Stochastic and Distributed Geometric Guidance for Spacecraft Formation Planning

by

Hermann K. Sipowa

B.A., University of Maryland, 2017

M.S., University of Colorado Boulder, 2019

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Aerospace Engineering Sciences

2022

Committee Members:

Jay W. McMahon, Chair

Prof. Hanspeter Schaub

Prof. Nisar Ahmed

Prof. Scott Palo

Dr. David Gaylor

Sipowa, Hermann K. (Ph.D., Aerospace Engineering Sciences)

Stochastic and Distributed Geometric Guidance for Spacecraft Formation Planning

Thesis directed by Prof. Jay W. McMahon

This research introduces a novel Guidance, Navigation, and Control (GNC) algorithm for spacecraft formation flying. The theoretical background of the proposed relative path planner is based upon nonlinear geometric control theory. The algorithm incorporates holonomic/nonholonomic constraints and a filter's information to generate collision-free trajectories despite the system's stochasticity. It is shown that the local fuel-optimal solutions are geodesics of the constrained solution manifold. The optimization problem is shown to be equivalent to a partial differential equation defined on a Riemannian manifold. Using Lie brackets, the relative problem is shown to be controllable. The total delta-v required to travel along the computed geodesic is small enough to do the orbital transfer using natural forces such as atmospheric drag and solar radiation pressure. Consequently, 6DOF fuel-free trajectories are generated for a flat plate deputy spacecraft around a cannonball chief. The computation is distributed among the different spacecraft in the formation to ensure that the relative GNC algorithm can be used with large formations. The algorithm's computation requirement is distributed among the different agents in the formation. Each spacecraft identifies potential obstacles (i.e., neighboring spacecraft) and uses this local information to independently compute a fuel-efficient path to avoid its neighbors. A relative navigation filter is also derived to keep the uncertainty associated with each deputy bounded. The filters operate on a distributed architecture, and each agent estimates its state using relative measurements with its neighbor. The covariance envelope associated with each filter is used as a keep-out ellipsoid, and a chance constraint formulation is used to transform the probabilistic collision constraint on the filters' covariance into a deterministic condition on the belief states of the system (i.e., mean of each filters' distribution). A covariance avoidance sequence is initiated every time two covariance envelopes intersect. The resulting GNC algorithm has two components (geometric path planner and relative navigation filter) that run in parallel. A Lyapunov feedback controller is demonstrated to track the computed optimal path in real-time. The algorithm is used for creating fuel-optimal and collision-free trajectories, and is demonstrated on a six-spacecraft formation around Earth.

Dedication

It takes a village to raise a child, and I am merely the product of those who have mentored, supported, and guided me on my life journeys. I want to dedicate this dissertation to all the people who helped me get to this point. There are not enough words to express my gratitude. Thank you, for I could not have done it without you.

Acknowledgements

I want to thank my advisor, Dr. Jay McMahon, for seeing potential in me (even when I doubted myself) from the very beginning and always supporting my wildest research ideas. You have been instrumental in my growth as a scholar and helped me gain confidence throughout my exciting and, at times, uncertain Ph.D. journey. Thanks to Dr. Christine Hartzell, Dr. Rhea Nedd, and Ms. Güll Branco for your guidance during my time at the University of Maryland. I also want to thank Dr. Ariel Paul and Dr. Lynne Harden, who have supported me every step of the way and helped me navigate life as a graduate student. Special thanks to my parents (Jean and Cecile) for all the sacrifices they have made for me to succeed and to my brothers for constantly pushing me to be a better version of myself. Shout out to the ORCCA lab for being like a family to me.

Contents

Chapter

Appendix

Tables

Table

Figures

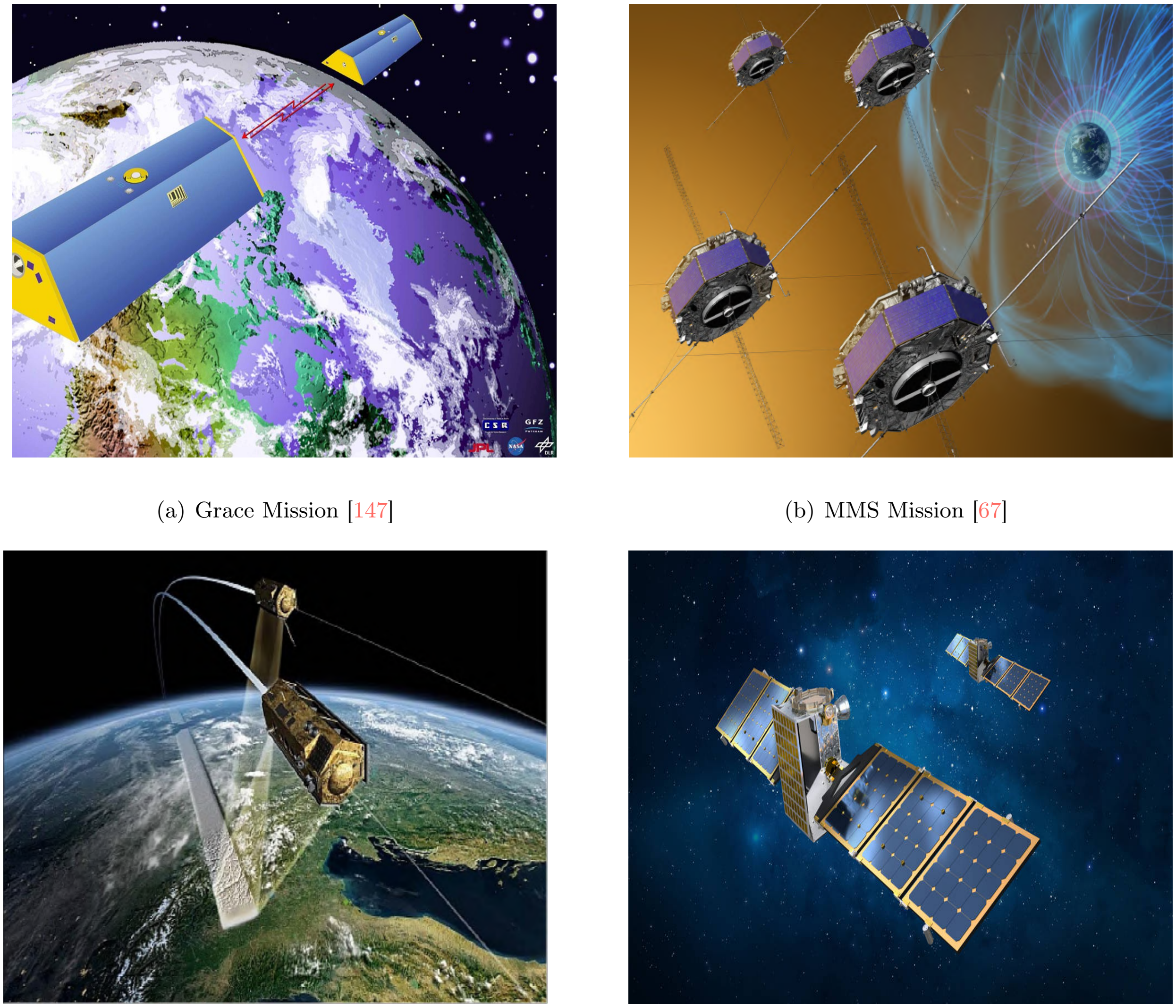
Figure

Chapter 1

Background and Motivation

1.1 Background and Motivation

A Distributed Space System (DSS) is a spacecraft architecture that reconfigures large and complex monolithic satellites into a cluster of smaller satellites. Intuitively, this new paradigm distributes the mission's payload and experiments across a network of smaller satellites, allowing for better science returns (more collected data), and cheaper mission costs (more affordable to manufacture) [25, 140]. The origins of DSS can be traced back to the Gemini program, and it was a key technology instrumental in landing the first man on the Moon [50]. Over the years, the Gravity Recovery and Climate Experiment (GRACE) [192], the TerraSAR-X add-on for Digital Elevation Measurement (TanDEM-X) [9], and the Magnetospheric Multiscale Mission (MMS) [214] are a few examples of flown missions that leveraged this approach to perform Earth gravity field recovery, synthetic aperture radar interferometry, and magnetospheric observation, respectively. Encouraged by the success of past missions, many space agencies, government research and development (R&D) centers, and universities have invested a considerable amount of time and effort in designing more than twenty DSS missions in the past two decades [173]. There are currently working on at least ten more such missions that will launch in the next decade, including the Janus mission launching later this year [58].



(a) Grace Mission [147]

(b) MMS Mission [67]

(c) TanDEM-X Mission [61]

(d) Janus Mission [148]

Figure 1.1: Example of Past and Future Distributed Space Systems (DSS) Missions

With these added benefits comes a new set of challenges. A small spacecraft has low fuel storage and limited actuation capability making it difficult to follow arbitrary relative trajectories; this induces the need to develop Guidance, Navigation, and Control (GNC) algorithms that leverage natural dynamics to achieve fuel-efficient relative trajectories [36, 145]. Spacecraft Formation Flying (SFF) is a framework where the dynamics of each agent in a distributed architecture are defined with respect to another agent; therefore, it is the appropriate framework for designing these kinds of GNC algorithms.

Over the years, numerous research efforts have developed fuel-efficient relative orbits such as passive apertures orbits. Both linear and nonlinear approaches exist to generate the initial conditions for such passive apertures. In the realm of linearized dynamics, many linear optimal control methods have been developed based on Hill-Clohessy-Wiltshire (HCW) and the Tschauner–Hempel (T-H) equations of motion [84, 11, 32, 33]. However, the performance of these methods decreases with large relative separations and as external perturbations act on the system [170, 6]. On the other hand, nonlinear methods involve using a numerical grid search to find a path optimizing a chosen cost function [198, 111, 48, 60, 34]. However, the performance of these nonlinear methods heavily relies on the choice of local coordinate representations. The algorithm developed in this study reframes the fuel-optimal two-point boundary value (2PBVP) problem into a curve minimization on a Riemannian manifold. The optimal path is shown to be a geodesic of the solution manifold, and it is obtained by solving a partial differential equation derived in Chapter 2. The large convergence radius of the proposed method is robust to bad initial guesses which is a big improvement over other methods.

Another challenge associated with SFF is the generation of collision-free trajectories, mainly when the spacecraft operate near one another. In recent years, Low Earth Orbit (LEO) overcrowding has become a primary concern for the scientific community [126, 112]. In addition, the increase of commercial endeavors (i.e., Starlink, OneWeb, and Project Kuiper) indicates that the spacecraft-overpopulation problem is likely to increase in the next decades [131], thus the need for collision avoidance algorithms. Many approaches are currently proposed in the literature for collision-free path planning. Conjunction assessment is one of the most recognizable approaches to deal with this issue. It consists of actively tracking hazardous objects, computing the probability of collision, and performing a collision avoidance maneuver should the collision probability exceed a specified threshold [3, 181, 56, 163, 42]. However, this approach is impractical for SFF collision avoidance because the agents' fly-bys are not short-term encounters, and their relative motion is not rectilinear in general. Zhang and Gurfil developed a cooperative control algorithm that converges on collision-free trajectories via consensus on the control objectives for the entire formation to address this shortcoming [219]. Mazal and Gurfil have also looked into controlling the agents' mean orbital elements (semi-major axis, inclination, and eccentricity) and imposing a minimum and maximum separation for extended time intervals [142]. Uriot et al. [197] have discussed the use of machine learning to predict the final risk of collision between two spacecraft at the time of closest approach (TCA). Wang and Schaub [208] have proposed a control strategy which uses cluster internal coulomb forces to ensure that deep-space spacecraft do not collide. D'Amico et al. have developed the eccentricity and inclination (e/i) vector separation method, which guarantees collision avoidance by placing the different agents in non-intersecting relative planes [51, 187, 70]. Other methods have used artificial potential strategies, which create a repulsive force as the agents get closer to each other [182, 86, 180]. Although these methods are effective for some applications, they do not take advantage of natural dynamics to reduce fuel expenditures.

More specifically, natural forces such as Solar Radiation Pressure (SRP) and atmospheric drag can be utilized to enable fuel-free trajectory design around Earth. Solar radiation pressure has been studied for trajectory control in the past. Namely, SRP-based feedback control strategies have been used for stabilizing motion at the libration points [186, 64] and transfers between them [62, 31]. Oguri and McMahon have developed optimal control methods that leverage SRP for visiting and landing on small bodies[155, 154]. Xu et al. have investigated the use of SRP for station-keeping formation flying in the Sun-Earth-Moon Circular Restricted Three-Body Problem [216]. Atmospheric drag has also sparked interest in enabling fuel-free missions. Trajectory design using aeroassist technology has been investigated for Earth orbit [74], Mars robotic[213, 167] and manned [204] missions, and Titan[129]. Ref. [75] analyzes these missions and provides a mass and monetary cost analysis driving home the effectiveness and appeal for aeroassisted orbit transfer.

Distributed autonomous guidance is now more than ever at the forefront of robotics and engineering research to provide innovative solutions to complex problems [192, 9, 214]. These problems affect every aspect of our lives and range from self-driving cars, marine and submarine studies and exploration, and debris space removal, to enumerate a few. The computation is distributed among the different deputy spacecraft to ensure that the relative GNC algorithm can be used with large formations. Therefore, each agent computes a locally optimal path given the information locally available.

The derivations also considered the stochastic nature of real-life dynamical systems because the decisions made by a guidance algorithm are only as good as the knowledge of its surroundings. Consequently, the proposed GNC algorithm incorporates navigation information via chance constraints. The chance constraint is a mathematical framework that transforms (unbounded) probabilistic constraints into deterministic rules. It has been extensively used in robotics [35, 194, 193] and has gained traction in spacecraft GNC [153, 202, 41, 71]. In this work, the chance constraint transforms the covariance (Gaussian distributions) associated with each spacecraft into a keep-out ellipsoid. Therefore, the collision avoidance constraint is enforced by ensuring that the covariance envelopes of any two agents do not intersect. A relative navigation algorithm is also proposed to provide the covariance matrices mentioned above to the stochastic geometric guidance algorithm.

Centralized and distributed navigation algorithms are the two main approaches currently used for SFF navigation. Centralized algorithms relay all the raw data (local state estimates, covariance, and relative measurements) from the different agents to a single node for processing. The processing node is then responsible for estimating every agent's state and communicating the information back across the network [59, 91, 92]. In such a centralized approach, the size of the state space increases with the number of agents in the formation (i.e., for a formation of p spacecraft each with n states, a centralized filter has to estimate $n * p$ states simultaneously). Therefore, the computational requirements associated with large formations make a centralized filter intractable for SFF estimation in general. Centralized algorithms are also vulnerable to a single point of failure, and local changes in the network topology affect the entire system's behavior, causing central filters to diverge.

Distributed algorithms, on the other hand, treat every agent as a separate computation center that processes its local measurements and only shares relevant information with the rest of the network. In addition, the nodes are easily replaceable, and the system's topology can be modified (nodes added or removed) without affecting the algorithm's performance. However, data fusion is a challenge for distributed network architectures. Notably, double-counting the information available in measurements is a common problem that leads to filter inconsistencies [37]. Numerous studies have leveraged the excellent mathematical properties of consensus algorithms (i.e., provable convergence) to address these filter inconsistencies [158, 217, 209, 135]. However, a consensus algorithm can be hard to apply in real-life scenarios due to underlying assumptions regarding the network's topology and communication capability (sufficiently fast communication rate) [196]. Another proposed solution is designing network architectures that intrinsically distribute specific state information to the concerned nodes [105]. Though effective, these methods require knowledge of the cross-correlation between agents. The covariance Intersection (CI) algorithm is a conservative yet efficient way to merge information collected by different sensors when their cross-correlation is unknown [161, 14, 13, 37]. The proposed navigation algorithm uses CI to remove the network-topology assumption and allow for ad hoc connection (random measurement and communication graph) between the different agents in the network.

Ref [45] bears many similarities with the work presented in this dissertation. Namely, The author uses insights from reachability theory to derive a fuel-optimal and impulsive guidance in a perturbed environment. The approach developed in this work uses relative orbital elements, and a closed-form solution is derived for Linear Time-Varying (LTV) systems. As it will be shown, however, the approach proposed in this dissertation is a nonlinear guidance algorithm that leverages natural forces to minimize fuel usage. In addition, the algorithm herein is applicable for collision avoidance with incorporative target and incorporates filter information to compute stochastic-aware trajectories.

1.2 Proposed Research

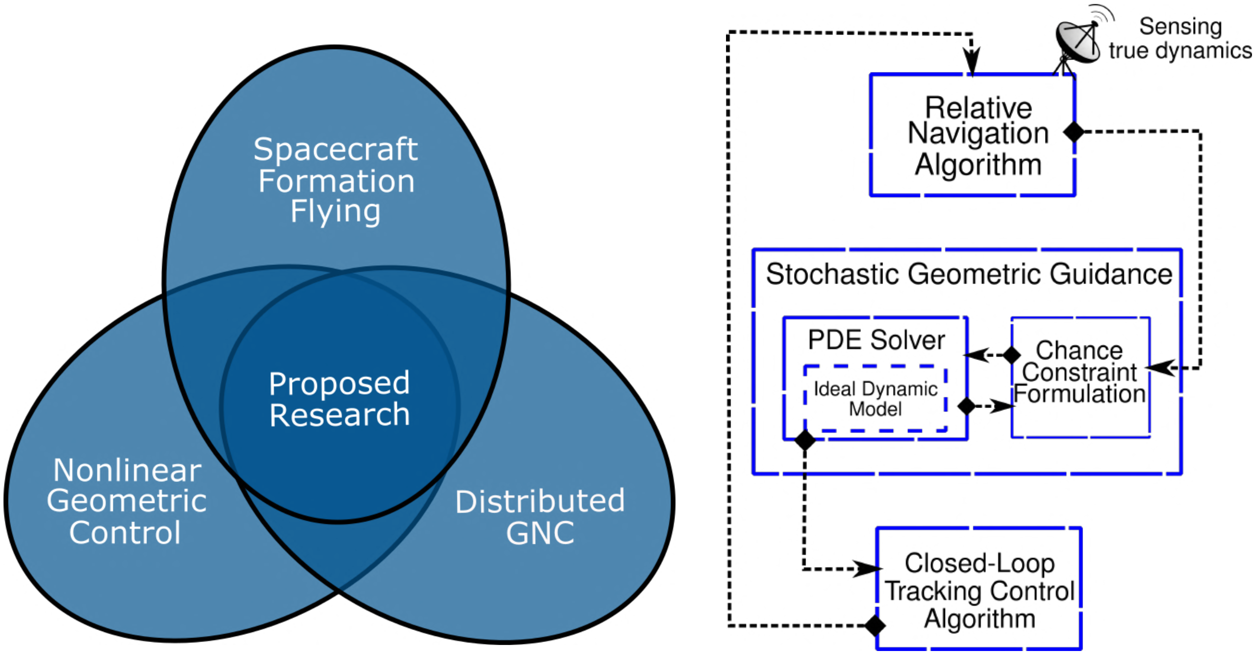
The abovementioned methods have advanced the state of the art in path planning for cluster flights. However, there is still a need for a relative GNC algorithm that leverages natural dynamics to minimize fuel while ensuring collision-free paths between the agents in a formation. Also, a consolidating approach that imposes more general holonomic and nonholonomic constraints is still missing. This work aims to fill that gap. Fig. 1.2(a) shows the different area of research combined in this study. The designed algorithm's flow chart is shown in Fig. 1.2(b). The controllability of the full relative motion is analyzed using Lie bracket theory. The geometric guidance leverages differential SRP and differential drag to generate six degrees of freedom (6-DOF) fuel-free transfer trajectories between a flat plate deputy spacecraft which changes its attitude to fly around a cannonball chief spacecraft. The proposed guidance recursively solves a 2BVP and computes locally fuel-optimal and collision-free trajectories. The algorithm developed in this work also considers uncertainties in the planning phase. A distributed navigation filter estimates the state of each agent, and the guidance algorithm incorporates the covariance information when generating transfer trajectories using a chance constraint formulation. The navigation algorithm allows spacecraft to navigate using solely relative measurements. The three characteristics that make the proposed algorithm unique are as follows:

(1) **Speed:** We show that the fuel-optimal solution of the 2BVP is the stationary curve for a partial differential equation (PDE). The PDE converges in seconds and allows active collision through replanning online.

(2) **Nonlinearity:** The nonlinear interactions between the states of the system and the natural forces (drag, SRP, higher-order gravitational perturbations) are leveraged to reduce fuel expenditure. In addition, the algorithm's large convergence radius provide robustness to bad initial guesses.

(3) **Modularity:** The algorithm is designed independently of the dynamics model. For controllable systems, the proposed method can be applied in different dynamical environments (underwater, air, etc.) without any simplifications/approximations to the dynamics or alteration to the main solver.

The resulting geometric path planning algorithm is fully distributed and capable of handling the stochastic nature of GNC problems. This work primarily focuses on formations between Low


Spacecraft Formation Flying
Proposed Research
Nonlinear Geometric Control
Distributed GNC
Sensing true dynamics
Relative Navigation Algorithm
Stochastic Geometric Guidance
PDE Solver
Ideal Dynamic Model
Chance Constraint Formulation
Closed-Loop Tracking Control Algorithm

(a) Venn Diagram

(b) Flowchart

Figure 1.2: Contribution of the Proposed Research

Earth Orbits (LEO) and Geostationary Earth Orbits (GEO), but the methods developed herein can be applied in any dynamical environment. The thesis statement is a follows:

> *Assuming a Spacecraft Formation Flying system with controllable dynamics, this research develops a distributed Guidance Navigation and Control (GNC) algorithm for fuel-optimal and collision-free trajectory generation in a stochastic-nonlinear environment.*

And the contribution of this work to the literature consists of the four following phases:

(1) First, a deterministic and centralized relative guidance algorithm is derived. This algorithm takes advantage of the full nonlinearity of the problem. Lie brackets theory is used to analyze the controllability of the 6DoF SFF problem where a flat-plate deputy maneuvers around a cannonball chief by modulating its attitude in the presence of SRP and drag. The

geometric guidance demonstrates how to design a six degree of freedom (6DoF) trajectory that harvests natural forces to enable fuel-free formation flying missions.

(2) The computational burden is separated across the different agents in the formation. The distributed algorithm relies on periodic information sharing among the agents to converge on a locally optimal solution. The generated paths obey both local constraints (specific to an agent – i.e., actuation requirement) and global constraints (constraints about the entire formation – i.e., collision avoidance).

(3) A novel distributed navigation filter is introduced for relative navigation of spacecraft in a formation. The algorithm merges the Unscented Information Filter (UIF) and the Covariance Intersection algorithm (CI) to estimate the spacecraft's states in a distributed manner. Each spacecraft tracks its states, and the CI algorithm allows the distributed architecture to converge to a consistent estimate while ignoring the cross-correlation between the different spacecraft estimates.

