is jointly constrained with the ith subsystem evolution, i.e., there exists a constraint of the form $y_c^x(X_i(k), X_j(k))$ or $y_c^u(u_i(k), u_j(k))$,

b) the parameter cost function $W(p(k))$ in (5.16) contains one or more terms coupling the ith and jth subsystems.

If subsystem $j \in \mathbb{N}_i$, then subsystem j is referred to as the "neighbor" of subsystem i. Note that $j \in \mathbb{N}_i \iff i \in \mathbb{N}_j$.

Definition 5.2 (Turn [69])**.** *A turn $\mathbb{T} \subset \mathbb{S}$ is a subset of non-neighboring subsystems, i.e.,*

$$\forall i, j \in \mathbb{T} \text{ such that } i \neq j, \text{it follows that } j \notin \mathbb{N}_i. \tag{5.23}$$

With these definitions, it is clear that for a given turn at time instant k, $\mathbb{T}_k$, if $i,\ j \in \mathbb{T}_k$ then $p_i(k)$ and $p_j(k)$ can be updated independently and concurrently while guaranteeing that all constraints will be satisfied. This fact leads to the basic mechanism of the turn-based strategy: at time k, update $p_i(k)$ for all $i \in \mathbb{T}_k$ and set $p_i(k) = p_i(k-1)$ for all $i \notin \mathbb{T}_k$.

At a given time instant k, updates to each $p_i(k)$, $i \in \mathbb{T}_k$, are determined by solving a

local optimization problem, i.e.,

$$\begin{aligned} \min_{p_i(k)} \quad & J(k, p(k), x(k)) \\ \text{subject to} \quad & X(k+\sigma|k) \in \mathbb{X},\ \sigma \in \mathbb{Z}_{[0,T]}, \\ & u(k+\sigma|k) \in \mathbb{U},\ \sigma \in \mathbb{Z}_{[0,T-1]}, \\ & p(k+\sigma|k) = p(k),\ \sigma \in \mathbb{Z}_{[0,T]}, \\ & p_i(k) \in \Gamma(p_i(k-1)), \\ & X(k|k) = X(k), \end{aligned} \tag{5.24}$$

where $\Gamma(p_i(k-1))$ is defined as follows: For $p_i^0 \in \mathbb{P}$,

$$\begin{aligned} \Gamma(p_i^0) = \{p_i^0\} \cup \quad & \big\{p_i \mid p_i = p_i^0 + \delta, \text{ for all } \delta \in \{[\pm\zeta^1,\ 0, ...,\ 0]^T, \\ & [0,\ \pm\zeta^2, 0, ...,\ 0]^T, ..., [0, ..., 0, \pm\zeta^n]^T\} \text{ such that } p_i \in \mathbb{P}\big\}. \end{aligned} \tag{5.25}$$

In (5.24), the adjustable variables are restricted to $p_i(k) \in \Gamma(p_i(k-1)) \subset \mathbb{P}$. This restriction is useful because it facilitates obtaining the solution even when $\mathbb{X}$ or $\mathbb{U}$ are non-convex.Specifically, the solution to (5.24) can be obtained by running a maximum of 3^n simulations, where n is the number parameters that can be adjusted for a single subsystem. These simulations can be carried out easily, even when non-convex constraints are considered.

Remark 5.3. *Note that if the optimization problem (5.24) is convex, e.g., the objective function is convex in p and the sets $\mathbb{X}$, $\mathbb{U}$ and $\mathbb{P}$ are convex, then it may be possible to quickly perform the optimization over all $p_i(k) \in \mathbb{P}$ using standard convex optimization algorithms, see [116]. Such an approach could yield faster convergence, or improved system performance compared to the method described above, when $p_i(k)$ is restricted to $p_i(k) \in \Gamma(p_i(k-1))$. The choice to restrict $p_i(k)$ to $p_i(k) \in \Gamma(p_i(k-1))$ was made because in typical formation control settings, the constraint set $\mathbb{X}$ is non-convex due to collision avoidance requirements.*

To implement the turn-based update strategy, the sequence of turns $\mathbb{T}_k$, $k \in \mathbb{Z}_{\geq 0}$ must be defined. As discussed in [69], in order to ensure convergence, any sequence of turns must periodically include each subsystem of the overall system, i.e.,

$$\exists\, k' \in \mathbb{Z}_{>0} \text{ such that for all } k > 0,\ \cup_{i=0}^{k'} \mathbb{T}_{k+i} \supset \mathbb{S} = \{1, 2, \cdots, q\}. \tag{5.26}$$

A straightforward way to determine a sequence of turns satisfying (5.26) is to generate a

sequence of turns $\mathbb{T}_1,\ \mathbb{T}_2,\ \cdots,\ \mathbb{T}_c$ such that

$$\cup_{i=1}^{c} \mathbb{T}_i \supset \mathbb{S} = \{1, 2, \cdots, q\}. \tag{5.27}$$

Then, the overall sequence of turns is obtained by simply repeating the sequence $\mathbb{T}_1,\ \mathbb{T}_2,\ \cdots,\ \mathbb{T}_c$ until convergence is achieved, i.e.,

$$\mathbb{T}_k = \mathbb{T}_{\text{mod}(k,c)}. \tag{5.28}$$

A sequence $\mathbb{T}_1,\ \mathbb{T}_2,\ \cdots,\ \mathbb{T}_c$ satisfying (5.27) can be obtained by solving a vertex coloring problem, [117], for the graph $\mathcal{G}(\mathbb{S}, \mathbb{E})$ with one node (vertex) corresponding to each subsystem and the edge set $\mathbb{E} \subset \mathbb{S} \times \mathbb{S}$ connecting neighboring subsystems, i.e., $(i, j) \in \mathbb{E}$ if and only if $j \in \mathbb{N}_i$. The general vertex coloring problem is stated as follows:

Definition 5.3. *[Vertex Coloring Problem (VCP)] Find an assignment of colors to each node (vertex) of $\mathcal{G}(\mathbb{S}, \mathbb{E})$ such that, for all $i, j \in \mathbb{S}$, if $(i, j) \in \mathbb{E}$, then i and j have different colors.*

Several algorithms exist to obtain solutions to the VCP; some of these are discussed in Appendix F. Note that a trivial solution to the VCP is to assign a different color to each node.

In terms of fast response, it is desirable to determine the sequence of turns that maximizes the frequency with which each subsystem parameter is updated, i.e., determining the shortest sequence of turns satisfying (5.27). Obtaining this shortest sequence involves solving the *minimal* VCP:

Definition 5.4. *[Minimal Vertex Coloring Problem (mVCP)] Find the minimal number of colors that can be assigned to nodes (vertices) of $\mathcal{G}(\mathbb{S}, \mathbb{E})$, such that, for all $i, j \in \mathbb{S}$, if $(i, j) \in \mathbb{E}$, then i and j have different colors.*

Remark 5.4. *Note that the solution to the mVCP (commonly referred to as the "chromatic number" of a graph) is also a solution to the VCP. The mVCP is an NP-hard problem [118]. For formations with a small number of agents, or for graphs $\mathcal{G}(\mathbb{S}, \mathbb{E})$ with simple structures, e.g., complete graphs or cycles, it may be possible to quickly solve the minimal vertex coloring problem to obtain the minimal sequence of turns. In the general case, it is possible to solve the mVCP using brute force algorithms, or to obtain approximate solutions using heuristic-based algorithms. See Appendix F for additional discussion.*

After obtaining a solution to either the VCP or the mVCP, the composition of each turn in the repeating sequence $\mathbb{T}_1,\ \mathbb{T}_2,\ \cdots, \mathbb{T}_c$, where c is the number of colors used to color

the graph, is defined by grouping all nodes (subsystems) with the same color into a single turn.

The turn-based parameter update method is summarized in the following procedure.

Procedure 5.1 [Turn-based Parameter Update Procedure]

Inputs: Initial state: $X(0)$,

Nominal control for reference trajectories: $\bar{u}_{di}(k)$

Initial states for reference trajectories: $\bar{X}_{di}^0$

Constant time-shifts specifying desired relative phasing: θ_i

The sequence of turns: $\mathbb{T}_1,\ \mathbb{T}_2,\ \cdots,\ \mathbb{T}_c$

Initialization at time instant $k = 0$:

1. Obtain any feasible solution, p^0 to (5.22) and set $p(0) = p^0$,
2. Send $p(0)$ to all subsystems,
3. For all $i \in \mathbb{S}$, apply $p_i(0)$.

Repeat at each time instant $k > 0$:

4. If $i \in \mathbb{T}_{\text{mod}(k,c)}$
   (a) Obtain $X_j(k)$, $p_j(k)$ for all neighbors, i.e., all $j \in \mathbb{N}_i$
   (b) Solve (5.24) to obtain $p_i(k)$.
   (c) Send $X_i(k)$ and $p_i(k)$ to all neighbors.
5. If $i \notin \mathbb{T}_{\text{mod}(k,c)}$
   (a) Set $p_i(k) = p_i(k-1)$
6. Apply $p_i(k)$.

End Procedure.

In the initialization steps (1-3), any feasible solution to (5.22) can be used. Such a feasible solution could be calculated offline and sent to all subsystems in advance. In Step (4a), the state $X_j(k)$ is obtained by propagating the last state vector received from subsystem j to the current time-instant, i.e., $X_j(k) = X_j(k|\tilde{k}, X_j(\tilde{k}), p_j(\tilde{k}))$, where $\tilde{k}$ is the last time instant at which the jth subsystem parameter was updated. The parameter value $p_j(k) = p_j(\tilde{k})$ since each subsystem parameter is held fixed in between turns.

Using Procedure 5.1 to update the parameter leads to the following desirable properties:

1. Parameter updates are distributed over time and between spacecraft, i.e., each spacecraft must only update its own parameter every c time-steps.

2. The updated parameter value can be obtained by solving (5.24) by running, at most, 3^n simulations, where n is the number of parameters that can be adjusted for a single subsystem. (Note that this can be reduced further to only 3 simulations, by only considering changes to a single element of the parameter vector for a given turn, and convergence properties of the parameter governor are maintained; see Remark 5.5).

3. The number of simulations, i.e., computation time, required to update the parameter is independent of the number of subsystems. Therefore, computations do not increase as additional subsystems are added and larger formations are considered.

4. Parameter updates are made based only on local information, i.e., information from neighbors. This information only needs to be exchanged when one neighbor updates its parameter. (Note that this does not necessarily imply that a subsystem is able to communicate directly with all neighbors after it makes an update. Rather, this implies that there must exist a communication path (possibly including other subsystems) between any subsystem updating its parameter and all of its neighbors). In other words, the communication graphs within each neighborhood must be connected.

In the next section, the convergence properties of Procedure 5.1 are established.

Remark 5.5. *It is possible to reduce computations even further, at the possible expense of increased convergence time. In the formulation described above, modifications to each element of the parameter vector $p_i(k)$ are considered each time $i \in \mathbb{T}_{mod(k,w)}$. However, it is possible to only consider modifications to a subset of $p_i(k)$, or even a single element of $p_i(k)$ at a time. Specifically, (5.24) is replaced by*

$$\begin{aligned} \underset{p_i^j(k)}{\textit{min}} \quad & J(k, p(k), X(k)) \\ \textit{subject to} \quad & X(k+\sigma|k) \in \mathbb{X},\ \sigma \in \mathbb{Z}_{[0,T]}, \\ & u(k+\sigma|k) \in \mathbb{U},\ \sigma \in \mathbb{Z}_{[0,T-1]}, \\ & p(k+\sigma|k) = p(k),\ \sigma \in \mathbb{Z}_{[0,T]}, \\ & p_i^j(k) \in \{p_i^j(k-1) - \zeta^j,\ p_i^j(k-1),\ p_i^j(k-1) + \zeta^j\} \cap \mathbb{P}_j, \\ & X(k|k) = X(k), \end{aligned} \tag{5.29}$$

where there is now only a single scalar adjustable variable, p_i^j, and the solution to (5.29) is obtained by running, at most, 3 simulations. Then, at the first time instant k such that $i \in \mathbb{T}_{mod(k,w)}$, adjustments to p_i^1 are considered, i.e., $j = 1$ in (5.29). At the next time

instant k such that $i \in \mathbb{T}_{mod(k,w)}$, $j = 2$, etc. This sequence continues until $j = n$, after which, at the next time instant k when $i \in \mathbb{T}_{mod(k,w)}$, the sequence repeats, i.e., $j = 1$ and adjustments to p_i^1 are again considered. Note that the convergence analysis presented in the next section still holds for this revised parameter update strategy because updates to each element of the parameter vector are still considered periodically.

5.2.5 Convergence Analysis

Define the set of all feasible states and parameters at time-instant k as follows:

$$\begin{aligned}\mathbb{D}(k) = \quad & \{(X^0, p^0) \mid X^0 \in \mathbb{R}^{ql}, p^0 \in \mathbb{P}^q, X(k) = X^0 : p(k+\sigma|k) = p^0 \\ & \implies X(k+\sigma|k) \in \mathbb{X}, u(k+\sigma|k) \in \mathbb{U}, \sigma \in \mathbb{Z}_{[0,T]}\}.\end{aligned} \tag{5.30}$$

Then, the set of feasible states, denoted $\mathbb{C}(k)$, is the projection of $\mathbb{D}(k)$ onto the state coordinates in $\mathbb{R}^{ql}$. Note that because the constraint sets $\mathbb{X}$ and $\mathbb{U}$ are compact, the sets $\mathbb{D}(k)$ and $\mathbb{C}(k)$ are also compact. Consider the following assumptions:

A1) $X(0) \in \mathbb{C}(0)$.

A2) There exists $\nu \in \mathbb{R}_{[0,1)}$ such that, for all $(X^0, p) \in \mathbb{D}(k)$ and all $k \in \mathbb{Z}_{\geq 0}$,

$$\Omega\big(k+1, p, X(k+1|k, p, X^0)\big) \leq \nu\Omega(k, p, X^0).$$

A3) There exist $\epsilon > 0$ and $k^* \in \mathbb{Z}_{\geq 0}$ such that, for all $k \geq k^*$, all $p^0 = [p_1^{0T} \cdots p_q^{0T}]^T \in \mathbb{P}^q$, and all $\phi(p^0) \in \Gamma(p_1^0) \times \cdots \times \Gamma(p_q^0)$, the following holds:

$$(X_d(k, p^0) + \epsilon\mathcal{B},\ \phi(p^0)) \subset \mathbb{D}(k).$$

A4) There exists $\rho_i \geq \zeta_{max}$, $M_i \geq 0$, $N_i \geq 0$ and $\epsilon > 0$, such that, if $\|\hat{p}_i - p_i\|_1 \leq \rho_i$, $\hat{p}_i \in \mathbb{P}$, then, for all $X \in \bigcup_{k \in \mathbb{Z}_{\geq 0}, p \in \mathbb{P}^q} X_d(k, p) + \epsilon\mathcal{B}$, and all k sufficiently large, it follows that

$$\begin{aligned}\big|E_i[X_i(k+\sigma|k) - X_{di}(k+\sigma|k, \hat{p}_i)] - E_i[X_i(k+\sigma|k) - X_{di}(k+\sigma|k, p_i)]\big| \leq M_i\|\hat{p}_i - p_i\|_1, \\ \big|U_i[u_i(k+\sigma|k) - u_{di}(k+\sigma|k, \hat{p}_i)] - U_i[u_i(k+\sigma|k) - u_{di}(k+\sigma|k, p_i)]\big| \leq N_i\|\hat{p}_i - p_i\|_1,\end{aligned} \tag{5.31}$$

for $i \in \mathbb{S}$.

A5) The prediction horizon $T > 0$ satisfies $(T+1)(M_i + N_i) < 1$ for $i \in \mathbb{S}$.

These assumptions are reasonable, as discussed in the following comments. Assumption (A1) ensures that the initial state is feasible. Assumption (A2) ensures that, if p is held

constant, Ω is strictly decreasing with time. This assumption is reasonable given the construction of Ω in (5.19) and the assumed stability property for the system, (P3), and is satisfied for sufficiently long prediction horizons. Assumption (A3) ensures that, if the system state is sufficiently close to the reference trajectory, the parameters p^i can be adjusted by at least one step, i.e., $\pm\zeta^i$ without causing constraint violation [recall that the sets $\Gamma(p_i^0)$ are defined in (5.25)]. This assumption can be thought of as a strict steady-state feasibility condition, and it is similar to the common assumption that constraint sets contain the origin in their interiors, e.g., Assumption (A2) in [119]. This assumption typically holds in the case of state and control constraints on individual subsystems, e.g., control vector magnitude limits, but may not hold when considering constraints coupling two or more subsystems. Assumption (A4) is a locally Lipschitz type assumption on the state and control penalty functions in the cost when the system state is close to the system reference, and is reasonable due to the construction of E_i and U_i described in Property (P7). Finally, Assumption (A5) is needed to ensure that changes in p close to the system reference lead to a decrease in the cost. This assumption can be satisfied through the construction of Ω, or by decreasing the parameter step-sizes, ζ^i.

In addition to the assumptions stated above, the convergence analysis relies on the following two results. Firstly, due to the assumed stability property of the system (P3), the compactness of $\mathbb{D}(k)$ and $\mathbb{P}^q$, and Assumption (A3), the following result holds:

Proposition 5.1. *There exists $T^* \in \mathbb{Z}_{>0}$ such that, for $(X^0, p) \in \mathbb{D}(k)$, $k \in \mathbb{Z}_{\geq 0}$, if $X(k+\sigma|k, p, X^0) \in \mathbb{X}$ and $u(k+\sigma|k, p, X^0) \in \mathbb{U}$ for $\sigma \in \mathbb{Z}_{[0,T^*]}$, then $X(k+\sigma|k, p, X^0) \in \mathbb{X}$ and $u(k+\sigma|k, p, X^0) \in \mathbb{U}$ for all $\sigma > T^*$.*

Proof. Suppose $(X^0, p) \in \mathbb{D}(k)$, $X(k+\sigma|k, p, X^0) \in \mathbb{X}$ and $u(k+\sigma|k, p, X^0) \in \mathbb{U}$ for $\sigma \in \mathbb{Z}_{[0,T^*]}$ and some $T^* > 0$. Based on Property (P3) and Assumption (A3), the desired property, i.e., $X(k+\sigma|k, p, X^0) \in \mathbb{X}$ and $u(k+\sigma|k, p, X^0) \in \mathbb{U}$ for all $\sigma > T^*$, is shown to hold for a sufficiently large T^*.

With p held fixed, by the compactness of $\mathbb{D}(k)$ and $\mathbb{P}^q$ and Property (P3), for any $\bar{\epsilon} > 0$, there exists a finite time-instant $T^* \geq 0$ such that

$$X(k+\sigma|k, p, X^0) \in X_d(k+\sigma, p) + \bar{\epsilon}\mathcal{B}, \text{ for all } \sigma \geq T^*. \tag{5.32}$$

Hence, there exists a finite T^* such that (5.32) holds with $\bar{\epsilon} = \epsilon$, where ϵ is defined in Assumption (A3). Without loss of generality, suppose this $T^* \geq k^*$, where k^* is defined in Assumption (A3). Then, by Assumption (A3), $(X(k+\sigma|k, p, X^0), p) \in \mathbb{D}(k+\sigma)$ for all $\sigma \geq T^*$, and therefore $X(k+\sigma|k, p, X^0) \in \mathbb{X}$ and $u(k+\sigma|k, p, X^0) \in \mathbb{U}$ for all $\sigma > T^*$. □

Secondly, Proposition 5.2 establishes that the cost is non-increasing when Procedure 5.1 is used to update the parameter.

Proposition 5.2. *Suppose the parameter $p(k)$ is updated using Procedure 5.1, and the prediction horizon satisfies $T \geq T^*$, where T^* is defined in Proposition 5.1. Then, the cost (5.16) is non-increasing, i.e.,*

$$J(k+1, p(k+1), X(k+1)) \leq J(k+1, p(k), X(k+1)). \tag{5.33}$$

Proof. Since $T \geq T^*$, the choice $p(k+1) = p(k)$ is always a feasible choice (guarantees constraint satisfaction). At time-instant $k+1$, parameter values $p_i(k+1)$ for each $i \in \mathbb{T}_{k+1}$ are chosen in step (4b) of Procedure 5.1 as the minimizer of $J(k+1, p(k+1), X(k+1))$ over $p_i(k+1) \in \Gamma(p_i(k))$. Note that:

a) $p_i(k) \in \Gamma(p_i(k))$, therefore the choice $p_i(k+1) = p_i(k)$ is allowed by Procedure 5.1.

b) By the definition of a turn, only subsystem parameters that are not coupled in the objective function J are updated at a given time-step. Therefore, a decrease in cost resulting from a single subsystem's parameter update cannot be negated by another subsystem update during the same turn.

Hence, since $p(k+1) = p(k)$ is a feasible choice, (5.33) holds. □

The following theorem summarizes the convergence properties of the PG.

Theorem 5.1. *Suppose Assumptions (A1)-(A5) hold, the prediction horizon T satisfies $T \geq T^*$, where T^* is defined in Proposition 5.1, and $p(k)$ is determined using Procedure 5.1. Then, the following properties hold:*

1. $X(k) \in \mathbb{C}(k), X(k) \in \mathbb{X}$, *and* $u(k) \in \mathbb{U}$ *for all* $k \geq 0$,
2. $\Omega(k, p(k), X(k)) \to 0$ *as* $k \to \infty$,
3. $u_i(k) \to u_{di}(k)$ *as* $k \to \infty$ *for* $i = 1, 2, ..., q$,
4. $e_i(k) = X_i(k) - X_{di}(k, p_i(k)) \to 0$ *as* $k \to \infty$ *for* $i = 1, 2, ..., q$,
5. *There exists* $\tilde{k} > 0$ *such that* $W(p(k)) = 0$ *for all* $k \geq \tilde{k}$,
6. $J(k) \to 0$ *as* $k \to \infty$.

Proof. The proof is similar to the proofs of Theorems 2 and 3 in [73]. The main steps are summarized here, and the detailed arguments are provided below. Property (1) is shown to hold due to Assumption (A1), which ensures the initial state is feasible, and because the prediction horizon, T, is sufficiently long to ensure recursive feasibility, i.e., $p(k)$ is always a feasible choice at time instant $k+1$. Properties (2)-(4) are shown to hold by demonstrating that, as a result of Assumption (A2) and Proposition 5.2, the total cost J converges to a limit as $k \to \infty$. Property (5) is shown by contradiction, i.e., assuming that for large k, $W(p(k)) > 0$, and applying Assumptions (A3)-(A5) to show that when $W(p(k)) > 0$ for large k, there always exists a feasible alternate parameter $\tilde{p}(k)$ that the PG can select to decrease the cost. Property (6) then follows directly from properties (2) and (5). The detailed arguments based on these main ideas proceed as follows.

By Assumption (A1), there exists a $p(0)$ such that $(X(0), p(0)) \in D(0)$. The parameter governor updates p to ensure that $X(k+\sigma|k, p(k), X(k)) \in \mathbb{X}$ and $u(k+\sigma|k, p(k), X(k)) \in \mathbb{U}$ for $\sigma \in \mathbb{Z}_{[0,T]}$, and, by Proposition 5.1 and the statement of the theorem, T is sufficiently large so that $X(k+\sigma|k, p(k), X(k)) \in \mathbb{X}$, $u(k+\sigma|k, p(k), X(k)) \in \mathbb{U}$ for all $\sigma \in \mathbb{Z}_{\geq 0}$. This implies that $X(k+1) = X(k+1|k, p(k), X(k)) \in \mathbb{X}$, $u(k+1) = u(k+1|k, p(k), X(k)) \in \mathbb{U}$, and a feasible choice $p(k+1) = p(k)$ exists at time $k+1$. Since k is arbitrary, property (1) holds.

To show properties (2)-(4), consider the change in the cost (5.16) over one time step with $p(k)$ held fixed. At time step $k+1$, the cost is given by

$$J(k+1, p(k), X(k+1)) = W(p(k)) + \Omega(k+1, p(k), X(k+1)). \tag{5.34}$$

Adding and subtracting $\Omega(k, p(k), X(k))$ to the right-hand side of (5.34) results in

$$\begin{aligned} J(k+1, p(k), X(k+1)) = \; & W(p(k)) + \Omega(k, p(k), X(k)) \\ & + \Omega(k+1, p(k), X(k+1)) - \Omega(k, p(k), X(k)). \end{aligned} \tag{5.35}$$

Using $J(k, p(k), X(k)) = W(p(k)) + \Omega(k, p(k), X(k))$, (5.35) is re-written, yielding the following expression relating $J(k+1, p(k), X(k+1))$ to $J(k, p(k), X(k))$:

$$J(k+1, p(k), x(k+1)) = J(k, p(k), X(k)) + \Omega(k+1, p(k), x(k+1)) - \Omega(k, p(k), X(k)). \tag{5.36}$$

By Assumption (A2), $\Omega(k+1, p(k), X(k+1)) \leq \nu \Omega(k, X(k), p(k))$, $\nu \in [0, 1)$. Hence (5.36) becomes an inequality

$$J(k+1, p(k), X(k+1)) \leq J(k, p(k), X(k)) - (1-\nu)\Omega(k, p(k), X(k)), \tag{5.37}$$

and, from (5.33) in Proposition 5.2, it follows that

$$J(k+1,p(k+1),X(k+1)) \leq J(k+1,p(k),X(k+1)) \leq J(k,p(k),X(k)) - (1-\nu)\Omega(k). \tag{5.38}$$

Because $\nu \in [0,1)$ and Ω is non-negative by construction, (5.38) shows that the sequence $J(k,p(k),X(k))$ is bounded and non-increasing with time, and therefore converges to a limit. Hence, $\Omega(k,p(k),X(k)) \to 0$ as $k \to \infty$ and Property (2) holds. From (5.19)-(5.21), it follows that $E_i(k) \to 0$, $U_i(k) \to 0$, $e_i(k) \to 0$ and $u_i(k) \to u_{di}(k)$ as $k \to \infty$. Hence, Properties (3) and (4) hold.

As a result of Properties (3) and (4), the subsystems asymptotically approach their specified reference points. Furthermore, since J reaches a limit and $\Omega \to 0$ as $k \to \infty$, this implies that $W(p(k))$ converges to a limit, denoted by β, as well. By the definition of the parameter set $\mathbb{P}$, and by the parameter update procedure defined in Procedure 5.1, any adjustment to $W(p(k))$ is by a finite amount. Hence, $W(p(k))$ is a discrete-valued sequence and therefore converges to its limit in finite time, i.e., there exists $\tilde{k}$ such that $W(p(k)) = \beta$ for all $k \geq \tilde{k}$. The property $\beta = 0$, i.e, Property (5), is shown by contradiction.