(4) A stochastic distributed geometric path planner is developed to blend the distributed geometric guidance and estimation theory. Here, the geometric guidance algorithm runs parallel with the relative filtering algorithm developed in Chapter 5. The navigation algorithm keeps the uncertainty of the system bounded while the geometric planner avoids collisions with the stochastic deputy via a chance constraint algorithm. The goal is to design a path planning algorithm capable of accounting for the uncertainty provided by the navigation algorithm.

1.3 Publications

The work done in this dissertation generated the following publicartions:

1.3.1 Journal Articles

(1) Hermann Kaptui Sipowa and Jay McMahon. Distributed Estimator for Spacecraft Cooperative Localization. **Journal of Guidance, Control, and Dynamics**, pages 1–6, feb

2022

(2) Hermann Kaptui Sipowa and Jay McMahon. Fuel-Optimal Geometric Guidance for Spacecraft Formation Flying. **Journal of Guidance, Control, and Dynamics**, 2021. Manuscript submitted for publication

(3) Hermann Kaptui Sipowa and Jay McMahon. Fuel-optimal geometric path planning algorithm for spacecraft formation flying. **Journal of Guidance, Control, and Dynamics**, 2022. Manuscript submitted for publication

(4) Hermann Kaptui Sipowa and Jay McMahon. Distributed and stochastic guidance for multi-agent systems. **Journal of Guidance, Control, and Dynamics**, 2022. Manuscript in preparation

1.3.2 Conference Papers

(1) Hermann Kaptui Sipowa and Jay McMahon. First order approximation of the effects of solar radiation pressure on relative motion using a linearized representation of relative orbital elements. In **AAS/AIAA Astrodynamics Specialist Conference**, volume 167, Aug. 2019. Paper AAS 18-474

(2) Hermann Kaptui Sipowa, Jay W. McMahon, and Taralicin Deka. Distributed unscented-information kalman filter (uikf) for cooperative localization in spacecraft formation flying. In **AIAA Scitech 2020 Forum**. Paper AIAA-2020-1917

(3) Hermann Kaptui Sipowa and Jay McMahon. Analysis of srp-disturbed relative motion using geometric nonlinear control theory. In **AAS/AIAA Astrodynamics Specialist Conference**, volume 175, Aug. 2020. Paper AAS 20-474

(4) Hermann Kaptui Sipowa and Jay McMahon. 6dof nonlinear guidance for spacecraft formation flying. In **AAS/AIAA Astrodynamics Specialist Conference**, Aug. 2021. Paper AAS 21-768

(5) Hermann Kaptui Sipowa and Jay McMahon. Fuel-efficient distributed path planning for spacecraft formation flying. In **IEEE Aerospace Conference**, Mar. 2022. Paper 2393

(6) Hermann Kaptui Sipowa and Jay McMahon. Stochastic distributed geometric motion planner for spacecraft formation flying. In **International Workshop on Satellite Constellations & Formation Flying**, Jun. 2022. (Submitted)

Chapter 2

Riemannian Geometry and Geometric Path Planner

The problem investigated in this chapter is a two-point boundary value problem (2PBVP) with fixed end conditions, and the guidance algorithm derived herein computes the locally fuel-optimal path of the solution manifold. The theoretical background of the proposed relative path planner is based upon nonlinear geometric control theory. The optimization problem is equivalent to a partial differential equation defined on a Riemannian manifold. Compared to the methods mentioned earlier, the proposed geometric guidance is agnostic to the choice of coordinates. The large convergence radius of the algorithm (shown numerically) removes the need to have a thorough knowledge of the dynamical system while generating the initial guess. The optimal trajectory is computed by updating an initial guess (homotopies iteration) to minimize the difference between the spacecraft acceleration and natural dynamics (i.e., reducing fuel usage). The derivations of the planning algorithm take advantage of the full nonlinearity of the problem so that generated paths obey the natural dynamics as closely as possible. Collision avoidance constraints are also seamlessly integrated into the manifold's construction, and the proposed approach generates collision-free relative trajectories with minor changes to the algorithm.

The remainder of this chapter is organized as follows. Section 2.1 presents the dynamical system under consideration. In Section 2.2, the fuel-optimal path reframes the path-planning problem into a PDE; this result is the the main contribution of this chapter to the trajectory design literature. In Section 2.3, the proposed approach is used to generate fuel-optimal and collision-free transfer trajectories for a spacecraft constellation flying around a highly eccentric reference orbit, which poses no issue for the proposed method. Finally, Section 2.4 draws conclusions of the chapter with a few closing remarks.

2.1 Nonlinear Relative Motion Dynamics

Let $\mathscr{E} : \{\hat{\boldsymbol{e}}_r, \hat{\boldsymbol{e}}_\theta, \hat{\boldsymbol{e}}_h\}$ denote the chief's Local-Vertical Local-Horizontal (LVLH) frame (c.f. fig. 2.1), and let the unit vectors $\hat{\boldsymbol{e}}_r, \hat{\boldsymbol{e}}_\theta, \hat{\boldsymbol{e}}_h$ be defined in Eq. (2.1) where $\vec{\boldsymbol{r}}_c, \vec{\boldsymbol{v}}_c$ are the chief's inertial position and velocity vectors, respectively.

$$\hat{\boldsymbol{e}}_r = \frac{\vec{\boldsymbol{r}}_c}{\|\vec{\boldsymbol{r}}_c\|}, \hat{\boldsymbol{e}}_h = \frac{\vec{\boldsymbol{r}}_c \times \vec{\boldsymbol{v}}_c}{\vec{\boldsymbol{r}}_c \times \vec{\boldsymbol{v}}_c}, \hat{\boldsymbol{e}}_\theta = \hat{\boldsymbol{e}}_h \times \hat{\boldsymbol{e}}_r \tag{2.1}$$

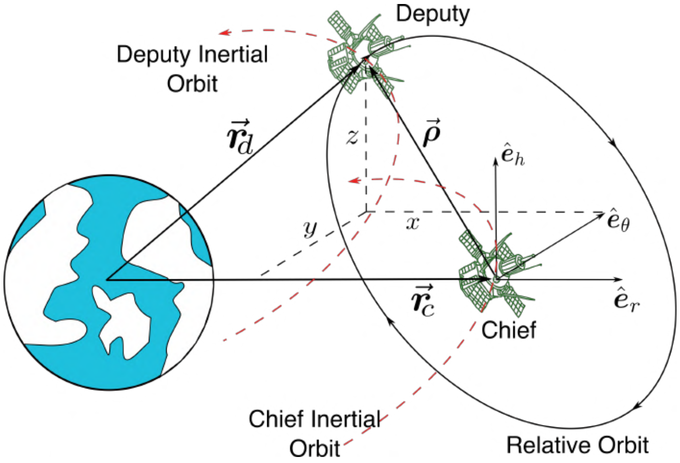


Figure 2.1: Chief's Local-Vertical Local-Horizontal Frame

The full nonlinear model for relative dynamics is written as:

$$\dot{\vec{\boldsymbol{x}}} = \boldsymbol{f}(\vec{\boldsymbol{x}}(t)) + \sum_{i=1}^{3} \boldsymbol{b}_i(\vec{\boldsymbol{x}}(t))u_i(t) \tag{2.2}$$

where $\vec{\boldsymbol{x}} = [\vec{\boldsymbol{\rho}}, \dot{\vec{\boldsymbol{\rho}}}]^\intercal$ is the vector composed of the relative position $\vec{\boldsymbol{\rho}}$ and relative velocity $\dot{\vec{\boldsymbol{\rho}}}$ in the LVLH frame. The multivariate function $\boldsymbol{f}(\cdot)$ is called the drift vector field, and it describes the behavior of the system in the absence of any controls. The multivariate function $\boldsymbol{b}_i(\cdot)$ is a control vector field and maps the i^{th} control input to the deputy's acceleration. In the cases studied in this chapter, $\boldsymbol{b}_i(\cdot)$ represents the thrusters' direction, and it is assumed the control vectors span $\mathbb{R}^3$ (i.e., an acceleration can be produced in all directions). The drift vector field is defined as:

$$\boldsymbol{f}(\vec{\boldsymbol{x}}(t)) = \begin{bmatrix} \dot{\vec{\boldsymbol{\rho}}} \\ \delta\vec{\boldsymbol{a}}^{g} + \vec{\boldsymbol{\Xi}} \end{bmatrix} \tag{2.3}$$

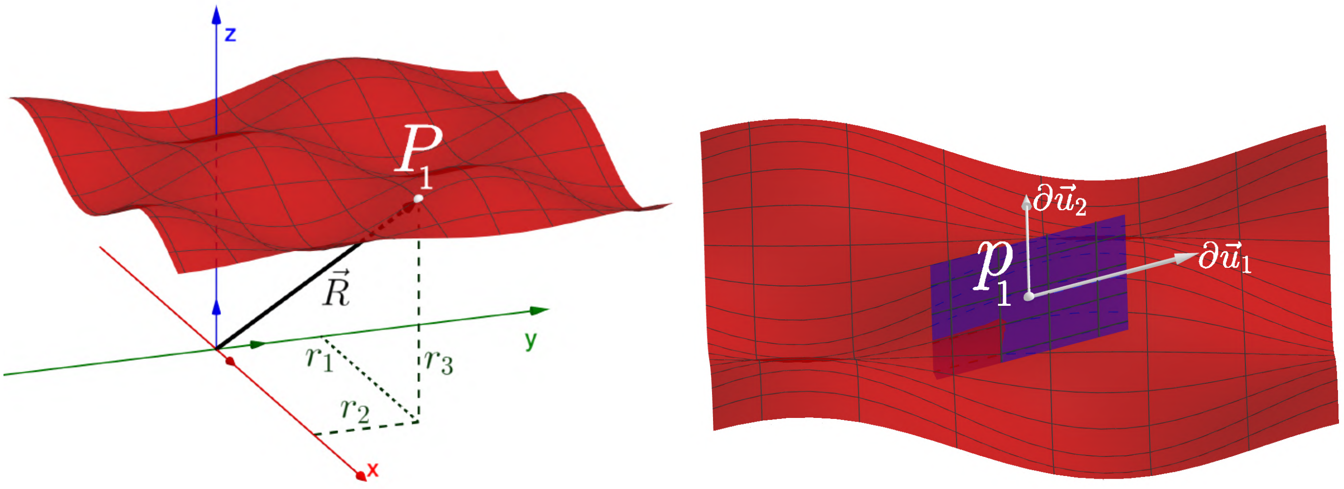
where $\vec{\boldsymbol{\Xi}} = -\vec{\boldsymbol{\Omega}}\times\vec{\boldsymbol{\rho}} - 2\dot{\vec{\boldsymbol{\Omega}}}\times\dot{\vec{\boldsymbol{\rho}}} - \vec{\boldsymbol{\Omega}}\times\vec{\boldsymbol{\Omega}}\times\vec{\boldsymbol{\rho}}$, and $\delta\vec{\boldsymbol{a}}^{g} = \left(\vec{\boldsymbol{a}}_{d}^{g} - \vec{\boldsymbol{a}}_{c}^{g}\right)$ is the relative gravitational acceleration between the chief and deputy spacecraft. In the chief's LVLH frame, the relative acceleration is written as follows [4] (assuming unperturbed 2BP dynamics):

$$\delta\vec{\boldsymbol{a}}^{g} = -\frac{\mu}{\left[(r+\rho_{1})^{2} + \rho_{2}^{2} + \rho_{3}^{2}\right]^{3/2}} \begin{bmatrix} r+\rho_{1} \\ \rho_{2} \\ \rho_{3} \end{bmatrix} + \frac{\mu}{r^{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{2.4}$$

where r is the instantaneous radius of the chief trajectory.

2.2 Optimal Path Planning

The solutions of differential equations are subsets of a curved surface also referred to as the solution manifold and denoted $\mathscr{M}$. These surfaces are typically modeled one of two ways, as shown in Fig. 2.2. The first approach embeds the manifold into a higher dimensional space with an additional constraint (see Fig.2.2(a)). This method allows one to define the manifold fully, but it makes the derivations more complicated because of the added constraints (i.e., more variables than degrees of freedom). The second approach defines a local coordinate around a reference to approximate the manifold's behavior (see Fig.2.2(b)). This method uses the minimum number of variables in the derivations, but breaks down for points away from the reference (due to singularities or non-valid approximations). In attitude dynamics, for instance, these two approaches correspond to using quaternions (no singularities but an extra constraint) and Euler angles (only three parameters but singularities), respectively.



(a) Embedding a 2D manifold into a 3D frame

(b) Local coordinate approximation

Figure 2.2: Classical treatment of curved surfaces

Riemannian geometry provides a systematic approach that consolidates the two methods mentioned above. It does so by defining local basis vectors (represented by $\partial\vec{\boldsymbol{u}}_1$ and $\partial\vec{\boldsymbol{u}}_2$ whose sizes equal that of the manifold) at every point p_i on the solution manifold $\mathscr{M}$ (see Fig. 2.3). These local basis vectors define the tangent space noted $T_{p_i}\mathscr{M}$. If $\vec{\gamma}(t)$ denotes a solution trajectory on $\mathscr{M}$ and $\dot{\vec{\gamma}}(t) = \frac{d}{dt}\vec{\gamma}(t)$ the time derivative associated with every point on $\vec{\gamma}(t)$, then the notion of Riemannian manifold is defined as follows:

Definition A Riemannian manifold is a real, smooth manifold $\mathscr{M}$ endowed with a positive-definite inner product $\langle\dot{\vec{\gamma}}, \dot{\vec{\gamma}}\rangle = \dot{\vec{\gamma}}^{\mathsf{T}}\mathbf{G}\dot{\vec{\gamma}}$ on the tangent space $T_{\vec{\boldsymbol{x}}}\mathscr{M}$ at each point $\vec{\boldsymbol{x}}$ on the manifold and for an $n \times n$ positive definite matrix $\mathbf{G}(\vec{\gamma})$ called the Riemannian metric and $\forall\dot{\vec{\gamma}} \in T_{\vec{\boldsymbol{x}}}\mathscr{M}$.

The behavior of any dynamical system can be represented by a smooth mapping that associates a real parameter (such as time) to a point on a Riemannian manifold. Such mapping defines the curve $\vec{\gamma}(t)$ evolving on $\mathscr{M}$, as shown in Fig. 2.3.

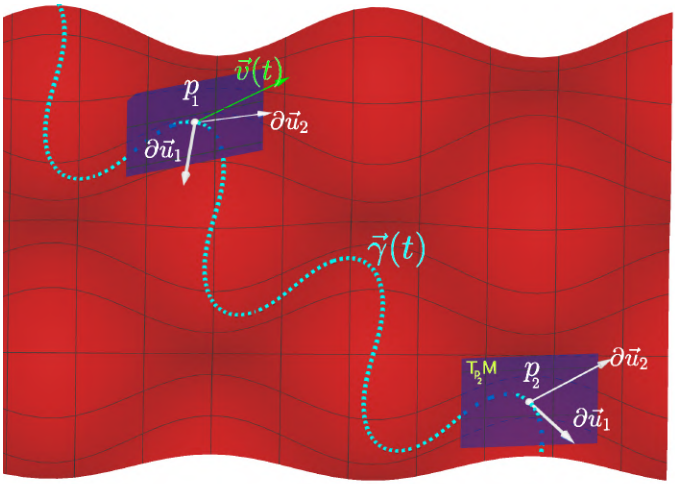


Figure 2.3: Illustration of a Riemannian manifold

2.2.1 Geodesic

Given a Riemannian manifold $(\mathcal{M}, \mathbf{G})$, a curve is defined by the following mapping:

$$\gamma : [0, T] \to M \mid \dot{\gamma} = f(\gamma(t), u(t)) \tag{2.5}$$

A geodesic is a minimum length curve connecting two points on a Riemannian manifold. On the manifold, the concept of distance is tied to the Riemannian metric, and the length of a curve connecting $\gamma(0) = \boldsymbol{x}_i$ to $\gamma(T) = \boldsymbol{x}_f$ in time T is written as:

$$\mathscr{L}(\gamma(t)) = \int_0^T \sqrt{\dot{\vec{\gamma}}^\intercal \mathbf{G} \dot{\vec{\gamma}}}dt = \int_0^1 \mathrm{d}s \tag{2.6}$$

where $\mathrm{d}s^2 = \sum_i \sum_j \mathbf{G}_{ij} dx_i dx_j$ is the infinitesimal length element. Conceptually, a geodesic is the extension of a straight line to a curved space.

Let's define a connection as a mapping between tangent spaces at different points on the manifold. The Levi-Civita connection $\nabla : T_{p_1}\mathscr{M} \to T_{p_2}\mathscr{M}, T_{p_1}, T_{p_2} \in \mathscr{M}$ is a special mapping that preserves the direction of the velocity vector along a curve (i.e., it is the mapping that keeps the velocity vector along a curve "straight"). The geodesic is described by the following parabolic PDE [29]:

$$\frac{\partial \boldsymbol{\gamma}(t, s)}{\partial s} = \nabla_{\dot{\gamma}(t,s)} \dot{\gamma}(t, s) = \mathbf{0} \tag{2.7}$$

From a control perspective, geodesics are zero-fuel trajectories connecting two points on a manifold. The motion planning problem boils down to finding the connecting curve which requires the minimum amount of controls to cancel any tangential accelerations. The PDE defined in Eq. (2.7) is typically solved numerically from given boundary conditions and an initial condition. The initial condition is updated by moving the curve in the direction of the tangential acceleration, thereby reducing its curvature. The resulting path is the minimum length curve between any two points on the Riemannian manifold. Ref. [89] provides a rigorous mathematical explanation why a geodesic is a locally minimizing curve. The term $\dot{\boldsymbol{\gamma}}(t,s) = \frac{\partial \boldsymbol{\gamma}(t,s)}{\partial t}$ is the derivative with respect to time. $\boldsymbol{\gamma}(0,s) = \boldsymbol{x}_i$ is the initial state, and $\boldsymbol{\gamma}(T,s) = \boldsymbol{x}_f$ represents the goal state. $\nabla_{\dot{\boldsymbol{\gamma}}(t,s)}\dot{\boldsymbol{\gamma}}(t,s)$ is the tangential acceleration felt by a particle traveling along the curve $\boldsymbol{\gamma}(t,s)$. $\nabla_X Y$ is the covariant derivative of the vector Y along the direction of the vector X, and it quantifies how much Y changes along the direction of X. The k^{th} component of $\nabla_X Y$ is given by

$$(\nabla_X Y)_k = \sum_{j=1}^{n}\left[\frac{\partial Y_k}{\partial x_j} X_j + \sum_{i=1}^{k}\left[\Gamma_{ij}^{k} X_i Y_j\right]\right] \tag{2.8}$$

where $X = \frac{\partial \boldsymbol{x}}{\partial t}$ is the time derivative of the local coordinate $\boldsymbol{x}$. X_i is the i^{th} component of the velocity vector X. Γ^k is an $n \times n$ matrix associated with k^{th} component of the local coordinate and is called the Christoffel symbols of $\boldsymbol{x}_k$. Christoffel symbols characterize the directional change of the basis vectors at each point of a curved manifold. Γ_{ij}^k is the (i,j) component Γ^k and is given by

$$\Gamma^i_{jk} = \frac{1}{2}\sum_{l=1}^{n} g^{il}\left(\frac{\partial g_{lj}}{\partial x_k} + \frac{\partial g_{lk}}{\partial x_j} - \frac{\partial g_{jk}}{\partial x_l}\right) \tag{2.9}$$

where g_{ij} is the (i,j) component of the Riemannian metric $\mathbf{G}$, and g^{ij} is the (i,j) component of $\mathbf{G}^{-1}$.

2.2.2 Geodesic and Fuel-Optimal Guidance

The initial and final conditions of the problem are defined as $\vec{\boldsymbol{\gamma}}(0,s) = \vec{\boldsymbol{x}}_i$ and $\vec{\boldsymbol{\gamma}}(T,s) = \vec{\boldsymbol{x}}_f$, respectively. The parameter t is the physical time it takes to travel from $\vec{\boldsymbol{x}}_i$ to $\vec{\boldsymbol{x}}_f$ along the curve $\vec{\boldsymbol{\gamma}}(t)$, and the parameter s is a real positive number that characterizes the homotopies of the initial guess. For a fixed homotopy number s, Eq. (2.7) is expressed in terms of local coordinates as follows

$$\frac{\mathrm{d}^2 x^i}{\mathrm{d}t^2} + \frac{\mathrm{d}x^m}{\mathrm{d}t}\Gamma^i_{mk}\frac{\mathrm{d}x^k}{\mathrm{d}t} = 0 \tag{2.10}$$

where the matrix elements Γ^i_{mk} are called Christoffel symbols defined in Eq. (2.8). The action integral associated with a geodesic is defined as:

$$\mathscr{A}(\vec{\gamma}(t)) = \int_0^T \boldsymbol{L}(\vec{\gamma}(t), \dot{\vec{\gamma}}(t))\mathrm{d}t = 0 \tag{2.11}$$

where $\boldsymbol{L}(\vec{\gamma}(t), \dot{\vec{\gamma}}(t))$ is the Lagrangian function associated with the manifold, and it is defined as:

$$\boldsymbol{L}(\vec{\gamma}(t), \dot{\vec{\gamma}}(t)) = \frac{1}{2}\dot{\vec{\gamma}}(t)^{\intercal}\mathbf{G}\dot{\vec{\gamma}}(t) = \frac{1}{2}g_{ik}\dot{x}^i\dot{x}^k \tag{2.12}$$

where g_{ik} are the components of the metric $\mathbf{G}$. Algorithmically, the solution of the PDE in Eq. (2.7) is obtained numerically from given boundary conditions and an initial condition. The initial condition is gradually updated (via a homotopy transformation) in a direction where the tangential acceleration along the curve $\nabla_{\dot{\vec{\gamma}}(t,s)}\dot{\vec{\gamma}}(t,s)$ approaches zero, thereby zeroing out the work done on the system along that solution curve. The resulting path is the minimum length curve between any two points on the Riemannian manifold. As shown in the Appendix A, the equation of a geodesic in Eq. (2.7) is equivalent to the Euler-Lagrange equations, the necessary condition for a curve $\vec{\gamma}^*$ to be a local optimizer of Eq. (2.11).