Suppose, $\beta > 0$, i.e., for large, finite k, $W(p(k)) > 0$. Without loss of generality, suppose $1 \in \mathbb{T}_{\text{mod}(k,w)}$, and, per (P6), an adjustment to $p_1^1(k)$ would result in a reduced value for W. Consider a new value for $p_1^1(k)$, denoted by $\hat{p}_1^1$, while all other parameter values, i.e., $p_1^2(k), \cdots, p_1^n(k)$, $p_2(k)$, $p_3(k)$,$\cdots$,$p_q(k)$, remain unchanged. Let $\hat{p}_1 = [\hat{p}_1^1,\ p_1^2(k),\ \cdots,\ p_1^n(k)]^T$, and $\hat{p}(k) = [\hat{p}_1,\ p_2(k),\ p_3(k),\ \cdots,\ p_q(k)]^T$. By Assumption (A3) and Property (P6), for large k, it is possible to select $\hat{p}_1 \in \mathbb{P}$ so that

$$W(\hat{p}(k)) \leq W(p(k)) - \zeta^1, \tag{5.39}$$

where ζ^1 is the parameter step-size for parameter p_1^1, and system trajectories remain feasible. The difference in total cost J resulting from the change in $p(k)$ is written as

$$\begin{aligned} J(k,\hat{p}(k),X(k)) - J(k,p(k),X(k)) &= W(\hat{p}(k)) - W(p(k)) \\ &\quad + \Omega(k,\hat{p}(k),X(k)) - \Omega(k,p(k),X(k)) \\ &\leq -\zeta^1 + \Omega(k,\hat{p}(k),X(k)) - \Omega(k,p(k),X(k)), \end{aligned} \tag{5.40}$$

where $\Omega(k, \hat{p}(k), X(k)) - \Omega(k, p(k), X(k))$ is given by (5.19) as

$$\begin{aligned}\Omega(k, \hat{p}(k), X(k)) - \Omega(k, p(k), X(k)) = \sum_{\sigma=0}^{T} & \{E_1[X_1(k+\sigma|k) - X_{d1}(k+\sigma|k, \hat{p}_1)] \\ & -E_1[X_1(k+\sigma|k) - X_{d1}(k+\sigma|k, p_1)] \\ & +U_1[u_1(k+\sigma|k) - u_{d1}(k+\sigma|k, \hat{p}_1)] \\ & -U_1[u_1(k+\sigma|k) - u_{d1}(k+\sigma|k, p_1)]\}.\end{aligned} \tag{5.41}$$

By Assumption (A4), there exist $M_1 > 0$ and $N_1 > 0$ such that for each time instant $\sigma = 0, 1, ..., T$,

$$|E_1[X_1(k+\sigma|k) - X_{d1}(k+\sigma|k, \hat{p}_1)] - E_1[X_1(k+\sigma|k) - X_{d1}(k+\sigma|k, p_1)]| \leq M_1 \|\hat{p}_1 - p_1(k)\|_1, \tag{5.42}$$

and

$$|U_1[u_1(k+\sigma|k) - u_{d1}(k+\sigma|k, \hat{p}_1)] - U_1[u_1(k+\sigma|k) - u_{d1}(k+\sigma|k, p_1)]| \leq N_1 \|\hat{p}_1 - p_1(k)\|_1. \tag{5.43}$$

Given (5.42), (5.43) and the fact that $\|\hat{p}_1 - p_1(k)\|_1 = \zeta^1$, over the entire prediction horizon, the change in Ω due to the change in p_1 is bounded by

$$|\Omega(k, \hat{p}(k), X(k)) - \Omega(k, p(k), X(k))| \leq (T+1)(M_1 + N_1)\zeta^1, \tag{5.44}$$

and, by Assumption (A5), it follows that

$$|\Omega(k, \hat{p}(k), X(k)) - \Omega(k, p(k), X(k))| \leq (T+1)(M_1 + N_1)\zeta^1 < \zeta^1. \tag{5.45}$$

Considering (5.45) and (5.40),

$$J(k, \hat{p}(k), X(k)) - J(k, p(k), X(k)) < 0, \tag{5.46}$$

and the cost would decrease by replacing $p(k)$ with $\hat{p}(k)$. This fact, along with Procedure 5.1 which ensures adjustments to $p(k)$ are made at least every c time instants whenever such an adjustment is feasible and leads to a decreased cost, contradicts the assumption that $W(p(k)) > 0$ for large k. Therefore, there exists $\tilde{k}$ such that $W(p(\tilde{k})) = 0$ and, due to (5.33), $W(p(k)) = 0$ for all $k \geq \tilde{k}$. Thus, Property (5) holds. Property (6) follows directly from Properties (2) and (5) □

Remark 5.6. *Due to the state-feedback nature of the nominal controller (5.4), sufficiently small disturbances are naturally accommodated. It is expected that bounded disturbances may be incorporated into the convergence analysis presented above using an ISS analysis*

similar to that used for other MPC controllers, e.g., [120]. While this extension is left to future work, simulation results in Section 5.3.6 show that two different parameter governors based on the general framework described above are able to generate and maintain desired formations of spacecraft even when bounded disturbances are present.

In the proof of Theorem 5.1, it is shown that $J \to 0$ and therefore the desired formation is attained. In this analysis, it is not required, or shown, that the parameter $p(k)$ converges to a limit. While not required, the property that $p(k)$ converges to a limit in finite time may be desirable because this implies that the parameter governor becomes inactive and no adjustments are made for large k. This property can be shown to hold in several situations, and/or under additional assumptions. Some of these results are provided in the corollaries below.

Corollary 5.1. *Suppose all conditions of Theorem 5.1 hold, and $W(p(k))$ is constructed such that there exists a unique $\tilde{p} \in \mathbb{P}^q$ such that $W(\tilde{p}) = 0$. Then, all conclusions of Theorem 5.1 hold and there exists a $\tilde{k}$ such that $p(k) = \tilde{p}$ for all $k \geq \tilde{k}$.*

Proof. In this case, $W(p(k)) = 0 \implies p(k) = \tilde{p}$. Therefore, the desired property follows directly from Property (5) in Theorem 5.1. □

Corollary 5.2. *Suppose all conditions of Theorem 5.1 hold, and Procedure 5.1 is modified such that parameter adjustments are only made when they result in a reduced parameter cost, i.e,*

$$W(p(k+1)) < W(p(k)), \text{ whenever } p(k+1) \neq p(k). \tag{5.47}$$

Then, all conclusions of Theorem 5.1 hold and there exists a $\tilde{k}$ and $\tilde{p}$ such that $p(k) = \tilde{p}$ for all $k \geq \tilde{k}$.

Proof. The desired result follows directly from the fact that $W(p)$ is non-negative by construction, Property (5) in Theorem 5.1 and (5.47). □

The proof of the next Corollary relies on the following two propositions. Consider a function $f(x(k)) : \mathbb{V} \to \mathbb{W}$.

Proposition 5.3. *Suppose the following properties hold:*

1. *$y(k) \to x(k)$ as $k \to \infty$,*
2. *f is globally Lipschitz on $\mathbb{V}$.*

Then, $f(y(k)) \to f(x(k))$ as $k \to \infty$.

Proof. See Appendix E.2. □

Proposition 5.4. *Suppose the following properties hold:*

1. $f(x(k)) \to f(y(k))$ *as* $k \to \infty$,
2. *The inverse of* $f(x(k))$ *exists for all* $x(k) \in \mathbb{V}$*, and this inverse is unique and globally Lipschitz, i.e., for all* $x(k) \in \mathbb{V}$*, there exists a function* $f^{-1} : \mathbb{W} \to \mathbb{V}$ *such that, for all* $x(k) \in \mathbb{V}$, $f(x(k)) = w(k) \iff f^{-1}(w(k)) = x(k)$.

Then, $x(k) \to y(k)$ *as* $k \to \infty$.

Proof. See Appendix E.2. □

Corollary 5.3. *Suppose all conditions of Theorem 5.1 hold, and the reference trajectory* X_{di} *satisfies the following property:*

P8) for all $k \in \mathbb{Z}_{\geq 0}$, $X_{di}(k, p_i)$ *is invertible with respect to* $p_i \in \mathbb{P}$*, and this inverse is unique and globally Lipschitz in* p_i.

Then, all conclusions of Theorem 5.1 hold and there exists a $\tilde{k}$ *such that* $p(k) = \tilde{p}$ *for all* $k \geq \tilde{k}$.

Proof. The desired property follows from Properties (3) and (4) from Theorem 5.1 along with Property (P8). Consider the state error dynamics for the ith subsystem,

$$e_i(k+1) = X_i(k+1) - X_{di}(k+1), \tag{5.48}$$

which is re-written using (5.1) and (5.6) as

$$e_i(k+1) = f(k, X_i(k), u_i(k)) - f(k, X_{di}(k, p_i(k+1)), u_{di}(k, p_i(k+1)). \tag{5.49}$$

Next, $f(k, X_{di}(k, p_i(k)), u_{di}(k, p_i(k)))$ is added and subtracted to the RHS of (5.49), resulting in

$$\begin{aligned} e_i(k+1) = \quad & f(k, X_i(k), u_i(k)) - f(k, X_{di}(k, p_i(k+1)), u_{di}(k, p_i(k+1))) \\ & + f(k, X_{di}(k, p_i(k)), u_{di}(k, p_i(k))) - f(k, X_{di}(k, p_i(k)), u_{di}(k, p_i(k))). \end{aligned} \tag{5.50}$$

From Properties (3) and (4) in Theorem 1, $e_i(k) = X_i(k) - X_{di}(k, p_i(k)) \to 0$ and $u_i(k) \to u_{di}(k)$ as $k \to \infty$. Hence, as $k \to \infty$, (5.50)$\to 0$, and by Proposition 5.3, because f is

globally Lipschitz in X and u (Property (P1)), the first and last terms on the RHS of (5.50) cancel each other. Therefore, (5.50) yields

$$\lim_{k\to\infty} f(k, X_{di}(k, p_i(k)), u_{di}(k, p_i(k))) - f(k, X_{di}(k, p_i(k+1)), u_{di}(k, p_i(k+1))) = 0. \tag{5.51}$$

Using (5.6), (5.51) is equivalently written as

$$\lim_{k\to\infty} X_{di}(k+1, p_i(k)) - X_{di}(k+1, p_i(k+1)) = 0. \tag{5.52}$$

Because $p_i(k),\ p_i(k+1) \in \mathbb{P}$, by Proposition 5.4 and Property (P8), (5.52) implies that

$$\lim_{k\to\infty} p_i(k) - p_i(k+1) = 0. \tag{5.53}$$

Because $p_i(k) \in \mathbb{P}$ and $\mathbb{P}$ is a discrete-set, $p_i(k)$ is a discrete-valued sequence. Therefore (5.53) implies that $p_i(k)$ converges to a limit, denoted $\tilde{p}_i$, in finite time, i.e., $p_i(k) = \tilde{p}_i$ for all $k \geq \tilde{k}$. □

Remark 5.7. *Corollary 5.1 applies, for example, to the parameter governors considered in [73]. In Corollary 5.2, the condition (5.47) reduces the flexibility of the parameter governor in choosing the parameter at each time instant; however, if this property holds, then convergence can be achieved using a cost function only containing the parameter cost, i.e., $J(k) = W(p(k))$. This may be desirable as it eliminates calculations required to determine the incremental cost Ω at each time-step. The Property (P8) in Corollary 5.3 can typically be satisfied if the parameter modifies the controller reference-point and the parameter set is properly bounded. Examples of this are the SSG and TSG PGs, discussed in the next section.*

Remark 5.8. *The convergence analysis above was based on the parameter being confined to a discrete set. As previously discussed, this is useful because even if the optimization problem (5.24) is non-convex, a solution can be obtained quickly by running a small number of simulations. However, convergence may still be achieved even if the parameter is allowed to vary continuously within a bounded set. A continuously varying parameter may be beneficial as this increases the flexibility of the PG. In this case, the convergence analysis is similar to that presented in Theorem 5.1 with the following changes:*

1. *Assumptions (A3) and (A4) are modified such that they hold for an arbitrarily small but finite adjustment to the parameter values, instead of adjustments by a single discrete step, ζ. Note that this change weakens Assumptions (A3) and (A4).*

2. *The Property (5) in Theorem 5.1, i.e., $W(p(k)) = 0$, only holds as $k \to \infty$, rather than for all $k \geq \tilde{k}$, and, if the parameter converges by Corollaries 5.1, 5.2, or 5.3, this convergence also only occurs as $k \to \infty$. Note that this implies that if the parameter is allowed to vary continuously, convergence may be slower.*

5.3 Parameter Governors for Coordinated Control of Spacecraft Formations

To demonstrate implementations of the general PG framework described above, two specific PGs are proposed and applied to the problem of spacecraft FF. These PGs, referred to as the SSG and TSG, can be used to guide an arbitrary number of spacecraft to specified NMTs in Hill's frame, and then maintain the spacecraft at the desired phasing along the NMTs, thereby generating and maintaining a spacecraft formation. The SSG and TSG are added onto a nominal inner-loop system in order to enforce constraints and improve performance. In Sections 5.3.1-5.3.2 below, this nominal inner-loop system is described, including the dynamics for each spacecraft in formation and the nominal controller for each satellite. Constraints on both control and state variables are also defined.

5.3.1 Nominal Inner-Loop System

The dynamics of each spacecraft (subsystem), $i \in \mathbb{S} = \{1, 2, \cdots, q\}$, are expressed using the LTI discrete-time CW equations of relative motion (2.20), repeated here for clarity:

$$X_i(k+1) = AX_i(k) + Bu_i(k). \tag{5.54}$$

In (5.54), the matrix A is given by (2.21), and the control input, u_i, corresponds to instantaneous velocity change (ΔV), hence the matrix B is given by (2.23). Note that the subsystem dynamics (5.54) satisfy property (P1), hence (5.54) is applicable to the general framework described above. To facilitate implementation for the TSG, described in Section 5.3.4, it is assumed that the nominal orbital period is an integer multiple of the discrete-time update period ΔT, i.e., $(2\pi)/(\omega\Delta T) \in \mathbb{Z}$, where ω is the mean motion for the nominal circular orbit.

The formations considered in this work are similar to the series of concentric passive relative orbits described in [121]. The objective of the SSG and TSG is to place the satellites onto specified closed, unforced NMTs in Hill's frame with the correct phasing while satisfying constraints. Such a target trajectory for the ith spacecraft in discrete-time is given

by

$$\bar{X}_{di}(k+1) = A\bar{X}_{di}(k), \tag{5.55}$$

with an initial condition $\bar{X}_{di}(0)$ selected such that the resulting trajectory is periodic, i.e., the initial condition satisfies (2.25). To achieve a prescribed position in the formation, the nominal target for the ith spacecraft is given as

$$X_{di}(k) = \bar{X}_{di}(k+\theta_i), \tag{5.56}$$

where $\theta_i \in \mathbb{Z}_{\geq 0}$ specifies the desired phase shift along the trajectory. Each spacecraft is controlled with a nominal inner-loop static state feedback law that tracks the prescribed target,

$$u_i(k) = K(X_i(k) - X_{di}(k)), \tag{5.57}$$

where K is gain matrix selected such that the matrix $\bar{A} = A + BK$ is Schur, and therefore the dynamics for each spacecraft are asymptotically stable. Recalling that $X(k)$ and $u(k)$ denote the state and control vectors for the overall system, i.e.,

$$X(k) = [X_1(k)^T,\ X_2(k)^T,\ \cdots,\ X_q(k)^T]^T,$$

and

$$u(k) = [u_1(k)^T,\ u_2(k)^T,\ \cdots,\ u_q(k)^T]^T,$$

a schematic of the inner-loop system for the formation as a whole is shown in Figure 5.3.

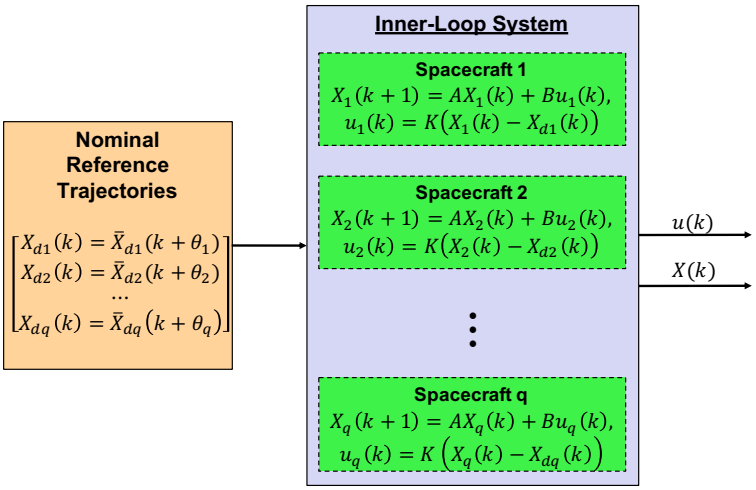


Figure 5.3: The inner-loop system for the formation

5.3.2 Constraints

Constraints on state and control variables are imposed as follows. Firstly, the control for each spacecraft is limited by an upper-bound at each time instant, i.e.,

$$y_{ci}^{u} = \|u_i(k)\|_2^2 - u_{max}^2 \leq 0,\ i = 1, 2, \cdots, q, \tag{5.58}$$

where u_{max} is the upper-bound on control. Secondly, a minimum separation distance is enforced between each pair of spacecraft at each time instant, i.e.,

$$y_{ci}^{x} = r_{min}^2 - \|S(X_j(k) - X_l(k))\|_2^2 \leq 0,\ i = 1, 2, \cdots, \tfrac{q!}{2(q-2)!}, \tag{5.59}$$

where j and l denote different spacecraft, i.e., $j,\ l \in \mathbb{S}$, $j \neq l$, the matrix S is given by $S = [I_{3\times3}\ 0_{3\times3}]$, and r_{min} is the minimum separation distance.

Remark 5.9. *Given the constraints (5.58), the control constraint set $\mathbb{U}$ is compact as required by the general formulation of Section 5.2. Given only the constraints (5.59), the state constraint set $\mathbb{X}$ is not compact. Additional state constraints can be added to make the set $\mathbb{X}$ compact. Because of the state feedback nature of the nominal control law (5.57), these constraints could be generated by reformulating the control constraint (5.58) in terms of the state variables. Alternatively, an artificial constraint could be introduced such as $X(k)^T X(k) - \delta^2 \leq 0$, for some large δ.*

Although the control law (5.57) is capable of guiding each spacecraft to its specified reference trajectory, and therefore generating and maintaining the desired formation, the response may not satisfy constraints (5.58)-(5.59). Hence, to enforce these constraints, a PG is added to the inner-loop system described above. In Sections 5.3.3 and 5.3.4, two specific PGs are described and implemented.

5.3.3 The Scale Shift Governor

The SSG enforces constraints by modifying the nominal target provided to each spacecraft according to

$$X_{di}(k, g_i(k)) = g_i(k)\bar{X}_{di}(k + \theta_i), \tag{5.60}$$

where $g_i(k)$, which is the parameter adjusted by the SSG, is a scale factor that enlarges or shrinks the nominal reference trajectory (5.56). Figure 5.4 illustrates the application of the SSG to the nominal inner-loop system.

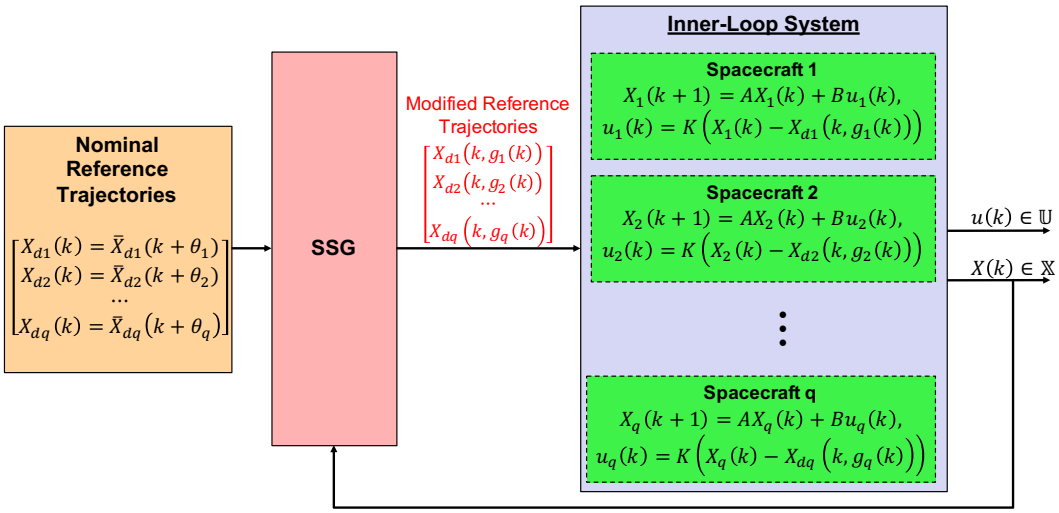


Figure 5.4: The Scale Shift Governor

The scale factor (parameter) $g_i(k)$ for each spacecraft is limited to a discrete set:

$$g_i(k) \in \mathbb{P}_{SSG} = \{g_{min},\ g_{min} + \zeta,\ g_{min} + 2\zeta,\ \cdots,\ g_{min} + \alpha\zeta\}, \tag{5.61}$$

where $g_{min} > 0$ is the minimum possible scale factor, $\zeta \in \mathbb{R}_{\geq 0}$ is the parameter step size, and $\alpha \in \mathbb{Z}_{>0}$ defines the maximum possible scale factor. Based on (5.57) and (5.60), the inner-loop controller for each spacecraft is given by

$$u_i(k) = K(X_i(k) - X_{di}(k, g_i(k))). \tag{5.62}$$

Note that the control law (5.62) satisfies property (P2), and, because the matrix $\bar{A} = A + BK$ is Schur, the required stability property for the inner-loop system (P3) also holds.

Defining $g(k) = [g_1(k)^T,\ g_2(k)^T, \cdots,\ g_q(k)^T]^T$, the cost function used by the SSG is given compactly as

$$J(k, g(k), X(k)) = W_g(g(k)) + \Omega(k, g(k), X(k)), \tag{5.63}$$

where

$$W_g(g(k)) = \sum_{i=1}^{q} |g_{di} - g_i(k)|, \tag{5.64}$$

the parameter g_{di} is the desired scale factor for the ith spacecraft, and

$$\Omega(k, g(k), X(k)) = \sum_{i=1}^{q} \sum_{\sigma=0}^{T} \left(\|X_i(k+\sigma|k) - X_{di}(k+\sigma|k)\|_{\Theta}^2 + \|u_i(k+\sigma|k)\|_{\Phi}^2 \right), \quad (5.65)$$

where Θ and Φ are symmetric positive definite weighting matrices, T is the prediction horizon, and the predicted state and control, i.e., $X_i(k+\sigma|k)$ and $u_i(k+\sigma|k)$, also depend on the same $g(k)$ and $X(k)$. Desired system performance, i.e., either fast convergence to the controller reference trajectory or low control (fuel) usage, can be encouraged through appropriate selection of the weighting matrices Θ and Φ, respectively. Note that the cost function defined by (5.63)-(5.65) satisfies the properties (P4)-(P7).

5.3.3.1 Parameter Update Strategy

The SSG updates the parameter $g(k)$ according to Procedure 5.1 and the turn-based strategy described in Section 5.2.4.2. Due to the constraints (5.59) enforcing a minimum separation distance between each pair of spacecraft in formation, the neighborhood of each spacecraft consists of the entire formation, i.e., $\mathbb{N}_i = \mathbb{S}$ for all $i \in \mathbb{S}$, and therefore, a) each turn consists of a single spacecraft, and b) the repeating sequence of turns used to periodically update each spacecraft parameter is given by

$$\mathbb{T}_1 = \{1\},\ \mathbb{T}_2 = \{2\},\ \cdots, \mathbb{T}_c = \{c\}, \text{where } c = q. \quad (5.66)$$

The initial parameter vector at $k = 0$, i.e., g^0, is determined based on a feasible solution to the following global optimization problem,

$$\begin{aligned} \min_{g(k)} \quad & J(k, g(k), X(k)) \\ \text{subject to} \quad & r_{min}^2 - \|S(X_i(k+\sigma|k) - X_j(k+\sigma|k))\|_2^2 \leq 0,\ i, j \in \mathbb{S},\ i \neq j,\ \sigma \in \mathbb{Z}_{[0,T]}, \\ & \|u_i(k+\sigma|k)\|_2^2 - u_{max}^2 \leq 0,\ i \in \mathbb{S},\ \sigma \in \mathbb{Z}_{[0,T-1]}, \\ & g(k+\sigma|k) = g(k),\ \sigma \in \mathbb{Z}_{[0,T]}, \\ & g(k) \in \mathbb{P}_{SSG}^q, \\ & X(k|k) = X(k), \end{aligned} \quad (5.67)$$

and updates to the parameter for individual spacecraft are made based on the solution to

the following local optimization problem:

$$\begin{aligned}
\min_{g_i(k)} \quad & J(k, g(k), X(k)) \\
\text{subject to} \quad & r_{min}^2 - \|S(X_i(k+\sigma|k) - X_j(k+\sigma|k))\|_2^2 \leq 0,\ j \in \mathbb{S}\backslash\{i\},\ \sigma \in \mathbb{Z}_{[0,T]}, \\
& \|u_i(k+\sigma|k)\|_2^2 - u_{max}^2 \leq 0,\ \sigma \in \mathbb{Z}_{[0,T-1]}, \\
& g(k+\sigma|k) = g(k),\ \sigma \in \mathbb{Z}_{[0,T]}, \\
& g_i(k) \in \Gamma(g_i(k-1)), \\
& X(k|k) = X(k),
\end{aligned} \tag{5.68}$$

where $\Gamma(g_i(k-1))$ is defined as follows: For $g_i^0 \in \mathbb{P}_{SSG}$,

$$\Gamma(g_i^0) = \{g_i^0 - \zeta,\ g_i^0,\ g_i^0 + \zeta\} \cap \mathbb{P}_{SSG}. \tag{5.69}$$

The parameter update strategy for the SSG is given by the Procedure 5.2 below. Note that Procedure 5.2 is an SSG-specific procedure, and is equivalent to Procedure 5.1 for the general framework.

Procedure 5.2 [SSG Parameter Update Procedure]

Inputs: Initial state: $X(0)$,
Initial states for reference trajectories: $\bar{X}_{di}^0$, $i \in \mathbb{S}$,
Constant time-shifts specifying desired relative phasing: θ_i, $i \in \mathbb{S}$,
SSG Parameter set: $\mathbb{P}_{SSG}$.

Initialization at time instant $k = 0$:

1. Obtain any feasible solution, g^0 to (5.67) and set $g(0) = g^0$,
2. Send $g(0)$ to all spacecraft,
3. For all $i \in \mathbb{S}$, apply $g_i(0)$.

Repeat at each time instant $k > 0$:

4. If $i = \text{mod}(k, q)$,
   (a) Obtain $X_j(k)$, $g_j(k)$ for all spacecraft in formation, i.e., all $j \in \mathbb{S}$
   (b) Solve (5.68) to obtain $g_i(k)$.
   (c) Send $X_i(k)$ and $g_i(k)$ to all spacecraft in formation.
5. If $i \neq \text{mod}(k, q)$,
   (a) Set $g_i(k) = g_i(k-1)$

6. Apply $g_i(k)$.

End Procedure.

Remark 5.10. *In the implementation for the SSG, because of the separation distance constraints coupling each pair of spacecraft in formation, only a single spacecraft updates its parameter at each time instant (step 4). The updated parameter value, i.e., the solution to (5.68), is obtained quickly by running, at most, 3 simulations and selecting the parameter value $g_i(k) \in \Gamma(g_i(k-1))$ that yields the lowest cost, subject to constraints. Note that, in step 4c, it is assumed that the communication graph for the entire formation is connected at every time-instant. This strong assumption can be weakened, and convergence properties are maintained. For example, it is sufficient for each spacecraft, after updating its parameter value, to send this updated value and the current values for all other spacecraft in formation (which are unchanged) only to the spacecraft that will update it's parameter at the next time instant. Further investigation into requirements for formation communication architectures is left to future work.*

5.3.3.2 Simulations

In this section, two simulations are presented to illustrate the implementation of the SSG to generate and maintain a formation of 3 spacecraft. Parameters used in the simulations are shown in Table 5.1. The parameter set for the SSG contains 50 distinct scale factors and is given by

$$\mathbb{P}_{SSG} = \{0.5,\ 0.6,\ \cdots,\ 5.4\}. \tag{5.70}$$

The first simulation shows an example in which the objective of the SSG is to place the spacecraft onto the same elliptical NMT, separated in phase by roughly 120^o. The reference trajectory for each spacecraft is given by

$$\bar{X}_{di}(k+1) = A\bar{X}_{di}(k+\theta_i),\ \bar{X}_{di}(0) = \bar{X}^0_{di}, \tag{5.71}$$

where $\bar{X}^0_{di} = [1,\ 0,\ 0,\ 0,\ -2\omega,\ 0]^T$, for $i = 1, 2, 3$, and the parameters specifying the spacecraft phasing along this reference trajectory are given by

$$\theta_1 = 16,\ \theta_2 = 0,\ \theta_3 = 33. \tag{5.72}$$

The desired scale factors (parameters) for each spacecraft are $g_{di} = 1.0$ for $i = 1, 2, 3$. Figure 5.5 shows the desired spacecraft formation.

Table 5.1: Parameters used in simulations

Parameter	**Symbol**	**Value**
Nominal orbital radius	R_0	$6,728$ km
Mean motion	ω	0.001144 rad/s
Discrete-time update period	ΔT	109.84 sec ($\frac{1}{50}$ of nom. orbital period)
State-error weighting matrix used to obtain LQR gain matrix, K	Q_{LQ}	$\text{diag}(1,\ 1,\ 1,\ 0.001,\ 0.001,\ 0.001)$
Control weighting matrix used to obtain LQR gain matrix, K	R_{LQ}	$10^8 I_{3\times 3}$
Maximum control	u_{max}	0.001 km/s
Minimum separation distance	r_{min}	1 km
Prediction Horizon	T	75 discrete-time steps
State-error weighting matrix for the incremental cost, Ω	Θ	$0.1 I_{6\times 6}$
Control weighting matrix for the incremental cost Ω	Φ	$I_{3\times 3}$

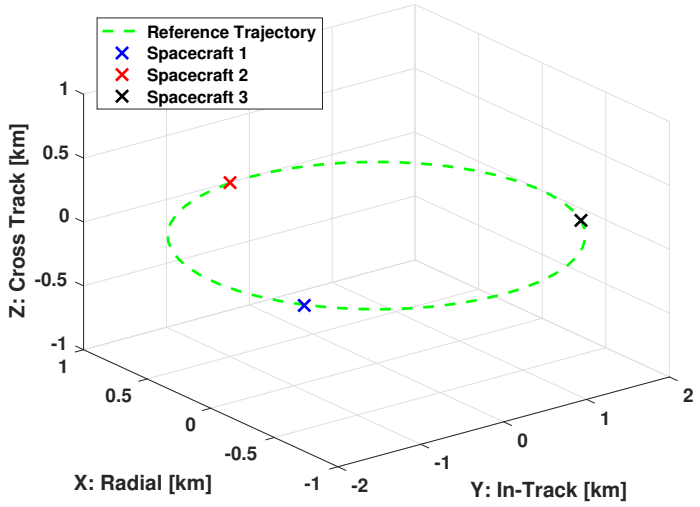


Figure 5.5: Desired formation: single reference trajectory

The spacecraft are initially located along a different elliptical NMT, with initial state vectors given by

$$\begin{aligned} X_1(0) = &\ [\ 0.5000, \quad 0.0000, \quad -0.8660, \quad 0.0000, \quad -0.0011, \quad 0.0000]^T, \\ X_2(0) = &\ [-0.2679, \quad 0.8443, \quad 0.4640, \quad 0.0005, \quad 0.0006, \quad -0.0008]^T, \\ X_3(0) = &\ [-0.2129, \quad -0.9048, \quad 0.3687, \quad -0.0005, \quad 0.0005, \quad 0.0009]^T, \end{aligned} \tag{5.73}$$

where units for position are km and units for velocity are km/s. The SSG is used to reposition the spacecraft to the desired NMT at the appropriate phasing. Figure 5.6 demonstrates the effectiveness of the SSG. Figure 5.6a shows that, with no SSG, two of the three minimum separation distance constraints are violated (constraints are violated if $y_{ci}^x(k) > 0$. Figure 5.6b shows that, after adding the SSG, all constraints are strictly enforced. Figure 5.6c illustrates how the SSG adjusts the parameter for each spacecraft with time, and Figure 5.6d shows the total cost, $J(k)$. Note that $W_g(g(k)) = 0$ for large k, and $J(k)$ approaches 0 as k increases, hence, the desired formation is attained. Figure 5.6e shows spacecraft trajectories, also illustrating that the desired formation is attained.

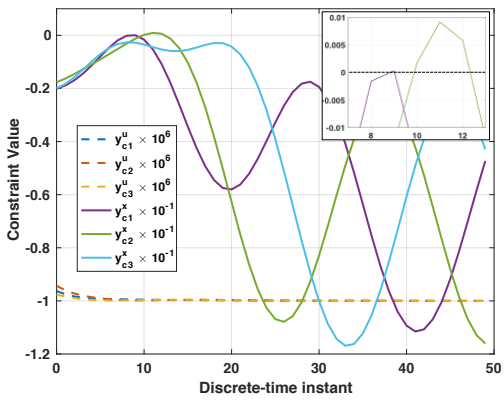
Next, a second simulation is shown in which the SSG objective is to place the spacecraft onto three concentric elliptical NMTs, again separated in phase by roughly 120^o. As in the first example simulation, the reference trajectory $\bar{X}_{di}$ and the phase parameters, θ_i are given by (5.71) and (5.72), respectively. However, unlike the first simulation where the desired scale factors (parameter) for each spacecraft was equal to 1, in this example the desired scale factors are $g_{d1} = 0.5$, $g_{d2} = 1.0$ and $g_{d3} = 1.5$. The desired spacecraft formation is shown in Figure 5.7.

The spacecraft are initially located at unforced equilibria along the in-track (y) axis, with initial state vectors given by

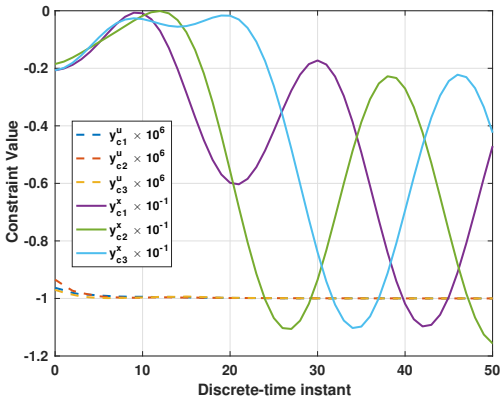
$$\begin{aligned} X_1(0) = &\ [0, \quad -6, \quad 0, \quad 0, \quad 0, \quad 0]^T, \\ X_2(0) = &\ [0, \quad -8, \quad 0, \quad 0, \quad 0, \quad 0]^T, \\ X_3(0) = &\ [0, \quad -10, \quad 0, \quad 0, \quad 0, \quad 0]^T, \end{aligned} \tag{5.74}$$

where units for position are km and units for velocity are km/s. Figure 5.8a shows that, with the SSG inactive, the control constraint for one spacecraft and one separation distance constraint are violated, while Figure 5.8b shows that with the SSG, constraints are strictly enforced. Figure 5.8c shows parameter updates and the parameter cost, $W_g(g(k))$, and Figure 5.8d shows the total cost, $J(k)$, illustrating that the desired formation is attained. Finally, spacecraft trajectories are shown in Figure 5.8e.

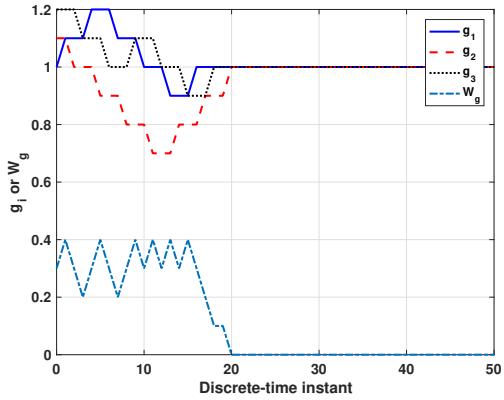
Remark 5.11. *For the simulations described in this section, using MATLAB 2016a and*



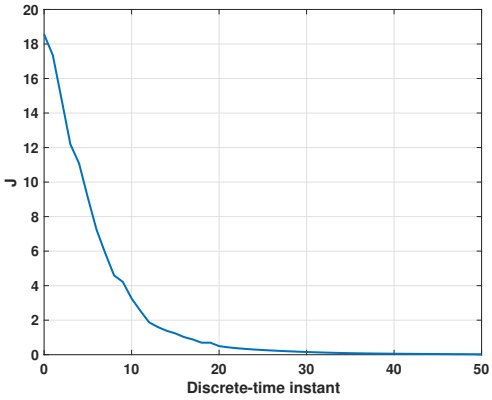
(a) Constraints with SSG inactive



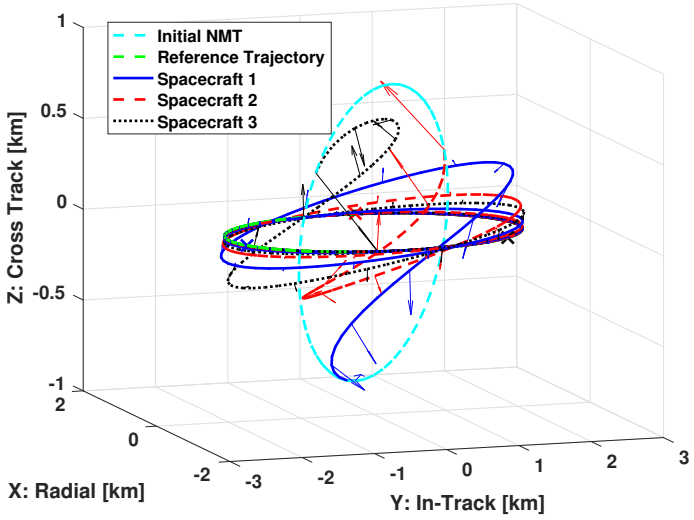
(b) Constraints with SSG active



(c) Parameter modifications



(d) Total cost



(e) Spacecraft trajectories. Arrows denote ΔV direction, X's denote final spacecraft positions.

Figure 5.6: Data for SSG simulation 1

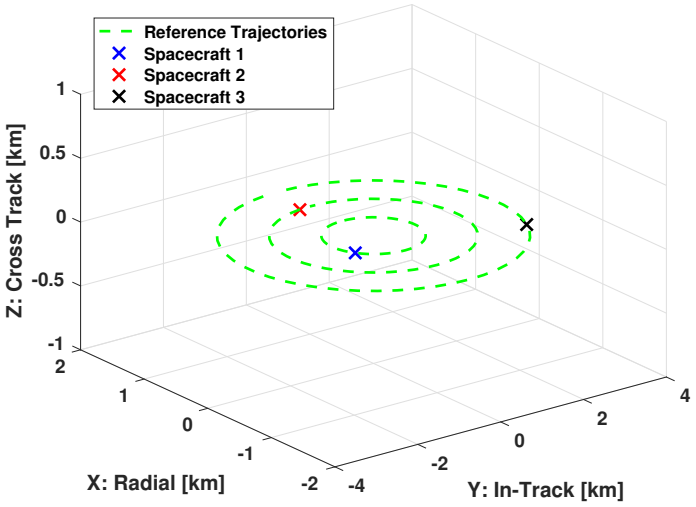
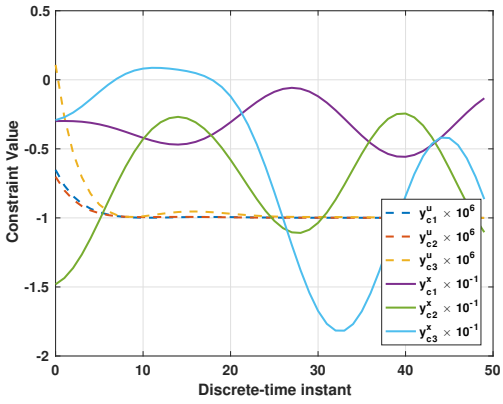
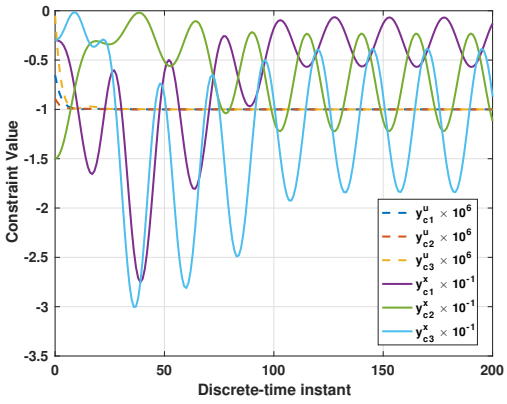


Figure 5.7: Desired formation: multiple concentric relative orbits

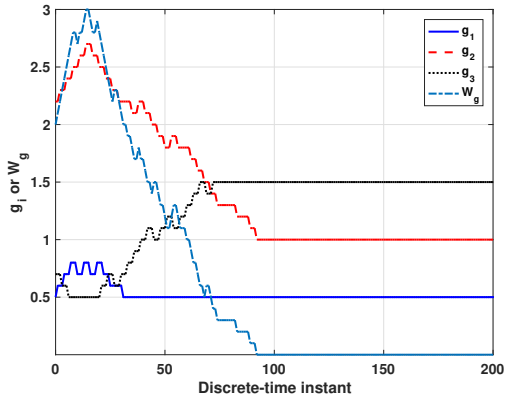
a 2.8 GHz processor, the worst-case computation time for an initial feasible solution at $k = 0$ *is approximately 1 minute, whereas the time required to update the parameter at each time-instant* $k > 0$ *is approximately 0.005 seconds.*



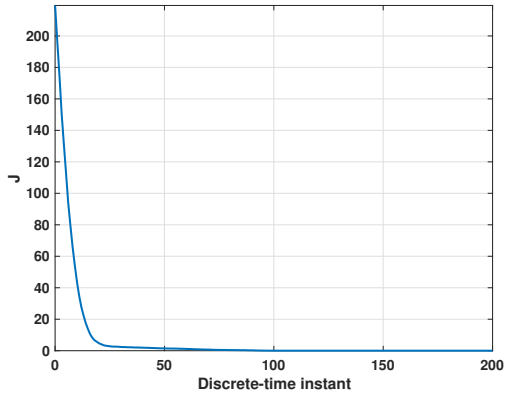
(a) Constraints with SSG inactive



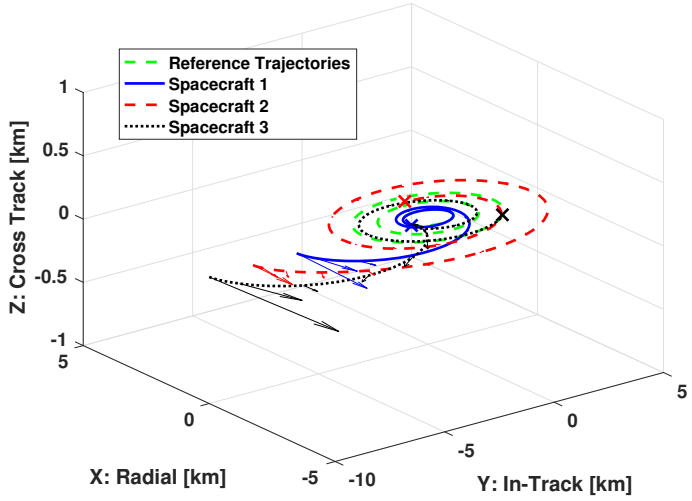
(b) Constraints with SSG active



(c) Parameter modifications



(d) Total cost



(e)Spacecraft trajectories. Arrows denote ΔV direction, X's denote final spacecraft positions.

Figure 5.8: Data for SSG simulation 2

5.3.4 The Time Shift Governor

The TSG enforces constraints by modifying the nominal target provided to each spacecraft according to

$$X_{di}(k, \tau_i(k)) = \bar{X}_{di}(k + \theta_i + \tau_i(k)), \tag{5.75}$$

where $\tau_i(k)$, which is the parameter adjusted by the TSG, is a time-shift along the nominal reference trajectory. Whereas the SSG expanded and contracted the nominal reference trajectory, the TSG leaves the nominal reference trajectory unchanged, and instead shifts the controller reference-point along the nominal trajectory at each time instant. Note that the desired formation is attained if and only if, for large k, $\tau_i(k) = \tau_j(k)$ for all $i, j \in \mathbb{S}$, see Remark 5.12. Figure 5.9 illustrates the application of the TSG to the nominal inner-loop system.

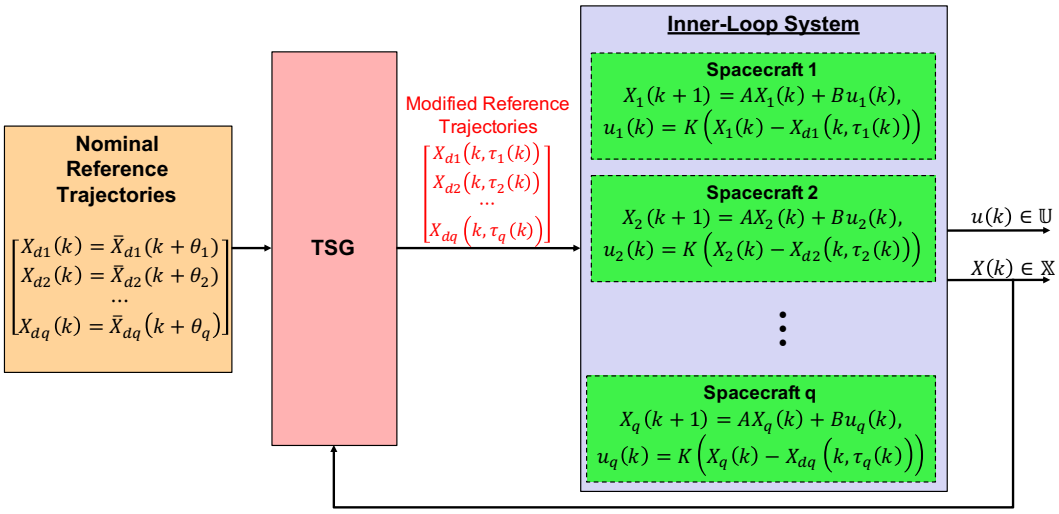


Figure 5.9: The Time Shift Governor

The allowable time-shifts for each spacecraft are limited to a single period of the nominal circular orbit, i.e.,

$$\tau_i(k) \in \mathbb{P}_{TSG} = \{\tau_i \in \mathbb{Z}_{\geq 0} \mid 0 \leq \tau_i < N\}, \tag{5.76}$$

where

$$N = \frac{2\pi}{\omega \Delta T}, \tag{5.77}$$

and where ω is the mean motion of the nominal circular orbit and ΔT is the discrete-time update period. From (5.75) and (5.76), it follows that the TSG is able to select any discrete state vector along the nominal closed reference trajectory ($\bar{X}_{di}$) as the controller reference

point ($X_{di}(k, \tau_i(k))$) at any time instant.

Remark 5.12. *Unlike in the SSG case, for the TSG there does not exist a unique set of parameters that yield the desired formation. Rather, the desired formation can be attained with any time-shift parameter value $\tilde{\tau} \in \mathbb{P}_{TSG}$, as long as all spacecraft share this same value, i.e., $\tau_i = \tilde{\tau}$ for all $i \in \mathbb{S}$.*

Based on (5.57) and (5.75), the inner-loop controller for each spacecraft is given by

$$u_i(k) = K(X_i(k) - X_{di}(k, \tau_i(k))). \tag{5.78}$$

Note that the control law (5.78) satisfies property (P2), and, because the matrix $\bar{A} = A + BK$ is Schur, the required stability property for the inner-loop system (P3) also holds.

Defining $\tau(k) = [\tau_1(k),\ \tau_2(k),\ \cdots,\ \tau_q(k)]^T$, the cost function used by the TSG is given compactly as

$$J(k, \tau(k), X(k)) = W_\tau(\tau(k)) + \Omega(k, \tau(k), X(k)), \tag{5.79}$$

where the parameter cost is given by

$$W_\tau(\tau(k)) = \sum_{i=1}^{q-1} \sum_{j=i+1}^{q} |\tau_i(k) - \tau_j(k)|, \tag{5.80}$$

and the incremental cost is given by

$$\Omega(k, \tau(k), X(k)) = \sum_{i=1}^{q} \sum_{\sigma=0}^{T} \left(\|X_i(k+\sigma|k) - X_{di}(k+\sigma|k)\|_\Theta^2 + \|u_i(k+\sigma|k)\|_\Phi^2 \right), \tag{5.81}$$

where Θ and Φ are symmetric positive definite weighting matrices, T is the prediction horizon, and the predicted state and control, i.e., $X_i(k+\sigma|k)$ and $u_i(k+\sigma|k)$, also depend on the same $\tau(k)$ and $X(k)$. Desired system performance, i.e., either fast convergence to the controller reference trajectory or low control (fuel) usage, can be encouraged through appropriate selection of the weighting matrices Θ and Φ, respectively. Note that the cost function defined by (5.79)-(5.81) satisfies properties (P4)-(P7).

Remark 5.13. *The parameter cost $W_\tau(\tau(k))$ in (5.80) is the sum of the absolute values of the difference between parameters τ_i for each pair of spacecraft in formation. For example, if there are three spacecraft in formation, i.e., $q = 3$, then*

$$W_\tau(\tau(k)) = |\tau_1(k) - \tau_2(k)| + |\tau_1(k) - \tau_3(k)| + |\tau_2(k) - \tau_3(k)|.$$

This parameter cost is 0 if and only if all parameters τ_i share a common value.

5.3.4.1 Parameter Update Strategy

The TSG updates the parameter $\tau(k)$ according to Procedure 5.1 and the turn-based strategy described in Section 5.2.4.2. Due to the constraints (5.59) enforcing a minimum separation distance between each pair of spacecraft in formation, and because of the coupling between spacecraft in the parameter cost (5.80), the neighborhood of each spacecraft consists of the entire formation, i.e., $\mathbb{N}_i = \mathbb{S}$ for all $i \in \mathbb{S}$, and therefore, a) each turn consists of a single spacecraft, and b) the repeating sequence of turns used to periodically update each spacecraft parameter is given by

$$\mathbb{T}_1 = \{1\},\ \mathbb{T}_2 = \{2\},\ \cdots, \mathbb{T}_c = \{c\}, \tag{5.82}$$

where $c = q$.

The initial parameter vector at $k = 0$, i.e., τ^0, is determined based on a feasible solution to the following global optimization problem,

$$\begin{aligned}
\min_{\tau(k)} \quad & J(k, \tau(k), X(k)) \\
\text{subject to} \quad & r_{min}^2 - \|S(X_i(k+\sigma|k) - X_j(k+\sigma|k))\|_2^2 \leq 0,\ i, j \in \mathbb{S},\ i \neq j,\ \sigma \in \mathbb{Z}_{[0,T]}, \\
& \|u_i(k+\sigma|k)\|_2^2 - u_{max}^2 \leq 0,\ \sigma \in \mathbb{Z}_{[0,T-1]}, \\
& \tau(k+\sigma|k) = \tau(k),\ \sigma \in \mathbb{Z}_{[0,T]}, \\
& \tau(k) \in \mathbb{P}_{TSG}^q, \\
& X(k|k) = X(k),
\end{aligned} \tag{5.83}$$

and updates to the parameter for individual spacecraft are made based on the solution to the following local optimization problem:

$$\begin{aligned}
\min_{\tau_i(k)} \quad & J(k, \tau(k), X(k)) \\
\text{subject to} \quad & r_{min}^2 - \|S(X_i(k+\sigma|k) - X_j(k+\sigma|k))\|_2^2 \leq 0,\ j \in \mathbb{S}\backslash\{i\},\ \sigma \in \mathbb{Z}_{[0,T]}, \\
& \|u_i(k+\sigma|k)\|_2^2 - u_{max}^2 \leq 0,\ \sigma \in \mathbb{Z}_{[0,T-1]}, \\
& \tau(k+\sigma|k) = \tau(k),\ \sigma \in \mathbb{Z}_{[0,T]}, \\
& \tau_i(k) \in \Gamma(\tau_i(k-1)), \\
& X(k|k) = X(k),
\end{aligned} \tag{5.84}$$

where $\Gamma(\tau_i(k-1))$ is defined as follows: For $\tau_i(k-1) \in \mathbb{P}_{TSG}$,

$$\Gamma(\tau_i(k-1)) = \{\text{mod}(\tau_i(k-1)-1, N+1),\ \tau_i(k-1),\ \text{mod}(\tau_i(k-1)+1, N+1)\}. \quad (5.85)$$

Remark 5.14. *Note that the set $\Gamma(\tau_i(k-1))$, as defined in (5.85) using the modulo function, is always a subset of the TSG parameter set, i.e., $\Gamma(\tau_i(k-1)) \subset \mathbb{P}_{TSG}$. Also, because of the periodic nature of the reference trajectories considered, the three integer values in the set $\Gamma(\tau_i(k-1))$ always correspond to controller reference points that are three successive state vectors along the reference trajectory.*

The parameter update strategy for the TSG is given by the Procedure 5.3 below. Note that Procedure 5.3 is a TSG-specific procedure, and is equivalent to Procedure 5.1 for the general framework.

Procedure 5.2 [TSG Parameter Update Procedure]

Inputs: Initial state: $X(0)$,
Initial states for reference trajectories: $\bar{X}^0_{di}$, $i \in \mathbb{S}$,
Constant time-shifts specifying desired relative phasing: θ_i, $i \in \mathbb{S}$,
TSG Parameter set: $\mathbb{P}_{TSG}$,

Initialization at time instant $k = 0$:

1. Obtain any feasible solution, τ^0 to (5.83) and set $\tau(0) = \tau^0$,
2. Send $\tau(0)$ to all spacecraft,
3. For all $i \in \mathbb{S}$, apply $\tau_i(0)$.

Repeat at each time instant $k > 0$:

4. If $i = \text{mod}(k, q)$,
   (a) Obtain $X_j(k)$, $\tau_j(k)$ for all spacecraft in formation, i.e., all $j \in \mathbb{S}$
   (b) Solve (5.84) to obtain $\tau_i(k)$.
   (c) Send $X_i(k)$ and $\tau_i(k)$ to all spacecraft in formation.
5. If $i \neq \text{mod}(k, q)$,
   (a) Set $\tau_i(k) = \tau_i(k-1)$
6. Apply $\tau_i(k)$.

End Procedure.

Remark 5.15. *In the implementation for the TSG, because of the coupling between spacecraft in the separation distance constraints and in the parameter cost $W(\tau(k))$, only a single spacecraft updates its parameter at each time-step (step 4). The updated parameter value, i.e., the solution to (5.84), is obtained quickly by running 3 simulations and selecting the parameter value $\tau_i(k) \in \Gamma(\tau_i(k-1))$ that yields the lowest cost, subject to constraints. Note that, in step 4c, it is assumed that the communication graph for the entire formation is connected at every time-instant. This strong assumption can be weakened, and convergence properties are maintained. For example, it is sufficient for each spacecraft, after updating its parameter value, to send this updated value and the current values for all other spacecraft in formation (which are unchanged) only to the spacecraft that will update it's parameter at the next time-instant. Further investigation into requirements for formation communication architectures is left to future work.*

5.3.4.2 Simulations

In this section, two simulations are presented to illustrate the implementation of the TSG to generate and maintain a formation of 3 spacecraft. Parameters used in the simulations are the same as those used in SSG simulations, and are shown in Table 5.1. Given the nominal circular orbit and discrete-time update period in Table 5.1, the parameter set for the TSG contains 50 elements and is given by

$$\mathbb{P}_{TSG} = \{0,\ 1,\ \cdots,\ 49\}. \tag{5.86}$$

The first simulation shows an example in which the objective of the TSG is to place the spacecraft onto the same elliptical NMT, separated in phase by roughly 120^o. The reference trajectory for each spacecraft is given by

$$\bar{X}_{di}(k+1) = A\bar{X}_{di}(k+\theta_i),\ \bar{X}_{di}(0) = \bar{X}^0_{di}, \tag{5.87}$$

where $\bar{X}^0_{di} = [1,\ 0,\ 0,\ 0, -2\omega,\ 0]^T$, for $i = 1, 2, 3$, and the parameters specifying the desired spacecraft phasing along this reference trajectory are given by

$$\theta_1 = 16,\ \theta_2 = 0,\ \theta_3 = 33. \tag{5.88}$$

The desired formation is shown in Figure 5.5.