Applied to the ODE defined in Eq. (2.2), one can compute the minimum fuel trajectory of the 2PBVP by defining the following Lagrangian:

$$\boldsymbol{L}(\vec{\boldsymbol{x}}, \dot{\boldsymbol{x}}) = \frac{1}{2}\left(\dot{\vec{\boldsymbol{x}}} - \boldsymbol{f}(\vec{\boldsymbol{x}})\right)^{\intercal}\mathbf{G}\left(\dot{\vec{\boldsymbol{x}}} - \boldsymbol{f}(\vec{\boldsymbol{x}})\right). \tag{2.13}$$

The solution of Eq. (2.7) is the steady-state of the following PDE [128]:

$$\frac{\partial\vec{\gamma}}{\partial s} = \mathbf{G}^{-1}(\vec{\gamma})\left(\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial\dot{\vec{\gamma}}}(\vec{\gamma}, \dot{\vec{\gamma}}) - \frac{\partial L}{\partial\vec{\gamma}}(\vec{\gamma}, \dot{\vec{\gamma}})\right) \tag{2.14}$$

where $\mathbf{G}$ is the Riemannian metric associated with the problem and, as mentioned above, allows for the compution of dot products (and thereby to define distances) on the manifold. This metric is defined as

$$\mathbf{G}(x) = \boldsymbol{\psi}(\vec{\boldsymbol{x}})\bar{\mathbf{F}}^{-\intercal}(\vec{\boldsymbol{x}})\mathbf{D}\bar{\mathbf{F}}^{-1}(\vec{\boldsymbol{x}}) \tag{2.15}$$

where $\mathbf{D} = \text{diag}([\underbrace{\lambda, \cdots, \lambda}_{\text{n-m}}, \underbrace{1, \cdots, 1}_{\text{m}}])$ is the penalty matrix. The scalar $\lambda > 0$ is a tuning parameter chosen large enough to penalize motion in prohibited directions. $\boldsymbol{\psi}(\vec{\boldsymbol{x}})$ is a non-zero function that is used to impose additional constraints, such as characterizing obstacles to be avoided during the path or defining the maximum achievable control. An example demonstrating the use of $\boldsymbol{\psi}(\vec{\boldsymbol{x}})$ will be shown in the next section. $\bar{\mathbf{F}}(\vec{\boldsymbol{x}})$ is an $n \times n$ matrix that is differentiable in x and invertible at every point on the manifold. It is defined as

$$\bar{\mathbf{F}}(x) = [\mathbf{F}_c(\vec{\boldsymbol{x}}) \mid \mathbf{F}(\vec{\boldsymbol{x}})] \in \mathbb{R}^{n \times n} \tag{2.16}$$

where $\mathbf{F}(\vec{\boldsymbol{x}})$ is an $m \times n$ matrix and encodes all directions where the system can instantaneously move by changing the control inputs. It is constructed as $\mathbf{F}(\vec{\boldsymbol{x}}) = [\boldsymbol{b}_1, \cdots, \boldsymbol{b}_m]$, where $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_m$ are the m control vector fields in (2.2). The set of all holonomic and nonholonomic constraints on the studied system can be encoded in a bounded x-dependent $(n - m) \times n$ matrix $\mathbf{F}_c(\vec{\boldsymbol{x}})$. The matrix $\mathbf{F}_c(\vec{\boldsymbol{x}})$ encodes all the directions the system cannot move in. For instance, in the case of a wheel rolling without slipping, the column of $\mathbf{F}_c(\vec{\boldsymbol{x}})$ will encode the nonholonomic constraint between the linear speed and the wheel rotational rate. To ensure that all the prohibited directions are penalized equally, it is recommended to orthogonalize the column of $\mathbf{F}_c(\vec{\boldsymbol{x}})$ using the Gram-Schmidt process.

Given the boundary conditions $\vec{\boldsymbol{x}}(0, s) = \vec{\boldsymbol{x}}_i, \vec{\boldsymbol{x}}(T, s) = \vec{\boldsymbol{x}}_f$, and initial condition $\vec{\boldsymbol{x}}(t, 0)$, $t \in [0, T]$, numerically solve the PDE defined in Eq. (2.14). The solution is noted as $\vec{\boldsymbol{x}}^*(t, s)$, and it is the path obeying to Hamilton's principle of critical action (i.e., the path of least resistance and therefore requiring the least amount of control). The action integral defined in Eq. (2.11) decreases for every feasible curve and gradually approaches the steady state solution $\vec{\boldsymbol{x}}^*(t) = \vec{\boldsymbol{x}}(t, s_{max})$ where it is equal to zero [128]. That is $\frac{\mathrm{d}}{\mathrm{d}s}\mathscr{A}(\vec{\boldsymbol{x}}(\cdot, s)) \leq 0$ for all admissible $\vec{\boldsymbol{x}}(t, s)$, and the equality holds at $\boldsymbol{x}^*(t, s)$. The solution is computed by performing homotopy of the initial condition, and the solution is updated in the direction of decreasing $\frac{\mathrm{d}}{\mathrm{d}s}\mathscr{A}(\vec{\boldsymbol{x}}(\cdot, s))$, and

$$\lim_{s \to \infty} \vec{\boldsymbol{x}}(t, s) = \vec{\boldsymbol{x}}^*(t, s) \tag{2.17}$$

Algorithm 1 defined below gives a consice overview of the algorithm discussed above.

Algorithm 1: Geometric Path Planning

```
input  : $\vec{\boldsymbol{x}}(t_0), \vec{\boldsymbol{x}}(t_f)$
// Set boundary conditions
$\vec{\gamma}(0,0) = \vec{\boldsymbol{x}}(t_0), \vec{\gamma}(T,0) = \vec{\boldsymbol{x}}(t_f)$
// Pick an initial guess
$\vec{\gamma}(t,0)$ is any function connecting the boundary conditions
// Define the Riemannian metric
$\mathbf{G}(\vec{\gamma}) = (1+\boldsymbol{\psi})\left(\bar{\mathbf{F}}^{-\intercal}(\vec{\gamma})\mathbf{D}\bar{\mathbf{F}}^{-1}(\vec{\gamma})\right)$
// Define the lagrangian
$\mathbf{L}(\vec{\gamma},\dot{\vec{\gamma}}) = \frac{1}{2}\left(\dot{\vec{\gamma}} - \boldsymbol{f}(\vec{\gamma})\right)^\intercal \mathbf{G}(\vec{\gamma})\left(\dot{\vec{\gamma}} - \boldsymbol{f}(\vec{\gamma})\right)$
// Solve the geodesic PDE until convergence
$\vec{\gamma}(t,\infty) \triangleq \mathbf{G}_i^{-1}(\vec{\gamma})\left(\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\vec{\boldsymbol{x}}}}(\vec{\gamma},\dot{\vec{\gamma}}) - \frac{\partial L}{\partial \vec{\boldsymbol{x}}}(\vec{\gamma},\dot{\vec{\gamma}})\right) = 0$
// Return the optimal path
$\vec{\boldsymbol{x}}^*(t) = \vec{\gamma}(t,\infty)$;
output: $\vec{\boldsymbol{x}}^*(t)$
```

2.2.3 Constrained Optimal Path Planning: Collision Avoidance

Reframing the path planning problem into an optimization problem on a Riemannian manifold allows for imposing different constraints by modifying the definition of the metric **G**. Collision avoidance is one such constraint that is of paramount importance to the success of formation flying missions. In this work, such the collision avoidance is imposed by using the barrier function defined in Eq. (2.18). However, other options can be used, as it will be shown in Chapters 4 and 6. The reader is referred to [179, 180] for more details on barrier functions and how they are constructed. A minimum approach distance can be imposed by augmenting the Riemannian metric (2.15) with the following functional:

$$\psi_j\left(\vec{\boldsymbol{\rho}}_j(t), \vec{\boldsymbol{\rho}}_i(t)\right) = \sum_{i=1}^{n} \frac{k_i}{(\delta\rho_{ij} - \alpha)^p}, \tag{2.18}$$

$$\delta\rho_{ij} = \|\vec{\boldsymbol{\rho}}_j(t) - \vec{\boldsymbol{\rho}}_i(t)\| \tag{2.19}$$

where k, α, and p are user defined parameters. The presence of the parameter k_i in the numerator is the relative importance of the i^{th} constraints. This is crucial to finding satisfactory paths from arbitrary initial guesses as it allows the guess to be chosen irrespective of whether the agents meet the minimum separation constraints. The parameter p controls how fast the constraints rise toward infinity, and α represents the distance of closest approach. The vectors $\vec{\boldsymbol{\rho}}_i$ and $\vec{\boldsymbol{\rho}}_j$ are the relative positions of any (i, j) deputies. From Eq. (2.15), the value of the Riemannian metric (and consequently the action integral) increases to an infinitely large value as the two spacecraft approach the relative keep-out distance.

2.2.4 Tracking Control

Chapter 3 introduces a rigorous check for controllability. However, the system is assumed to be controllable in this section. The computed optimal trajectory can be tracked using many control approaches. Let $\vec{\boldsymbol{u}}^*(t)$ be the tracking control that allows the deputy to follow the fuel-optimal path $\vec{\boldsymbol{x}}^*(t)$ computed from Algorithm 1. Below, two potential methods are discussed to compute the tracking control $\vec{\boldsymbol{u}}^*(t)$: an open-loop least-squares control and a closed-loop Lyapunov controller. The open-loop control law $\vec{\boldsymbol{u}}^*_{LS}(t)$ is computed by projecting the difference in acceleration between the converged trajectory and the natural dynamics onto the columns space of $\mathbf{F}(\vec{\boldsymbol{x}}) = [\boldsymbol{b}_1, \cdots, \boldsymbol{b}_m]$, as defined below

$$\vec{\boldsymbol{u}}^*_{LS}(t) = [0 \;\; \boldsymbol{I}_{m\times m}] \left[\bar{\mathbf{F}}\left(\vec{\boldsymbol{x}}(t, s_{max})\right)\right]^{-1} \left(\dot{\vec{\boldsymbol{x}}}^*(t, s_{max}) - \boldsymbol{f}_d(\vec{\boldsymbol{x}}(t, s_{max}))\right) \tag{2.20}$$

The control law $\vec{\boldsymbol{u}}^*_{LS}(t)$ is straight forward to compute. However, the trajectory it generates may not faithfully follow the optimal path $\vec{\boldsymbol{x}}^*(t, s_{\max})$ because it is a least-squares approximation. Due to its open-loop nature, such a control is also not recommended if there are perturbations in the system.

An alternative closed-loop tracking control $\vec{\boldsymbol{u}}^*_{CL}(t)$ can be derived using Lyapunov control theory. Here the aim is to actively minimize the errors between the spacecraft states (relative position $\delta\vec{\boldsymbol{r}}$ and velocity $\delta\vec{\boldsymbol{v}}$) and the desired trajectory (the computed optimal trajectory computed from Eq. (2.14)) using a closed-loop control law. To this end, the following Lyapunov function is considered:

$$V(\delta\vec{\boldsymbol{r}}, \delta\vec{\boldsymbol{v}}) = \frac{1}{2}\delta\vec{\boldsymbol{v}}^{\intercal}\delta\vec{\boldsymbol{v}} + \frac{1}{2}\delta\vec{\boldsymbol{r}}^{\intercal}[\boldsymbol{K}_1]\delta\vec{\boldsymbol{r}} \tag{2.21}$$

where $\delta\vec{\boldsymbol{r}}(t) = \vec{\boldsymbol{r}}_i(t) - \vec{\boldsymbol{r}}^*(t, s_{\max})$ is the difference between the instantaneous position of the deputy $\vec{\boldsymbol{r}}_i(t)$ and the desired position $\vec{\boldsymbol{r}}^*(t, s_{\max})$. Likewise, $\delta\vec{\boldsymbol{v}}(t) = \vec{\boldsymbol{v}}_i(t) - \vec{\boldsymbol{v}}^*(t, s_{\max})$ is deviation between the instantaneous velocity $\vec{\boldsymbol{v}}_i(t)$and the desired velocity $\vec{\boldsymbol{v}}^*(t, s_{\max})$. By setting the time derivative of the Lyapunov function to $\frac{\mathrm{d}}{\mathrm{d}t}V(\delta\vec{\boldsymbol{r}}, \delta\vec{\boldsymbol{v}}) = -\frac{1}{2}\delta\vec{\boldsymbol{v}}^{\intercal}[\boldsymbol{P}_1]\delta\vec{\boldsymbol{v}}$, the closed-loop control can be expressed as (see Appendix B):

$$\vec{\boldsymbol{u}}^*_{CL}(t) = -\delta\vec{\boldsymbol{a}} - [\boldsymbol{K}_1]\delta\vec{\boldsymbol{r}} - [\boldsymbol{P}_1]\delta\vec{\boldsymbol{v}} \tag{2.22}$$

where $\delta\vec{\boldsymbol{a}}$ is the difference between the instantaneous acceleration of the deputy and the acceleration along the optimal path. The control matrices $[\boldsymbol{K}_1]$ and $[\boldsymbol{P}_1]$ are the sensitivity of the controller to deviation in position and velocity, respectively. The closed-loop control in Eq. (2.22) ensures that the cost function in Eq. (2.21) decreases over time, and the resulting closed-loop system (i.e., $\delta\vec{\boldsymbol{a}} + [\boldsymbol{P}_1]\delta\vec{\boldsymbol{v}} + [\boldsymbol{K}_1]\delta\vec{\boldsymbol{r}} = 0$) is globally asymptotically convergent. For additional information on how to derive this Lyapunov control law, the reader is referred to Ref. [177].

Both control laws were applied in this analysis. However, there were no appreciable differences between their respective performances because dynamical perturbations are not considered in this analysis and there is no error in the initial condition of the trajectory-tracking problem. Consequently, only the results computed from the Lyapunov closed-loop control will be presented in the results section.

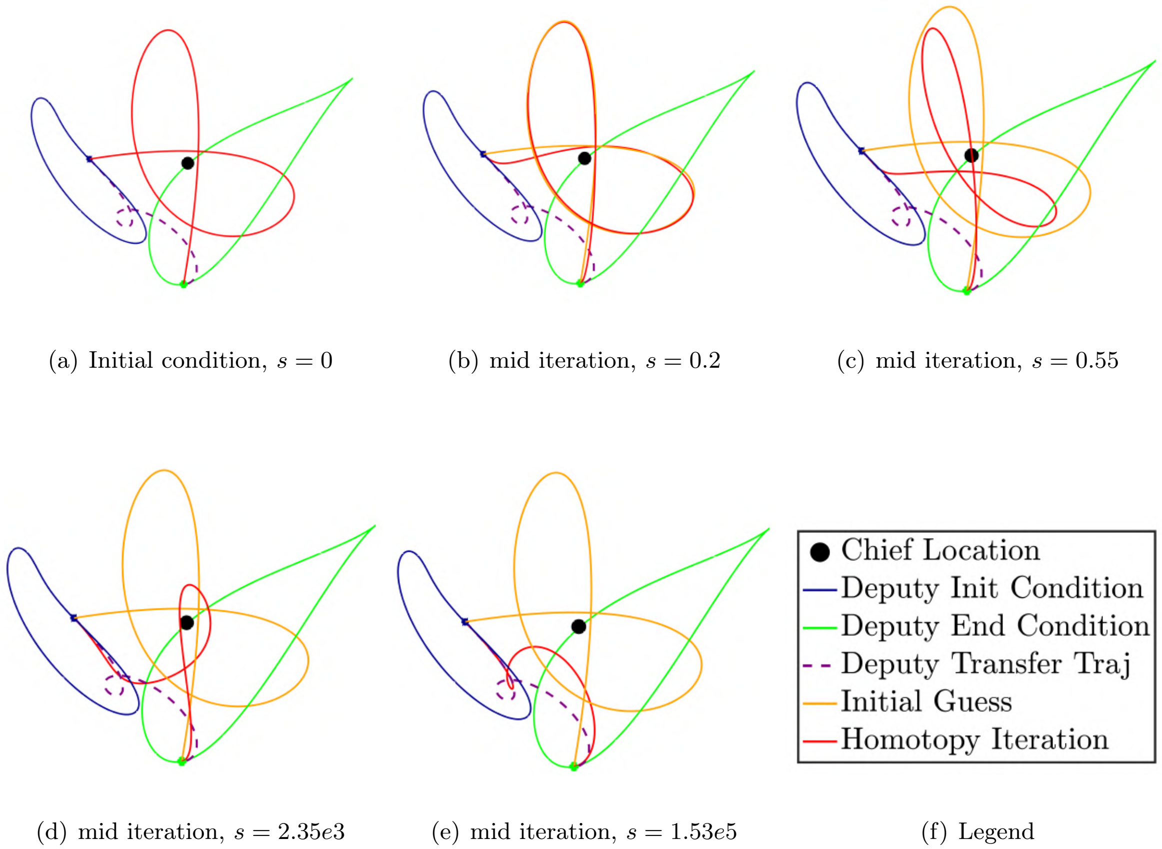
2.3 Simulation, Results and Discussion

2.3.1 Trajectory Design for One Spacecraft in Formation

The following section looks at path planning for the nonlinear system described in Eq. (2.2). The geometric path planner introduced in this study converges on an optimal solution, starting from a wide variety of initial guesses connecting the boundary conditions. To illustrate the large convergence radius of the proposed guidance algorithm, the following initial guess was chosen:

$$\vec{\boldsymbol{x}}(t, 0) = \vec{\boldsymbol{x}}_0 \cos(\frac{\lambda_1 t}{T}) + \vec{\boldsymbol{x}}_T \sin(\frac{\lambda_1 t}{T}) - (\vec{\boldsymbol{x}}_0 - \vec{\boldsymbol{x}}_T)\sin(\frac{\lambda_2 t}{T}) \tag{2.23}$$

where $\lambda_1 = 5\pi/2$, $\lambda_2 = 3\pi$, $T = t_f$ is the final time. $\vec{\boldsymbol{x}}_0$ and $\vec{\boldsymbol{x}}_T$ are the initial and final states, respectively. For the orbital transfer scenario defined in Tables 2.1 and 3.2, Fig. 2.4 shows the initial condition in gold and the converged solution (the fuel-optimal path) in dashed purple. The initial and final relative orbits are blue and green, respectively. Note that, even though the initial condition does not represent a suitable orbital trajectory that follows the dynamics, the geometric planner performs homotopies to reshape the bad initial guess and converges to the optimal path. Here, the homotopy iterations are shown in red. The red and gold curves are superimposed at the beginning of the optimization process, as seen in Fig. 2.4(a). The algorithm proceeds to compute the gradient defined in Eq. (2.14) and updates the red curve in the direction that minimizes the cost function defined in Eq. (2.11) (i.e., minimizes the control required to travel along the solution curve). The red curve becomes more aligned with the natural dynamics as more iterations are performed. Fig. shows the value of the cost function defined in Eq. (2.11) as a function of the iteration parameter s. The algorithm is iterated until the cost function reaches a local minimum.



(a) Initial condition, $s = 0$

(b) mid iteration, $s = 0.2$

(c) mid iteration, $s = 0.55$

(d) mid iteration, $s = 2.35e3$

(e) mid iteration, $s = 1.53e5$

(f) Legend

Figure 2.4: Evolution of the Geometric Planner as a Function of the Homotopy Parameter s

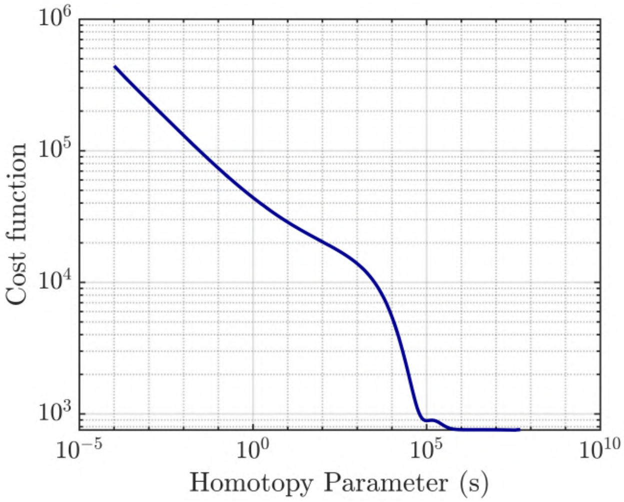


Figure 2.5: Cost funtion vs homotopy iterations

The approach described above was used for an orbit transfer problem where a deputy is tasked to travel between two relative orbits around the chief in a specified time of flight. The initial conditions of the chief spacecraft are given in Table 2.1

Table 2.1: Reference Orbit in Orbital Elements

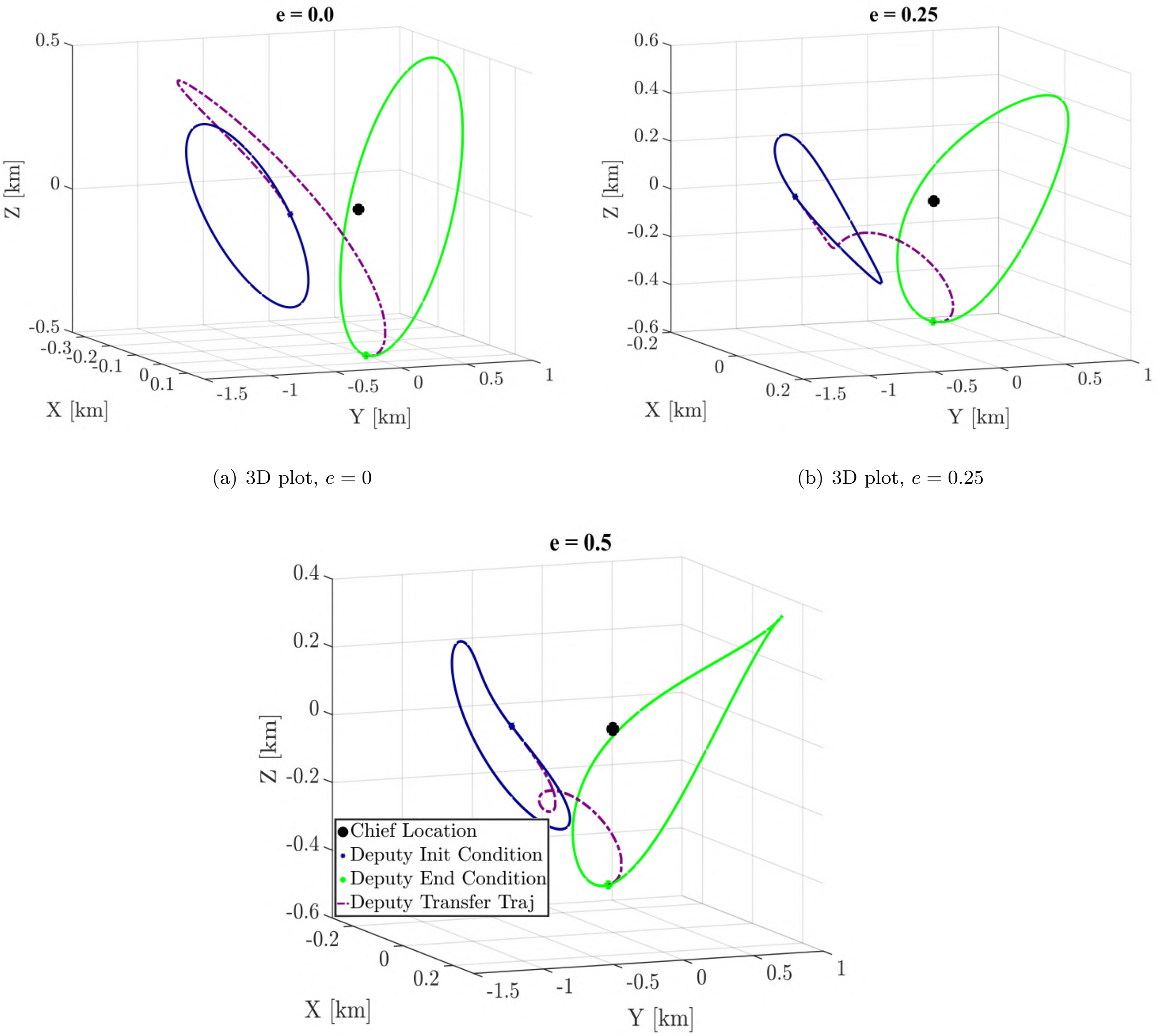
Initial OE	a [km]	e	i [rad]	Ω [rad]	ω [rad]	f [rad]
Reference	1.42e04	$\{0.0, 0.25, 0.5\}$	50	10	10	0.0

The three cases studied were created by varying the chief's eccentricity between $e = \{0.0, 0.25, 0.5\}$. The initial and final conditions of the deputy spacecraft are represented as relative orbital elements and are given in Table 3.2.

Table 2.2: Deputy Orbit in Relative Orbital Elements

ΔOE	δa [km]	δe	δi [rad]	$\delta\Omega$ [rad]	$\delta\omega$ [rad]	δf [rad]
Initial	0.0	$1/(6a)$	$-1/(3a)$	0.0	$-2\pi * 1e - 5$	$\pi * 1e - 6$
Final	0.0	$1/(6a)$	$1/(2a)$	0.0	0.0	0.0

The transfer was prescribed to occur at $t_f = T = 0.75P$, where P is the period of the chief's orbital period. Fig. 2.6 shows the deputy's optimal trajectories for each case. For every scenario, closed-loop tracking control is computed from the converged PDE solution, using Eq. (2.22). Fig. 2.7 shows the components of closed-loop tracking controls expressed in the chief's LVLH frame.

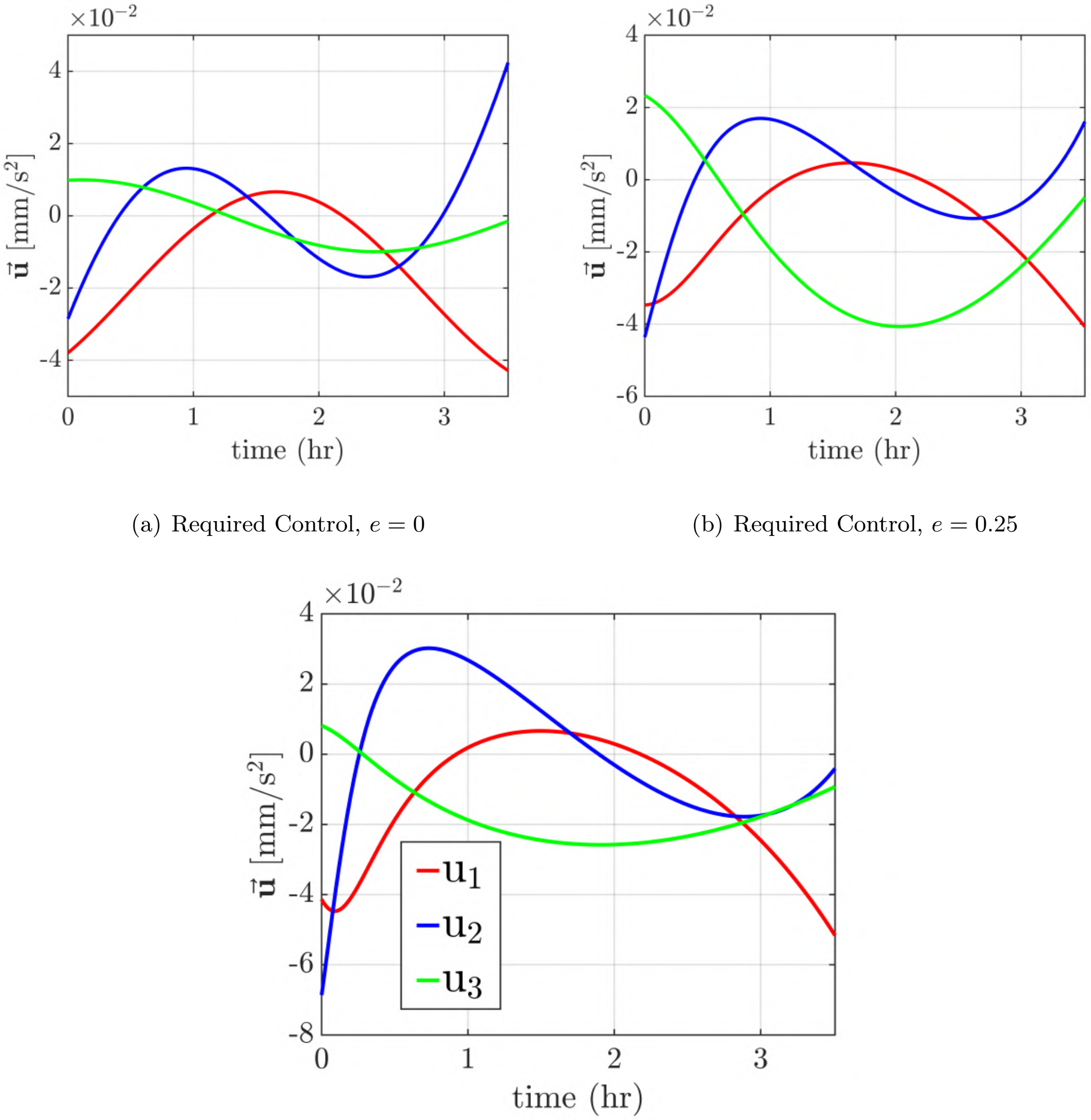


(a) 3D plot, $e = 0$

(b) 3D plot, $e = 0.25$

(c) 3D plot, $e = 0.5$

Figure 2.6: Converged solution for single deputy transfer, for different values of chief's eccentricity



(a) Required Control, $e = 0$

(b) Required Control, $e = 0.25$

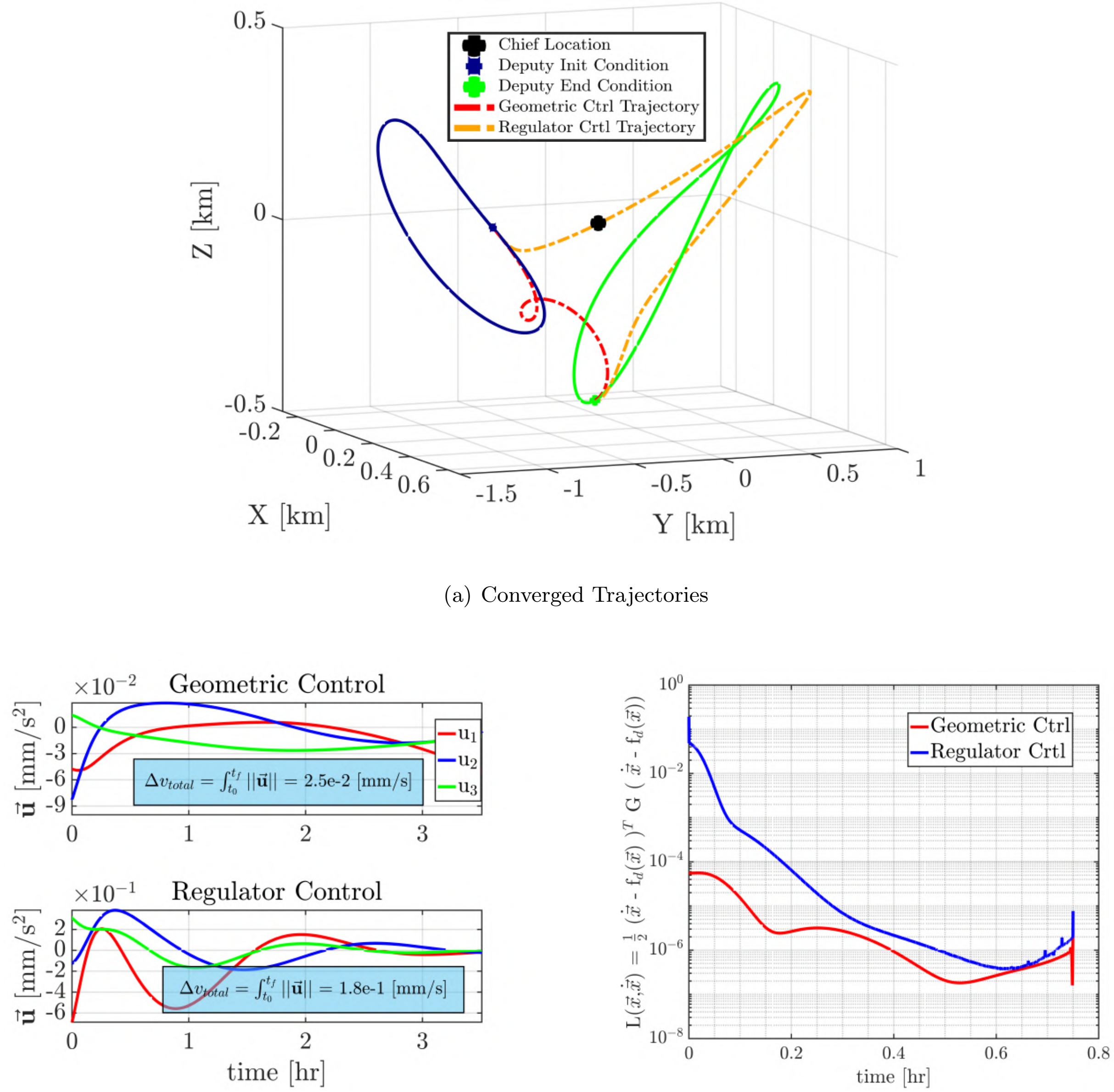
(c) Required Control, $e = 0.5$

Figure 2.7: Required control for single deputy transfer, for different values of chief's eccentricity

The performance of the proposed guidance is compared to another path planning method, called regulator guidance. The control required to follow the regulator's path is computed using the steps laid out in Section 2.2.4, and it is given by the following expression:

$$\boldsymbol{u}_r(t) = \boldsymbol{f}^*(\vec{\boldsymbol{x}}_d) - \boldsymbol{f}(\vec{\boldsymbol{x}}) - [\mathbf{K}_2]\delta\vec{\boldsymbol{r}} - [\mathbf{P}_2]\delta\vec{\boldsymbol{v}} \tag{2.24}$$

where $\vec{\boldsymbol{x}}_d(t) = [\vec{\boldsymbol{r}}_d,\ \vec{\boldsymbol{v}}_d]^\intercal$ is the desired state to be tracked, $\vec{\boldsymbol{x}}(t) = [\vec{\boldsymbol{r}},\ \vec{\boldsymbol{v}}]^\intercal$ is the state of the deputy. $\boldsymbol{f}^*(\vec{\boldsymbol{x}}_d)$ represents gravitational acceleration associated with the desired state while $\boldsymbol{f}(\vec{\boldsymbol{x}})$ is the drift vector field of the deputy. The vectors $\delta\vec{\boldsymbol{r}} = \vec{\boldsymbol{r}} - \vec{\boldsymbol{r}}_d$ and $\delta\vec{\boldsymbol{v}} = \vec{\boldsymbol{v}} - \vec{\boldsymbol{v}}_d$ are the error vectors between the deputy's state and the desired state. The gain matrices $[\mathbf{K}]$ and $[\mathbf{P}]$ penalize deviations in the position and the velocity. Both the control gains in Eqs. (2.22) and (2.24) were set to $[\mathbf{K}_1] = [\mathbf{K}_2] = \text{diag}([1e-6, 1e-6, 1e-6,])$ and $[\mathbf{P}_1] = [\mathbf{P}_2] = \text{diag}([6e-4, 6e-4, 6e-4,])$. Furthermore, these gains were selected so that the controls are active throughout both orbital transfers; this allows for a fair comparison between the two methods. Fig. 2.8(a) shows the trajectories obtained from both methods, and Fig. 2.8(b) shows the control required to travel on each trajectory. Unlike the geometric guidance, the regulator's path does not follow the natural dynamics at all times, especially in the middle of its transfer. The total Δv required to perform the transfer is $\Delta v =$ 2.524e-2 mm/s for the geometric planner as opposed to $\Delta v =$ 1.805e-1 mm/s for the regulator guidance. That is, the proposed method ensures convergence with a control that is one order of magnitude smaller than the regulator guidance. Also, the profile of geometric controls is small throughout the transfer, allowing for this transfer to be performed by a low-thrust engine [76] or using natural forces such as differential solar radiation pressure or differential drag. Figure 2.8(c) compares, as a performance index, the weighted product of the difference between the deputy's acceleration and the acceleration induced by the natural dynamics for both the regulator guidance (in blue) and the geometric guidance (in red). The regulator control continuously fights the natural dynamics at the beginning of the trajectory, resulting in poor performance. As discussed earlier, the regulator control is not concerned with the amount of fuel burned so long as it hits its targeted state. The proposed geometric path planning finds the optimal trajectory for minimum fuel consumption by iterating through different homotopies. The cost function for both methods is the same at the end of the transfer as they both hit the targeted end conditions.



(a) Converged Trajectories

(b) Required Controls

(c) Lagrangian

Figure 2.8: Comparison Between Path Planning Approaches

2.3.2 Constrained Trajectory Design for Multiple Spacecraft

The geometric path planner is applied to a collision avoidance scenario in this section. The following example looks at the transfer of two deputies between different parking orbits. The paths for both spacecraft are computed simultaneously from a centralized algorithm. The collision avoidance is achieved by enforcing the minimum-distance barrier function derived in Eq. (2.18). The boundary conditions are given in Tables 2.3 and 2.4. The parameters of the barrier function were set to $k = 1e - 7$, $p = 1$, and the distance of the closest approach $\alpha = 0.45$km.

Table 2.3: Reference Orbit in Orbital Elements

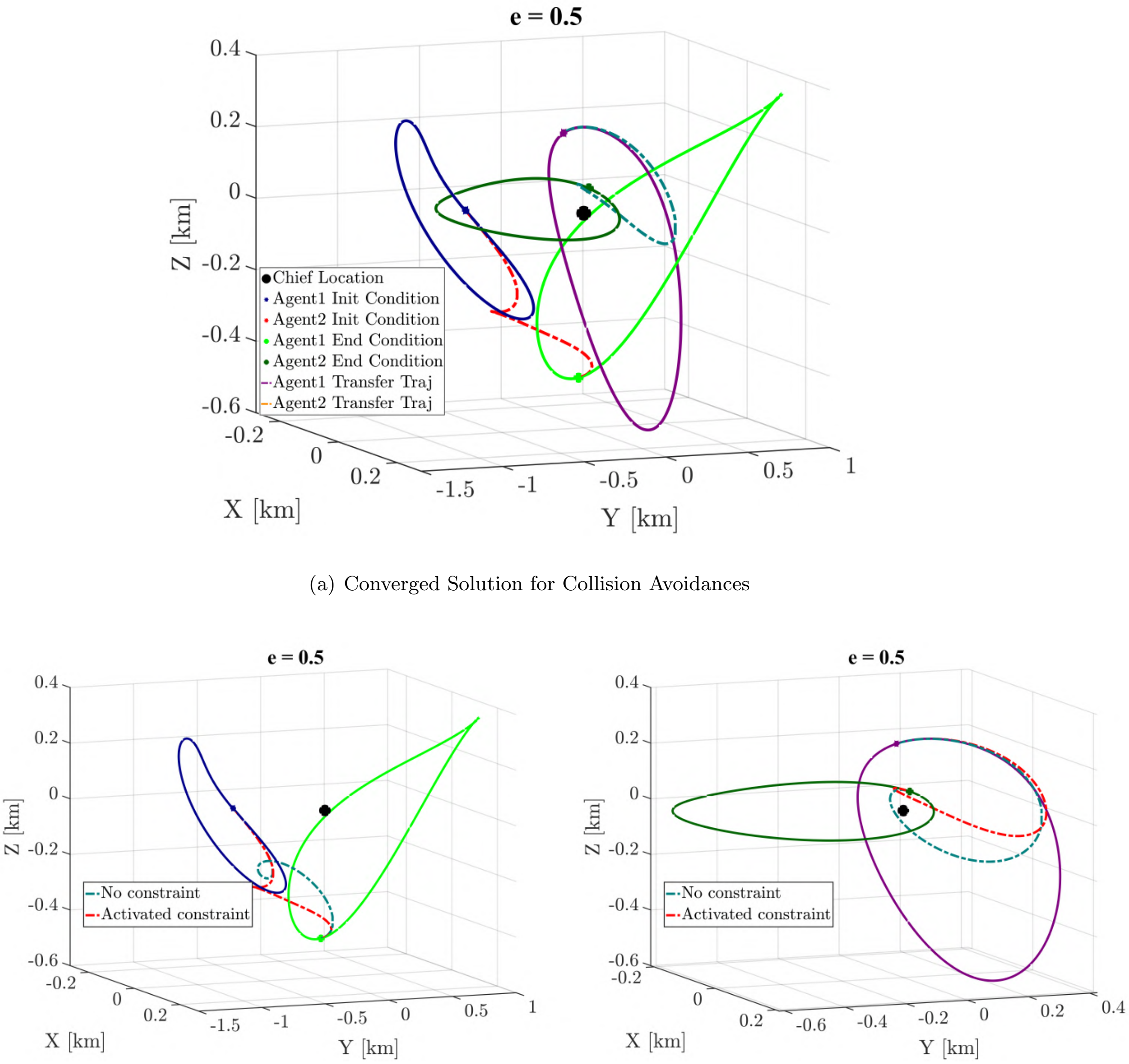
Initial OE	a [km]	e	i [rad]	Ω [rad]	ω [rad]	f [rad]
Reference	1.42e04	0.5	50	10	10	0.0

Table 2.4: Initial Differential Orbital Elements

ΔOE	δa [km]	δe	δi [rad]	$\delta\Omega$ [rad]	$\delta\omega$ [rad]	δf [rad]
Deputy1 (initial)	0.0	$1/(6a)$	$-1/(3a)$	0.0	$-2\pi * 1e - 5$	$\pi * 1e - 6$
Deputy2 (initial)	0.0	$1/(8a)$	$2/(10a)$	$-\pi * 1e - 5$	$\pi * 1e - 5$	$\pi * 1e - 6$
Deputy1 (final)	0.0	$1/(6a)$	$1/(2a)$	0.0	0.0	0.0
Deputy2 (final)	0.0	$-1/(8a)$	$-1/(20a)$	$\pi * 1e - 6$	$-5 * \pi * 1e - 7$	0.0

Figure 2.9(a) shows the collision-free trajectory of the two spacecraft in formation. To better understand the effect of the collision avoidance constraint, the collision-free paths were compared to transfer trajectories that did not consider the minimum allowable distance. Figures 2.9(b) and 2.9(c) illustrate the paths each spacecraft follows when the constraint is considered (dashed red line) and not considered (dashed green line). Imposing the minimum allowable distance caused the two agents to deviate from the trajectories they would have followed otherwise. The effect of the path change also appears in the controls. Figs. 2.9(b) and 2.9(c) show the control efforts for agents 1 and 2, respectively. The first row represents the control to follow the trajectory without

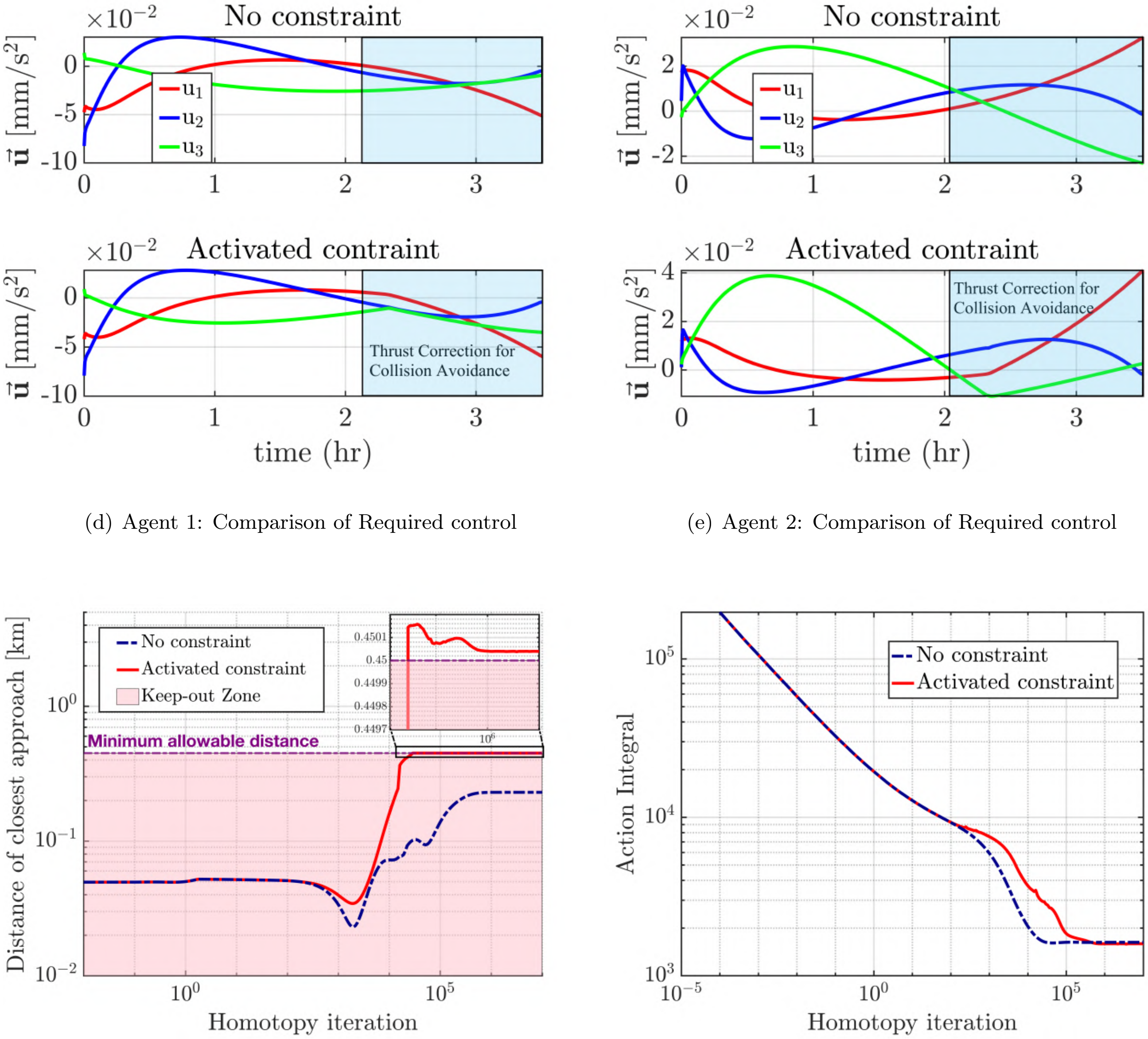
constraint, and the bottom row is the control required to follow the constrained trajectory. When the two spacecraft come close to one other, they perform a thrust correction. Figure 2.9(f) shows the distance of the closest approach at each homotopy for each of the studied scenarios. The constrained path's minimum distance is shown in red, and the unconstrained case is shown in blue. The pink background represents the keep-out zone, and successful collision-free paths should have their closest approach outside the pink area. At the beginning of the iteration, both the collision-free and the unconstrained paths follow the same trend. During that phase, the emphasis is placed on minimizing the control inputs (i.e., minimizing the error between the control trajectory and the natural dynamics), because it is the main factor driving the cost function. Once the cost gets low enough, the minimum separation constraints become the dominant concern, and the red solution obeys the imposed constraints. Fig 2.9(g) shows a comparison between the cost function resulting from generating independent paths versus generating collision-free paths for the two-deputy case. The cost function associated with the collision-free case does not fall as rapidly to its steady-state value because the separation constraint becomes dominant toward the end. Consequently, more homotopy iterations are required before the solution reaches a solution that ensures that the spacecraft stay at a safe distance from each other. The process discussed here can be extended to a constellation with any arbitrary number of spacecraft.