The spacecraft are initially located at unforced equilibria along the in-track (y) axis,

with initial state vectors given by

$$\begin{aligned} X_1(0) = &\ [0, \quad -8, \ 0, \ 0, \ 0, \ 0]^T, \\ X_2(0) = &\ [0, \ -10, \ 0, \ 0, \ 0, \ 0]^T, \\ X_3(0) = &\ [0, \ -12, \ 0, \ 0, \ 0, \ 0]^T, \end{aligned} \tag{5.89}$$

where units for position are km and units for velocity are km/s. Figure 5.10 demonstrates the effectiveness of the TSG. Figure 5.10a shows that, with no TSG, the control constraint for spacecraft 3 and one of the separation distance constraints are violated, while Figure 5.10b shows that, after adding the TSG, all constraints are strictly enforced. Figure 5.10c illustrates how the TSG adjusts the parameter for each spacecraft with time, and Figure 5.10d shows the total cost, $J(k)$. Note that $W_\tau(\tau(k)) = 0$ for large k, and that $J(k)$ approaches 0 as k increases, hence, the desired formation is attained. Figure 5.10e shows spacecraft trajectories, also illustrating that the desired formation is attained.

Next, a second simulation is shown in which the TSG objective is to place the spacecraft onto three concentric elliptical NMTs, again separated in phase by roughly 120^o. The reference trajectory for each spacecraft is given by (5.87), where

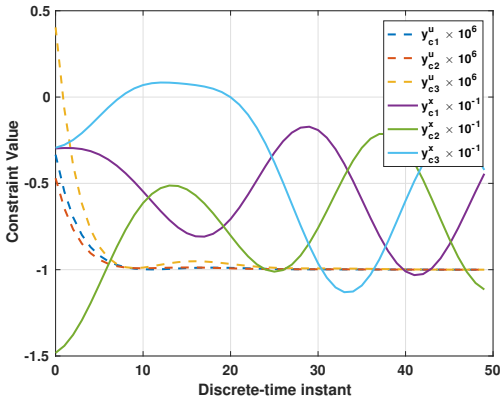
$$\begin{aligned} \bar{X}^0_{d1} &= 0.5[1, \ 0, \ 0, \ 0, \ -2\omega, \ 0]^T, \\ \bar{X}^0_{d2} &= [1, \ 0, \ 0, \ 0, \ -2\omega, \ 0]^T, \\ \bar{X}^0_{d3} &= 1.5[1, \ 0, \ 0, \ 0, \ -2\omega, \ 0]^T, \end{aligned} \tag{5.90}$$

and the parameters, θ_i, specifying the spacecraft phasing along these reference trajectories are given by (5.88). The desired spacecraft formation is shown in Figure 5.7.

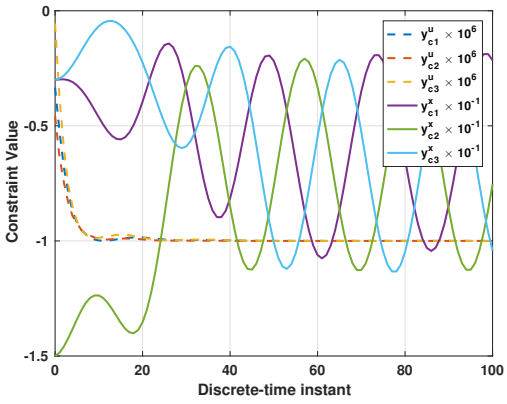
The spacecraft are initially located along a different elliptical NMT, with initial state vectors given by

$$\begin{aligned} X_1(0) = &\ [-2.6791, \quad 8.4433, \quad 0.000, \quad 0.0048, \quad 0.0061, \quad 0.0000]^T, \\ X_2(0) = &\ [-2.1289, \ -9.0483, \ 0.0000, \ -0.0052, \quad 0.0049, \ 0.0000]^T, \\ X_3(0) = &\ [\ 5.0000, \quad 0.0000, \ 0.0000, \quad 0.0000, \ -0.0114, \ 0.0000]^T, \end{aligned} \tag{5.91}$$

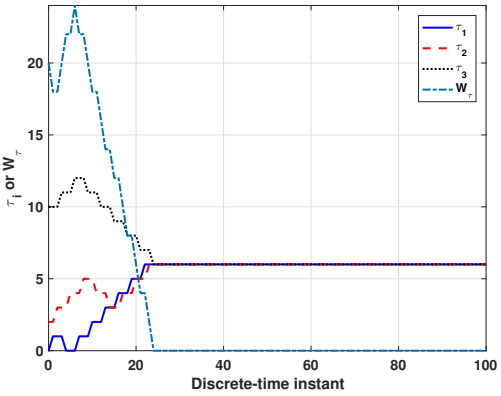
where units for position are km and units for velocity are km/s. Figure 5.11a shows that, with the TSG inactive, the control constraint for spacecraft 3 and one of the separation distance constraints are violated, while Figure 5.11b shows that with the TSG, constraints are strictly enforced. Figure 5.11c shows parameter updates, the parameter cost, $W_\tau(\tau(k))$, and Figure 5.11d shows the total cost, $J(k)$, illustrating that the desired formation is attained. Finally, spacecraft trajectories are shown in Figure 5.11e.



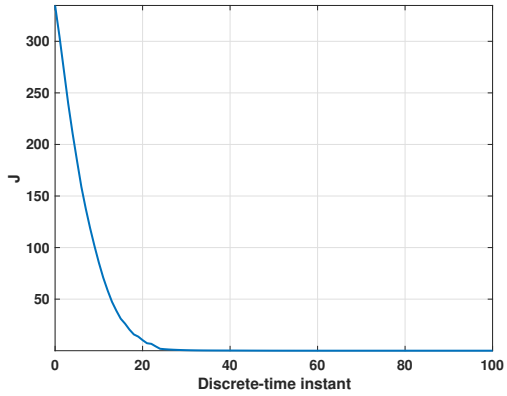
(a) Constraints with TSG inactive



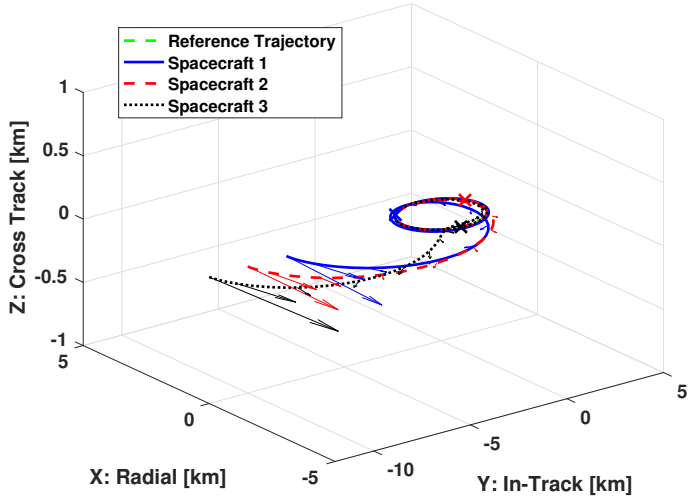
(b) Constraints with TSG active



(c) Parameter modifications

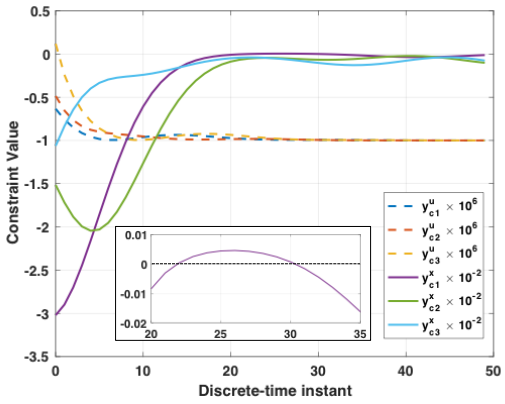


(d) Total cost

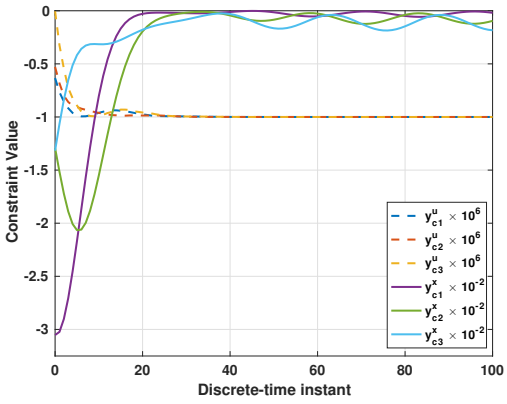


(e)Spacecraft trajectories. Arrows denote ΔV direction, X's denote final spacecraft positions.

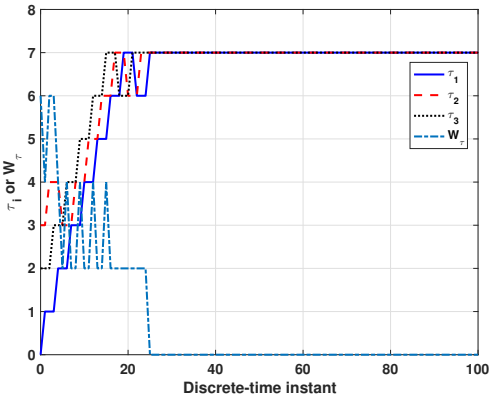
Figure 5.10: Data for TSG simulation 1



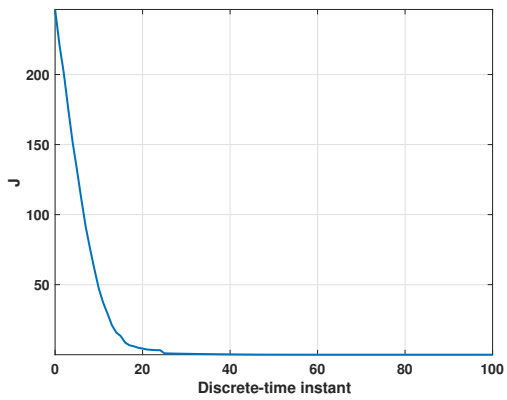
(a) Constraints with TSG inactive



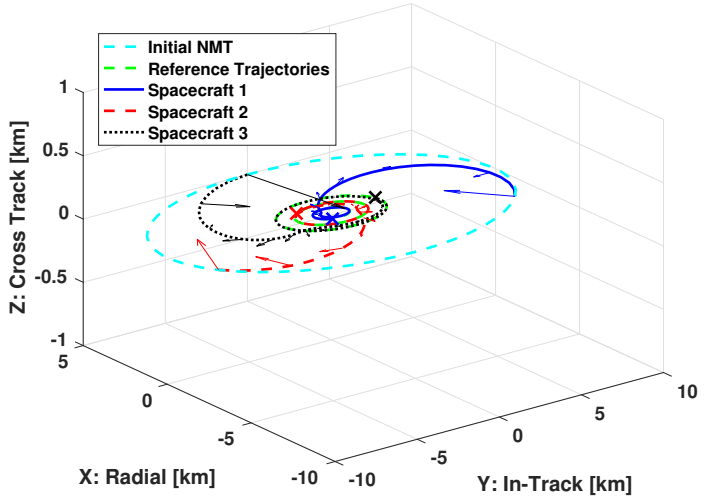
(b) Constraints with TSG active



(c) Parameter modifications



(d) Total cost



(e) Spacecraft trajectories. Arrows denote ΔV direction, X's denote final spacecraft positions.

Figure 5.11: Data for TSG simulation 2

5.3.5 Applicability of Convergence Analysis

In this section, the applicability of the convergence analysis for the general framework, i.e., Theorem 5.1 and Corollaries 5.1-5.3, to the specific cases of the SSG and TSG is analyzed. As noted above in Sections 5.3.1-5.3.4, the system dynamics, controller and cost function considered for the SSG and TSG satisfy the properties (P1)-(P7), discussed in Section 5.2 for the general framework. For the convergence results of Theorem 5.1 to hold, Assumptions (A1)-(A5) must also be satisfied.

Assumption (A1) states that the initial state is feasible. In the current implementation, this assumption is verified in Step 1 of Procedures 5.2 and 5.3 for the SSG and TSG, respectively. In future implementations, it may be possible to pre-compute (offline) sets of states for which certain parameter values (g^0 or τ^0) are known to be feasible, and preposition the spacecraft at these locations before activating the PG to generate the desired formation. Procedures to generate such sets of feasible states are left to future work.

Assumptions (A2), which guarantees the incremental cost is strictly decreasing with time, and (A5), which ensures adjustments made to the parameter near the reference trajectories result in a decreased cost, are satisfied by an appropriately selected prediction horizon, T. Simulation results suggest that selecting T to be equivalent to 1.5 times the period of the nominal circular orbit, i.e., $T = 1.5\frac{2\pi}{\omega\Delta T}$, is sufficient for this purpose. Simulation results also suggest that this choice of prediction horizon is sufficient to satisfy Proposition 5.1 as well.

Assumption (A3) ensures that, if the spacecraft states are sufficiently close to the reference trajectories, then the parameter values can be adjusted by at least one step without causing constraint violation. This assumption is satisfied for both the TSG and SSG if only control constraints are considered. If separation distance constraints coupling the spacecraft in formation are also considered, then Assumption (A3) can still be satisfied if the desired spacecraft formation and set of admissible parameter values are such that a) the desired formation has sufficient spacing between spacecraft, and b) the spacecraft remain sufficiently far apart when near their respective controller reference points, $X_{di}(k, p_i(k))$, for any admissible parameter value. Item b) (and therefore Assumption (A3)) is less restrictive for the SSG compared to the TSG. For example, for a formation with each spacecraft travelling along the same reference trajectory, e.g., the formation shown in Figure 5.5, Assumption (A3) is not satisfied for the TSG with the parameter set defined as in (5.86), which allows the controller reference point to be any state vector along the reference trajectory at any time instant. This limitation can be seen in simulations. For many initial conditions with a feasible initial solution, the TSG is able to guide the spacecraft to the specified reference trajectory, but is unable to converge, i.e., obtain the correct phasing, and therefore is

unable to generate the desired formation. In contrast, simulations for the SSG suggest that the SSG is able to generate the desired formation from any initial condition with a feasible initial solution.

Assumption (A4), which is a local-Lipschitz type requirement on the penalty functions E_i and U_i in the incremental cost, Ω, is satisfied due to the form of Ω used by the SSG and TSG, given in (5.65) and (5.81), respectively, and the periodic nature of the reference trajectories.

Finally, note that for both the SSG and TSG, if Assumptions (A1)-(A5) are satisfied, the parameter is guaranteed to converge to a limit, and, therefore, the SSG and TSG become inactive for large k. Specifically Corollary 5.1 applies for the SSG, and Corollary 5.3 applies for both the SSG and the TSG.

5.3.6 Robustness to Disturbances

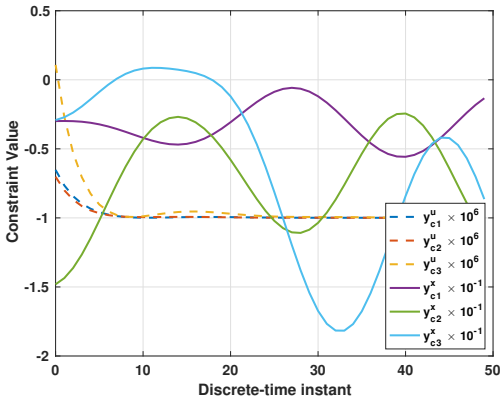
To demonstrate that the SSG and TSG are robust to sufficiently small disturbances, a new simulation is run for the SSG using the same set-up as SSG simulation 2, i.e., the desired formations and spacecraft initial conditions are the same, and for the TSG, a simulation is run using the same set-up as TSG simulation 1. In these new simulations, disturbances are added at each time-step, i.e., the state of each spacecraft evolves according to

$$X_i(k+1) = AX_i(k) + Bu_i(k) + Bw_i(k), \tag{5.92}$$

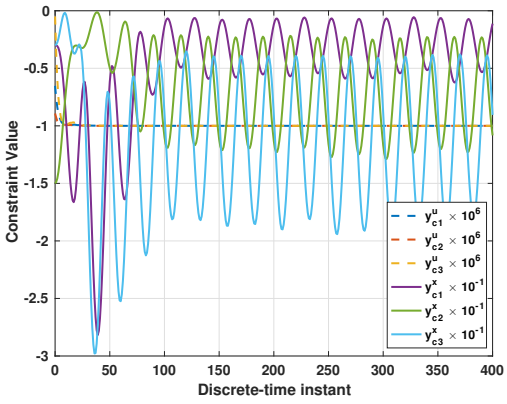
where $w_i(k) \in \mathbb{R}^3$ is the disturbance vector randomly sampled from a uniform distribution over an infinity-norm ball centered at the origin with a radius of 0.025 m/sec.

Data for the SSG simulation with disturbances is shown in Figure 5.12. Constraints with and without the SSG are shown in Figures 5.12a and 5.12b, respectively, demonstrating that the SSG enforces all constraints. The parameter for each spacecraft and the parameter cost $W_g(g(t))$ are shown in Figure 5.12c, demonstrating that $W_g(g(t))$ goes to zero, and Figure 5.12d shows that the 2-norm of the state error, $e_i(k) = X_i(k) - X_{di}(k, g_i(k)))$, decays to a small value for each spacecraft, illustrating that the desired formation is attained. Spacecraft trajectories are shown in Figure 5.12e. Note that with disturbances, the spacecraft do not asymptotically converge to their specified reference trajectories. Rather, they converge to and then stay within a small tube centered on the reference trajectories.

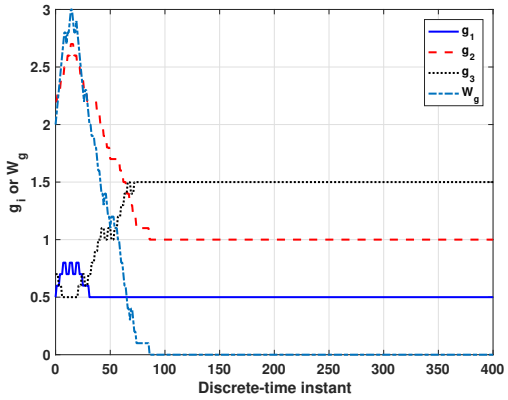
Similar data for the TSG simulation with disturbances are shown in Figure 5.13.



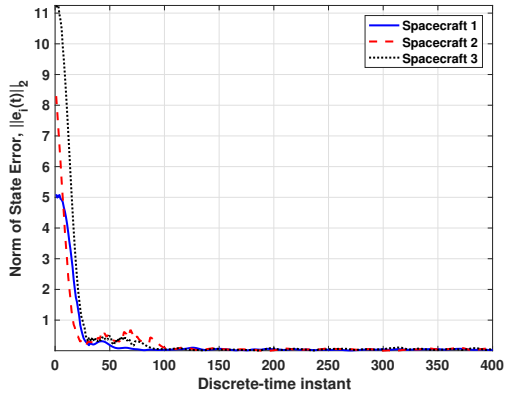
(a) Constraints with SSG inactive



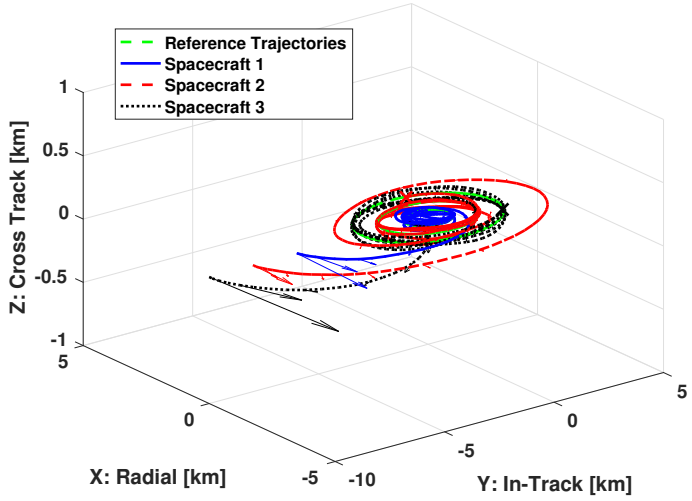
(b) Constraints with SSG active



(c) Parameter modifications

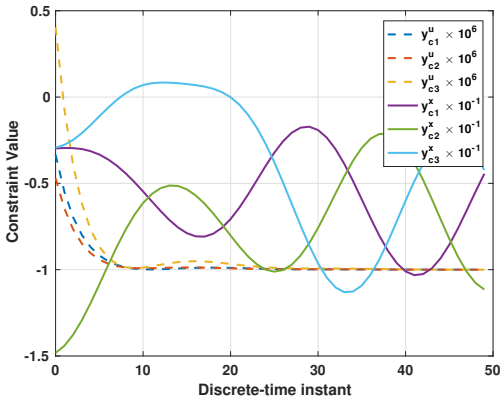


(d) State error

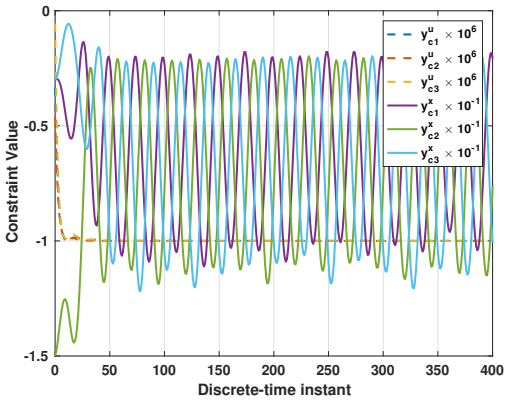


(e)Spacecraft trajectories. Arrows denote ΔV direction, X's denote final spacecraft positions.

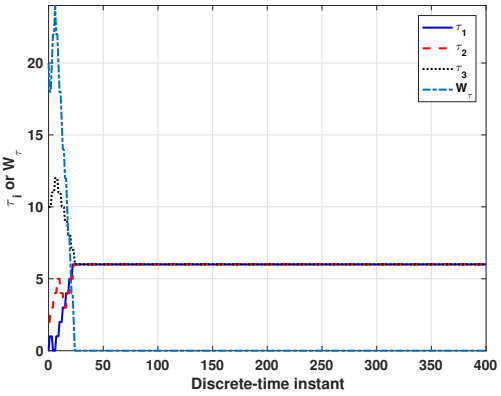
Figure 5.12: Data for SSG simulation with disturbances



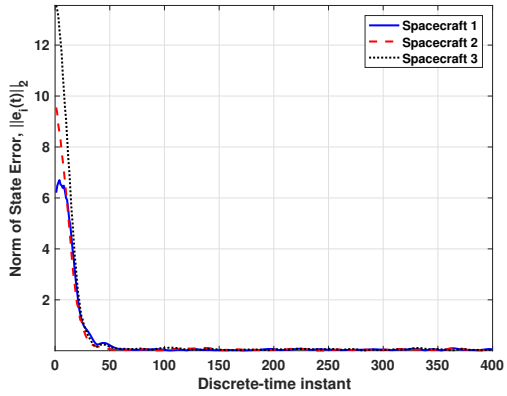
(a) Constraints with TSG inactive



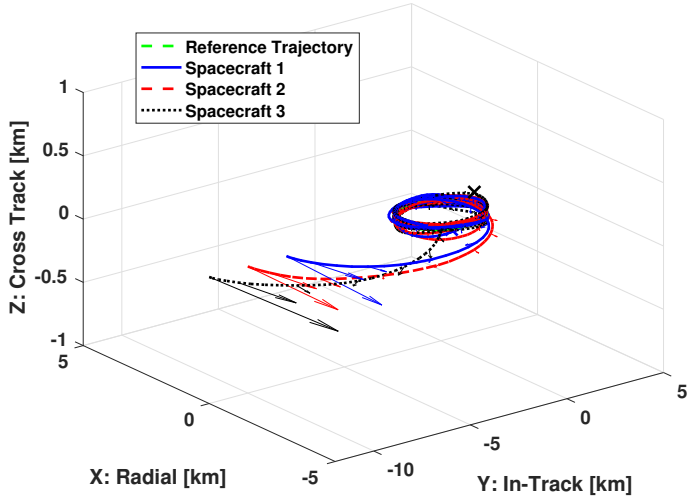
(b) Constraints with TSG active



(c) Parameter modifications



(d) State error



(e) Spacecraft trajectories. Arrows denote ΔV direction, X's denote final spacecraft positions.

Figure 5.13: Data for TSG simulation with disturbances

5.3.7 Combined SSG/TSG

The preceding sections have demonstrated that the SSG and TSG, individually, are capable of generating and maintaining formations of spacecraft. However, both the SSG and TSG have limitations when used alone, for example, as discussed in Section 5.3.5, the TSG can become "stuck", i.e., unable to make modifications to the parameter when all spacecraft are travelling along the same reference trajectory. Motivated by these limitations, in this section, a simulation is presented to demonstrate that it is possible, and in certain cases beneficial, to combine the SSG and TSG into a single PG.

Consider three spacecraft with initial conditions given by

$$\begin{aligned} X_1(0) &= [0, \ -12, \ 0, \ 0, \ 0, \ 0]^T, \\ X_2(0) &= [0, \ -10, \ 0, \ 0, \ 0, \ 0]^T, \\ X_3(0) &= [0, \ \ -8, \ 0, \ 0, \ 0, \ 0]^T, \end{aligned} \tag{5.93}$$

and a desired formation with all three spacecraft evenly spaced along a single elliptical NMT, i.e., the formation shown in Figure 5.5. Figure 5.14 shows constraints with no PG active, illustrating that two control constraints and one separation distance constraint are violated.

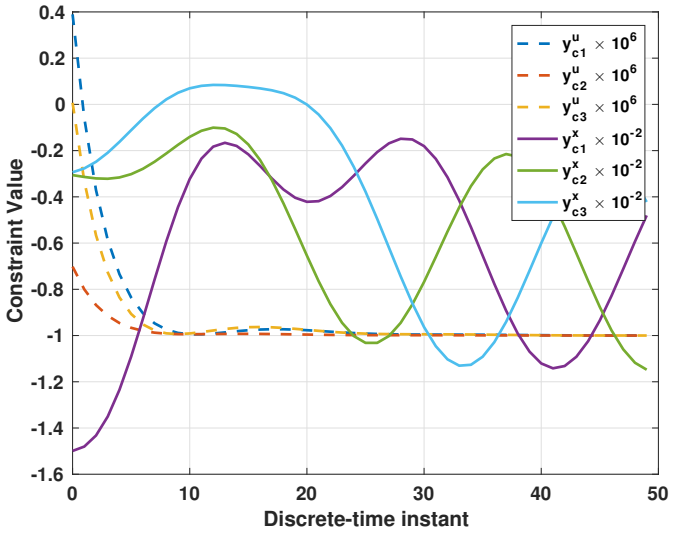


Figure 5.14: Constraints with no PG

From the initial conditions given in (5.93), there is no feasible initial solution for the SSG, i.e., there does not exist a $g(0) \in \mathbb{P}^3_{SSG}$ such that the optimization problem (5.67)

is feasible. For the TSG, there is a feasible initial solution,[1] however, from this initial solution, the TSG is unable to converge, i.e., $W_\tau(\tau(k))$ does not go to 0.

The preceding paragraph has demonstrated that neither the SSG or TSG alone, as currently designed, are capable of obtaining the desired formation. However, the desired formation *is* obtainable if the SSG and TSG are combined into a single PG, capable of adjusting both scale and time-shifts applied to the reference trajectory. An initial implementation of such a combined SSG/TSG is as follows:

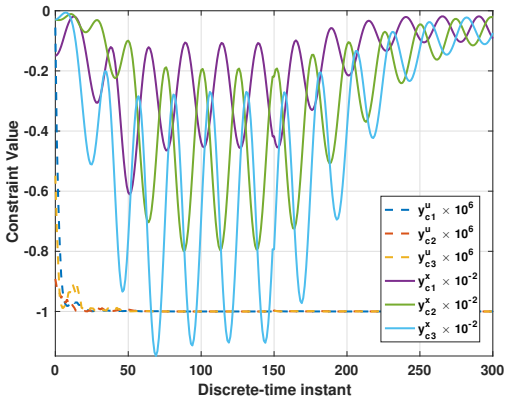
1. Initial scale factors are selected such that the corresponding reference trajectories are sufficiently far apart to satisfy the separation distance constraints. For the simulation currently being considered, these values are chosen as $g_1 = 1.5,\ g_2 = 2.5,\ g_3 = 3.5$.

2. Using these scale factors and corresponding reference trajectories, the TSG is engaged, using Procedure 5.3, and the simulation is continued until $W_\tau(\tau(k)) = 0$ and $\Omega(k) \leq \alpha$, where α is some small positive value. Note that these conditions imply that the spacecraft have obtained the desired phasing along their respective reference trajectories, and have converged close to their current controller reference points. For the simulation currently being considered, $\alpha = 2.5 \times 10^{-7}$.

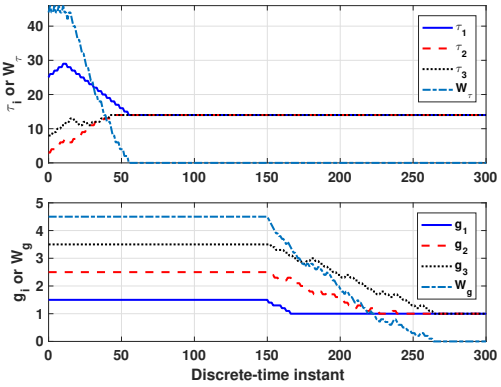
3. With time-shifts τ_i held fixed, the SSG is then engaged, using Procedure 5.2, to generate the desired formation.

Figure 5.15 shows an example simulation using this implementation. Figure 5.15a shows that all constraints are strictly enforced by the combined PG. Figure 5.15b shows how the combined PG makes adjustments first to the time-shifts τ_i, and then to the scale-shifts g_i, such that W_τ and W_g both converge to 0, and Figure 5.15c shows that the total cost, calculated as $J = W_\tau + W_g + \Omega$, also goes to 0 for large k, and hence the desired formation is obtained. Figure 5.15d shows spacecraft trajectories.

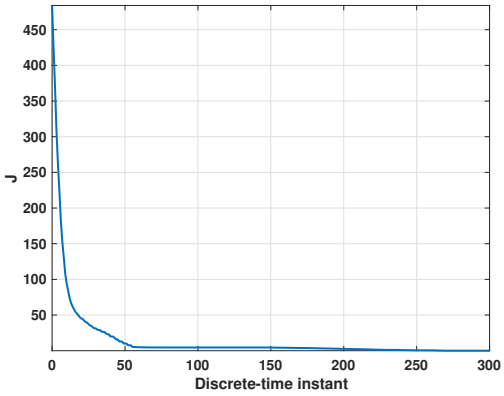
[1] The feasible initial solution referred to here is the first τ obtained that is feasible for (5.83), when a grid search is performed over all $\tau_1,\ \tau_2,\ \tau_3 \in \mathbb{P}_{TSG}$.



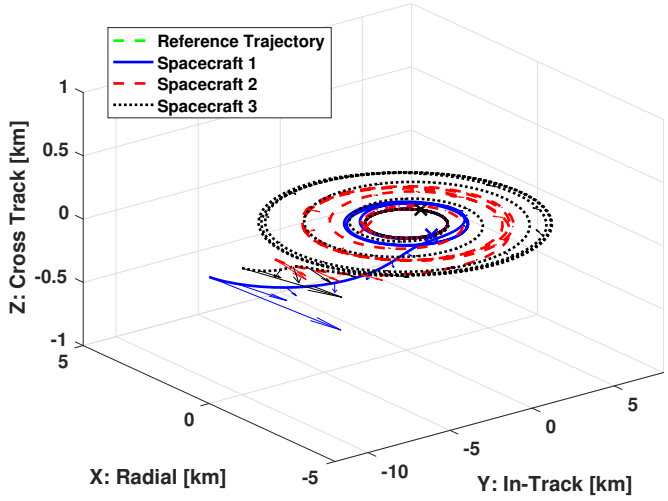
(a) Constraints with combined PG active



(b) Parameter modifications



(c) Total cost



(d) Spacecraft trajectories. Arrows denote ΔV direction, $X's$ denote final spacecraft positions.

Figure 5.15: Data for combined SSG/TSG simulation

CHAPTER 6

Conclusions and Future Work

6.1 Conclusions

This dissertation focused on the problem of constrained SRMP. Two specific cases of SRMP were considered: bSRMP in which a single spacecraft executes maneuvers in close proximity to obstacles or other non-maneuvering spacecraft, and FF in which multiple spacecraft execute maneuvers simultaneously to generate and maintain a specified formation. The SRMP methods proposed in this dissertation provide several advantages compared to many existing optimization or path planning techniques. Specifically, they accommodate non-linear and non-convex constraints, are robust to sufficiently small disturbances, and are implemented using fast and straightforward calculations. The contributions of the specific SRMP methods developed in this dissertation are summarized below.

Invariance-based Satellite Relative Motion Planning

A bSRMP method was developed to plan maneuvers to transition a spacecraft between specified NMTs. This method was based on a graph search applied to a "virtual net" composed of periodic (closed) and non-periodic (open) NMTs. Adjacency between NMTs in the virtual net was determined by conditions based on SPI tubes built around each NMT. The method was shown to enforce constraints on both control and state variables, including maximum control limits and non-convex constraints formulated to ensure avoidance of exclusion zones. Robustness to set-bounded disturbances was guaranteed through construction of the virtual net. These disturbances were used to accommodate a minimum control limit, similar to the minimum impulse bit constraints common to spacecraft.

Maneuvers developed using this methodology were fuel efficient by nature, i.e., they consisted only of NMTs and transfers between them. Fuel efficiency was also promoted through use of an adjacency matrix weighted with the control (fuel) cost of transfer between adjacent NMTs. Simulations showed that the computations required to generate

trajectories *after* the virtual net was formed were minimal. These computations may be implementable on-board a spacecraft. A method to speed up the calculations required to form the virtual net was proposed, and shown to reduce the time required for the calculations by approximately 97%, with minor tradeoffs related to maneuver planning flexibility.

Relative Motion Planning for Satellite Inspection

A bSRMP method was developed for satellite inspection applications. In this setting, an inspector spacecraft maneuvers to obtain information about a prescribed point on a target spacecraft, while avoiding collisions between spacecraft. An information collection model was developed to express the rate of information collection as a function of both distance and angle to target. The information collection model was used to formulate a rapidly computable LG-based analytical control law. This control law was shown to drive the inspector spacecraft on a path along which the rate of information collection was strictly increasing.

Based on an analysis of closed-loop trajectories using the LG control law, a state feedback control law was developed, and rules were proposed to determine when to switch from the LG control law to the state feedback control law. Simulations demonstrated that the use of this switched control scheme was effective in generating trajectories useful for information collection, while satisfying constraints. A comparison study showed that modifications to parameters in the LG control law or in the control switching rules can be made to promote mission priorities such as fuel efficiency or fast information collection. Robustness to disturbances was demonstrated through simulation.

Parameter Governors for Constrained Control of Multi-agent Formations

A general PG-based control scheme was developed to generate and maintain formations containing an arbitrary number of agents (vehicles) or subsystems. The PG was added to a nominal inner-loop system, composed from the set of subsystems, and used to enforce constraints and improve performance by modifying parameters applied to each subsystem. These parameters either changed the reference points given to the inner-loop controllers, or modified the inner-loop system dynamics. Parameter values were calculated based on a solution to a receding horizon constrained optimization problem at each time instant.

By utilizing a turn-based parameter update strategy based on graph colorability theory, parameter updates were distributed over time and between subsystems. This approach yielded fast computations. Furthermore, by limiting the possible parameter values to a discrete set of small cardinality, updated parameter values were obtained quickly, even if

non-linear and non-convex constraints were included. A convergence analysis proved that, under reasonable assumptions, the PG is guaranteed to generate and maintain the desired formation.

Two specific PGs, referred to as the SSG and TSG, were proposed and applied to the specific problem of FF. The SSG and TSG were used to enforce non-linear and non-convex constraints by modifying either a scale- or time-shift applied to the target trajectory provided to each spacecraft in formation. Desired performance related to convergence time and control (fuel) use were promoted through selection of weighting matrices in the SSG and TSG cost functions. Simulation showed that both the SSG and TSG were capable of generating formations composed of three spacecraft, and that both methodologies can accommodate sufficiently small disturbances.

6.2 Future Work

There are many possibilities to continue and expand the work presented in this dissertation. Some of these are summarized in what follows.

Invariance-based Satellite Relative Motion Planning

The SRMP method presented in Chapter 3 is able to accommodate set-bounded disturbances through the construction of SPI tubes. Further investigation of the handling of specific disturbances, such as J_2 or slightly elliptical reference orbits, would be useful. Simulation case studies could be carried out to verify that by choosing an appropriate bound for the disturbance set, the method can handle these common disturbances. Expected orbital perturbations, such as J_2, could also be explicitly calculated along predicted transfer trajectories used to weight the virtual net. This may improve the accuracy of predicted control (fuel) use. Furthermore, to explicitly accommodate elliptical reference orbits, it may be possible to design a virtual net based on the linear time-periodic TH equations, rather than the LTI CW equations. This may require the use of a time-periodic shape matrix used to define safe ellipsoidal sets and SPI tubes.

Further investigation into methods to select NMTs for the virtual net is also warranted. The current method of choosing NMTs to be "evenly spaced" throughout a prescribed region does not account for the locations of known obstacles. This method also does not consider how far apart NMTs in the virtual net should be spaced to ensure a high degree of adjacency, while also keeping the computational load related to the total number of NMTs in the virtual net manageable.

Another direction for future research is to expand the method to incorporate forced trajectories, in addition to NMTs. This change would increase the flexibility of the method, but would also increase calculations by, e.g., requiring a separate calculation for the safe scale factor ρ^u for each state vector along forced trajectories. Finally, ways to incorporate moving obstacles into the framework could be investigated, based on the work in [48] for a virtual net composed of forced equilibria.

Relative Motion Planning for Satellite Inspection

In Chapter 4, several possibilities for rules to govern switching between the LG control law and the state feedback control law were discussed, and the ability of the proposed method to handle disturbances was briefly investigated through simulation. Additional investigation in both of these directions is warranted. Further development to incorporate the LG/state feedback control law into more realistic mission settings would also be useful. For example, missions in which information about multiple target points is desired could be considered. This extension could involve development of rules to determine the optimal ordering of points for information collection, i.e., which points should be visited first, second, etc. Additionally, rules could be developed to govern multiple switches between LG and state-feedback controllers based on each target point. Finally, investigating ways to provide stronger assurances of exclusion zone avoidance, and to incorporate additional exclusion zones not co-located with the target spacecraft would be useful. One possibility in this direction is to consider combining the LG control law developed here with existing methods for obstacle avoidance such as the use of artificial potential functions, see, e.g., [122].

Parameter Governors for Constrained Control of Multi-agent Formations

To illustrate the flexibility of the general PG framework developed in Section 5.2, applications to other types of formations, such as UAVs or self-driving vehicles, could be developed. Additionally, the proposed PG could be compared to related control schemes for distributed systems, e.g., distributed command governors [68], to better understand relative strengths and weaknesses. Useful research directions related to theoretical developments include developing methods to calculate or estimate the required prediction horizon, T^*, investigating ways to weaken the assumptions required for the convergence analysis, and incorporating disturbances into this analysis.

For applications specific to SRMP, investigations could be carried out to further explore combining the SSG and TSG into a single PG. This could involve developing rules to govern when adjustments to scale- and time-shifts are made to provide desired perfor-

mance. Finally, adding additional adjustable parameters to the combined PG may improve the PG's ability to satisfy constraints. For example, combining the SSG/TSG with a PG similar to the gain governor proposed in [73] would directly provide additional capability to accommodate control saturation limits, and may also assist in accommodating separation distance constraints. Specifically, by setting the controller gain to zero (or nearly zero) initially, a gain governor may allow certain spacecraft to "wait" at their initial states for a certain time period, thereby allowing separation distances with other spacecraft to increase, before beginning their maneuvers.

APPENDIX A

Initial Conditions for NMTs

Consider the following formulation of the solution to the continuous-time CW equations, known as the "traveling ellipse" formulation [84, 85]:

$$\begin{aligned} x(t) &= \bar{X} + b\sin(\omega t + \phi), & \dot{x}(t) &= b\omega\cos(\omega t + \phi), \\ y(t) &= \bar{Y} - \tfrac{3}{2}\omega t\bar{X} + 2b\cos(\omega t + \phi), & \dot{y}(t) &= -\tfrac{3}{2}\omega\bar{X} - 2b\omega\sin(\omega t + \phi), \\ z(t) &= c\sin(\omega t + \psi), & \dot{z}(t) &= c\omega\cos(\omega t + \psi). \end{aligned} \tag{A.1}$$

From this formulation, it is apparent that motion in the z direction will be oscillatory, and motion in the x-y plane will trace out a 2×1 ellipse, with semi-minor axis b. Furthermore, this ellipse will translate horizontally (along the y-axis) if the parameter $\bar{X} \neq 0$. In (A.1), the parameters $\bar{X}$ and $\bar{Y}$, and time t, define the instantaneous center of the elliptical x-y plane motion, the parameter c defines the magnitude of oscillation in the z direction, and ϕ and ψ are the phase angles for the motion in the x-y plane, and in the z direction, respectively. These parameters are related to the initial condition $X_0 = [x_0,\ y_0,\ z_0,\ \dot{x}_0,\ \dot{y}_0,\ \dot{z}_0]^T$ by the following expressions:

$$\begin{aligned} \bar{X} &= 4x_0 + \tfrac{2}{\omega}\dot{y}_0, & \bar{Y} &= y_0 - \tfrac{2}{\omega}\dot{x}_0, \\ b\cos(\phi) &= \tfrac{\dot{x}_0}{\omega}, & b\sin(\phi) &= -3x_0 - \tfrac{2}{\omega}\dot{y}_0, \\ c\sin(\psi) &= z_0, & c\cos(\psi) &= \tfrac{\dot{z}_0}{\omega}. \end{aligned} \tag{A.2}$$

From (A.1) and (A.2), initial conditions for specific types of NMTs can be obtained.

Closed NMTs

If the initial condition is chosen to satisfy $\bar{X} = 0$, or equivalently

$$\dot{y}_0 = -2\omega x_0, \tag{A.3}$$

then the resulting NMT will be periodic (closed).

A "stationary point" closed NMT along the y-axis may be generated by setting $\bar{X} = b = c = 0$. These conditions correspond to an initial condition X_0 satisfying

$$\begin{aligned} y_0 &= y_0, \\ x_0 = z_0 = \dot{x}_0 = \dot{y}_0 = \dot{z}_0 &= 0. \end{aligned} \tag{A.4}$$

Note that these "stationary point" NMTs correspond to unforced equilibria, located at point y_0 along the y-axis.

A "periodic line segment" closed NMT in the y-z plane may be generated by setting $\bar{X} = b = 0$, and choosing a non-zero value for c. These conditions correspond to an initial condition X_0 satisfying

$$\begin{aligned} y_0 = y_0, \;\; z_0 = c\sin(\psi), \;\; \dot{z}_0 = \omega c\cos(\psi), \\ x_0 = \dot{x}_0 = \dot{y}_0 = 0, \end{aligned} \tag{A.5}$$

where $c > 0$ defines the magnitude of oscillation, i.e., one-half the length of the line segment, y_0 gives the location of intersection with the y-axis, and the phase angle ψ is arbitrarily chosen.

A closed elliptical NMT centered at the point $x = \bar{X} = 0$, $y = \bar{Y}$ and $z = 0$ may be generated by setting $\bar{X} = 0$, and choosing values for $b > 0$ and $c \geq 0$. These conditions correspond to an initial condition X_0 satisfying

$$\begin{aligned} x_0 &= b\sin(\phi), & \dot{x}_0 &= b\omega\cos(\phi), \\ y_0 &= \bar{Y} + 2b\cos(\phi), & \dot{y}_0 &= -2b\omega\sin(\phi), \\ z_0 &= c\sin(\psi), & \dot{z}_0 &= c\omega\cos(\psi). \end{aligned} \tag{A.6}$$

These elliptical NMTs correspond to sections of a $2b \times b$ cylinder with the central axis parallel to the z-axis passing through the y-axis at $y = \bar{Y}$. Figure A.1 provides an illustration of such a cylinder, and shows a few examples of elliptical NMTs as sections of the cylinder. Obtaining an initial condition for an elliptical NMT with a desired orientation within this cylinder is not straightforward using (A.6) alone. This task is simplified by introducing two angles, θ_1 and θ_2, to represent the orientation of the ellipse [84].

The angles θ_1 and θ_2 are measured from the center of the ellipse, with respect to the ellipse's "normal vector", $\hat{h}$; i.e., the vector $\hat{h}$ is perpendicular to plane defined by the ellipse, as shown in Figure A.2.

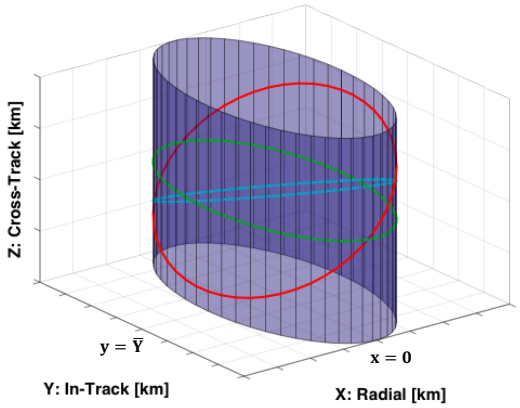


Figure A.1: Illustration showing example NMTs as sections of a cylinder

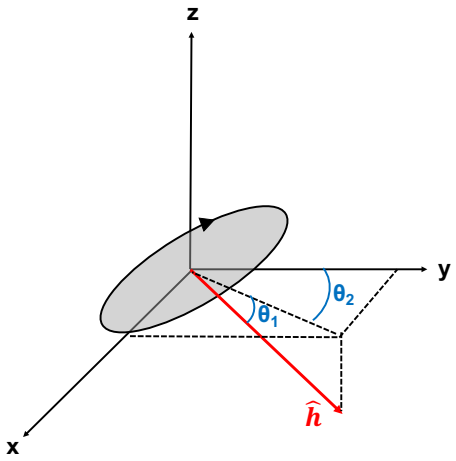


Figure A.2: Depiction of the angles θ_1 and θ_2 used to parametrize elliptical NMTs

Given desired values for θ_1, θ_2, b, and $\bar{Y}$, an initial condition for an elliptical NMT with the desired orientation can be obtained through the following steps:

Procedure A.1 [Initial Condition Calculation for Elliptical NMT]

Inputs: Desired NMT parameters: θ_1, θ_2, b, $\bar{Y}$.

Output: NMT Initial Condition: $X_0 = [x_0,\ y_0,\ z_0,\ \dot{x}_0,\ \dot{y}_0,\ \dot{z}_0]^T$.

1. Arbitrarily choose a value for $\phi > 0$,

2. Calculate ψ using
$$\tan(\phi - \psi) = 2\frac{\cos(\theta_1)}{\tan(\theta_2)}, \tag{A.7}$$

3. Calculate c using
$$c = \frac{b}{\sin(\theta_1)}\sqrt{\tan^2(\theta_2) + 4\cos^2(\theta_1)}, \tag{A.8}$$

4. Calculate the initial condition using (A.6), i.e.,
$$\begin{aligned} x_0 &= b\sin(\phi), & \dot{x}_0 &= b\omega\cos(\phi), \\ y_0 &= \bar{Y} + 2b\cos(\phi), & \dot{y}_0 &= -2b\omega\sin(\phi), \\ z_0 &= c\sin(\psi), & \dot{z}_0 &= c\omega\cos(\psi). \end{aligned} \tag{A.9}$$

End Procedure.

Figure A.3 shows three example elliptical NMTs centered at the origin along with the corresponding values for the parameters $\bar{Y}$, b, θ_1 and θ_2.

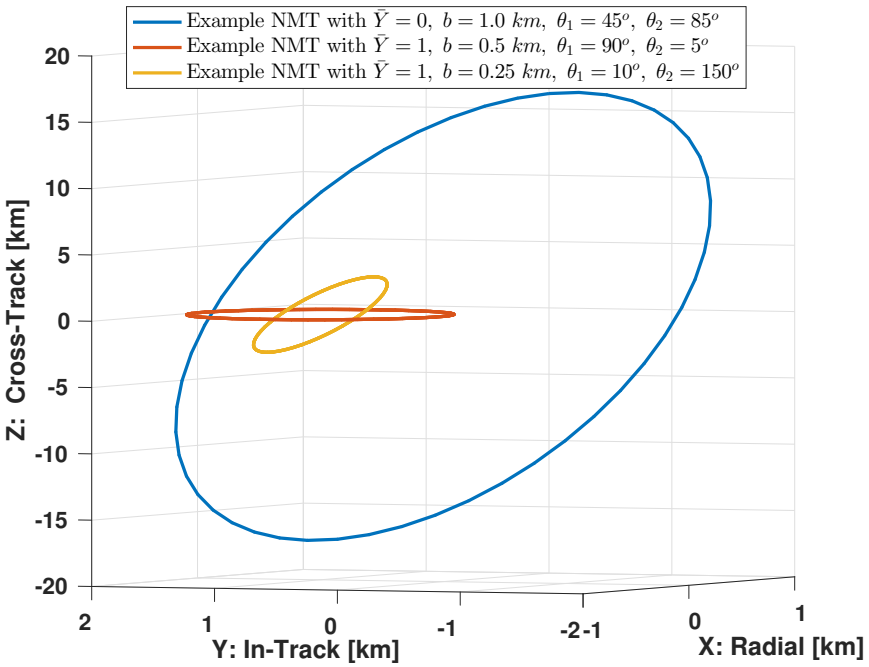


Figure A.3: Examples of elliptical NMTs

Open NMTs

If the initial condition is chosen such that $\bar{X} \neq 0$, and therefore (A.3) is not satisfied, the resulting NMT will be non-periodic (open). Two specific types of open NMTs useful for SRMP applications are straight line segment NMTs, and helical (or spiral) NMTs.

Straight line segment NMTs parallel to the y-axis in the x-y plane, can be generated by setting $b = c = 0$, and choosing a non-zero value for $\bar{X}$. These conditions correspond to an initial condition X_0 satisfying

$$\begin{aligned} x_0 &= \bar{X}, \quad \dot{x}_0 = 0, \\ y_0 &= \bar{Y}, \quad \dot{y}_0 = -\tfrac{3}{2}\omega\bar{X}, \\ z_0 &= 0, \quad \dot{z}_0 = 0. \end{aligned} \tag{A.10}$$

where $\bar{Y}$ can take any desired value. Note that if $\bar{X} > 0$, the spacecraft is "above" the target, and will drift in the $-y$ direction in the x-y plane along the line defined by $x = \bar{X}$. Conversely, if $\bar{X} < 0$, the spacecraft is "below" the target, and will drift in the $+y$ direction in the x-y plane along the line defined by $x = \bar{X}$. These straight line NMTs are useful in a SRMP setting as they allow the maneuvering satellite to drift along a line either directly above or below the target.

Helical NMTs are trajectories with motion in the x-y plane corresponding to an ellipse, with the center of the ellipse translating along the y direction. These types of NMTs, with the desired orientation, can be generated based on a nominal periodic elliptical NMT. Specifically, a nominal initial condition can be obtained using Procedure A.1. Then, the x and $\dot{y}$ components of this initial condition are modified using

$$x_0 = \bar{x}_0^{nom} + \bar{X}, \quad \dot{y}_0 = \dot{y}_0^{nom} - \tfrac{3}{2}\omega\bar{X}, \tag{A.11}$$

where the superscript ^{nom} denotes the nominal value. The resulting NMT will be a "travelling ellipse", with the center of motion beginning at the point $(\bar{X}, \bar{Y})$ in the x-y plane, and translating along the line defined by $x = \bar{X}$. If $\bar{X} > 0$, the ellipse translates in the $-y$ direction, whereas if $\bar{X} < 0$, the ellipse translates in the $+y$ direction. Note that larger values of $\bar{X}$ (in terms of absolute value) correspond to faster rates of translation. These "traveling ellipse" NMTs resemble helices or spirals when plotted in Hill's frame, as illustrated in Figure A.4.

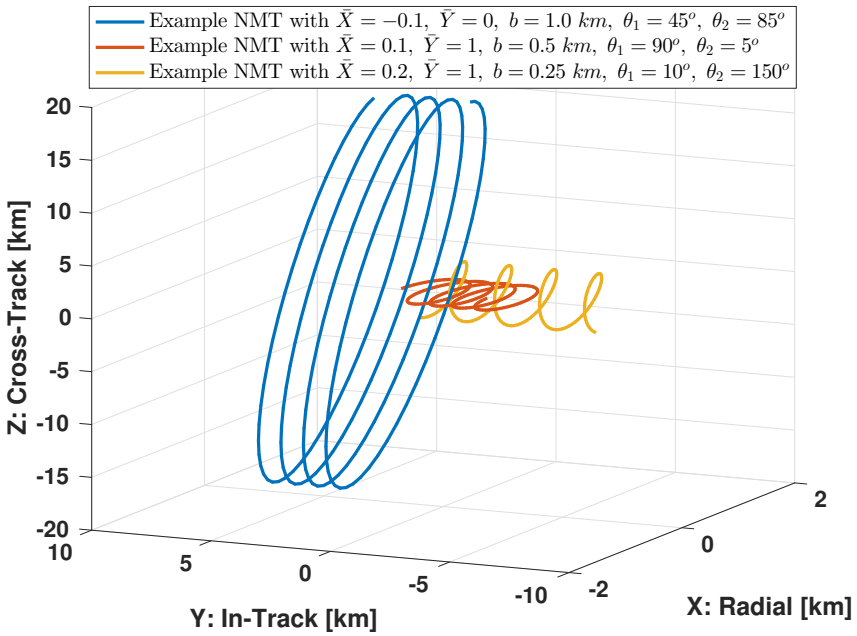


Figure A.4: Examples of helical NMTs

APPENDIX B

Safe Ellipsoidal Set Scale Factor Calculations

B.1 Maximum Scale Factor based on Control Constraint

To calculate the control limit on the scale factor ρ_k^s, denoted by ρ^u, note that the control constraint (3.12) can be equivalently expressed as

$$\eta_i^T K(X - X_n(k)) \leq u_{max},\ i = 1, 2, \cdots, 6, \tag{B.1}$$

where

$$\begin{gathered}\eta_1 = [1,\ 0,\ 0]^T,\ \eta_2 = [0,\ 1,\ 0]^T,\ \eta_3 = [0,\ 0,\ 1]^T, \\ \eta_4 = [-1,\ 0,\ 0]^T,\ \eta_5 = [0,\ -1,\ 0]^T,\ \eta_6 = [0,\ 0,\ -1]^T.\end{gathered} \tag{B.2}$$

The value for ρ^u, is found by considering the following convex optimization problem for $i = 1, 2, \cdots, 6$:

$$\begin{aligned}\max_{X_i} \quad & \eta_i^T K(X_i - X_n(k)), \\ \text{subject to} \quad & \tfrac{1}{2}(X_i - X_n(k))^T P(X_i - X_n(k)) \leq \alpha.\end{aligned} \tag{B.3}$$

If a value for α is found such that the solutions X_i^* of problem (B.3) satisfy

$$\max_i \{\eta_i^T K(X_i^* - X_n(k))\} = u_{max}, \tag{B.4}$$

then

$$\rho^u = 2\alpha. \tag{B.5}$$

To obtain the solution to problem (B.3), the matrix P is diagonalized as

$$P = V^T \Lambda V, \tag{B.6}$$

where V is an orthogonal matrix and Λ is a diagonal matrix with eigenvalues of P on the

diagonal. Then, by defining parameters ζ_i and h_i from

$$X_i - X_n(k) = V^T \Lambda^{-\frac{1}{2}} \zeta_i, \tag{B.7}$$

and

$$h_i^T = \eta_i^T K V^T \Lambda^{-\frac{1}{2}}, \tag{B.8}$$

the optimization problem is reformulated by substituting (B.6)-(B.8) into (B.3), yielding

$$\begin{aligned} \max_{\zeta_i} \quad & h_i^T \zeta_i, \\ \text{subject to} \quad & \tfrac{1}{2} \zeta_i^T \zeta_i \leq \alpha. \end{aligned} \tag{B.9}$$

The constrained optimization problem (B.9) corresponds to maximizing the inner product of two vectors over a two-norm ball, and has a solution given by

$$\zeta_i^* = \frac{h_i}{\|h_i\|_2} \sqrt{2\alpha}, \tag{B.10}$$

see [123]. Therefore, the maximum value for the objective function in (B.9) is given by

$$\|h_i\|_2 \sqrt{2\alpha}, \tag{B.11}$$

and, setting (B.11) equal to u_{max} and solving for α yields

$$\alpha = \frac{1}{2} \frac{u_{max}^2}{\|h_i\|_2^2}. \tag{B.12}$$

Therefore, by the logic discussed in (B.4) and (B.5), the safe ellipsoidal set scale factor limit based on the control constraint is given by

$$\rho^u = \min_i \frac{u_{max}^2}{\|h_i\|_2^2}. \tag{B.13}$$

Note that the scale factor ρ^u corresponds to the largest ellipsoidal set for which the control constraint (3.12) is satisfied pointwise within the set. However, the constraint (3.12) will also be satisfied pointwise in any ellipsoidal set with $\rho_k^s \leq \rho^u$.