(a) Converged Solution for Collision Avoidances

(b) Agent 1: Comparison of Transfer Trajectory

(c) Agent 2: Comparison of Transfer Trajectory



(d) Agent 1: Comparison of Required control

(e) Agent 2: Comparison of Required control

(f) Distance of closest approach for collision avoidance

(g) Action Integral for collision avoidance

Figure 2.9: Collision Avoidance: Two Deputies

2.4 Conclusion

This work presented a novel nonlinear method for fuel-optimal and constrained trajectory designs. The path planning method was derived using theoretical development from nonlinear geometric control. The novel path planning technique reframes the fuel-optimal trajectory design problem into a curve minimization problem on a Riemannian manifold. The method also allows the incorporation of holonomic and nonholonomic constraints. The optimal path was obtained by solving a partial differential equation. This optimal path is the geodesic of the solution manifold and was shown to be equivalent to the solution of the Euler-Lagrange equations. The proposed methodology was applied to trajectory design for distributed space systems. The case of collision avoidance was investigated to illustrate how constraints can be imposed on a multi-agent path planning problem. The chapter offered a new perspective on computing fuel-efficient and collision-free trajectories for orbital transfer within the spacecraft formation flying problem. The (closed-loop) control required to follow the generated trajectories was shown to be one order of magnitude smaller than a regulator guidance. Because the geometric planner is concerned with reducing the acceleration requirement imposed on the thrusters, the magnitude of the required control is relatively constant and suitable for designing low-thrust trajectories.

Chapter 3

Fuel-Efficient Distributed Path Planning for Spacecraft Formation Flying

The algorithm discussed in the previous chapter computes the minimum-fuel path that connects two points on the solution manifold. However, when multiple agents are in the system, it is imperative to incorporate collision avoidance constraints when planning each spacecraft's path. This chapter extends the geometric path planning algorithm to account for any potential collisions among the computed trajectories while keeping the computational requirements to a minimum (i.e., the algorithm's computation time is agnostic to the number of agents in the formation). Throughout this chapter, it is assumed that every agent is fully controllable and has perfect state knowledge. It will be shown how collision avoidance constraints can be imposed using artificial potential functions. The proposed distributed path planning method adjusts for detected obstacles and re-constructs a new path that avoids those obstacles. Effectively, each agent constructs a local solution manifold and iteratively solves for a locally fuel-optimal solution in the presence of obstacles. The newly computed geodesic retains the fuel-saving characteristics while avoiding new obstacles detected along the way. A (Lyapunov) feedback controller is used to track the optimal path to the goal state. The distributed guidance is an online algorithm that actively avoids other agents infringing on a defined keep-out zone. The remainder of this chapter is structured as follows: In section 3.1, the distributed guidance algorithm is derived. In Section 3.2, simulation results show examples where the distributed guidance algorithm generates collision-free paths for a multi-agent system (i.e., avoidance of moving obstacles). Finally, Section 3.3 offers a summary and provides a few concluding remarks.

3.1 Distributed Motion Planner

3.1.1 Geometric Planning in the Presence of Moving Obstacles

Let N spacecraft be tasked to transfer between distinct parking orbits. On its journey to its goal, agent i is assumed to know its surroundings fully. Every agent has a safety/avoidance sphere that should not be infringed upon by the other agents (see Fig. 3.3). Let R_ψ be the radius of the avoidance sphere. Any agent entering the avoidance sphere is considered to be a potential obstacle. Lastly, define a priority level $\wp$ to decide which agent should be responsible for performing the re-planning maneuver (i.e., if $\wp_j > \wp_i$, then agent i will conduct an avoidance maneuver).

In the presence of an obstacle, the collision avoidance constraint is imposed by augmenting the Riemannian metric to the following expression:

$$G_i(\vec{\boldsymbol{x}}) = \left(1 + \sum_{j=1}^{N_{obs}^i} \boldsymbol{\psi}_j\right)\left(\bar{\mathbf{F}}^{-\intercal}(\vec{\boldsymbol{x}})\mathbf{D}\bar{\mathbf{F}}^{-1}(\vec{\boldsymbol{x}})\right) \tag{3.1}$$

where N_{obs}^i is the number of obstacles that agent i has identified, and the function $\boldsymbol{\psi}_j$ is a C^∞ function that increases as agent i approaches the obstacle j. The function chosen for this analysis is defined below:

$$\boldsymbol{\psi}_j(\vec{\boldsymbol{\rho}}_i, \vec{\boldsymbol{\rho}}_j) = K_j \exp\left(-\left(\frac{\delta\rho_{ij}}{\sqrt{2}P_{ij}}\right)^2\right) \tag{3.2}$$

Here $\delta\rho_{ij} = \left\|\delta\vec{\boldsymbol{\rho}}_{ij}\right\|$ and $\delta\vec{\boldsymbol{\rho}}_{ij} = \vec{\boldsymbol{\rho}}_j - \vec{\boldsymbol{\rho}}_i$ is the relative separation between the agent i and the dynamic obstacle j. The parameter P_{ij} controls the rate of increase in the cost function as the spacecraft i and j approach each other. K_j is a scalar weight that imposes the relative importance of each constraint. If agent j is identified as an obstacle, the new cost function to minimize has the classic Bell curve shape shown in Fig. 3.1.

The geodesic PDE is solved using a gradient descent method. Therefore, selecting the parameter P_{ij} affects the algorithm's performance by defining the slope of the cost function for different relative separation distances. The larger P_{ij}, the smaller the slope of the cost and the larger the missed distance between the agents. However, a larger P_{ij} also implies that the algorithm takes

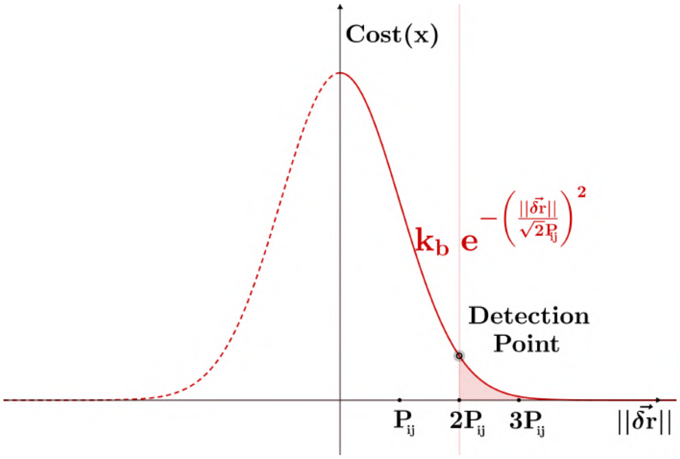


Figure 3.1: Augmented Cost Between Agents i and j

longer to converge. P_{ij} is picked so that the slope of the cost function provides the highest separation distance while ensuring minimum computation time. Fig. 3.2 shows the effects of P_{ij} on the algorithm run time and achieved separation distance at convergence, obtained from heuristic simulation data. Setting the initial separation radius to four standard deviations away from the collision point provides a good balance between maximum separation distance and minimum computation time. The parameter P_{ij} can be written as a function of the initial separation radius $\|\delta \vec{r}_0\|$ as follows:

$$P_{ij} = \frac{\|\delta \vec{r}_0\|}{4} \tag{3.3}$$

3.1.2 Receding Horizon Path Planner for Multi-Agent Transfers

The proposed receding horizon path planner is a guidance algorithm that aims to proactively correct the agents' paths and reach their destinations without colliding with each other. The guidance algorithm is discretized so that k designates a guidance epoch. Each new trajectory computed during a guidance epoch is done using the geometric path planner described in Chapter 2. The operating concept is shown in Fig. 3.3, where the re-planning agent, noted "Agent i", has an avoidance sphere of radius $\mathscr{R}_\psi$ defined in Eq. (3.4).

$$\mathscr{R}^i_\psi = \sup \left\| \mathscr{R}\left(\vec{\gamma}^{\,i}_{k-1}(t_k), \leq T_\psi\right) \right\| \tag{3.4}$$

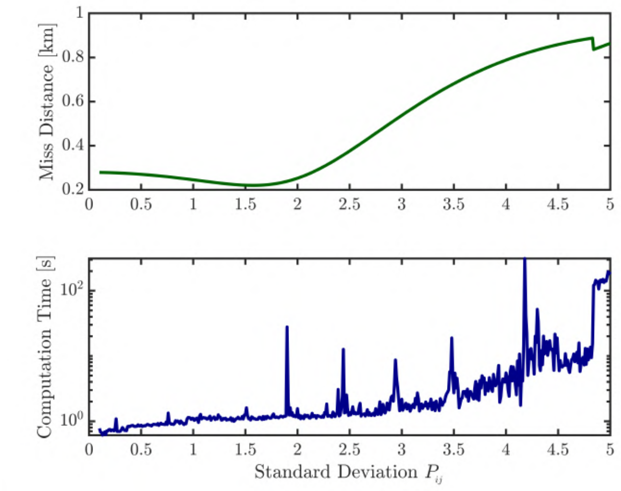
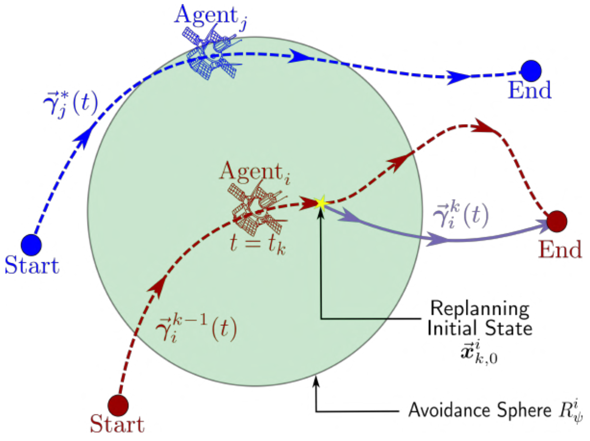


Figure 3.2: Effect of P_{ij} on the miss distance and the computation time

Here $\vec{\boldsymbol{\gamma}}^{i}_{k-1}(t)$ is the optimal path $\vec{\boldsymbol{\gamma}}^{*}_{i}(t)$ followed by Agent i before the guidance update at epoch k, and $\mathscr{R}\left(\vec{\boldsymbol{\gamma}}^{i}_{k-1}(t), \leq T_{\psi}\right)$ is the set of states that can be reached in time $t_k + T_{\psi}$ from the state at the detection time $\vec{\boldsymbol{\gamma}}^{k-1}_{i}(t_k)$. Any neighboring spacecraft, noted "Agent j", would be considered as an obstacle if it were to enter the keep-out sphere (i.e., $\left\|\delta\vec{\boldsymbol{\rho}}_{ij}\right\| \leq \mathscr{R}^{i}_{\psi}$). When an obstacle is detected, a re-planning sequence is initiated. The initial condition of the re-planning sequence is chosen to be the state of the spacecraft at time $t_k + t_{\xi}$ should it continue to follow the old optimal path (i.e., $\vec{\boldsymbol{x}}^{i}_{k,0} = \vec{\boldsymbol{\gamma}}^{i}_{k-1}(t_{\xi})$). Selecting the initial state downstream allows accounting for the continuous motion of the spacecraft during the time interval it takes to compute a new path. Each spacecraft is assigned a priority factor $\wp$ that determines which Agent is responsible for performing the avoidance maneuver. If Agent j (of higher priority level – i.e., $\wp_j > \wp_i$) is in the avoidance sphere, Agent i will be required to re-plan its path from $\vec{\boldsymbol{x}}^{k}_{i,0}$ to the goal while considering Agent j as a moving obstacle. On the other hand, if $\wp_j < \wp_i$, Agent i will ignore Agent j in the computation of its new path. This is ok because then when Agent j replays its trajectory it will avoid agent i, so one craft will always ensure collision avoidance is achieved.

The following two-agent scenario was used as a test case to study the effect of the avoidance sphere's radius on the algorithm's performance (i.e., the maximum separation distance achieved). In



Agent$_j$
$\vec{\gamma}_j^*(t)$
End
Start
Agent$_i$
$t = t_k$
$\vec{\gamma}_i^k(t)$
End
$\vec{\gamma}_i^{k-1}(t)$
Replanning Initial State $\vec{x}_{k,0}^i$
Start
Avoidance Sphere R_ψ^i

Figure 3.3: Concept of Receding Horizon Path Planner

this example, two spacecraft are tasked to transfer between two boundary conditions while avoiding each other. The motions of the two deputies are described around an eccentric reference orbit. The transfer was prescribed to occur at $t_f = \frac{3}{4}P$, where P is the period of the reference orbit. The initial conditions of the reference orbit are given in Table 1 in terms of classical orbital elements. Orbital elements are an alternative representation for a spacecraft trajectory, and they are utilized here because of the geometric insights they provide about the shape of an orbit. The reader is referred to Ref. [200] to learn more about the classical orbital elements and how they map to Cartesian coordinates.

Table 3.1: Reference Orbit in Orbital Elements

Initial OE	a [km]	e	i [rad]	Ω [rad]	ω [rad]	f [rad]
Reference	1.42e04	0.5	50	10	10	0.0

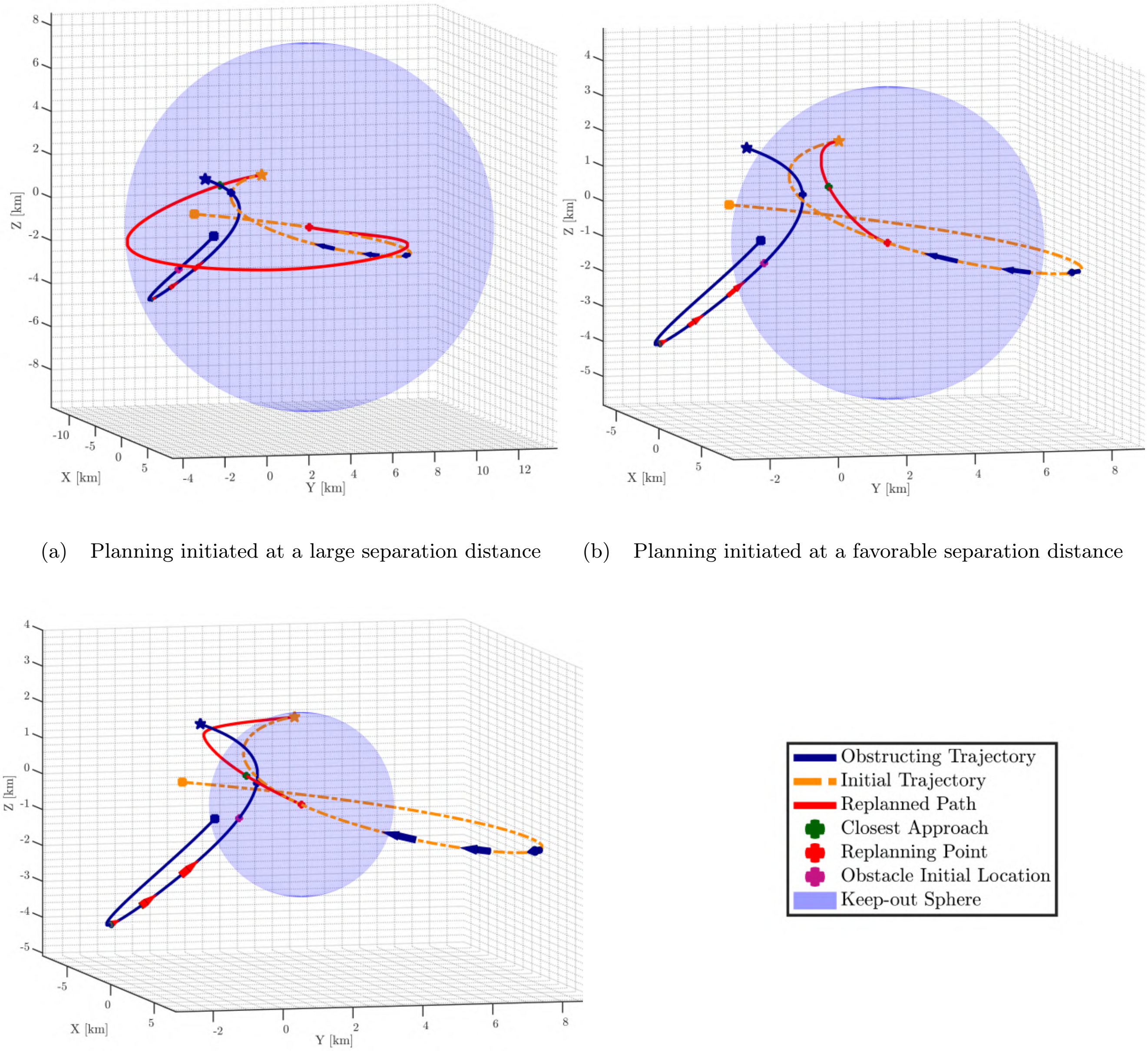
Likewise, the initial and final conditions of the deputy spacecraft are given in Table 3.2, and these boundary conditions are represented as relative orbital elements. The reader is referred to Ref.[177] to learn more about relative orbital elements.

Table 3.2: Deputies Orbit in Relative Orbital Elements

ΔOE	δa [km]	δe	δi [rad]	$\delta\Omega$ [rad]	$\delta\omega$ [rad]	δf [rad]
Deputy1 (initial)	0.0	$-4/(3a)$	$-9/(2a)$	$-2\text{E}{-5} \, * \, \pi$	0.0	$1\text{E}{-6} \, * \, \pi$
Deputy2 (initial)	0.0	$9/(2a)$	$-5/(3a)$	$-\pi * 1e - 5$	$\pi * 1e - 5$	$\pi * 1e - 6$
Deputy1 (final)	0.0	$5/(2a)$	$-9/(4a)$	0.0	$1\text{E}{-6} \, * \, \pi$	$-1\text{E}{-5} \, * \, \pi$
Deputy2 (final)	0.0	$-5/(2a)$	$-9/(4a)$	$-5\text{E}{-6} \, * \, \pi$	$1\text{E}{-6} \, * \, \pi$	$5\text{E}{-5} \, * \, \pi$

Three possible scenarios are possible depending on how far the obstacle is when it is detected. The re-planning agent can be too close, too far, or a "favorable" distance from the moving obstacle. Each of these scenarios is shown in Fig. 3.4. The planning horizon is specified by the avoidance sphere, represented here as a blue sphere. Suppose the re-planning occurs too far from the obstacle. In that case, the algorithm generates an utterly different trajectory which may become problematic as more obstacles are introduced in the simulation and would require more fuel to get to the boundary at time $t = t_f$(cf. Fig. 3.4(a)).

On the other hand, if the re-planning is initiated too close to the impediment, the new trajectory requires significant control to move away due to its inertia (cf. Fig. 3.4(c)). In addition, the miss distance achieved is small because of the momentum of the maneuvering agent. Lastly, if the planning occurs at a relatively favorable distance from the obstacle, the re-planning algorithm generates a new path that obeys all the constraints that have been imposed (cf. Fig. 3.4(b)). The relative separation for each case is shown in Fig. 3.5.



(a) Planning initiated at a large separation distance

(b) Planning initiated at a favorable separation distance

(c) Planning initiated at a small separation distance

(d) legend

Figure 3.4: Effects of the initial separation distance on the re-planned path

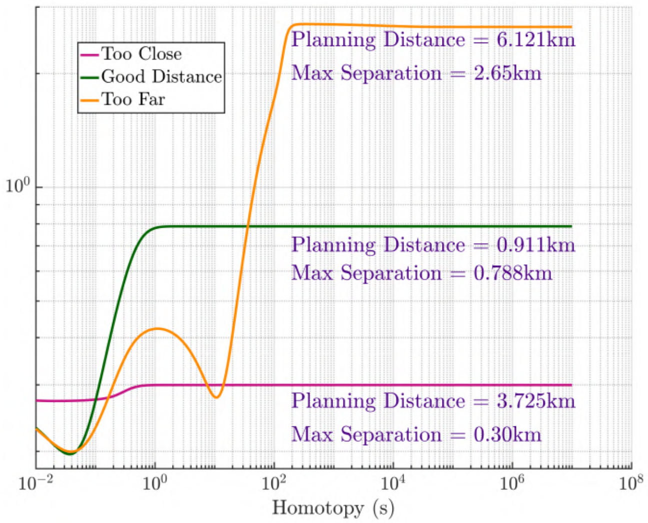


Figure 3.5: Achieved Separation Distance

A directional cone replaces the avoidance sphere to account for motion direction and avoid triggering a planning maneuver when agents are not moving in the same direction (see fig. 3.6). The cone is centered at the avoiding agent and points in the direction of its velocity vector. The requirement for initiating a re-planning sequence is

$$\left\{ \|\delta\vec{\boldsymbol{\rho}}_{ij}\| \leq \mathscr{R}^i_\psi \right\} \cap \left\{ \measuredangle(\dot{\vec{\boldsymbol{\rho}}}_i, \delta\vec{\boldsymbol{\rho}}_{ij}) < \alpha_i \right\} \tag{3.5}$$

where $\measuredangle(\dot{\vec{\boldsymbol{\rho}}}_i, \delta\vec{\boldsymbol{\rho}}_{ij})$ is the angle between the relative position vector $\delta\vec{\boldsymbol{\rho}}_{ij}$ and the velocity of the detecting spacecraft $\dot{\vec{\boldsymbol{\rho}}}_i$. The maximum angle α_i depends on sensor capability.