Remark B.1. *From the derivation above, it is clear that the scale factor limit ρ^u is independent of the state vector $X_n(k)$, i.e., the state vector along an NMT corresponding to the center of the ellipsoidal set. Hence, ρ^u is constant for all $X_n(k)$, and for all NMTs $\mathcal{N}$, and must only be calculated once. This differs from the case considered in [48], in which the centers of ellipsoidal sets were forced equilibria and the value of ρ^u depended on the*

chosen forced equilibria.

B.2 Maximum Scale Factor based on Exclusion Zones

The maximum safe value for the ellipsoidal set scale factor considering the ith exclusion zone constraint (3.14), denoted by $\rho_k^{\mathcal{O}_i}$, is determined by solving a convex optimization problem in which the minimum-sized ellipsoidal set, centered at $X_n(k)$, is sought, that shares a common point with exclusion zone $\mathcal{O}_i(s_i, S_i)$. Defining

$$\bar{X} = X - X_n(k), \tag{B.14}$$

this optimization problem is given by

$$\begin{aligned} \min_{\bar{X},\, \rho_k^{\mathcal{O}_i}} \quad & \rho_k^{\mathcal{O}_i}, \\ \text{subject to} \quad & \bar{X}^T P \bar{X} \leq \rho_k^{\mathcal{O}_i}, \\ & (\Phi(\bar{X} + X_n(k)) - s_i)^T S_i (\Phi(\bar{X} + X_n(k)) - s_i) \leq 1. \end{aligned} \tag{B.15}$$

The solution to problem (B.15) is obtained via Karush-Kuhn-Tucker (KKT) conditions [116].

The Lagrangian for problem (B.15) is given by

$$L = \rho_k^{\mathcal{O}_i} + \lambda_1 (\bar{X}^T P \bar{X} - \rho_k^{\mathcal{O}_i}) + \lambda_2 \left[(\Phi(\bar{X} + X_n(k)) - s_i)^T S_i (\Phi(\bar{X} + X_n(k)) - s_i) - 1 \right], \tag{B.16}$$

where λ_1 and λ_2 are scalar Lagrange multipliers, and the KKT conditions are:

$$\frac{\partial L}{\partial \rho_k^{\mathcal{O}_i}} = 0, \tag{B.17a}$$

$$\frac{\partial L}{\partial \bar{X}} = 0, \tag{B.17b}$$

$$\lambda_i \geq 0,\ i = 1, 2, \tag{B.17c}$$

$$\lambda_1 (\bar{X}^T P \bar{X} - \rho_k^{\mathcal{O}_i}) = 0, \tag{B.17d}$$

$$\lambda_2 \left[(\Phi(\bar{X} + X_n(k)) - s_i)^T S_i (\Phi(\bar{X} + X_n(k)) - s_i) - 1 \right] = 0. \tag{B.17e}$$

$$\bar{X}^T P \bar{X} - \rho_k^{\mathcal{O}_i} \leq 0, \tag{B.17f}$$

$$(\Phi(\bar{X} + X_n(k)) - s_i)^T S_i (\Phi(\bar{X} + X_n(k)) - s_i) - 1 \leq 0. \tag{B.17g}$$

The condition (B.17a) yields the value for λ_1,

$$\frac{\partial L}{\partial \rho_k^{\mathcal{O}_i}} = 0 = 1 - \lambda_1 \implies \lambda_1 = 1, \tag{B.18}$$

and the condition (B.17b), evaluated at $\lambda_1 = 1$, yields the following expression for $\bar{X}$:

$$\bar{X} = -(P + \lambda_2 \Phi^T S_i \Phi)^{-1} \Phi^T S_i (\Phi X_n(k) - s_i) \lambda_2. \tag{B.19}$$

Note that since $P = P^T \succ 0$ and $\Phi^T S_i \Phi \succeq 0$, and $\lambda_2 \geq 0$, the matrix $(P + \lambda_2 \Phi^T S_i \Phi)$ is invertible.

Next, the value for the Lagrange multiplier $\lambda_2 \geq 0$ must be determined. There are two cases to consider, $\lambda_2 = 0$ and $\lambda_2 > 0$:

Case 1 $\lambda_2 = 0$**:** In this case, condition (B.17e) is trivially satisfied, and condition (B.19) implies that $\bar{X} = 0$. If condition (B.17g) is satisfied with $\bar{X} = 0$, it is concluded that $\lambda_2 = 0$ and, from condition (B.17d), it follows that $\rho_k^{\mathcal{O}_i} = 0$.

Remark B.2. *Note that the case of* $\lambda_2 = 0$ *yielding* $\rho_k^{\mathcal{O}_i} = 0$ *corresponds to the situation where the point* $X_n(k)$ *lies within an exclusion zone, i.e.,* $X_n(k) \in \mathcal{O}_i(S_i, s_i)$.

Case 2 $\lambda_2 > 0$**:** In this case, the term $[\cdot]$ in condition (B.17e) must be equal to zero. Therefore, the problem reduces to determining the value for $\lambda_2 > 0$ that is the root of

$$F(\lambda_2) = (\Phi(\bar{X} + X_n(k)) - s_i)^T S_i (\Phi(\bar{X} + X_n(k)) - s_i) - 1 = 0, \tag{B.20}$$

where $\bar{X}$ is given by (B.19). This scalar root-finding problem can be solved via a Newton-Rhapson iteration. Given an initial guess for λ_2, denoted λ_2^0 (see Remark B.3), this guess is updated as

$$\lambda_2^{c+1} = \lambda_2^c - \frac{F(\lambda_2)}{\frac{\partial F}{\partial \lambda_2^c}\Big|_{\lambda_2^c}}, \quad c = 0, 1, 2, \cdots \tag{B.21}$$

where

$$\frac{\partial F(\lambda_2)}{\partial \lambda_2} = \frac{\partial F(\lambda_2)}{\partial \bar{X}} \frac{\partial \bar{X}}{\partial \lambda_2}, \tag{B.22}$$

$$\frac{\partial F(\lambda_2)}{\partial \bar{X}} = 2(\Phi(\bar{X} + X_n(k)) - s_i)^T S_i \Phi, \tag{B.23}$$

and

$$\frac{\partial \bar{X}}{\partial \lambda_2} = (P + \lambda_2 \Phi^T S_i \Phi)^{-1} (-\Phi^T S_i (\Phi X_n(k) - s_i) - \Phi^T S_i \Phi \bar{X}). \tag{B.24}$$

This iteration is continued until the successive values for λ_2 stop changing to within some small tolerance.

Having obtained the value for λ_2, this value is plugged into (B.19) to obtain $\bar{X}$, and by condition (B.17d), $\rho_k^{\mathcal{O}_i}$ is given by

$$\bar{X}^T P \bar{X} = \rho_k^{\mathcal{O}_i} > 0. \tag{B.25}$$

Figure B.1 shows a projection of a safe set ellipsoidal set, with scale factor $\rho_k^{\mathcal{O}_i}$ and centered at a state vector $X_n(k)$ along an example closed NMT. The scale factor $\rho_k^{\mathcal{O}_i}$ was calculated using the method described above. Notice that the safe ellipsoidal set (in blue) has exactly one point of intersection with the exclusion zone (in black).

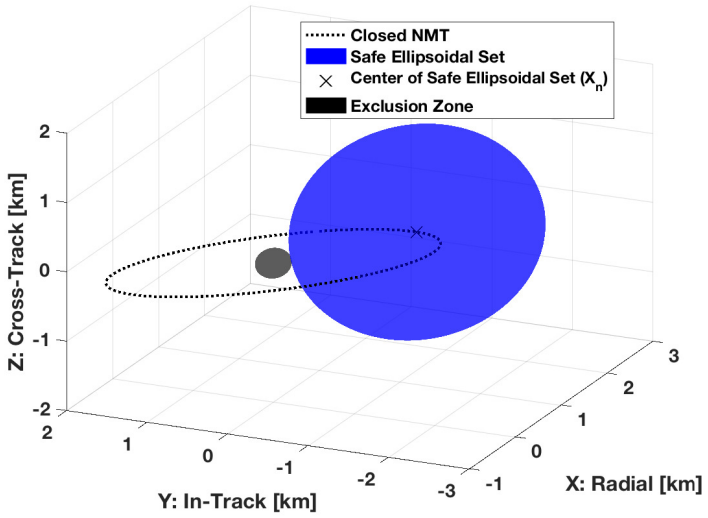


Figure B.1: A safe set with scale factor $\rho_k^{\mathcal{O}_i}$ projected onto the position space, $\mathbb{R}^3$

Remark B.3. *A separate value for $\rho_k^{\mathcal{O}_i}$ must be calculated for each state vector $X_n(k)$ along an NMT. This process can be sped up by re-using the solution to the Newton-iteration described above found for $X_n(k)$ as an initial guess to determine the value for $X_n(k+1)$. Specifically, for the first state vector $X_n(0)$ considered along an NMT, an initial guess for λ_2 is selected (simulations show that choosing $\lambda_2^0 = 1000$ yields good performance). Then, after the iteration has converged and the true value for λ_2 is found, this value is used as the initial guess in the Newton iteration to determine the value for λ_2 for state vector $X_n(1)$, etc.*

APPENDIX C

QCQPs

C.1 Calculation of $d^i(\rho_{k+1})$

The value of each $d^i(\rho_{k+1})$, given by

$$\begin{aligned} d^i(\rho_{k+1}) = \quad & \min_{e^+} \quad e^T P e - e^{+T} P e^+ \qquad i = 1, 2, \cdots, n_w, \\ & \text{subject to} \quad e^{+T} P e^+ = \rho_{k+1}, \\ & \qquad\qquad e = \bar{A}^{-1}(e^+ - B w^i), \end{aligned} \tag{C.1}$$

is obtained by reducing the problem to a scalar root finding problem via the method described in [97]. The objective function in (C.1) is re-written by replacing e with the expression in line 2 of the constraints, yielding

$$\begin{aligned} d^i(\rho_{k+1}) = \quad & \min_{e^+} \quad e^{+T} Q e^+ + 2 p^{iT} e^+ + v^i \\ & \text{subject to} \quad e^{+T} P e^+ - \rho_{k+1} = 0, \end{aligned} \tag{C.2}$$

where

$$\begin{aligned} Q &= \bar{A}^{-1T} P \bar{A}^{-1} - P \succ 0, \\ p^{iT} &= -w^{iT} B^T \bar{A}^{-1T} P \bar{A}^{-1}, \\ v^i &= w^{iT} B^T \bar{A}^{-1T} P \bar{A}^{-1} B w^i. \end{aligned} \tag{C.3}$$

Problem (C.2) is reformulated by simultaneously diagonalizing the matrices Q and P with an invertible matrix L such that $LQL^T = I_{6\times 6}$ and $LPL^T = P_D$, where P_D is a diagonal matrix [124]. The matrix L is calculated as follows:

$$L = (TU)^{-1}, \tag{C.4}$$

where

- $T = VD$,

- The matrix V has columns corresponding to the normalized eigenvectors of the matrix Q,
- D is a diagonal matrix with diagonal entries consisting of the square roots of the eigenvalues of the matrix Q,
- The matrix U has columns corresponding to the normalized eigenvectors of $T^{-1}PT^{-T}$.

Defining two vectors y and b^i from

$$\begin{aligned} e^+ &= L^T y, \text{ and} \\ b^{iT} &= p^{iT} L^T, \end{aligned} \tag{C.5}$$

Problem (C.2) becomes

$$\begin{aligned} d^i(\rho_{k+1}) = \quad & \min_{y} \quad y^T y + 2b^{iT} y + v^i \\ & \text{subject to} \quad y^T P_D y - \rho_{k+1} = 0. \end{aligned} \tag{C.6}$$

Next, define a vector z^i as

$$z^i = y + b^i. \tag{C.7}$$

With this substitution, (C.6) becomes

$$\begin{aligned} d^i(\rho_{k+1}) = \quad & \min_{z^i} \quad z^{iT} z^i + r^i \\ & \text{subject to} \quad z^{iT} P_D z^i + 2c^{iT} z^i + s^i = 0, \end{aligned} \tag{C.8}$$

where

$$\begin{aligned} r^i &= -b^{iT} b^i + v^i, \\ c^i &= -P_D b^i, \\ s^i &= -\rho_{k+1} + b^{iT} P_D b^i. \end{aligned} \tag{C.9}$$

Note that the minimizer to (C.8) is equivalent to the minimizer to

$$\begin{aligned} & \min_{z^i} \quad z^{iT} z^i \\ & \text{subject to} \quad z^{iT} P_D z^i + 2c^{iT} z^i + s^i = 0, \end{aligned} \tag{C.10}$$

since r^i does not depend on z^i. Therefore, the minimizer to (C.10) is sought using standard conditions for optimality.

The Lagrangian for (C.10) is

$$\mathcal{L} = z^{iT} z^i - \lambda(z^{iT} P_D z^i + 2c^{iT} z^i + s^i). \tag{C.11}$$

The first-order necessary conditions for optimality give the following:

$$\frac{\partial \mathcal{L}}{\partial z^i} = 0 \implies (I_{6\times 6} - \lambda P_D) z^i - \lambda c^i = 0, \tag{C.12}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \implies z^{iT} P_D z^i + 2 c^{iT} z^i + s^i = 0, \tag{C.13}$$

and the second-order condition to ensure a minimizer is

$$\frac{\partial^2 \mathcal{L}}{\partial z^{i2}} = I_{6\times 6} - \lambda P_D \succeq 0. \tag{C.14}$$

From (C.14), it follows that the optimal Lagrange multiplier, λ^* satisfies

$$\lambda^* \leq \tfrac{1}{P_{Dmax}}, \tag{C.15}$$

where P_{Dmax} is the maximum diagonal value of P_D. Further, if the sub-vector of c^i corresponding to P_{Dmax} is non-zero (note that simulations show this condition holds for the calculations being considered), then it is guaranteed that

$$\lambda^* < \tfrac{1}{P_{Dmax}}, \tag{C.16}$$

and therefore, the matrix

$$D_{\lambda^*} = I_{6\times 6} - \lambda^* P_D \tag{C.17}$$

is invertible.

From (C.12), the minimizer z^{i*} is given as a function of λ^*,

$$z^{i*} = \lambda^* D_{\lambda^*}^{-1} c^i. \tag{C.18}$$

Substituting (C.18) into (C.13) yields

$$f(\lambda) = \lambda^2 c^{iT} D_\lambda^{-1T} P_D D_\lambda^{-1} c^i + 2\lambda c^{iT} D_\lambda^{-1} c^i + s^i = 0. \tag{C.19}$$

In [97], it is shown that there exists a unique, finite root for (C.19) on the open interval $(-\inf, \frac{1}{P_{Dmax}})$. This root, corresponding to λ^*, can be found numerically using standard algorithms, such as bisection.

After λ^* is determined, the value for $d^i(\rho_{k+1})$ is obtained through the following steps:

- The minimizer to (C.10) (and to C.8) is calculated from (C.18) as $z^{i*} = \lambda^* D_{\lambda^*}^{-1} c^i$.
- The solution to (C.8) is given by $d^i(\rho_{k+1}) = z^{i*T} z^{i*} + r^i$.

C.2 Calculation of Transfer Trajectory Adjacency Requirements

In this appendix, a method is presented to obtain solutions to the optimization problem (3.104), repeated here for clarity:

$$\begin{aligned} F^l(\bar{k}, k_i, \hat{k}_j) = \quad & \max_{Y} \ \left[\bar{A}^{\bar{k}} e(0, \hat{k}_j) + g + f^l\right]^T P \left[\bar{A}^{\bar{k}} e(0, \hat{k}_j) + g + f^l\right], \\ \text{subject to} \quad & (Y - X_{ni}(k_i))^T P (Y - X_{ni}(k_i)) = \rho_{min} + \alpha, \end{aligned} \tag{C.20}$$

$l = 1, 2, \cdots, n_w$, where

$$\begin{gathered} e(0, \hat{k}_j) = Y - X_{nj}(\hat{k}_j), \\ g = g(\bar{k}, \hat{k}_j, k_j) = X_{nj}(\bar{k} + \hat{k}_j) - X_{nj}(k_j), \\ f^l = (\bar{A}^{\bar{k}-1} + \bar{A}^{\bar{k}-2} + \cdots + \bar{A}^0) B w^l. \end{gathered}$$

To obtain solutions to (C.20), simultaneous diagonalization is used to reformulate the problem. Then, using the Lagrangian for the reformulated problem and necessary conditions for optimality, a solution is obtained by solving a scalar root-finding problem. This method, described below, is adapted from the developments in [97].

The objective and constraint functions in (C.20) are expanded, and the notation is simplified, resulting in

$$\begin{aligned} F^l = \quad & \max_{Y} \quad Y^T Q Y + 2 q^T Y + d \\ & \text{subject to} \quad Y^T P Y + 2 p^T Y + h = 0, \end{aligned} \tag{C.21}$$

where

$$\begin{aligned} &Q = (\bar{A}^{\bar{k}})^T P (\bar{A}^{\bar{k}}), && q^T = (f^l + g)^T P \bar{A}^{\bar{k}} - \hat{X}^T Q, \\ &d = \hat{X}^T Q \hat{X} - 2 \hat{X}^T (\bar{A}^{\bar{k}})^T P (f^l + g) + (f^l + g)^T P (f^l + g), && \hat{X} = X_{nj}(\hat{k}_j), \\ &p^T = -X^T P, && h = X^T P X - (\rho_{min} + \alpha), \\ &X = X_{ni}(k_i). \end{aligned} \tag{C.22}$$

The matrices Q and P are simultaneously diagonalized with an invertible matrix, L, such that $LQL^T = I_{6\times 6}$ and $LPL^T = P_D$, where P_D is a diagonal, positive-definite matrix. The matrix L is obtained as follows:

$$L = (TU)^{-1}, \tag{C.23}$$

where

- $T = VD$,
- The matrix V has columns corresponding to the normalized eigenvectors of the matrix Q,
- D is a diagonal matrix with diagonal entries consisting of the square roots of the eigenvalues of the matrix Q,
- The matrix U has columns corresponding to the normalized eigenvectors of $T^{-1}PT^{-1T}$.

Using the matrix L, (C.21) is equivalently expressed as

$$\begin{aligned} F^l = \quad & \max_{\bar{Y}} \quad \bar{Y}^T\bar{Y} + 2b^TY + d \\ & \text{subject to} \quad \bar{Y}^T P_D \bar{Y} + 2m^T\bar{Y} + h = 0, \end{aligned} \tag{C.24}$$

where

$$Y = L^T\bar{Y}, \quad b = Lq, \quad m = Lp. \tag{C.25}$$

Next, let $Z = \bar{Y} + b$. With this substitution, (C.24) becomes

$$\begin{aligned} F^l = \quad & \max_{Z} \quad Z^TZ + r \\ & \text{subject to} \quad Z^T P_D Z + 2c^TZ + s = 0, \end{aligned} \tag{C.26}$$

where

$$r = d - b^Tb, \quad c = m - P_Db, \quad s = h + b^TP_Db - 2m^Tb. \tag{C.27}$$

Now, note that the maximizer of (C.26) is not affected by r, therefore, to find the maximizer to (C.26), it is sufficient to find the maximizer of

$$\begin{aligned} F^l = \quad & \max_{Z} \quad Z^TZ \\ & \text{subject to} \quad Z^T P_D Z + 2c^TZ + s = 0. \end{aligned} \tag{C.28}$$

The Lagrangian for (C.28) is given by

$$\mathcal{L} = Z^TZ - \lambda(Z^TP_DZ - 2c^TZ + s). \tag{C.29}$$

First-order necessary conditions for optimality yield the following expressions,

$$\frac{\partial \mathcal{L}}{\partial Z} = 0 \implies (I_{6\times 6} - \lambda P_D)Z - \lambda c = 0, \tag{C.30}$$

and

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \implies Z^T P_D Z - 2c^T Z + s = 0. \tag{C.31}$$

The second order condition to ensure a maximizer yields the following expression:

$$\frac{\partial^2 \mathcal{L}}{\partial Z^2} \leq 0 \implies D_\lambda = I_{6\times 6} - \lambda P_D \preceq 0. \tag{C.32}$$

From (C.32), because P_D is positive definite, it can be concluded that the optimal value for λ, denoted by λ^*, satisfies

$$\lambda^* \geq \frac{1}{P_{Dmin}}, \tag{C.33}$$

where P_{Dmin} is the minimum diagonal value of P_D. Furthermore, simulations show that

$$\lambda^* > \frac{1}{P_{Dmin}}, \tag{C.34}$$

and therefore the matrix $D_{\lambda^*} = I_{6\times 6} - \lambda^* P_D$ is invertible.

Next, condition (C.30) yields,

$$D_\lambda Z = \lambda c, \tag{C.35}$$

which is solved for Z:

$$Z = \lambda D_\lambda^{-1} c. \tag{C.36}$$

The value for Z in (C.36) is substituted for Z in (C.31), yielding the following scalar function of λ:

$$f(\lambda) = \lambda^2 c^T D_\lambda^{-1T} P_D D_\lambda^{-1} c + 2\lambda c^T D_\lambda^{-1} c + s = 0. \tag{C.37}$$

Hence, the solution to (C.28) is reduced to a scalar root finding problem. Considering (C.34), the root finding problem is posed as follows: Determine λ^*, the root of $f(\lambda)$ given by (C.37), over the open interval $(\frac{1}{P_{Dmin}},\ +\infty)$. This root can be determined via bisections.

After λ^* is determined, the value for F^l is obtained through the following steps:

- The maximizer to (C.28) (and to C.26) is calculated from (C.36) as $Z^* = \lambda^* D_{\lambda^*}^{-1} c$.
- The solution to C.26 is given by $F^l = Z^{*T} Z + r$.

APPENDIX D

Newton-Like Iteration for Fast Adjacency Calculations

This appendix describes a method to obtain local minima of the objective function in (3.120), repeated here for clarity,

$$G_c = \min_{t_i, \hat{t}_j} (X^*(t_i, \hat{t}_j) - X_{nj}(\hat{t}_j))^T P (X^*(t_i, \hat{t}_j) - X_{nj}(t_j)) - \rho_{\hat{t}_j}. \tag{D.1}$$

These local minima are obtained using a Newton-like iteration, based on work in [125].

Define the objective function in (D.1) as $F = F(t_i, \hat{t}_j)$, i.e.,

$$F = F(t_i, \hat{t}_j) = (X^*(t_i, \hat{t}_j) - X_{nj}(\hat{t}_j))^T P (X^*(t_i, \hat{t}_j) - X_{nj}(t_j)) - \rho_{\hat{t}_j}. \tag{D.2}$$

In (D.2),

$$X_{nj}(\hat{t}_j) = \exp(A_c \hat{t}_j) X_{0j}, \tag{D.3}$$

$$X_{ni}(t_i) = \exp(A_c t_i) X_{0i}, \tag{D.4}$$

where $X_{0(\cdot)}$ is the initial condition for NMT $\mathcal{N}_{(\cdot)}$, and

$$X^*(t_i, \hat{t}_j) = X_{nj}(\hat{t}_j) + \lambda^* r(t_i, \hat{t}_j), \tag{D.5}$$

where

$$r(t_i, \hat{t}_j) = X_{ni}(t_i) - X_{nj}(\hat{t}_j), \tag{D.6}$$

and

$$\lambda^* = 1 + \sqrt{\frac{\rho_{min} + \alpha}{r(t_i, \hat{t}_j)^T P r(t_i, \hat{t}_j)}}. \tag{D.7}$$

Plugging (D.7) and (D.5) into (D.2), the function F is re-written as

$$F = r(t_i,\hat{t}_j)^T P r(t_i,\hat{t}_j) + (\rho_{min}+\alpha) + 2\sqrt{\rho_{min}+\alpha}\sqrt{r(t_i,\hat{t}_j)^T P r(t_i,\hat{t}_j)} - \rho_{\hat{t}_j}. \quad \text{(D.8)}$$

Therefore, a local minima of (D.8) is sought, starting from an initial guess $(t_i,\hat{t}_j)$, via a Newton iteration. This iteration requires the first and second partial derivatives of F with respect to both t_i and $\hat{t}_j$. The derivatives for the first three terms on the RHS of (D.8) can be easily obtained using (D.3), (D.4) and (D.6). Recall that the fourth term on the RHS of (D.8), i.e., $\rho_{\hat{t}_j}$, corresponds to the SPI scale factor for state vector $X_{nj}(\hat{t}_j)$. A continuously differentiable expression for $\rho_{\hat{t}_j}$ is not available. However, we do have discrete values $\rho_{\hat{k}_j}$ for $\hat{k}_j = [0, k_{max}(or\ k_{max}^{\bar{N}_j})]$. Therefore, to approximate $\rho_{\hat{t}_j}$ at any given $\hat{t}_j$, and therefore obtain the required derivatives, a quadratic function is fit to the three discrete $\rho_{\hat{k}_j}$ values closest to $\rho(\hat{t}_j)$. This process is described in Section (D.1). Given the approximations of $\rho_{\hat{t}_j}$, $\frac{d\rho_{\hat{t}_j}}{d\hat{t}_j}$ and $\frac{d^2\rho_{\hat{t}_j}}{d\hat{t}_j^2}$, the Newton-like iterative method to obtain the minimum of (D.8) proceeds as follows.