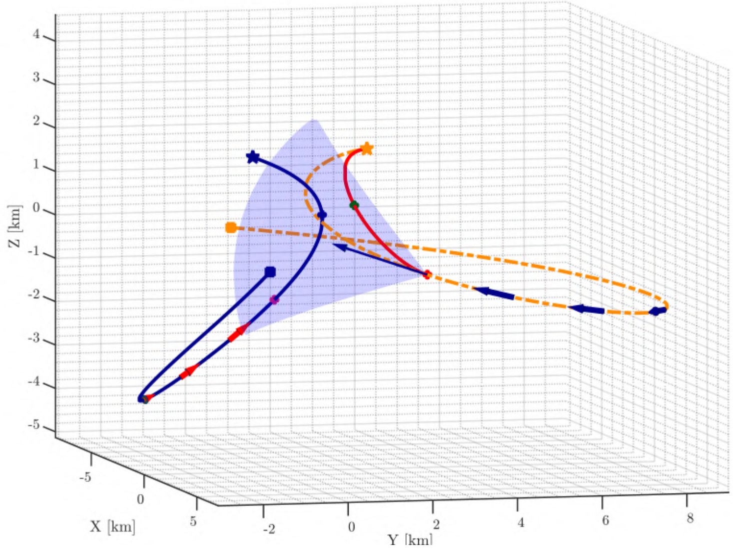


Figure 3.6: Implementation of the Direction Cone

The inner workings of the proposed guidance algorithm operate as shown in Algorithm 2.

Algorithm 2: Receding Horizon Algorithm (for Agent$_i$)

```
input  : $\vec{\boldsymbol{x}}^i_{k,0}$, $\vec{\gamma}^i_{k-1}(t)$
if $\left\{ \|\delta\vec{\boldsymbol{\rho}}_{ij}\| \leq \mathscr{R}^i_{\psi} \right\} \cap \left\{ \measuredangle(\dot{\vec{\boldsymbol{\rho}}}_i, \delta\vec{\boldsymbol{\rho}}_{ij}) < \alpha_i \right\}$ then
    if $\wp_j > \wp_i$, $j \in N^i_{obs}$ then
        // New Riemannian Metric
        $\mathbf{G}_i(x) = \left(1 + \sum_{j=1}^{N^i_{obs}} \boldsymbol{\psi}_j\right) \left(\bar{\mathbf{F}}^{-\intercal}(\vec{\boldsymbol{x}})\mathbf{D}\bar{\mathbf{F}}^{-1}(\vec{\boldsymbol{x}})\right)$
        // Agent$_i$'s new optimal path
        $\mathbf{G}_i^{-1}(\vec{\gamma}^i_k) \left( \frac{d}{dt} \frac{\partial L}{\partial \dot{\vec{\boldsymbol{x}}}}(\vec{\gamma}^i_k, \dot{\vec{\gamma}}^i_k) - \frac{\partial L}{\partial \vec{\boldsymbol{x}}}(\vec{\gamma}^i_k, \dot{\vec{\gamma}}^i_k) \right) = 0$;
    else
        // No action needed
        $\vec{\gamma}^i_k(t) = \vec{\gamma}^i_{k-1}(t)$;
    end
else
    // No action needed
    $\vec{\gamma}^i_k(t) = \vec{\gamma}^i_{k-1}(t)$;
end
output: $\vec{\gamma}^i_k(t)$
```

3.2 Application to Large Constellation: Results and Discussions

Algorithm 2 can be implemented on arbitrarily large constellations, and Fig. 3.8 shows an example with six deputy spacecraft around the eccentric reference orbit specified in Table 2.1. The avoidance sphere radius $\mathscr{R}_{\psi}^{i} = 5$km for each agent, and the angle of the direction cone constraint was $\alpha_i = 30°$. Here the six spacecraft were tasked to transfer between two sets of boundary conditions chosen at random. The green lines in Fig. 3.7 represent the initial trajectories computed with no regard to the potential presence of obstacles (i.e., $\vec{\gamma}_0^i(t)$). In contrast, the dashed orange lines represent the corrected paths generated after all the guidance updates (i.e., $\vec{\gamma}_\infty^i(t)$). It is also important to note that not all spacecraft had to re-plan their maneuver. Some agents had their original path clear of obstacles while others had a higher hierarchy level in the re-planning sequence; the hierarchy used in here is $\wp_6 > \wp_5 > \wp_4 > \wp_3 > \wp_2 > \wp_1$.

Fig. 3.8(a) shows a comparison of the control magnitude required to travel along each trajectory. Despite actively changing course to avoid moving obstacles, the amount of control necessary to follow the updated paths is not too different from the initial path $\vec{\gamma}_0^i(t)$. This fact can be attributed to the computed trajectories still inherently leveraging the natural dynamics to minimize fuel usage.

The only significant difference in fuel usage (more than twice the original amount) occurs during the transfer of agent 4. Fig. 3.8(b) provides more insights on each of the re-planning sequences Agent 4 initiated on its way to its end condition. At the beginning of the transfer, Agent 5 is identified as an obstacle. Therefore, Agent 4's guidance algorithm computed a new path to avoid Agent 5. The new trajectory deviates from the original path and moves away from Agent 5. Later, Agent 4 detects Agent 6 in its avoidance cone. It then initiates three avoidance maneuvers. Because the detection of Agent 6 occurs close to the final condition for agent 4, more control effort is required to change course and make sure the two spacecraft do not run into each other.

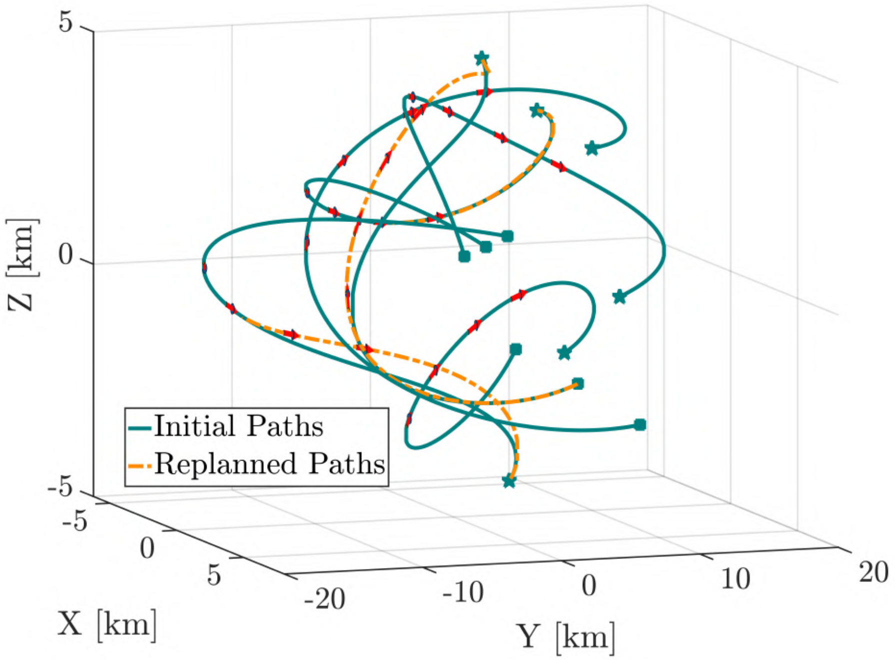
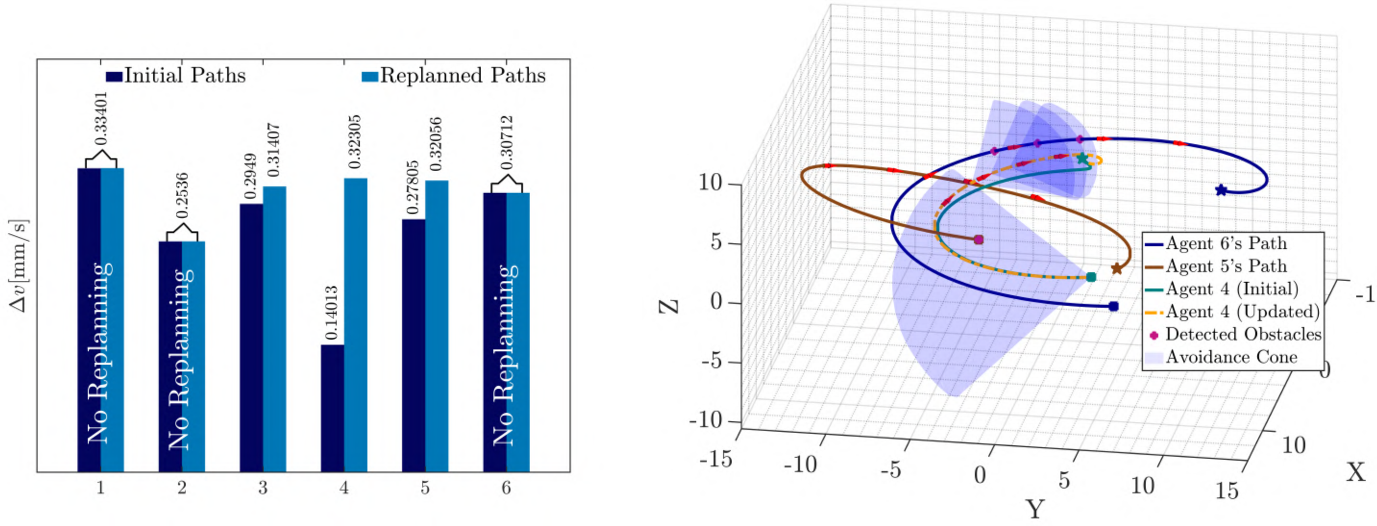


Figure 3.7: 3D Trajectories for All the Agents



(a) Comparison of Required Control

(b) Zooming on Agent 4 and its Obstacles

Figure 3.8: Distributed Geometric Path Planner for Six-Agent's Transfers

3.3 Summary

This chapter presents a distributed guidance algorithm that generates minimum-fuel and collision-free trajectories for multi-agent systems. The mathematical foundation of the path planning algorithm provides a seamless framework to introduce collision avoidance constraints on moving obstacles. The algorithm is implemented on a six-agent spacecraft constellation where each spacecraft was tasked to transfer from one parking orbit to another. It was shown that the distributed guidance algorithm allows each spacecraft to safely get to its destination while minimizing the fuel expenditure.

Chapter 4

Nonlinear Controllability and 6DoF Fuel-free Transfer Using Differential Solar Radiation Pressure and Differential Drag

This chapter extends earlier findings and leverages natural forces such as differential solar radiation pressure (SRP) and differential drag (between the chief spacecraft and the deputy spacecraft) to generate fuel-free transfer trajectories. This work will primarily focus on trajectories between Low Earth Orbit (LEO) and Medium Earth Orbit (MEO). The deputy is modeled as a flat plate, while the chief is assumed to be a cannonball. The differential accelerations are modified by modulating the attitude of the deputy. It is assumed that the deputy is equipped with a fully actuated gyro system, and the controllability of the relative path planning problem is investigated using Lie bracket theory. The contribution of this chapter is two fold: (1) provide tools to analyze the controllability of nonlinear affine problems, and those tools are used to determine conditions for which the spacecraft formation flying problem is controllable in a drag-plus-SRP-perturbed environment; and (2) the geometric motion planner introduced in Chapter 2 is extended to generate trajectories that solely require the accelerations from differential drag and differential SRP as the translational propulsion capability. This chapter is organized as follows: Section 4.1 presents the controllability results derived using the Lie bracket theory. Section 4.2 presents some simulation results showcasing the fuel-free trajectory generation. Section 4.3 closes the chapter with some last remarks.

4.1 Nonlinear Controllability Analysis of the Deputy's Motion

The following section presents the controllability analysis for the relative dynamic problem wherein the deputy spacecraft is modeled as a flat plate. In contrast, the chief spacecraft is modeled as a cannonball. The purpose of the following analysis is to investigate whether the deputy spacecraft can arbitrarily change its position and velocity by modifying its attitude. The differential acceleration (from SRP and drag forces) is the sole propulsive capability driving the deputy's motion, and it is assumed that the deputy is equipped with a fully actuated control gyro system. First, the dynamic model is presented, then the Lie brackets criterion for controllability is used to assess the feasibility of the path planning problem.

4.1.1 6DOF Dynamic Model for the Deputy Relative Motion

In the LVLH frame, the full nonliear dynamics of the deputy can be described by Eq. (4.1):

$$\dot{\vec{\boldsymbol{x}}} = \boldsymbol{f}(\vec{\boldsymbol{x}}) + \boldsymbol{b}_1(\vec{\boldsymbol{x}})u_1 + \boldsymbol{b}_2(\vec{\boldsymbol{x}})u_2 + \boldsymbol{b}_3(\vec{\boldsymbol{x}})u_3 \tag{4.1}$$

where

$$\boldsymbol{f} = \begin{bmatrix} \frac{1}{4}\boldsymbol{B}(\vec{\boldsymbol{\sigma}})\vec{\boldsymbol{\omega}} \\ [I]^{-1}\left(-\vec{\boldsymbol{\omega}} \times [I]\vec{\boldsymbol{\omega}}\right) \\ \vec{\boldsymbol{\rho}}\prime \\ \delta\vec{\boldsymbol{a}}^{g} + \delta\vec{\boldsymbol{a}}^{s} + \delta\vec{\boldsymbol{a}}^{d} + \vec{\boldsymbol{\Xi}} \end{bmatrix}, \quad \boldsymbol{b}_1 = \begin{bmatrix} 0_{3\times1} \\ \boldsymbol{i}_1 \\ 0_{3\times1} \\ 0_{3\times1} \end{bmatrix}, \quad \boldsymbol{b}_2 = \begin{bmatrix} 0_{3\times1} \\ \boldsymbol{i}_2 \\ 0_{3\times1} \\ 0_{3\times1} \end{bmatrix}, \quad \boldsymbol{b}_3 = \begin{bmatrix} 0_{3\times1} \\ \boldsymbol{i}_3 \\ 0_{3\times1} \\ 0_{3\times1} \end{bmatrix} \tag{4.2}$$

$\vec{\boldsymbol{x}} = [\vec{\boldsymbol{\sigma}}, \vec{\boldsymbol{\omega}}, \vec{\boldsymbol{\rho}}, \vec{\boldsymbol{\rho}}\prime]$ where $\vec{\boldsymbol{\sigma}}$ is the Modified Rodrigues Parameters (MRP) set representing the orientation of the flat plate, $\vec{\boldsymbol{\omega}}$ is the body frame angular velocity, and the relative position and velocity are $\vec{\boldsymbol{\rho}}$ and $\vec{\boldsymbol{\rho}}\prime$. The kinematic matrix $\boldsymbol{B}$ maps the effects of the angular velocity of the plate on the MRP's rate of change, and it is written as $\boldsymbol{B}(\vec{\boldsymbol{\sigma}}) = \big((1 - \vec{\boldsymbol{\sigma}}^{\intercal}\vec{\boldsymbol{\sigma}})[\mathbb{I}_{3\times3}] + 2[\tilde{\boldsymbol{\sigma}}] + 2\vec{\boldsymbol{\sigma}}\vec{\boldsymbol{\sigma}}^{\intercal}\big)$. The skew symmetric matrix $[\tilde{\boldsymbol{\sigma}}]$ is made of components of $\vec{\boldsymbol{\sigma}}$ and represents the cross product operation. It is assumed that the body frame is aligned with the principal body axis; thus, the moment of inertial $[I]$ is a diagonal matrix when expressed in the body frame (i.e.,$[I] = [\boldsymbol{i}_1,\ \boldsymbol{i}_2,\ \boldsymbol{i}_3]$, where $\boldsymbol{i}_1 = [\mathrm{i}_1,\ 0,\ 0]^{\intercal}$, $\boldsymbol{i}_2 = [0,\ \mathrm{i}_2,\ 0]^{\intercal}$, $\boldsymbol{i}_3 = [0,\ 0,\ \mathrm{i}_3]^{\intercal}$, and $\mathrm{i}_1, \mathrm{i}_2, \mathrm{i}_3 \in \mathbb{R}$). The vector $\vec{\boldsymbol{\Xi}}$ represents the effects of the rotating LVLH frame and is defined as $\vec{\boldsymbol{\Xi}} = -\dot{\vec{\boldsymbol{\Omega}}} \times \vec{\boldsymbol{\rho}} - 2\vec{\boldsymbol{\Omega}} \times \vec{\boldsymbol{\rho}}\prime - \vec{\boldsymbol{\Omega}} \times \vec{\boldsymbol{\Omega}} \times \vec{\boldsymbol{\rho}}$. The vectors $\vec{\boldsymbol{\Omega}}$ and $\dot{\vec{\boldsymbol{\Omega}}}$ are the angular velocity and acceleration with respect to the chief's LVLH frame. The vectors $\delta\vec{\boldsymbol{a}}^{g}$, $\delta\vec{\boldsymbol{a}}^{s}$, and $\delta\vec{\boldsymbol{a}}^{d}$ are the differential accelerations due to the effects of gravitational, solar radiation pressure, and drag, respectively. Each of these accelerations are defined in further detail below.

(1) ***Differential gravitational acceleration***: This quantity is defined in Eq.(2.4), but it is repeated here for convenience. The full nonlinear expression for the relative gravitational acceleration, assuming point mass gravity, is given below as components in the chief LVLH frame is written as follows:

$$\delta\vec{\boldsymbol{a}}^{g} = -\frac{\mu}{\left[(r+x)^2 + y^2 + z^2\right]^{3/2}} \begin{bmatrix} r+x \\ y \\ z \end{bmatrix} + \frac{\mu}{r^2} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{4.3}$$

(2) ***Differential solar radiation pressure acceleration***: The two SRP force models are written as follows [65, 201]:

(a) ***Cannonball Force Model (Chief):***

$$\vec{\boldsymbol{a}}_{c}^{s} = -\frac{P_{srp} C_r A_c}{m_c} \delta\hat{\boldsymbol{r}} \tag{4.4}$$

where $\delta\vec{\boldsymbol{r}} = \vec{\boldsymbol{r}}_c - \vec{\boldsymbol{r}}_{sun}$ is the relative distance between the Sun and the chief, $\vec{\boldsymbol{r}}_c$ is the position vector of the spacecraft, and $\vec{\boldsymbol{r}}_{sun}$ is the position vector of the sun. Both of these quantities are expressed in the inertial frame. The unit vector $\delta\hat{\boldsymbol{r}}$ is the unit vector pointing from the Sun to the spacecraft. P_{srp} is magnitude of the force exerted on an object at a distance $R = \|\vec{\boldsymbol{r}}_{sun}\|$ from the Sum and is given by:

$$P_{srp} = \frac{P_0}{c}\left(\frac{R_0}{R}\right)^2 \tag{4.5}$$

where $R_o = 1AU$ is the distance from Earth to the Sun, $P_0 = 1367\text{W/m}^2$ is the solar flux at 1 AU, and $c = 299792458\text{m/s}$ is the speed of light.

(b) ***Flat Plate Force Model (Deputy):***

$$\vec{\boldsymbol{a}}_d^{s} = -\frac{P_s}{m_d} A_d \langle \hat{\mathbf{n}}, \hat{\mathbf{r}}_d \rangle \left[(1-\rho_s)\hat{\mathbf{r}}_d + 2\left(\rho_s \langle \hat{\mathbf{n}}, \hat{\mathbf{r}}_d \rangle + \frac{\rho_d}{3}\right)\hat{\mathbf{n}}\right] \tag{4.6}$$

where $\hat{\mathbf{r}}_d$ is the unit vector from the deputy to the sun, $\hat{\mathbf{n}}$ is the normal vector of the plate, A_d is the reflecting surface of the plate. ρ_s and ρ_d are the specular and diffusion rates respectively.

(c) ***Differential SRP acceleration:***

$$\delta\vec{\boldsymbol{a}}^{s} = [CB]\vec{\boldsymbol{a}}_d^{s} - \vec{\boldsymbol{a}}_c^{s} \tag{4.7}$$

where $[CB]$ is the Direct Cosine Matrix (DCM) that maps vectors from the deputy body frame B into the chief's LVLH frame C.

(3) ***Drag acceleration***: The drag model used in this study is given by

$$\vec{\boldsymbol{a}}^{d} = \frac{C_d \beta A}{2}\left(\delta\vec{\boldsymbol{v}}_{st}^{\intercal}\delta\vec{\boldsymbol{v}}_{st}\right)^{\frac{1}{2}}\delta\vec{\boldsymbol{v}}_{st} \tag{4.8}$$

where $\delta\vec{\boldsymbol{v}}_{st} = \vec{\boldsymbol{v}}_{st} - \vec{\boldsymbol{v}}$ is the difference between the free stream velocity $\vec{\boldsymbol{v}}_{st}$ and the spacecraft velocity $\vec{\boldsymbol{v}}$. Here Earth is assumed to be spherical, and the air particles are fixed in the Earth-Fixed-Earth-Rotating (ECEF) frame (i.e., the air particles rotate at the same spin rate as the Earth). C_d is the drag coefficient, and the parameter β represents the air density, and it is computed using the U.S. Standard Atmosphere of 1976 [149]. The reference area A is the effective area facing the free stream flow and is given by:

$$A = \begin{cases} A_c, & \text{for the cannonball (chief)} \\ A_d\langle \hat{\mathbf{n}}, \delta\vec{\boldsymbol{v}}_{st}\rangle & \text{for the flat-plate (deputy)} \end{cases} \tag{4.9}$$

4.1.2 Lie Brackets and Accessibility Algebra Criteria

A brief review of the geometric nonlinear control tools is provided. The following section discusses the utilization of Lie brackets in characterizing the controllability of an affine control system (as the studied system is of this form). Let the differential equation of an n-dimensional system be represented by the following affine control system:

$$\dot{\vec{\boldsymbol{x}}}(t) = \boldsymbol{f}(\vec{\boldsymbol{x}}(t)) + \sum_{i=1}^{m} \boldsymbol{b}_i(\vec{\boldsymbol{x}}(t))u_i(t) \tag{4.10}$$

where $\vec{\boldsymbol{x}}(t)$ is a curve on a solution manifold $\mathscr{M}$. The multivariate function $\boldsymbol{f}(\cdot)$ is called the drift vector field and describes the behavior of the system in the absence of any controls. The multivariate function $\boldsymbol{b}_i(\cdot)$ is the control vector field and defines how the i^{th} control input affects the behavior of the system. $\vec{\boldsymbol{u}} \in \mathscr{U} \subset \mathbb{R}^m$ is the control input, where $\mathscr{U}$ is the set of admissible control. The triplet $\Sigma = \{\mathscr{M}, \mathscr{F} = \{\boldsymbol{f}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_m\}, \mathscr{U}\}$ is typically used in the literature to represent affine control systems.

Let $T \in \mathbb{R}^+$. The set of states that can be reached in time $t \leq T$ from the initial state $\vec{\boldsymbol{x}}_0$ is called the time T reachable set from $\vec{\boldsymbol{x}}_0$ and is defined as:

$$\mathscr{R}_\Sigma(\vec{\boldsymbol{x}}_0, \leq T) = \bigcup_{t \in [0,T]} \mathscr{R}_\Sigma(\vec{\boldsymbol{x}}_0, t) \tag{4.11}$$

where $\mathscr{R}_\Sigma(\vec{\boldsymbol{x}}_0, t) = \{\vec{\gamma}(t, \vec{\boldsymbol{x}}, \vec{\boldsymbol{u}}) | \vec{\gamma}(t, \vec{\boldsymbol{x}}, \vec{\boldsymbol{u}})$ is a control trajectory for $\Sigma, \vec{\boldsymbol{u}} \in \mathscr{U},$ and $\vec{\gamma}(0, \vec{\boldsymbol{x}}, \vec{\boldsymbol{u}}) = \vec{\boldsymbol{x}}_0\}$, and $[0, T]$ the time interval on which the system is simulated. The set $\mathscr{R}_\Sigma(\vec{\boldsymbol{x}}_0)$ denotes the of states that are reachable from $\vec{\boldsymbol{x}}_0$ in infinite time (i.e., $T = \infty$).

In nonlinear geometric control, the concept of controllability is intrinsically linked to the characteristics of the reachable sets. Given an affine control system $\Sigma = \{\mathscr{M}, \mathscr{F} = \{\boldsymbol{f}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_m\}, \mathscr{U}\}$, the different forms of controllability are as follows [172]:

(1) ***Accessible*:** Σ is accessible from $\vec{\boldsymbol{x}}_0$ if $int(\mathscr{R}_\Sigma(\vec{\boldsymbol{x}}_0)) \neq \emptyset$

(2) ***Strongly Accessible*:** Σ is strongly accessible from $\vec{\boldsymbol{x}}_0$ if $int(\mathscr{R}_\Sigma(\vec{\boldsymbol{x}}_0, \leq T)) \neq \emptyset \ \forall \ T > 0$

(3) ***Locally Controllable*:** Σ is locally controllable from $\vec{\boldsymbol{x}}_0$ if $\vec{\boldsymbol{x}}_0 \in int(\mathscr{R}_\Sigma(\vec{\boldsymbol{x}}_0, T))$

(4) ***Small Time Locally Controllable(STLC)*:** Σ is STLC from $\vec{\boldsymbol{x}}_0$ if there exists $T > 0$ so that $\vec{\boldsymbol{x}}_0 \in int(\mathscr{R}_\Sigma(\vec{\boldsymbol{x}}_0, \leq t) \ \forall \ t \in [0, T])$

(5) ***Globally Controllable*:** Σ is globally controllable from $\vec{\boldsymbol{x}}_0$ if $\mathscr{R}_\Sigma(\vec{\boldsymbol{x}}_0) = \mathscr{M}$.

Here $int\{\cdot\}$ means the interior set. These controllability definitions are gradually more stringent, and they are determined by assessing the span of vector fields on the solution manifold. These vector fields are computed by successive commutation of the drift and control vector fields, via the Lie bracket operator shown in Fig. 4.1 and defined below:

$$\begin{aligned} [\boldsymbol{f}, \boldsymbol{b}_i](\vec{\boldsymbol{x}})\colon \mathscr{F} &\to \mathscr{F} \\ \vec{\boldsymbol{x}} &\mapsto \frac{\partial \boldsymbol{b}_i}{\partial \vec{\boldsymbol{x}}}(\vec{\boldsymbol{x}})\boldsymbol{f}(\vec{\boldsymbol{x}}) - \frac{\partial \boldsymbol{f}}{\partial \vec{\boldsymbol{x}}}(\vec{\boldsymbol{x}})\boldsymbol{b}(\vec{\boldsymbol{x}}) \end{aligned} \tag{4.12}$$

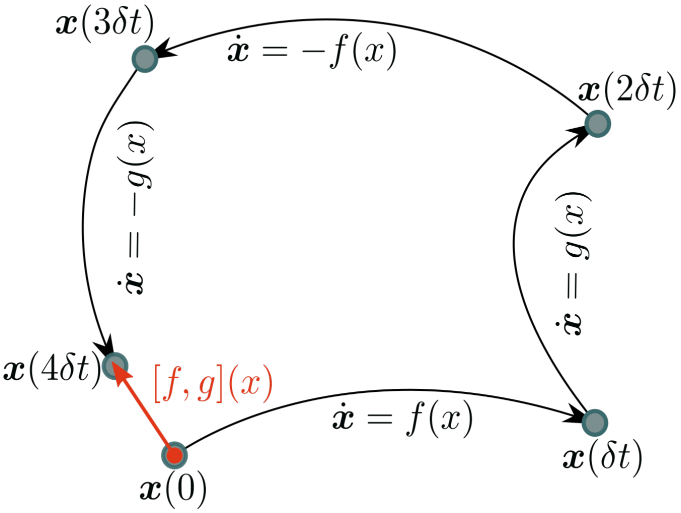

$\boldsymbol{x}(3\delta t)$
$\dot{\boldsymbol{x}} = -f(x)$
$\boldsymbol{x}(2\delta t)$
$\dot{\boldsymbol{x}} = -g(x)$
$\dot{\boldsymbol{x}} = g(x)$
$\boldsymbol{x}(4\delta t)$
$[f, g](x)$
$\dot{\boldsymbol{x}} = f(x)$
$\boldsymbol{x}(\delta t)$
$\boldsymbol{x}(0)$

Figure 4.1: Geometric interpretation of the Lie bracket

Given the vector fields $\vec{\boldsymbol{X}}, \vec{\boldsymbol{Y}}, \vec{\boldsymbol{Z}}$ on a smooth manifold $\mathscr{M}$ and the smooth functions $\xi, \psi : \mathscr{M} \mapsto \mathbb{R}$, a Lie bracket operator has the following properties [123]:

(1) The flow map $(\vec{\boldsymbol{X}}, \vec{\boldsymbol{Y}}) \mapsto [\vec{\boldsymbol{X}}, \vec{\boldsymbol{Y}}]$ is $\mathbb{R}$-bilinear[1]

(2) $[\vec{\boldsymbol{X}}, \vec{\boldsymbol{Y}}] = -[\vec{\boldsymbol{Y}}, \vec{\boldsymbol{X}}]$ (Skew Symmetry)

(3) $[\vec{\boldsymbol{X}}, [\vec{\boldsymbol{Y}}, \vec{\boldsymbol{Z}}]] + [\vec{\boldsymbol{Z}}, [\vec{\boldsymbol{Y}}, \vec{\boldsymbol{Y}}]] + [\vec{\boldsymbol{Y}}, [\vec{\boldsymbol{Z}}, \vec{\boldsymbol{X}}]] = \vec{\boldsymbol{0}}$ (Jacobi identity)

(4) $[\xi\vec{\boldsymbol{X}}, \psi\vec{\boldsymbol{Y}}] = \xi\psi[\vec{\boldsymbol{X}}, \vec{\boldsymbol{Y}}] + \xi\mathscr{L}_{\vec{\boldsymbol{X}}}(\psi)\vec{\boldsymbol{Y}} - \psi\mathscr{L}_{\vec{\boldsymbol{Y}}}(\xi)\vec{\boldsymbol{X}}$, where $\mathscr{L}_{\vec{\boldsymbol{X}}}(\psi) = \frac{\partial\psi}{\partial\vec{\boldsymbol{X}}}(\vec{\boldsymbol{X}})$

[1] A bilinear map is a function combining elements of two vector spaces to yield an element of a third vector space, and is linear in each of its arguments

The mapping defined Eq. (4.12) is called the Lie bracket between the vectors fields $\boldsymbol{f}$ and $\boldsymbol{b}_i$ and represents a new attainable direction if it is independent of its generators $\boldsymbol{f}$ and $\boldsymbol{b}_i$. Conceptually, the Lie bracket between two vectors fields is a vector product generating another field which, if independent of its generators, represents a new attainable direction for a curve to flow. A good example is parallel parking a car. By alternating between the allowable velocities (vector fields), one can generate a lateral motion.

At every point $\vec{\boldsymbol{x}} \in \mathscr{M}$, one can define a tangent vector as the smooth mapping $\mathscr{V} \colon \boldsymbol{C}^{\infty}(\mathscr{M}) \to \mathbb{R}^n$. The set of all tangent vectors at $\vec{\boldsymbol{x}}$ is denoted by $T_{\vec{\boldsymbol{x}}}\mathscr{M}$ and called the tangent space at $\vec{\boldsymbol{x}}$. The family of vector field (i.e., the set of tangent vectors) describing the behavior of the dynmical system defined in Eq. (4.10) is $\mathscr{F}_{\vec{\boldsymbol{x}}} = \{\boldsymbol{f}(\vec{\boldsymbol{x}}), \boldsymbol{b}_1(\vec{\boldsymbol{x}}), \ldots, \boldsymbol{b}_m(\vec{\boldsymbol{x}})\}$. The set of independent vector fields associated with $\mathscr{F}_{\vec{\boldsymbol{x}}}$ denoted $\Delta(\mathscr{F}_{\vec{\boldsymbol{x}}})$ is defined by [115]:

$$\Delta(\mathscr{F}_{\vec{\boldsymbol{x}}}) = \{X(\vec{\boldsymbol{x}}) \mid X \in \mathscr{L}(\mathscr{F}_{\vec{\boldsymbol{x}}})\} \subseteq T_{\vec{\boldsymbol{x}}}\mathscr{M} \tag{4.13}$$

$\Delta(\mathscr{F}_{\vec{\boldsymbol{x}}})$ and is called the accessibility distribution, and $\mathscr{L}(\mathscr{F}_{\vec{\boldsymbol{x}}})$ is the maximum set of linearly independent vector fields obtained from successively taking Lie brackets of vectors in $\mathscr{F}_{\vec{\boldsymbol{x}}}$. When dealing with control affine systems with drift, as in Eq. (4.10), the accessibility distribution should be constructed with no bad Lie brackets[2] as those obstruct controllability by not allowing the flow to be reversed by changing the sign of the control [189]. The P.Hall basis can be used to compute good brackets. Indeed, the P.Hall basis of order k is a set of linearly independent Lie brackets of order k. Table 4.1 show the P. Hall basis for four vector fields up to order three.

One of the classical results in geometric control theory, independently proven by W.L. Chow and P.K. Rashevskii, provides sufficient condition for controllability of a nonlinear affine system based on the properties of the accessibility distribution [29, 22, 172]. The Chow-Rashevskii Theorem states that given a smooth manifold $\mathscr{M}$ and a smooth vector field $\mathscr{F}_{\vec{\boldsymbol{x}}} = \{\boldsymbol{f}(\vec{\boldsymbol{x}}), \boldsymbol{b}_1(\vec{\boldsymbol{x}}), \ldots, \boldsymbol{b}_m(\vec{\boldsymbol{x}})\}$ defined on $T_{\vec{\boldsymbol{x}}}\mathscr{M}$, the control system defined in Eq. (4.10) is locally controllable in any time (or STLC) at every point of $\mathscr{M}$ if $\Delta(\mathscr{F}_{\vec{\boldsymbol{x}}}) = T_{\vec{\boldsymbol{x}}}\mathscr{M}$ for all $\vec{\boldsymbol{x}} \in \mathscr{M}$. The idea of local controllability,

[2] A bad Lie bracket is a vector field where the control $\boldsymbol{g}$ appears in the construction of the vector field (such as $[g_i; [f; g_i]]$).

Table 4.1: P.Hall basis of order three for four vector fields

Bracket length	**Lie Brackets**
one	$\boldsymbol{f}, \boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3$
two	$[\boldsymbol{f}, \boldsymbol{b}_1], [\boldsymbol{f}, \boldsymbol{b}_2], [\boldsymbol{f}, \boldsymbol{b}_3], [\boldsymbol{b}_1, \boldsymbol{b}_2], [\boldsymbol{b}_1, \boldsymbol{b}_3], [\boldsymbol{b}_2, \boldsymbol{b}_3]$
three	$[\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{b}_1]], [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{b}_2]], [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{b}_3]], [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{b}_1]],$ $[\boldsymbol{f}, [\boldsymbol{b}_1, \boldsymbol{b}_2]], [\boldsymbol{f}, [\boldsymbol{b}_1, \boldsymbol{b}_3]], [\boldsymbol{f}, [\boldsymbol{b}_2, \boldsymbol{b}_3]],$ $[\boldsymbol{b}_1, [\boldsymbol{f}, \boldsymbol{b}_1]], [\boldsymbol{b}_1, [\boldsymbol{f}, \boldsymbol{b}_2]], [\boldsymbol{b}_1, [\boldsymbol{f}, \boldsymbol{b}_3]],$ $[\boldsymbol{b}_1, [\boldsymbol{b}_1, \boldsymbol{b}_2]], [\boldsymbol{b}_1, [\boldsymbol{b}_1, \boldsymbol{b}_3]], [\boldsymbol{b}_1, [\boldsymbol{b}_2, \boldsymbol{b}_3]],$ $[\boldsymbol{b}_2, [\boldsymbol{f}, \boldsymbol{b}_2]], [\boldsymbol{b}_2, [\boldsymbol{f}, \boldsymbol{b}_3]], [\boldsymbol{b}_2, [\boldsymbol{b}_1, \boldsymbol{b}_2]],$ $[\boldsymbol{b}_2, [\boldsymbol{b}_2, \boldsymbol{b}_3]], [\boldsymbol{b}_3, [\boldsymbol{f}, \boldsymbol{b}_3]], [\boldsymbol{b}_3, [\boldsymbol{b}_1, \boldsymbol{b}_3]], [\boldsymbol{b}_3, [\boldsymbol{b}_2, \boldsymbol{b}_3]]$

depicted in Fig. 4.2, is formally defined as follows: For every $\vec{\boldsymbol{x}}_0 \in \mathscr{M}$, every $T > 0$, and every neighborhood $\mathscr{N}$ of $\vec{\boldsymbol{x}}_0$, there is a neighborhood $\mathscr{O} \subset \mathscr{N}$ of $\vec{\boldsymbol{x}}_0$ such that for any $\vec{\boldsymbol{x}}_1, \vec{\boldsymbol{x}}_2 \in \mathscr{O}$, there is a control $\vec{\boldsymbol{u}} \in \mathscr{U}$ such that the trajectory $\vec{\boldsymbol{\gamma}}(t, \vec{\boldsymbol{x}}, \vec{\boldsymbol{u}}) : [0, T] \to \mathscr{M}$ remains in $\mathscr{N}$ and steers $\vec{\boldsymbol{x}}_1$ to $\vec{\boldsymbol{x}}_2$ (i.e. $\vec{\boldsymbol{\gamma}}(0, \vec{\boldsymbol{x}}, \vec{\boldsymbol{u}}) = \vec{\boldsymbol{x}}_1$ and $\vec{\boldsymbol{\gamma}}(T, \vec{\boldsymbol{x}}, \vec{\boldsymbol{u}}) = \vec{\boldsymbol{x}}_2$).

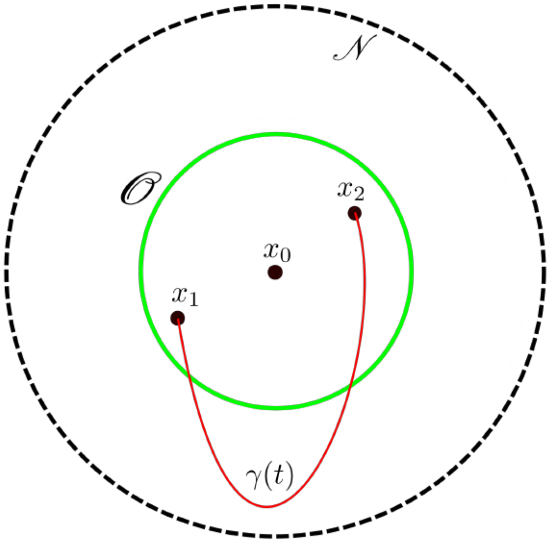


Figure 4.2: Pictorial Meaning of a Locally Controllability Curve $\boldsymbol{\gamma}(t)$

The condition $\Delta(\mathscr{F})_{\vec{\boldsymbol{x}}} = T_{\vec{\boldsymbol{x}}}\mathscr{M}$ is known as the Lie Algebra Rank Condition (LARC) and is equivalent to the Kalman rank condition for controllability in linear system theory. For a simply-connected solution manifold[3] , the notion of local controllability implies global controllability [12].

4.1.3 Controllability Analysis of the 6DoF Relative Motion

The controllability of the path planning problem was studied for an SRP-perturbed, drag-perturbed, and a drag-plus-SRP-perturbed environment, in that order.

(1) ***6DoF Controllability of the SRP-Perturbed Relative Dynamics***

Proposition 1 The SRP-perturbed relative dynamics problem is locally controllable every time the SRP acceleration on the flat plat is nonzero.

Proof To access the controllability of the solar sailing problem, the accessibility distribution is defined as follows:

$$\boldsymbol{\Delta} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, ad^1_{\boldsymbol{f}}\boldsymbol{b}_1, ad^1_{\boldsymbol{f}}\boldsymbol{b}_2, ad^1_{\boldsymbol{f}}\boldsymbol{b}_3, ad^2_{\boldsymbol{f}}\boldsymbol{b}_1, ad^2_{\boldsymbol{f}}\boldsymbol{b}_2, ad^2_{\boldsymbol{f}}\boldsymbol{b}_3, ad^3_{\boldsymbol{f}}\boldsymbol{b}_1, ad^3_{\boldsymbol{f}}\boldsymbol{b}_2, ad^3_{\boldsymbol{f}}\boldsymbol{b}_3] \tag{4.14}$$

where $ad^k_{\boldsymbol{f}}\boldsymbol{b}_i = [\boldsymbol{f}, ad^{k-1}_{\boldsymbol{f}}\boldsymbol{b}_i]$, $ad^1_{\boldsymbol{f}}\boldsymbol{b}_i = [\boldsymbol{f}, ad^1_{\boldsymbol{f}}\boldsymbol{b}_i] = \left[\frac{\partial \boldsymbol{f}}{\partial X}\right]\boldsymbol{b}_i - \left[\frac{\partial \boldsymbol{b}_i}{\partial X}\right]\boldsymbol{f}$, and the acceleration components of the vector field $\boldsymbol{f}$ does not contain the differential drag acceleration $\delta\vec{\boldsymbol{a}}^d$.

The Jacobian with respect to the state $\boldsymbol{X} = [\vec{\boldsymbol{\sigma}}, \vec{\boldsymbol{\omega}}, \vec{\boldsymbol{\rho}}, \vec{\boldsymbol{\rho}}/]^\intercal$ is given by

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{X}} = \begin{bmatrix} \frac{\partial}{\partial \vec{\boldsymbol{\sigma}}}\dot{\vec{\boldsymbol{\sigma}}} & \frac{\partial}{\partial \vec{\boldsymbol{\omega}}}\dot{\vec{\boldsymbol{\sigma}}} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \frac{\partial}{\partial \vec{\boldsymbol{\omega}}}\dot{\vec{\boldsymbol{\omega}}} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \frac{\partial}{\partial \vec{\boldsymbol{\rho}}'}\vec{\boldsymbol{\rho}}' \\ \frac{\partial}{\partial \vec{\boldsymbol{\sigma}}}\vec{\boldsymbol{\rho}}'' & 0_{3\times3} & \frac{\partial}{\partial \vec{\boldsymbol{\rho}}}\vec{\boldsymbol{\rho}}'' & \frac{\partial}{\partial \vec{\boldsymbol{\rho}}'}\vec{\boldsymbol{\rho}}'' \end{bmatrix} \tag{4.15}$$

The first order Lie bracket between the drift vector field $\boldsymbol{f}_d$ and the torque vector field $[\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3]$ is defined as follows:

$$[ad^1_{\boldsymbol{f}}\boldsymbol{b}_1, ad^1_{\boldsymbol{f}}\boldsymbol{b}_2, ad^1_{\boldsymbol{f}}\boldsymbol{b}_3]^\intercal = \begin{bmatrix} [\mathrm{I}]\frac{\partial}{\partial \vec{\boldsymbol{\omega}}}\dot{\vec{\boldsymbol{\sigma}}} & [\mathrm{I}]\frac{\partial}{\partial \vec{\boldsymbol{\omega}}}\dot{\vec{\boldsymbol{\omega}}} & 0_{3\times3} & 0_{3\times3} \end{bmatrix}^\intercal \tag{4.16}$$

[3] A solution manifold is to be simply connected if a path can connect every two points, and any closed path connecting any two points can be shrunk to a point while remaining in the manifold (i.e., the solution manifold does not have any holes).

The second order Lie bracket of the i^{th} torque vector field is given by

$$\begin{aligned} ad^2_{\boldsymbol{f}}\boldsymbol{b}_i = [\boldsymbol{f}, ad^1_{\boldsymbol{f}}\boldsymbol{b}_i] &= \left[\frac{\partial \boldsymbol{f}}{\partial X}\right] ad^1_{\boldsymbol{f}}\boldsymbol{b}_i - \left[\frac{\partial ad^1_{\boldsymbol{f}}\boldsymbol{b}_i}{\partial X}\right]\boldsymbol{f} \\ &= \begin{bmatrix} \frac{\partial}{\partial \vec{\boldsymbol{\sigma}}}\dot{\vec{\boldsymbol{\sigma}}} & \frac{\partial}{\partial \vec{\boldsymbol{\omega}}}\dot{\vec{\boldsymbol{\sigma}}} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \frac{\partial}{\partial \vec{\boldsymbol{\omega}}}\dot{\vec{\boldsymbol{\omega}}} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \frac{\partial}{\partial \vec{\boldsymbol{\rho}}'}\vec{\boldsymbol{\rho}}' \\ \frac{\partial}{\partial \vec{\boldsymbol{\sigma}}}\vec{\boldsymbol{\rho}}'' & 0_{3\times3} & \frac{\partial}{\partial \vec{\boldsymbol{\rho}}}\vec{\boldsymbol{\rho}}'' & \frac{\partial}{\partial \vec{\boldsymbol{\rho}}'}\vec{\boldsymbol{\rho}}'' \end{bmatrix} ad^1_{\boldsymbol{f}}\boldsymbol{b}_i - \begin{bmatrix} [-]_{3\times3} & 0_{3\times3} & 0_{3\times6} \\ 0_{3\times3} & [-]_{3\times3} & 0_{3\times6} \\ 0_{6\times3} & 0_{6\times3} & 0_{6\times6} \end{bmatrix} \boldsymbol{f} \\ &= \begin{bmatrix} [-]_{3\times1} \\ [-]_{3\times1} \\ 0_{3\times1} \\ \left[\frac{\partial \delta \vec{\boldsymbol{a}}^s}{\partial \vec{\boldsymbol{\sigma}}}\right] ad^1_{\boldsymbol{f}}\boldsymbol{b}_i \end{bmatrix} \end{aligned} \tag{4.17}$$

where [−] s a nonzero matrix of appropriate size, and each instance of these quantities is completely unrelated to the other. The second order Lie bracket can be consolidated as

$$\left[ad^2_{\boldsymbol{f}}\boldsymbol{b}_1, ad^2_{\boldsymbol{f}}\boldsymbol{b}_2, ad^2_{\boldsymbol{f}}\boldsymbol{b}_3\right]^{\mathsf{T}} = \left[[-]_{3\times3} \quad [-]_{3\times3} \quad 0_{3\times3} \quad [\mathrm{I}]\left[\frac{\partial \delta \vec{\boldsymbol{a}}^s}{\partial \vec{\boldsymbol{\sigma}}}\right]\left[\frac{\partial \dot{\vec{\boldsymbol{\sigma}}}}{\partial \vec{\boldsymbol{\omega}}}\right]\right]^{\mathsf{T}} \tag{4.18}$$

Continuing on the same path, one can show that the third order Lie bracket due to torque vector field is defined as

$$\left[ad^3_{\boldsymbol{f}}\boldsymbol{b}_1, ad^3_{\boldsymbol{f}}\boldsymbol{b}_2, ad^3_{\boldsymbol{f}}\boldsymbol{b}_3\right]^{\mathsf{T}} = \left[[-]_{3\times3} \quad [-]_{3\times3} \quad [\mathrm{I}]\left[\frac{\partial \delta \vec{\boldsymbol{a}}^s}{\partial \vec{\boldsymbol{\sigma}}}\right]\left[\frac{\partial \dot{\vec{\boldsymbol{\sigma}}}}{\partial \vec{\boldsymbol{\omega}}}\right] \quad [-]_{3\times3}\right]^{\mathsf{T}} \tag{4.19}$$

where [−] s is a nonzero matrix of appropriate size, and each instance of these quantities is entirely unrelated to the other.