Using (D.8), and the shortened notation $X(t_i) = X_{ni}(t_i)$, $X(t_j) = X_{nj}(t_j)$ and $X(\hat{t}_j) = X_{nj}(\hat{t}_j)$, the required partial derivatives of F are given by

$$\frac{\partial F}{\partial t_i} = \left(1 + \frac{\sqrt{\rho_{min}+\alpha}}{\sqrt{r(t_i,\hat{t}_j)^T P r(t_i,\hat{t}_j)}}\right) 2r(t_i,\hat{t}_j)^T P A_c X(t_i), \quad \text{(D.9)}$$

$$\frac{\partial F}{\partial \hat{t}_j} = -\left(1 + \frac{\sqrt{\rho_{min}+\alpha}}{\sqrt{r(t_i,\hat{t}_j)^T P r(t_i,\hat{t}_j)}}\right) 2r(t_i,\hat{t}_j)^T P A_c X(\hat{t}_j) - \frac{\partial \rho_{\hat{t}_j}}{\partial \hat{t}_j}, \quad \text{(D.10)}$$

$$\begin{aligned}\frac{\partial^2 F}{\partial t_i^2} = \;& 2X(t_i)^T A_c^T P A_c X(t_i) + 2r(t_i,\hat{t}_j)^T P A_c A_c X(t_i) \\ & + 2\frac{\sqrt{\rho_{min}+\alpha}}{\sqrt{r(t_i,\hat{t}_j)^T P r(t_i,\hat{t}_j)}}\left[X(t_i)^T A_c^T P A_c X(t_i) + r(t_i,\hat{t}_j)^T P A_c A_c X(t_i)\right] \\ & - 2\frac{\sqrt{\rho_{min}+\alpha}}{[r(t_i,\hat{t}_j)^T P r(t_i,\hat{t}_j)]^{3/2}}\left[r(t_i,\hat{t}_j)^T P A_c X(t_i)\right]^2,\end{aligned} \quad \text{(D.11)}$$

$$\begin{aligned}\frac{\partial^2 F}{\partial \hat{t}_j^2} = \;& 2X(\hat{t}_j)^T A_c^T P A_c X(\hat{t}_j) - 2r(t_i,\hat{t}_j)^T P A_c A_c X(\hat{t}_j) \\ & + 2\frac{\sqrt{\rho_{min}+\alpha}}{\sqrt{r(t_i,\hat{t}_j)^T P r(t_i,\hat{t}_j)}}\left[X(\hat{t}_j)^T A_c^T P A_c X(\hat{t}_j) - r(t_i,\hat{t}_j)^T P A_c A_c X(\hat{t}_j)\right] \\ & - 2\frac{\sqrt{\rho_{min}+\alpha}}{[r(t_i,\hat{t}_j)^T P r(t_i,\hat{t}_j)]^{3/2}}\left[r(t_i,\hat{t}_j)^T P A_c X(\hat{t}_j)\right]^2 - \frac{\partial^2 \rho_{\hat{t}_j}}{\partial \hat{t}_j^2},\end{aligned} \quad \text{(D.12)}$$

$$\begin{aligned}\frac{\partial^2 F}{\partial t_i \partial \hat{t}_j} = \frac{\partial}{\partial t_i}\frac{\partial F}{\partial \hat{t}_j} = \;& -2X(\hat{t}_j)^T A_c^T P A_c X(t_i) - 2\frac{\sqrt{\rho_{min}+\alpha}}{\sqrt{r(t_i,\hat{t}_j)^T P r(t_i,\hat{t}_j)}}\left[X(\hat{t}_j)^T A_c^T P A_c X(t_i)\right] \\ & + 2\frac{\sqrt{\rho_{min}+\alpha}}{[r(t_i,\hat{t}_j)^T P r(t_i,\hat{t}_j)]^{3/2}}\left[r(t_i,\hat{t}_j)^T P A_c X(t_i)\right]\left[r(t_i,\hat{t}_j)^T P A_c X(\hat{t}_j)\right],\end{aligned} \quad \text{(D.13)}$$

and

$$\frac{\partial^2 F}{\partial \hat{t}_j \partial t_i} = \frac{\partial}{\partial \hat{t}_j}\frac{\partial F}{\partial t_i} = \begin{array}{l} -2X(t_i)^T A_c^T P A_c X(\hat{t}_j) - 2\frac{\sqrt{\rho_{min}+\alpha}}{\sqrt{r(t_i,\hat{t}_j)^T P r(t_i,\hat{t}_j)}}\left[X(t_i)^T A_c^T P A_c X(\hat{t}_j)\right] \\ +2\frac{\sqrt{\rho_{min}+\alpha}}{[r(t_i,\hat{t}_j)^T P r(t_i,\hat{t}_j)]^{3/2}}\left[r(t_i,\hat{t}_j)^T P A_c X(\hat{t}_j)\right]\left[r(t_i,\hat{t}_j)^T P A_c X(t_i)\right]. \end{array} \tag{D.14}$$

Note from (D.13) and (D.14) that $\frac{\partial^2 F}{\partial t_i \partial \hat{t}_j} = \frac{\partial^2 F}{\partial \hat{t}_j \partial t_i}$.

Remark D.1. *Because $\rho_{\hat{t}_j}$ is approximated by a quadratic function for each value of $\hat{t}_j$, the derivatives $\frac{\partial \rho_{\hat{t}_j}}{\partial \hat{t}_j}$ and $\frac{\partial^2 \rho_{\hat{t}_j}}{\partial \hat{t}_j^2}$, in (D.10) and (D.12), respectively, are always defined.*

To determine t_i^* and $\hat{t}_j^*$ that correspond to a local minima of F, the following iterative method is used. Let $\chi = [t_i\ \hat{t}_j]^T$, and choose an initial guess $\chi_0 = [t_{i0}\ \hat{t}_{j0}]^T$. This guess is updated as follows, for $c \geq 0$:

$$\chi_{c+1} = \chi_c + \nu_c p_c, \tag{D.15}$$

where p_c is the "Newton step" and $\nu_c > 0$ is the step length. Calculations to obtain p_c and ν_c are now described in detail.

The Newton step p_c is calculated such that the iteration will converge to local minima, i.e.,

$$p_c = -\bar{H}_c^{-1} g_c, \tag{D.16}$$

where

$$g_c = \left[\begin{array}{c} \frac{\partial F}{\partial t_i} \\ \frac{\partial F}{\partial \hat{t}_j} \end{array}\right]\Bigg|_{\chi_c}, \tag{D.17}$$

and the matrix $\bar{H}_c$ is calculated as follows. The Hessian matrix of F is given by

$$H_c = \left[\begin{array}{cc} \frac{\partial^2 F}{\partial t_i^2} & \frac{\partial^2 F}{\partial \hat{t}_j \partial t_i} \\ \frac{\partial^2 F}{\partial t_i \partial \hat{t}_j} & \frac{\partial^2 F}{\partial \hat{t}_j^2} \end{array}\right]\Bigg|_{\chi_c}, \tag{D.18}$$

and the spectral decomposition of H_c is given by $H_c = V\Lambda V^{-1}$ where the columns of V correspond to eigenvectors of H_c and $\Lambda = \text{diag}(\lambda)$ where λ is a vector containing eigenvalues of $\mathcal{H}_c$.

To ensure the step p_c is in a descent direction, the matrix $\bar{H}_c$ is given by

$$\bar{H}_c = V\bar{\Lambda}V^{-1}, \tag{D.19}$$

where $\bar{\Lambda} = \text{diag}(\bar{\lambda})$ and

$$\bar{\lambda}_i = \begin{cases} |\lambda_i| & \text{if } |\lambda_i| \geq \delta > 0, \\ \delta > 0 & \text{otherwise.} \end{cases} \tag{D.20}$$

In (D.20), the parameter $\delta > 0$ can be tuned as desired. Simulations show that setting $\delta = 1 \times 10^{-16}$ yields good performance. Note that the matrix $\bar{H}_c$ is always invertible.

The step size ν_c in (D.15) is calculated based Wolfe's condition for sufficient descent [126], using a backtracking line-search. Starting at $\nu_c = 1$, the largest possible ν_c is sought that satisfies

$$F(\chi_{c+1}) \leq F(\chi_c) - \mu \nu_c g_c^T p_c, \tag{D.21}$$

where $\chi_{(\cdot)} = [t_{i(\cdot)} \ \hat{t}_{j(\cdot)}]^T$, the notation $F(\chi_{(\cdot)})$ is equivalent to $F(t_{i(\cdot)}, \hat{t}_{j(\cdot)})$, and $\mu > 0$ is a tunable parameter (simulations show that $\mu = 1/10$ provides good results). The value for ν_c is updated as $\nu_c = 0.75\nu_c$ as needed until the first value satisfying (D.21) is found.

Iteration termination conditions

The iteration (D.15) is performed until a desired tolerance is reached, that is, until

$$| \ \|\chi_k\|_2 - \|\chi_{k-1}\|_2 \ | \leq \epsilon, \tag{D.22}$$

where $\epsilon > 0$. In simulations, the value $\epsilon = 1 \times 10^{-6}$ is used.

Simplified cases of (D.8)

In general, local minima of (D.8) must be found with respect to both t_i and $\hat{t}_j$, where both t_i and $\hat{t}_j$ may vary. Therefore, the Newton iteration must be carried out as described above. However, for certain adjacency types, and depending on the nodes involved, one or both of t_i and $\hat{t}_j$ is fixed, and this simplifies the calculations. These special cases are as follows:

1. For adjacency types (A2) and (A4), the origin node is a single state vector, $X_{ni}(k_i)$. Therefore, the variable t_i in (D.8) has the fixed value $t_i = k_i \Delta T$. In this case, the Newton iteration to solve (D.8) is carried over only the single variable $\hat{t}_j$. This makes calculations simpler and faster. Specifically, choose an initial guess $\hat{t}_{j0}$ and update the guess as

$$\hat{t}_{j(c+1)} = \hat{t}_{j(c)} - \nu_c \left. \frac{\frac{\partial F}{\partial \hat{t}_j}}{\left| \frac{\partial^2 F}{\partial \hat{t}_j^2} \right|} \right|_{\hat{t}_{j(c)}}, \tag{D.23}$$

   where $\nu_k \in (0, 1]$ is chosen via backtracking line search to be the largest possible value which leads to a decrease in the function value. The iteration is terminated when $|\hat{t}_{j(k)} - \hat{t}_{j(k-1)}| \leq \epsilon$.

2. For adjacency types (A1) and (A3), if the origin node is a stationary point "periodic

NMT," then the variable t_i in (D.8) can be arbitrarily fixed at any $t_i \in [0, k_{max}\Delta T]$. Hence, the Newton iteration for this case is the same as that described in number (1) above.

3. For adjacency type (A1), if the destination node is a stationary point "periodic NMT," then the variable $\hat{t}_j$ in (D.8) can be arbitrarily fixed at any $\hat{t}_j \in [0, k_{max}\Delta T]$. In this case, the Newton iteration to solve (D.8) is carried over only the single variable t_i. Specifically, choose an initial guess t_{i0} and update the guess as

$$t_{i(c+1)} = t_{i(c)} - \nu_c \left. \frac{\frac{\partial F}{\partial t_i}}{\left| \frac{\partial^2 F}{\partial t_i^2} \right|} \right|_{t_{i(c)}}, \tag{D.24}$$

where $\nu_c \in (0, 1]$ is chosen via backtracking line search to be the largest possible value which leads to a decrease in the function value. The iteration is terminated when when $|t_{i(c)} - t_{i(c-1)}| \leq \epsilon$.

4. For adjacency type (A2), if the destination node is a stationary point "periodic NMT," then the variable and $\hat{t}_j$ can be arbitrarily fixed at any $\hat{t}_j \in [0, k_{max}\Delta T]$. In this case, the function $F(t_i, \hat{t}_j)$ is constant and (D.8) can be solved by a single function evaluation at $t_i = k_i \Delta T$, where k_i corresponds to the origin node (state vector), and a chosen value of $\hat{t}_j$.

5. For adjacency type (A1), if both the origin and destination nodes are stationary point "periodic NMTs", then the variables t_i and $\hat{t}_j$ can be arbitrarily fixed at any $t_i \in [0, k_{max}\Delta T]$ and any $\hat{t}_j \in [0, k_{max}\Delta T]$. In this case the function $F(t_i, \hat{t}_j)$ is constant and (D.8) can be solved by a single function evaluation at the chosen t_i and $\hat{t}_j$.

D.1 Approximation of $\rho_{\hat{t}_j}$

Consider an arbitrary value for $\hat{t}_j$. The discrete-time instant closest to this value is given by

$$k_2 = \text{round}(\tfrac{\hat{t}_j}{\Delta T}). \tag{D.25}$$

Then, the three discrete-time instant values closest to $\hat{t}_j$ are simply

$$k_1 = k_2 - 1, \quad k_2, \text{ and } k_3 = k_2 + 1. \tag{D.26}$$

Remark D.2. *If the destination NMT is closed and $k_1,\ k_2$, or k_3 are outside of the interval*

$[0, k_{max}]$, *they can be moved into the interval due to the periodicity. If the destination NMT is open and* $k_1,\ k_2$, *or* k_3 *are outside of the interval* $[0, k_{max}^{\bar{N}_j}]$, *then they are set to the appropriate extreme values at either end of this interval.*

The continuous-time values that correspond to these discrete-time instances are

$$\begin{aligned} \hat{t}_{j1} &= \Delta T k_1, \\ \hat{t}_{j2} &= \Delta T k_2, \\ \hat{t}_{j3} &= \Delta T k_3. \end{aligned} \tag{D.27}$$

Hence, we seek to fit a quadratic function through the three points given by

$$(\hat{t}_{j1}, \rho_{k_1}),\ \ (\hat{t}_{j2}, \rho_{k_2}),\ \ (\hat{t}_{j3}, \rho_{k_3}). \tag{D.28}$$

Let the quadratic function be given by

$$\rho_{\hat{t}_j} = a_{\hat{t}_j} \hat{t}_j^2 + b_{\hat{t}_j} \hat{t}_j + c_{\hat{t}_j}. \tag{D.29}$$

The coefficients $a_{\hat{t}_j}$, $b_{\hat{t}_j}$, and $c_{\hat{t}_j}$ may vary with $\hat{t}_j$ and are determined as

$$\begin{bmatrix} a_{\hat{t}_j} \\ b_{\hat{t}_j} \\ c_{\hat{t}_j} \end{bmatrix} = \begin{bmatrix} \hat{t}_{j1}^2 & \hat{t}_{j1} & 1 \\ \hat{t}_{j2}^2 & \hat{t}_{j2} & 1 \\ \hat{t}_{j3}^2 & \hat{t}_{j3} & 1 \end{bmatrix}^{-1} \begin{bmatrix} \rho_{k_1} \\ \rho_{k_2} \\ \rho_{k_3} \end{bmatrix}. \tag{D.30}$$

Then, the first and second derivatives of $\rho_{\hat{t}_j}$ are given by

$$\frac{d\rho_{\hat{t}_j}}{d\hat{t}_j} = 2a_{\hat{t}_j} \hat{t}_j + b_{\hat{t}_j}, \tag{D.31}$$

and

$$\frac{d^2\rho_{\hat{t}_j}}{d\hat{t}_j^2} = 2a_{\hat{t}_j}. \tag{D.32}$$

APPENDIX E

Proofs

E.1 Proof of Proposition 3.1

Proposition 3.1. *The following holds:*

$$\max_{e_1^+ \in D_1^+} \; w^T B^T P e_1^+ = \sqrt{\rho_{k+1}} \sqrt{w^T B^T P B w} \tag{E.1}$$

Proof. The result follows from "standard" conditions for optimality [87]. The Lagrangian for (E.1) is

$$\mathcal{L} = w^T B^T P e_1^+ + \mu(e_1^{+T} P e_1^+ - \rho_{k+1}), \tag{E.2}$$

where μ is a Lagrange multiplier. First-order necessary conditions for optimality yield the following:

$$\frac{\partial \mathcal{L}}{\partial e_1^+} = 0 \implies PBw + 2\mu P e_1^+ = 0, \tag{E.3}$$

$$\frac{\partial \mathcal{L}}{\partial \mu} = 0 \implies e_1^{+T} P e_1^+ - \rho_{k+1} = 0. \tag{E.4}$$

Equation (E.3) is solved for e_1^+,

$$e_1^+ = -\frac{1}{2\mu} Bw, \tag{E.5}$$

and this value is plugged into (E.4) yielding

$$\frac{1}{4\mu^2} w^T B^T P B w - \rho_{k+1} = 0. \tag{E.6}$$

Equation (E.6) is solved for μ,

$$\mu = \pm \frac{1}{2} \sqrt{\frac{w^T B^T P B w}{\rho_{k+1}}}. \tag{E.7}$$

Because we are searching for the maximum value of (E.1), we select the negative value for μ. This value for μ is plugged into (E.5) yielding

$$e_1^+ = \sqrt{\frac{\rho_{k+1}}{w^T B^T P B w}} B w, \tag{E.8}$$

which is then plugged back into the objective function of (E.1) which yields

$$\begin{aligned} w^T B^T P e_1^+ &= w^T B^T P \sqrt{\tfrac{\rho_{k+1}}{w^T B^T P B w}} B w \\ &= \sqrt{\tfrac{\rho_{k+1}}{w^T B^T P B w}} w^T B^T P B w, \\ &= \sqrt{\rho_{k+1}} \sqrt{w^T B^T P B w}. \end{aligned} \tag{E.9}$$

To verify that this value is a maximum, the second order condition for optimality, $\partial^2 L / \partial e_1^{+2} \prec 0$ is checked:

$$\frac{\partial^2 L}{\partial e_1^{+2}} = 2\mu P = -\sqrt{\frac{w^T B^T P B w}{\rho_{k+1}}} P \prec 0. \tag{E.10}$$

Note that (E.10) holds since $P \succ 0$. □

E.2 Proofs of Propositions 5.3 and 5.4

Recall the following definitions:

Definition E.1 (Continuous Function)**.** *A function $\phi : \mathbb{V} \to \mathbb{W}$ is continuous at $\bar{x} \in \mathbb{V}$ if for all $\epsilon > 0$ there exists $\delta(\epsilon, \bar{x})$ such that*

$$\|x - \bar{x}\| < \delta \implies \|\phi(x) - \phi(\bar{x})\| < \epsilon.$$

Definition E.2 (Global Lipschitz Continuity)**.** *A function $\phi : \mathbb{V} \to \mathbb{W}$ is globally Lipschitz continuous on $\mathbb{V}$ if there exists $L < \infty$ such that*

$$\textit{for all } x,\ y \in \mathbb{V},\ \|\phi(x) - \phi(y)\| \leq L\|x - y\|.$$

For a sequence, $x(k)$, $k \in \mathbb{Z}_{\geq 0}$,

Definition E.3 (Limit of a sequence)**.**

$$\lim_{k \to \infty} x(k) = \bar{x} \implies \textit{for all } \epsilon > 0,\ \exists\, N(\epsilon) < \infty \textit{ such that for all } k \geq N,\ \|x(k) - \bar{x}\| < \epsilon.$$

Note that the limit $\lim_{k\to\infty} x(k) = \bar{x}$ can also be written as $x(k) \to \bar{x}$ as $k \to \infty$.

Consider a function $f(x(k)) : \mathbb{V} \to \mathbb{W}$. Note that Proposition 5.3 provides a sufficient condition for the property

$$y(k) \to x(k) \text{ as } k \to \infty \implies f(y(k)) \to f(x(k)) \text{ as } k \to \infty, \tag{E.11}$$

while Proposition 5.4 provides a sufficient condition to the inverse to (E.11), i.e.,

$$f(y(k)) \to f(x(k)) \text{ as } k \to \infty \implies y(k) \to x(k) \text{ as } k \to \infty. \tag{E.12}$$

The proofs to Propositions 5.3 and 5.4 are obtained by first considering Proposition E.1 below. Note that Propositions E.1 and 5.3 and are based on results found in [127].

Proposition E.1. *Suppose the following properties hold:*

1. $y(k) \to x(k)$ *as* $k \to \infty$*,*
2. *There exists a function* $\phi : \mathbb{R} \to \mathbb{R}$*, continuous at 0 and satisfying* $\phi(0) = 0$*, such that* $\|f(y(k)) - f(x(k))\| \leq \phi(\|y(k) - x(k)\|)$.

Then, $f(y(k)) \to f(x(k))$ *as* $k \to \infty$.

Proof. From item (2) above, and the definition of a continuous function, for all $\epsilon > 0$, there exists a $\delta > 0$ such that

$$\|y(k) - x(k)\| < \delta \implies \|f(y(k)) - f(x(k))\| \leq \phi(\|y(k) - x(k)\| < \epsilon. \tag{E.13}$$

Suppose for a given ϵ, the corresponding δ is obtained. Next, from item (1), it follows that

$$[y(k) - x(k)] \to 0 \text{ as } k \to \infty. \tag{E.14}$$

From (E.14) and the definition of the limit of a sequence, there exists $N < \infty$ such that for all $k \geq N$,

$$\|y(k) - x(k)\| < \delta. \tag{E.15}$$

Combining (E.13) and (E.15), it follows that for all $\epsilon > 0$ there exists $N < \infty$ such that, for all $k \geq N$,

$$\|f(y(k)) - f(x(k))\| \leq \phi(\|y(k) - x(k)\|) < \epsilon, \tag{E.16}$$

therefore $\lim_{k\to\infty} f(y(k)) - f(x(k)) = 0$, i.e., $f(y(k)) \to f(x(k))$ as $k \to \infty$. □

The proof for Proposition 5.3 now follows from Proposition E.1. Recall Proposition 5.3:
Proposition 5.3. *Suppose the following properties hold:*

1. $y(k) \to x(k)$ as $k \to \infty$,
2. f *is globally Lipschitz on* $\mathbb{V}$.

Then, $f(y(k)) \to f(x(k))$ as $k \to \infty$.

Proof. From item (2), there exists $L < \infty$ such that, for all $y(k), x(k) \in \mathbb{V}$,

$$\|f(y(k)) - f(x(k))\| \leq L\|y(k) - x(k)\|. \tag{E.17}$$

Define a function ϕ as

$$\phi(\|y(k) - x(k)\|) = L\|y(k) - x(k)\|. \tag{E.18}$$

Therefore,

$$\|f(y(k)) - f(x(k))\| \leq \phi(\|y(k) - x(k)\|). \tag{E.19}$$

Note that $\phi(0) = 0$, and ϕ is continuous at 0 (since for all $\epsilon > 0$, $\|y(k) - x(k)\| < \frac{\epsilon}{L} \implies \|\phi(\|y(k) - x(k)\|)\| = L\|y(k) - x(k)\| < L\frac{\epsilon}{L} = \epsilon$). Therefore, by Proposition E.1, $f(y(k)) \to f(x(k))$ as $k \to \infty$. □

The proof for Proposition 5.4 now follows from Propositions E.1 and 5.3. Recall Proposition 5.4:
Proposition 5.4. *Suppose the following properties hold:*

1. $f(x(k)) \to f(y(k))$ as $k \to \infty$,
2. *the inverse of* $f(x(k))$ *exists for all* $x(k) \in \mathbb{V}$*, and this inverse is unique and globally Lipschitz, i.e., for all* $x(k) \in \mathbb{V}$*, there exists a function* $f^{-1} : \mathbb{W} \to \mathbb{V}$ *such that, for all* $x(k) \in \mathbb{V}$, $f(x(k)) = w(k) \iff f^{-1}(w(k)) = x(k)$.

Then, $x(k) \to y(k)$ as $k \to \infty$.

Proof. Suppose item (1) holds, i.e., $f(x(k)) \to f(y(k))$ as $k \to \infty$, where $x(k), y(k) \in \mathbb{V}$ are arbitrary. Let

$$f(x(k)) = w_x(k) \in \mathbb{W} \text{ and } f(y(k)) = w_y(k) \in \mathbb{W}. \tag{E.20}$$

From Proposition 5.3, and because f^{-1} is globally Lipschitz (item (2) above), it follows that

$$w_x(k) \to w_y(k) \implies f^{-1}(w_x(k)) \to f^{-1}(w_y(k)) \text{ as } k \to \infty, \tag{E.21}$$

and, because the inverse f^{-1} is unique, it follows that

$$f^{-1}(w_x(k)) = x(k), \text{ and } f^{-1}(w_y(k)) = y(k). \tag{E.22}$$

Substituting (E.20) and (E.22) into (E.21) yields

$$f(x(k)) \to f(y(k)) \implies x(k) \to y(k) \text{ as } k \to \infty. \tag{E.23}$$

Because $x(k)$ and $y(k)$ are arbitrary, the proof is complete. □

APPENDIX F

Solution Methods for the Vertex Coloring Problem

For an undirected graph $\mathcal{G}(\mathbb{S}, \mathbb{E})$ where $\mathbb{S}$ is the vertex (node) set and $\mathbb{E}$ is the edge set, the VCP and mVCP are defined as follows:

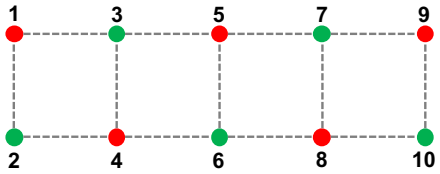
Vertex coloring problem (VCP): Find an assignment of colors to each node (vertex) of $\mathcal{G}(\mathbb{S}, \mathbb{E})$ such that, for all $i, j \in \mathbb{S}$, if $(i, j) \in \mathbb{E}$, then i and j have different colors.

Minimal Vertex coloring problem (mVCP): Find the *minimal* number of colors that can be assigned to nodes (vertices) of $\mathcal{G}(\mathbb{S}, \mathbb{E})$, such that, for all $i, j \in \mathbb{S} \times \mathbb{S}$, if $(i, j) \in \mathbb{E}$, then i and j have different colors.

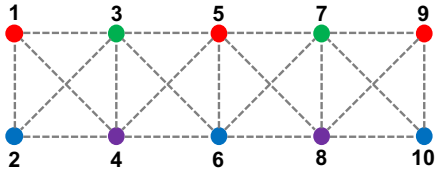
Recall from Section 5.2.4.2 that any solution to the VCP can be used to generate the required sequence of turns for the parameter update procedure (Procedure 5.1), however, the sequence of turns obtained from the solution to the mVCP is desirable because this may provide faster convergence times.

The solution to the mVCP is commonly referred to as the *chromatic number* of $\mathcal{G}(\mathbb{S}, \mathbb{E})$, denoted by $c_{\mathcal{G}}$. While solving the mVCP problem is NP-Hard, for graphs with a small number of vertices or graphs with a simple structure, e.g., complete graphs, the solution to the mVCP can be obtained by inspection. For example, Figure F.1 shows three graphs for which the chromatic number, and corresponding vertex coloring, is easily obtained.

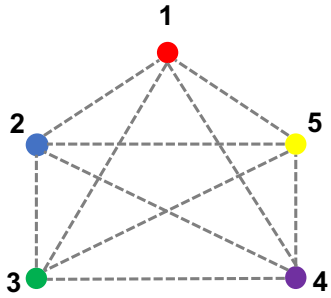
For more complex graphs, several exact algorithms to solve the mVCP exist based on integer programming and branch-and-bound techniques, see, e.g., [118, 128]. However, the time required to run these algorithms can become untenable for graphs with a large number of nodes. To overcome this limitation, many heuristics have been developed based on greedy algorithms, and local search techniques, among others. Some of these heuristic methods, and associated required computation times, are summarized in [118, 129].



(a) Example with 10 nodes and $c_{\mathcal{G}} = 2$.



(b) Example with 10 nodes and $c_{\mathcal{G}} = 4$.



(c) Example with 5 nodes and $c_{\mathcal{G}} = 5$.

Figure F.1: Simple graphs showing solution to mVCP obtained by inspection

If convergence time for the parameter governor is unimportant, it may be acceptable to determine the sequence of turns based on any feasible solution to the VCP (as opposed to using the exact solution to the mVCP). The advantage of this approach is that such feasible solutions can be obtained quickly. For example, the trivial solution of using a different color for each node in a graph is always a feasible solution to the VCP.

Additionally, it is possible to determine an upper-bound on the chromatic number of a graph (and therefore on the number of colors needed for a feasible solution to the VCP), and then, in certain cases, use this upper bound to obtain such a feasible solution. Specifically, due to Brook's Theorem [117, 130], the chromatic number of any graph is upper-bound by

$$c_{\mathcal{G}} \leq \Delta_{\mathcal{G}} + 1, \tag{F.1}$$

where $\Delta_{\mathcal{G}}$ is the maximum degree of the graph, defined as the maximum number of edges incident to any vertex in the graph. Furthermore, $c_{\mathcal{G}} = \Delta_{\mathcal{G}} + 1$ if and only if a) $\Delta_{\mathcal{G}} \neq 2$ and $\mathcal{G}$ has a complete graph with $\Delta_{\mathcal{G}} + 1$ nodes as a connected component, or b) $\Delta_{\mathcal{G}} = 2$ and $\mathcal{G}$ has an odd cycle as a connected component. For a graph that does not satisfy items (a) or (b), it is possible to obtain a feasible solution to VCP with $\Delta_{\mathcal{G}}$ colors using a linear-time algorithm, as described in [131]. Finally, note that MATLAB toolboxes exist to exactly solve the mVCP and to obtain feasible solutions to the VCP, e.g., the Matgraph toolbox [132].

BIBLIOGRAPHY

[1] Chamberlin, J. A. and Rose, J. T., "Gemini rendezvous program," *Journal of Spacecraft and Rockets*, Vol. 1, No. 1, 1964, pp. 13–18.

[2] Goodman, J. L., "History of space shuttle rendezvous and proximity operations," *Journal of Spacecraft and Rockets*, Vol. 43, No. 5, 2006.

[3] "U.S. Air Force Research Laboratory., Fact Sheet: Automated Navigation and Guidance Experiment for Local Space (ANGELS)," http://www.kirtland.af.mil/Portals/52/documents/AFD-131204-039.pdf?ver=2016-06-28-105617-297 [retrieved 9 January 2018], July 2014.

[4] Kawano, I., Mokuno, M., Kasai, T., and Suzuki, T., "Result of autonomous rendezvous docking experiment of engineering test satellite-VII," *Journal of Spacecraft and Rockets*, Vol. 38, No. 1, 2001, pp. 105–111.

[5] Friend, R. B., "Orbital express program summary and mission overview," *Sensors and Systems for Space Applications II*, Vol. 6958, International Society for Optics and Photonics, 2008, p. 695803.

[6] Martin, E., Dupuis, E., Piedboeuf, J.-C., and Doyon, M., "The TECSAS mission from a Canadian perspective," *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, 2005.

[7] Reintsema, D., Thaeter, J., Rathke, A., Naumann, W., Rank, P., and Sommer, J., "DEOS–the German robotics approach to secure and de-orbit malfunctioned satellites from low earth orbits," *Proceedings of the i-SAIRAS*, 2010, pp. 244–251.

[8] Obermark, J., Creamer, G., Kelm, B. E., Wagner, W., and Henshaw, C. G., "SUMO/FREND: vision system for autonomous satellite grapple," *Sensors and Systems for Space Applications*, Vol. 6555, International Society for Optics and Photonics, 2007, p. 65550Y.

[9] Henshaw, C. G., "The DARPA Phoenix spacecraft servicing program: Overview and plans for risk reduction," *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, 2014.

[10] Brown, O., Eremenko, P., and Bille, M., "Fractionated space architectures: tracing the path to reality," *Proceedings of the 24th Annual Conference on Small Satellites, AIAA/Utah State University, Logan, UT*, 2009.

[11] "Program Aims to Facilitate Robotic Servicing of Geosynchronous Satellites," http://www.darpa.mil/news-events/2016-03-25 [retrieved 19 Sep 2016], March 2016.

[12] Scharf, D. P., Hadaegh, F. Y., and Ploen, S. R., "A survey of spacecraft formation flying guidance and control, part ii: control," *Proceedings of the American Control Conference*, Vol. 4, Institute of Electrical and Electronics Engineers, 2004, pp. 2976–2985.

[13] Kirschner, M., Montenbruck, O., and Bettadpur, S., "Flight dynamics aspects of the GRACE formation flying," *Proceedings of the 2nd International Workshop on Satellite Constellations and Formation Flying*, 2001, pp. 19–20.

[14] Krieger, G., Moreira, A., Fiedler, H., Hajnsek, I., Werner, M., Younis, M., and Zink, M., "TanDEM-X: A satellite formation for high-resolution SAR interferometry," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 45, No. 11, 2007, pp. 3317–3341.

[15] Burch, J. and Angelopoulos, V., *The THEMIS mission*, Springer, 2009.

[16] Gangestad, J., Hardy, B., and Hinkley, D., "Operations, orbit determination, and formation control of the AeroCube-4 CubeSats," *Proceedings of the 27th Annual AIAA/Utah State University Conference on Small Satellites, Logan, UT*, 2013.

[17] Bandyopadhyay, S., Foust, R., Subramanian, G. P., Chung, S.-J., and Hadaegh, F. Y., "Review of formation flying and constellation missions using nanosatellites," *Journal of Spacecraft and Rockets*, Vol. 53, 2016, pp. 567–578.

[18] Latombe, J.-C., *Robot motion planning*, Vol. 124, Springer Science & Business Media, 2012, pp. 1–57.

[19] Patterson, M. A. and Rao, A. V., "GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 41, No. 1, 2014, pp. 1.

[20] Zagaris, C., Baldwin, M., Jewison, C., and Petersen, C., "Survey of spacecraft rendezvous and proximity guidance algorithms for on-board implementation," *Advances in the Astronautical Sciences (AAS/AIAA Spaceflight Mechanics 2015)*, Vol. 155, 2015, pp. 131–150.

[21] Bandyopadhyay, S., Subramanian, G. P., Foust, R., Morgan, D., Chung, S., and Hadaegh, F., "A review of impending small satellite formation flying missions," *Proceedings of the 53rd AIAA Aerospace Sciences Meeting, Kissimmee FL*, 2015.

[22] Alfriend, K., Vadali, S. R., Gurfil, P., How, J., and Breger, L., *Spacecraft formation flying: Dynamics, control and navigation*, Vol. 2, Butterworth-Heinemann, 2009.

[23] Wang, D., Wu, B., and Kee Poh, E., *Satellite Formation Flying: Relative Dynamics, Formation Design, Fuel Optimal Maneuvers and Formation Maintenance*, Vol. 87, Springer, 2017.

[24] Fehse, W., *Automated rendezvous and docking of spacecraft*, Vol. 16, Cambridge university press, 2003.

[25] Jezewski, D., Brazzel Jr, J., Prust, E., Brown, B., Mulder, T., and Wissinger, D., "A survey of rendezvous trajectory planning," *Astrodynamics 1991*, 1992, pp. 1373–1396.

[26] Betts, J. T., "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.

[27] Sullivan, J., Grimberg, S., and DAmico, S., "Comprehensive Survey and Assessment of Spacecraft Relative Motion Dynamics Models," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 8, 2017, pp. 1837–1859.

[28] Scharf, D. P., Hadaegh, F. Y., and Ploen, S. R., "A survey of spacecraft formation flying guidance and control, part 1: guidance," *Proceedings of the American Control Conference*, Vol. 2, Institute of Electrical and Electronics Engineers, 2003, pp. 1733–1739.

[29] Di Mauro, G., Lawn, M., and Bevilacqua, R., "Survey on Guidance Navigation and Control Requirements for Spacecraft Formation-Flying Missions," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 3, 2018, pp. 581–602.

[30] Richards, A., Schouwenaars, T., How, J. P., and Feron, E., "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 755–764.

[31] ILOG Inc., Mountain View, C., "ILOG AMPL CPLEX System Version 10.0 User's Guide," https://ampl.com/BOOKLETS/amplcplex100userguide.pdf [retrieved 11 January 2018], 2006.

[32] Liu, X. and Lu, P., "Solving nonconvex optimal control problems by convex optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 750–765.

[33] Lu, P. and Liu, X., "Autonomous trajectory planning for rendezvous and proximity operations by conic optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 2, 2013, pp. 375–389.

[34] Cetin, B., Bikdash, M., and Hadaegh, F., "Hybrid mixed-logical linear programming algorithm for collision-free optimal path planning," *IET Control Theory & Applications*, Vol. 1, No. 2, 2007, pp. 522–531.

[35] Kim, Y., Mesbahi, M., and Hadaegh, F. Y., "Dual-spacecraft formation flying in deep space: Optimal collision-free reconfigurations," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 2, 2003, pp. 375–379.

[36] LaValle, S. M., "Rapidly-Exploring Random Trees: A New Tool for Path Planning," Tech. Rep. TR 98-11, Computer Science Department, Iowa State University, 1998.

[37] Even, S., *Graph algorithms*, Cambridge University Press, 2011, p. 15.

[38] Karaman, S. and Frazzoli, E., "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, Vol. 30, No. 7, 2011, pp. 846–894.

[39] Aoude, G. S., How, J. P., and Garcia, I. M., "Two-stage path planning approach for solving multiple spacecraft reconfiguration maneuvers," *The Journal of the Astronautical Sciences*, Vol. 56, No. 4, 2008, pp. 515–544.

[40] Frazzoli, E., "Quasi-random algorithms for real-time spacecraft motion planning and coordination," *Acta Astronautica*, Vol. 53, No. 4-10, 2003, pp. 485–495.

[41] Weiss, A., Danielson, C., Berntorp, K., Kolmanovsky, I., and Di Cairano, S., "Motion planning with invariant set trees," *2017 IEEE Conference on Control Technology and Applications (CCTA),*, IEEE, 2017, pp. 1625–1630.

[42] Starek, J. A., Barbee, B. W., and Pavone, M., "A Sampling Based Approach to Spacecraft Autonomous Maneuvering with Safety Specifications," *38th Annual American Astronomical Society Conference on Guidance and Control*, American Astronomical Society, 2015.

[43] Starek, J. A., Schmerling, E., Maher, G. D., Barbee, B. W., and Pavone, M., "Real-time, propellant-optimized spacecraft motion planning under Clohessy-Wiltshire-Hill dynamics," *Aerospace Conference, 2016 IEEE*, IEEE, 2016, pp. 1–16.

[44] Starek, J. A., Schmerling, E., Maher, G. D., Barbee, B. W., and Pavone, M., "Fast, Safe, Propellant-Efficient Spacecraft Motion Planning Under Clohessy–Wiltshire–Hill Dynamics," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2016, pp. 418–438.

[45] Baldini, F., Bandyopadhyay, S., Foust, R., Chung, S.-J., Rahmani, A., de la Croix, J.-P., Bacula, A., Chilan, C. M., and Hadaegh, F. Y., "Fast Motion Planning for Agile Space Systems with Multiple Obstacles," *AIAA/AAS Astrodynamics Specialist Conference*, 2016, p. 5683.

[46] Tedrake, R., Manchester, I. R., Tobenkin, M., and Roberts, J. W., "LQR-trees: Feedback motion planning via sums-of-squares verification," *The International Journal of Robotics Research*, Vol. 29, No. 8, 2010, pp. 1038–1052.

[47] Singh, S., Majumdar, A., Slotine, J.-J., and Pavone, M., "Robust online motion planning via contraction theory and convex optimization," *International Conference on Robotics and Automation submission*, 2017.

[48] Weiss, A., Petersen, C., Baldwin, M., Erwin, R. S., and Kolmanovsky, I., "Safe Positively Invariant Sets for Spacecraft Obstacle Avoidance," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 4, 2014, pp. 720–732.

[49] Camacho, E. F. and Alba, C. B., *Model predictive control*, Springer Science & Business Media, 2013.

[50] Weiss, A., Kolmanovsky, I., Baldwin, M., and Erwin, R. S., "Model predictive control of three dimensional spacecraft relative motion," *American Control Conference (ACC), 2012*, IEEE, 2012, pp. 173–178.

[51] Di Cairano, S., Park, H., and Kolmanovsky, I., "Model predictive control approach for guidance of spacecraft rendezvous and proximity maneuvering," *International Journal of Robust and Nonlinear Control*, Vol. 22, No. 12, 2012, pp. 1398–1427.

[52] Singh, L., Bortolami, S., and Page, L., "Optimal guidance and thruster control in orbital approach and rendezvous for docking using model predictive control," *AIAA Guidance, Navigation, and Control Conference*, 2010, p. 7754.

[53] Manikonda, V., Arambel, P. O., Gopinathan, M., Mehra, R. K., and Hadaegh, F., "A model predictive control-based approach for spacecraft formation keeping and attitude control," *Proceedings of the American Control Conference*, Vol. 6, Institute of Electrical and Electronics Engineers, 1999, pp. 4258–4262.

[54] Breger, L., How, J., and Richards, A., "Model predictive control of spacecraft formations with sensing noise," *Proceedings of the American Control Conference*, Vol. 4, Institute of Electrical and Electronics Engineers, 2005, pp. 2385–2390.

[55] Sauter, L. and Palmer, P., "Analytic model predictive controller for collision-free relative motion reconfiguration," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 4, 2012, pp. 1069–1079.

[56] Morgan, D., Chung, S., and Hadaegh, F. Y., "Model predictive control of swarms of spacecraft using sequential convex programming," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 1725–1740.

[57] Breger, L. and How, J. P., "Gauss's variational equation-based dynamics and control for formation flying spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 2, 2007, pp. 437–448.

[58] Alamir, M., *Stabilization of nonlinear systems using receding-horizon control schemes: a parametrized approach for fast systems*, Vol. 339, Springer, 2006.

[59] Alamir, M., "A low dimensional contractive NMPC scheme for nonlinear systems stabilization: Theoretical framework and numerical investigation on relatively fast systems," *Assessment and Future Directions of Nonlinear Model Predictive Control*, Springer, 2007, pp. 523–535.

[60] Wang, L., "Discrete model predictive controller design using Laguerre functions," *Journal of process control*, Vol. 14, No. 2, 2004, pp. 131–142.

[61] Ong, C.-J. and Wang, Z., "Reducing variables in Model Predictive Control of linear system with disturbances using singular value decomposition," *Systems & Control Letters*, Vol. 71, 2014, pp. 62–68.

[62] Hara, N. and Kojima, A., "Reduced order model predictive control for constrained discrete-time linear systems," *International Journal of Robust and Nonlinear Control*, Vol. 22, No. 2, 2012, pp. 144–169.

[63] Garone, E., Di Cairano, S., and Kolmanovsky, I., "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, Vol. 75, 2017, pp. 306–328.

[64] Petersen, C. and Kolmanovsky, I., "Coupled translational and rotational dynamics for precise constrained rendezvous and docking with periodic reference governors," *Proceedings of 26th AAS/AIAA Space Flight Mechanics Meeting*, Vol. 158, 2016, pp. Paper AAS 16–507.

[65] Kalabić, U., Kolmanovsky, I., and Gilbert, E., "Reference governors for linear systems with nonlinear constraints," *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, IEEE, 2011, pp. 2680–2686.

[66] Petersen, C., Jaunzemis, A., Baldwin, M., Holzinger, M. J., and Kolmanovsky, I. V., "Model Predictive Control and extended command governor for improving robustness of relative motion guidance and control," *Proc. AAS/AIAA Space Flight Mechanics Meeting*, 2014.

[67] Garone, E., Tedesco, F., and Casavola, A., "Sensorless supervision of linear dynamical systems: The feed-forward command governor approach," *Automatica*, Vol. 47, No. 7, 2011, pp. 1294–1303.

[68] Casavola, A., Garone, E., and Tedesco, F., "A distributed multi-agent command governor strategy for the coordination of networked interconnected systems," *IEEE Transactions on Automatic Control*, Vol. 59, No. 8, 2014, pp. 2099–2112.

[69] Casavola, A., Garone, E., and Tedesco, F., "Scalability and performance improvement of distributed sequential command governor strategies via graph colorability theory," *IFAC Proceedings Volumes*, Vol. 47, No. 3, 2014, pp. 9400–9405.

[70] Casavola, A., Garone, E., and Tedesco, F., "A distributed command governor based on graph colorability theory," *International Journal of Robust and Nonlinear Control*, Vol. 28, No. 8, 2018, pp. 3056–3072.

[71] Tedesco, F., Raimondo, D. M., and Casavola, A., "Collision avoidance command governor for multi-vehicle unmanned systems," *International Journal of Robust and Nonlinear Control*, Vol. 24, No. 16, 2014, pp. 2309–2330.

[72] Bacconi, F., Mosca, E., and Casavola, A., "Hybrid constrained formation flying control of micro-satellites," *Institution of Engineering and Technology Control Theory & Applications*, Vol. 1, No. 2, 2007, pp. 513–521.

[73] Kolmanovsky, I. V. and Sun, J., "Parameter governors for discrete-time nonlinear systems with pointwise-in-time state and control constraints," *Automatica*, Vol. 42, No. 5, 2006, pp. 841–848.

[74] Frey, G. R., Petersen, C. D., Leve, F. A., Kolmanovsky, I. V., and Girard, A. R., "Constrained Spacecraft Relative Motion Planning Exploiting Periodic Natural Motion Trajectories and Invariance," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 12, 2017, pp. 3100–3115.

[75] Frey, G. R., Petersen, C. D., Leve, F. A., Kolmanovsky, I. V., and Girard, A. R., "Incorporating periodic and non-periodic natural motion trajectories into constrained invariance-based spacecraft relative motion planning," *Control Technology and Applications (CCTA), 2017 IEEE Conference on*, IEEE, 2017, pp. 1811–1816.

[76] Frey, G., Petersen, C., Leve, F., Girard, A., and Kolmanovsky, I., "Invariance-based spacecraft relative motion planning incorporating bounded disturbances and minimum thrust constraints," *American Control Conference (ACC)*, IEEE, 2018, pp. 658–663.

[77] Frey, G., Petersen, C., Leve, F., Girard, A., and Kolmanovsky, I., "Safe relative motion trajectory planning for satellite inspection," *Proceedings of 27th AAS/AIAA Space Flight Mechanics Meeting*, 2017, pp. Paper AAS 17–411.

[78] Frey, G. R., Petersen, C. D., Leve, F. A., Garone, E., Kolmanovsky, I. V., and Girard, A. R., "Time Shift Governor for Coordinated Control of Two Spacecraft Formations," *IFAC-PapersOnLine*, Vol. 49, No. 18, 2016, pp. 296–301.

[79] Frey, G. R., Petersen, C. D., Leve, F. A., Garone, E., Kolmanovsky, I. V., and Girard, A. R., "Parameter Governors for Coordinated Control of n-Spacecraft Formations," *Journal of Guidance, Control, and Dynamics*, 2017, pp. 1–6.

[80] Hempel, P. and Tschauner, J., "Rendezvous with a target in an elliptical orbit," *Acta Astronautica*, Vol. 11, 1965, pp. 104–109.

[81] Vallado, D. A., *Fundamentals of astrodynamics and applications*, Microcosm Press, 2013, p. 846.

[82] Clohessy, W. H. and Wiltshire, R. S., "Terminal guidance system for satellite rendezvous," *Journal of the Aerospace Sciences*, Vol. 27, No. 9, 1960, pp. 653–658.

[83] Shibata, M. and Ichikawa, A., "Orbital rendezvous and flyaround based on null controllability with vanishing energy," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 4, 2007, pp. 934–945.

[84] Kim, S. C., *Mission design and trajectory analysis for inspection of a host spacecraft by a microsatellite*, Master's thesis, Massachesetts Institute of Technology, 2006.

[85] Shepperd, S. W., "Constant covariance in local vertical coordinates for near-circular orbits," *Journal of Guidance, Control, and Dynamics*, Vol. 14, 1991, pp. 1318–1322.

[86] Jiang, Z.-P. and Wang, Y., "Input-to-state stability for discrete-time nonlinear systems," *Automatica*, Vol. 37, No. 6, 2001, pp. 857–869.

[87] Kabamba, P. T. and Girard, A. R., *Fundamentals of aerospace navigation and guidance*, Cambridge University Press, 2014.

[88] Bittanti, S., Laub, A. J., and Willems, J. C., *The Riccati Equation*, Springer Science & Business Media, 2012, p. 44.

[89] Delchamps, D. F., *State space and input-output linear systems*, Springer Science & Business Media, 2012.

[90] Kolmanovsky, I. and Gilbert, E. G., "Theory and computation of disturbance invariant sets for discrete-time linear systems," *Mathematical problems in engineering*, Vol. 4, No. 4, 1998, pp. 317–367.

[91] Kolmanovsky, I. and Gilbert, E. G., "Multimode regulators for systems with state & control constraints and disturbance inputs," *Control using logic-based switching*, Springer, 1997, pp. 104–117.

[92] Raković, S., Fontes, F. A., and Kolmanovsky, I. V., "Reachability and Invariance for Linear Sampled–data Systems," *IFAC-PapersOnLine*, Vol. 50, No. 1, 2017, pp. 3057–3062.

[93] Blanchini, F., "Survey paper: Set invariance in control," *Automatica*, Vol. 35, No. 11, 1999, pp. 1747–1767.

[94] Blanchini, F. and Miani, S., *Set-theoretic methods in control*, Springer, 2008.

[95] Açıkmeşe, B., Carson, J. M., and Blackmore, L., "Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem," *IEEE Transactions on Control Systems Technology*, Vol. 21, No. 6, 2013, pp. 2104–2113.

[96] Cannon, M., Kouvaritakis, B., Rakovic, S. V., and Cheng, Q., "Stochastic tubes in model predictive control with probabilistic constraints," *IEEE Transactions on Automatic Control*, Vol. 56, No. 1, 2011, pp. 194–200.

[97] Hmam, H., "Quadratic Optimization with One Quadratic Equality Constraint," http://www.dtic.mil/dtic/tr/fulltext/u2/a528339.pdf [retrieved 20 Feb 2017], 2010.

[98] Kurzhanskiy, A. A. and Varaiya, P., "Ellipsoidal techniques for reachability analysis of discrete-time linear systems," *IEEE Transactions on Automatic Control*, Vol. 52, No. 1, 2007, pp. 26–38.

[99] Borrelli, F., Bemporad, A., and Morari, M., *Predictive control for linear and hybrid systems*, Cambridge University Press, 2017.

[100] Maler, O., "Computing reachable sets: An introduction," *Technical Report, French National Center of Scientific Research*, 2008.

[101] Girard, A. and Le Guernic, C., "Efficient reachability analysis for linear systems using support functions," *IFAC Proceedings Volumes*, Vol. 41, No. 2, 2008, pp. 8966–8971.

[102] Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V., *Linear matrix inequalities in system and control theory*, Vol. 15, Siam, 1994.

[103] Gleich, D., "MatlabBGL," http://www.mathworks.com/matlabcentral/fileexchange/10922-matlabbgl [retrieved 10 Oct 2016], October 2008.

[104] Kurzhanskiy, A. A. and Varaiya, P., "Ellipsoidal Toolbox," Tech. Rep. UCB/EECS-2006-46, EECS Department, University of California, Berkeley, May 2006.

[105] Cochran, J., Siranosian, A., Ghods, N., and Krstic, M., "3-d source seeking for underactuated vehicles without position measurement," *IEEE Transactions on Robotics*, Vol. 25, No. 1, 2009, pp. 117–129.

[106] Ogren, P., Fiorelli, E., and Leonard, N. E., "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic control*, Vol. 49, No. 8, 2004, pp. 1292–1302.

[107] Bachmayer, R. and Leonard, N. E., "Vehicle networks for gradient descent in a sampled environment," *Proceedings of the 41st IEEE Conference on Decision and Control*, Vol. 1, IEEE, 2002, pp. 112–117.

[108] Kurokawa, H., "Survey of theory and steering laws of single-gimbal control moment gyros," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1331–1340.

[109] Leve, F. A. and Fitz-Coy, N. G., "Hybrid steering logic for single-gimbal control moment gyroscopes," *Journal of guidance, control, and dynamics*, Vol. 33, No. 4, 2010, pp. 1202–1212.

[110] Leve, F. and Jah, M., "Spacecraft actuator alignment determination through null-motion excitation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 50, No. 3, 2014, pp. 2336–2342.

[111] Shannon, C. E., "A mathematical theory of communication," *Bell system technical journal*, Vol. 27, No. 3, 1948, pp. 379–423.

[112] Klesh, A. T., Kabamba, P. T., and Girard, A. R., "Path planning for cooperative time-optimal information collection," *American Control Conference (ACC)*, IEEE, 2008, pp. 1991–1996.

[113] Klesh, A. T., Kabamba, P. T., and Girard, A. R., "Optimal path planning for uncertain exploration," *American Control Conference (ACC)*, IEEE, 2009, pp. 2421–2426.

[114] Chang, Y.-h., Hyun, B., and Girard, A. R., "Path planning for information collection tasks using bond-energy algorithm," *American Control Conference (ACC), 2012*, IEEE, 2012, pp. 703–708.

[115] Liberzon, D., *Switching in systems and control*, Springer Science & Business Media, 2012, p. 14.

[116] Boyd, S. and Vandenberghe, L., *Convex optimization*, Cambridge university press, 2004, p. 244.

[117] Jensen, T. R. and Toft, B., *Graph coloring problems*, Vol. 39, John Wiley & Sons, 2011.

[118] Malaguti, E. and Toth, P., "A survey on vertex coloring problems," *International transactions in operational research*, Vol. 17, No. 1, 2010, pp. 1–34.

[119] Wang, C. and Ong, C.-J., "Distributed model predictive control of dynamically decoupled systems with coupled cost," *Automatica*, Vol. 46, No. 12, 2010, pp. 2053–2058.

[120] Marruedo, D. L., Alamo, T., and Camacho, E., "Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties," *Proceedings of the 41st IEEE Conference on Decision and Control*, Vol. 4, IEEE, 2002, pp. 4619–4624.

[121] Morgan, D., Chung, S.-J., Blackmore, L., Acikmese, B., Bayard, D., and Hadaegh, F. Y., "Swarm-keeping strategies for spacecraft under J2 and atmospheric drag perturbations," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 5, 2012, pp. 1492–1506.

[122] Tatsch, A. R., *Artificial Potential-function Guidance for Autonomous In-space Operations*, University of Florida, 2006.

[123] Calafiore, G. C. and El Ghaoui, L., *Optimization models*, Cambridge university press, 2014, p. 34.

[124] Jiang, R. and Li, D., "Simultaneous Diagonalization of Matrices and Its Applications in Quadratically Constrained Quadratic Programming," *SIAM Journal on Optimization*, Vol. 26, No. 3, 2016, pp. 1649–1668.

[125] Murray, W., "Newton-Type Methods," *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[126] Wright, S. and Nocedal, J., *Numerical optimization*, Springer, New York, 2006, p. 33.

[127] Kolmanovsky, I. V., "On Property Y of Yildiray Yildiz," Unpublished Notes.

[128] Malaguti, E., Monaci, M., and Toth, P., "An exact approach for the vertex coloring problem," *Discrete Optimization*, Vol. 8, No. 2, 2011, pp. 174–190.

[129] Galinier, P., Hamiez, J.-P., Hao, J.-K., and Porumbel, D., "Recent advances in graph vertex coloring," *Handbook of optimization*, Springer, 2013, pp. 505–528.

[130] Brooks, R. L., "On colouring the nodes of a network," *Classic papers in combinatorics*, Springer, 2009, pp. 118–121.

[131] Skulrattanakulchai, S., "Δ-List vertex coloring in linear time," *Information processing letters*, Vol. 98, No. 3, 2006, pp. 101–106.

[132] Scheinerman, E., "Matgraph," https://www.mathworks.com/matlabcentral/fileexchange/19218-matgraph [retrieved 20 March 2018], 2008.