The distribution matrix in (4.14) is represented more succinctly as following expression:

$$\boldsymbol{\Delta} = \begin{bmatrix} 0_{3\times3} & \Delta_{12} & \Delta_{13} & \Delta_{14} \\ \Delta_{21} & \Delta_{22} & \Delta_{23} & \Delta_{24} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \Delta_{34} \\ 0_{3\times3} & 0_{3\times3} & \Delta_{43} & \Delta_{44} \end{bmatrix} \tag{4.20}$$

where the sub-matrix Δ_{ij} is a 3×3 fully populated matrix. The rows of $\boldsymbol{\Delta}$ can be rearranged to produce the following upper-block matrix:

$$\boldsymbol{\Phi} = \begin{bmatrix} \Delta_{21} & \Delta_{22} & \Delta_{23} & \Delta_{24} \\ 0_{3\times3} & \Delta_{12} & \Delta_{13} & \Delta_{14} \\ 0_{3\times3} & 0_{3\times3} & \Delta_{43} & \Delta_{44} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \Delta_{34} \end{bmatrix} \tag{4.21}$$

A sufficient condition for a general upper-block matrix to be full rank is that its subdiagonal matrices have to be invertible [26]. In the case of the matrix $\boldsymbol{\Phi}$, all sub-blocks are 3 by 3. It is worth mentioning that due to the sparse structure of the Jacobian $\left[\frac{\partial \boldsymbol{f}}{\partial X}\right]$, $\Delta_{43} = \Delta_{34}$. The rank of the accessibility distribution is given by

$$\boxed{rank(\boldsymbol{\Delta}) = rank(\boldsymbol{\Phi}) = rank(\Delta_{21}) + rank(\Delta_{12}) + 2\ rank(\Delta_{43}) = n = 12 \text{ if } \vec{\boldsymbol{a}}^{s}_{B} \neq \mathbf{0}}$$

Where the ranks of the submatrices Δ_{21}, Δ_{12}, Δ_{43} are studied in Appendix D. ▮

(2) *6DoF Controllability of the Drag-Perturbed Relative Dynamics*

Proposition 2 The drag-perturbed relative dynamics problem is not locally controllable.

Proof The proof of Proposition 2 follows that of Proposition 1. In this instance, the vector field $\boldsymbol{f}$ does not contain the differential SRP acceleration $\delta\vec{\boldsymbol{a}}^{s}$. The Jacobian w.r.t to the state $\boldsymbol{X} = [\vec{\boldsymbol{\sigma}}, \vec{\boldsymbol{\omega}}, \vec{\boldsymbol{\rho}}, \vec{\boldsymbol{\rho}}\prime]^{\intercal}$ is given by

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{X}} = \begin{bmatrix} \frac{\partial}{\partial \vec{\boldsymbol{\sigma}}}\dot{\vec{\boldsymbol{\sigma}}} & \frac{\partial}{\partial \vec{\boldsymbol{\omega}}}\dot{\vec{\boldsymbol{\sigma}}} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \frac{\partial}{\partial \vec{\boldsymbol{\omega}}}\dot{\vec{\boldsymbol{\omega}}} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \frac{\partial}{\partial \vec{\boldsymbol{\rho}}'}\vec{\boldsymbol{\rho}}' \\ \frac{\partial}{\partial \vec{\boldsymbol{\sigma}}}\vec{\boldsymbol{\rho}}'' & 0_{3\times3} & \frac{\partial}{\partial \vec{\boldsymbol{\rho}}}\vec{\boldsymbol{\rho}}'' & \frac{\partial}{\partial \vec{\boldsymbol{\rho}}'}\vec{\boldsymbol{\rho}}'' \end{bmatrix} \tag{4.22}$$

The second and third order Lie bracket in this instance are given by the following expressions:

$$\left[ad^2_{\boldsymbol{f}}\boldsymbol{b}_1, ad^2_{\boldsymbol{f}}\boldsymbol{b}_2, ad^2_{\boldsymbol{f}}\boldsymbol{b}_3\right]^{\intercal} = \left[[-]_{3\times3} \quad [-]_{3\times3} \quad 0_{3\times3} \quad [\mathrm{I}]\left[\frac{\partial\delta\vec{\boldsymbol{a}}^{d}}{\partial\vec{\boldsymbol{\sigma}}}\right]\left[\frac{\partial\dot{\vec{\boldsymbol{\sigma}}}}{\partial\vec{\boldsymbol{\omega}}}\right]\right]^{\intercal} \tag{4.23}$$

$$\left[ad^3_{\boldsymbol{f}}\boldsymbol{b}_1, ad^3_{\boldsymbol{f}}\boldsymbol{b}_2, ad^3_{\boldsymbol{f}}\boldsymbol{b}_3\right]^{\intercal} = \left[[-]_{3\times 3} \quad [-]_{3\times 3} \quad [\mathrm{I}]\left[\frac{\partial \delta \vec{\boldsymbol{a}}^{d}}{\partial \vec{\boldsymbol{\sigma}}}\right]\left[\frac{\partial \dot{\vec{\boldsymbol{\sigma}}}}{\partial \vec{\boldsymbol{\omega}}}\right] \quad [-]_{3\times 3}\right]^{\intercal} \tag{4.24}$$

The partial $\left[\frac{\partial \delta \vec{\boldsymbol{a}}^{d}}{\partial \vec{\boldsymbol{\sigma}}}\right] = -\frac{C_d \beta A_d}{2}\left(\delta \vec{\boldsymbol{v}}_{st}^{\intercal} \delta \vec{\boldsymbol{v}}_{st}\right)^{\frac{1}{2}}\left(\delta \vec{\boldsymbol{v}}_{st} \delta \vec{\boldsymbol{v}}_{st}^{\intercal}\right) \frac{\partial \hat{\boldsymbol{n}}_3}{\partial \vec{\boldsymbol{\sigma}}}$, and the span of its eigenspace is 1 because rank $\left(\left[\delta \vec{\boldsymbol{v}}_{st} \delta \vec{\boldsymbol{v}}_{st}^{\intercal}\right]\right) = 1$. By reconstructing the upper diagonal matrix $\boldsymbol{\Phi}$, it is trivial to see that

$$\boxed{rank(\boldsymbol{\Delta}) = rank(\boldsymbol{\Phi}) \neq n \text{ because } rank(\Delta_{43}) = rank(\Delta_{34}) = 1}$$

This result implies that the drag alone cannot be used to connect arbitrarily chosen boundary conditions. $\blacksquare$

(3) *6DoF Controllability of the Drag-Plus-SRP-perturbed Relative Dynamics*

Proposition 3 The drag-plus-SRP-perturbed relative dynamics problem is locally controllable.

Proof Following the same derivation process, the submatrices $\Delta_{43} = \Delta_{34} = [\mathrm{I}]\left[\frac{\partial \delta \vec{\boldsymbol{a}}^{d}}{\partial \vec{\boldsymbol{\sigma}}} + \frac{\partial \delta \vec{\boldsymbol{a}}^{s}}{\partial \vec{\boldsymbol{\sigma}}}\right]\left[\frac{\partial \dot{\vec{\boldsymbol{\sigma}}}}{\partial \vec{\boldsymbol{\omega}}}\right]$, and

$$\boxed{rank(\boldsymbol{\Delta}) = rank(\boldsymbol{\Phi}) = n \text{ if } \vec{\boldsymbol{a}}^{s}_{B} \neq \boldsymbol{0}}$$

$\blacksquare$

4.2 Simulation, Results and Discussions

A numerical analysis was conducted for each of the two controllable cases discussed above to ascertain the claims made in Section 4.1. Namely, the following section demonstrates a Medium Earth Orbit (MEO) orbital transfer using an SRP (where the effects drag are negligible), and a Low Earth Orbit (LEO) orbital transfer in a drag-plus-SRP perturbed environment. The deputy attitude description is switched from MRP (used in the controllability analysis) to quaternions (used in the simulations) because quaternions' continuity is better suited to the ODE integrator used in this analysis. The attitude computed by the geometric guidance is tracked, and no additional propulsion capability (other than SRP acceleration) is used to track the resulting position and velocity. The generated paths are therefore fuel-free. The feedback steering control is developed using Lyapunov theory and is given by the expression below [177]:

$$\vec{\boldsymbol{\tau}} = -k\vec{\boldsymbol{\epsilon}} - [\mathbf{P}]\delta\vec{\boldsymbol{\omega}} + [\mathbf{I}]\left(\dot{\vec{\boldsymbol{\omega}}}_r - [\tilde{\boldsymbol{\omega}}]\boldsymbol{\omega}_r\right) + [\tilde{\boldsymbol{\omega}}][\mathbf{I}]\boldsymbol{\omega} \tag{4.25}$$

where the parameter k is a positive gain, $\vec{\boldsymbol{\epsilon}} = [q_1,\ q_2,\ q_3]^\intercal$is the quaternion error, $[\mathbf{P}]$ is the (positive definite) gain matrix. The body-frame angular velocity of the flat plate is given by $\boldsymbol{\omega}$ while $\boldsymbol{\omega}_r = \vec{\boldsymbol{\omega}}_{rel}$ is the optimal angular velocity computed by the guidance algorithm. $\delta\vec{\boldsymbol{\omega}} = \boldsymbol{\omega} - \boldsymbol{\omega}_r$ is the angular velocity error, and $\dot{\vec{\boldsymbol{\omega}}}_r = -[\mathbf{I}]^{-1}\left([\tilde{\boldsymbol{\omega}}][\mathbf{I}]\boldsymbol{\omega}_r\right)$.

The initial orbital elements of the chief's trajectory are given in Table 4.2, whereas the deputy's initial and final relative orbital elements are given in Table 4.3. The initial and final conditions for the plate attitude and angular velocity are given in Table 4.4. The deputy mass was $m = 4$kg. These parameters are the same for both simulations.

Table 4.2: Reference Orbit in Orbital Elements

Initial OE	a [km]	e	i [rad]	Ω [rad]	ω [rad]	f [rad]
Reference	2e4	0.5	50	10	10	0.0

Table 4.3: Initial differential orbital elements

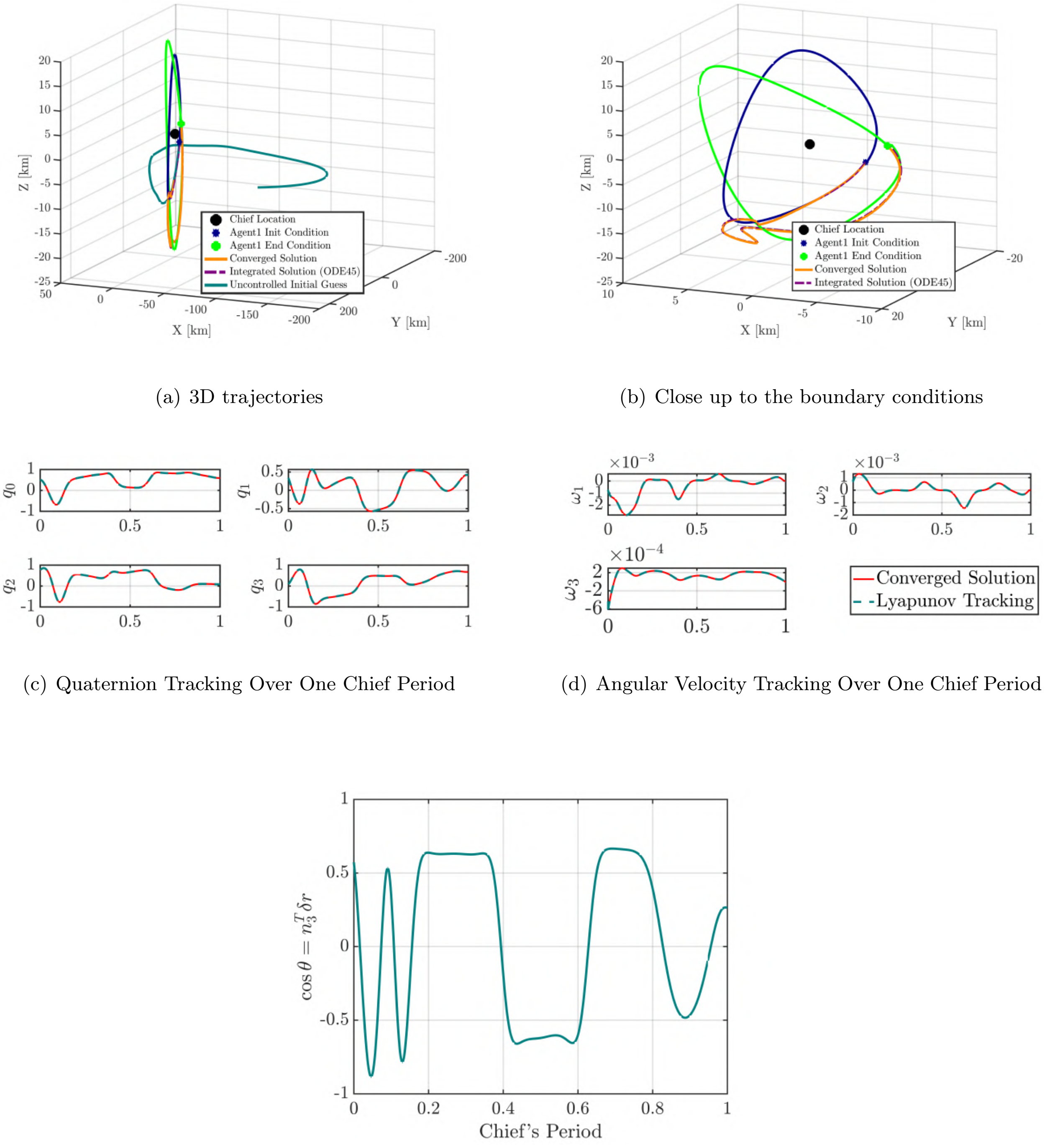
ΔOE	δa [km]	δe	δi [rad]	$\delta\Omega$ [rad]	$\delta\omega$ [rad]	δf [rad]
Deputy1 (initial)	0.0	$1/(5a)$	$-50/(3a)$	0.0	$\pi \times 1\text{e}{-4}$	$\pi \times 1\text{e}{-4}$
Deputy1 (final)	0.0	$20/(3a)$	$25/a$	$-2 * \pi \times 1\text{e}{-6}$	$-5 * \pi \times 1\text{e}{-6}$	$\pi \times 1\text{e}{-4}$

Table 4.4: Boundary Conditions for the Deputy's Attitude and Angular Velocity

q_{t_0} (Not normalized)	ω_{t_0}	q_{t_f} (Not normalized)	ω_{t_0}
$[6, 4, 9, 1]$	$[-45, 40, -35] \times 1\text{e}{-3}$	$[9, 5, 1, 8]$	$[-9, 9, -3] \times 1\text{e}{-4}$

4.2.1 Fuel-Free SRP-Perturbed 6DoF Orbital Transfer at MEO

In this scenario, the deputy is modeled as a square with side $l = 10$m. The transfer time was set to one chief's orbital period. Figures 4.3(a) and 4.3(b) show 3D renderings of the initial guess and the converged solution to the orbital transfer studied in this work. The initial and final parking orbits (green and blue elliptic relative curves) are computed without incorporating the effect of SRP. They are only added here to aid conveying the idea behind the simulation. The uncontrolled trajectory of the deputy is shown in Fig. 4.3(a). The attitude of the deputy spacecraft is represented in Fig. 4.3(c). The sun's position changes in the deputy body frame throughout the transfer, and Fig. 4.3(e) shows the pointing of the plate's normal relative to the Sun's position. This metric is used to assess the controllability of the system throughout the transfer. Although the system is not controllable when the plate rotates edge to the sun (i.e., $\cos(\theta) = \boldsymbol{n}_3^\intercal \delta \boldsymbol{r} = 0$), the problem remains controllable long enough to allow the path planner to generate a trajectory that connects the boundaries conditions of the problem when using solar radiation pressure.



(a) 3D trajectories

(b) Close up to the boundary conditions

(c) Quaternion Tracking Over One Chief Period

(d) Angular Velocity Tracking Over One Chief Period

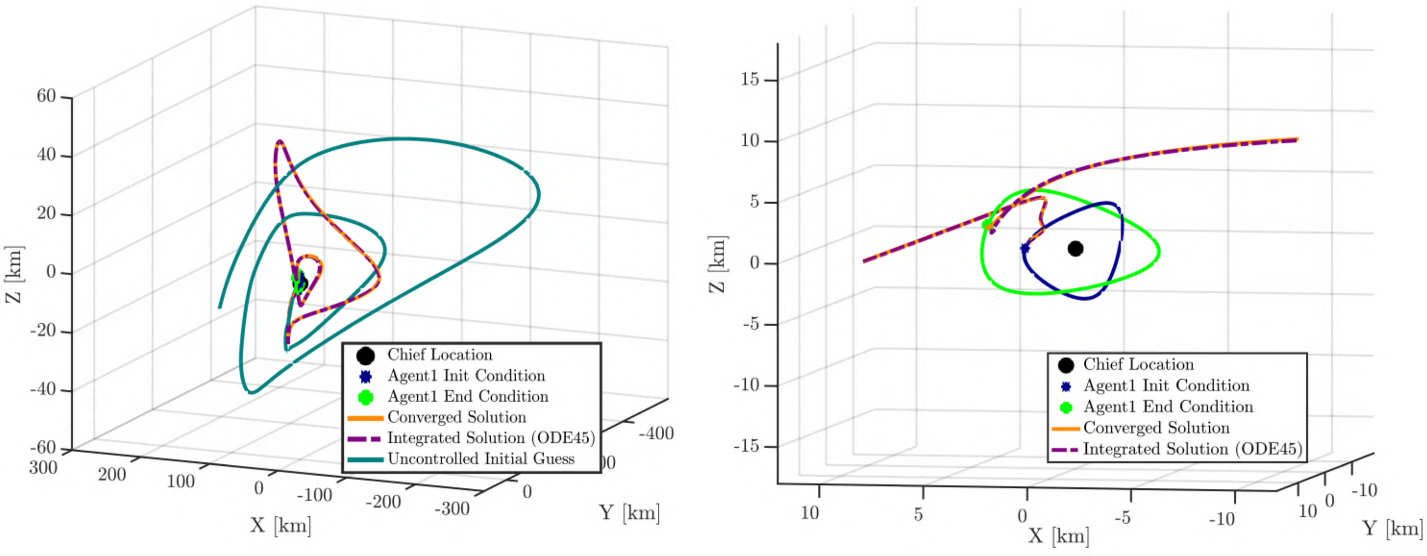
(e) Cosine of the Conning Angle

Figure 4.3: Path Planning in an SRP-Perturbed Environment

4.2.2 Fuel-Free Drag-SRP-Perturbed 6DoF Orbital Transfer at LEO

The analysis conducted above is extended to a Low Earth Orbit transfer. The chief parameters are the same as in Table 4.2 except for the semi-major axis which is brought down to $a = 3.689\text{e}3\text{km}$ (i.e., the periapsis altitude is $h = 1000\text{km}$). The boundary conditions of the deputy are kept the same as in Tables 4.3 and 4.4. However, the flat plate is modeled as a square with side $l = 50\text{m}$, and the transfer time is set to three times the chief's orbital period. It should be noted that a deputy with a smaller area-to-mass ratio can still complete this transfer, but it will do so in a longer time of flight than prescribed in this simulation.

Fig. 4.4(a) shows the trajectory computed by the geometric guidance algorithm. The uncontrolled trajectory is shown in green and can be seen to escape because of drag and SRP's effects. The optimal solution computed by the geometric planner is shown in orange. The dashed-magenta line is the trajectory the deputy follows by tracking the attitude of the optimal trajectory. Fig. 4.4(b) shows a close-up on the boundary conditions, confirming that the computed trajectory meets the boundary conditions despite the relatively large detour it takes. The trajectory traveled by the deputy is long because the SRP acceleration is small compared to drag; therefore, the planner computes a path that minimizes the effect of drag which removes energy in the system. Going lower in the altitude requires an exponentially larger plate size because the atmospheric density increases as the periapsis radius decreases. Still, there exists an altitude at which the effect of drag is just too large to overcome with SRP only. Determining that altitude and the minimum time of flight (for a given transfer) are still open-ended questions at this time. Nonetheless, the presented fuel-free guidance can allow a flat-plate servicer spacecraft to bring defunct spacecraft to reentering orbit and go back to higher altitude by solely using the acceleration provided by SRP. By being fuel-free, such missions would be a sustainable way to address the space debris problem.



(a) 3D trajectories

(b) Close up to the boundary conditions

Figure 4.4: Path Planning in a Drag-SRP-Perturbed Environment

4.3 Conclusion

The 6DoF nonlinear guidance proposed in this analysis allows performing fuel-free orbital transfers using natural forces. The algorithm performance was assessed on fuel-free transfer where a flat plate deputy used differential solar radiation pressure to travel around a cannonball chief. The problem was shown to be controllable using Lie bracket analysis. Then the path planning problem was reframed into a gradient descent problem by leveraging ideas from Riemannian geometry and the calculus of variations. An initial guess is iteratively updated to minimize a given cost function. Simulation results show that the computed path allows the deputy to reach the specified boundary conditions at the specified time.

Chapter 5

Distributed Estimator For Spacecraft Cooperative Localization

5.1 Introduction

This chapter briefly introduces a novel relative navigation algorithm developed for this dissertation. The developed filter allows each deputy spacecraft to navigate using relative measurements between themselves. Each filter runs independently, and communication among the agents allows to estimate the states of the whole formation conjointly. The UIF algorithm was chosen because it is a hybrid between the unscented Kalman filter and the Fisher Information filter. The time update of the UIF uses an unscented transform to accurately capture the moment parameters (mean and associated covariance) of the estimated process. The time history of the chief spacecraft is assumed to be known by all the deputies. Its states are treated as deterministic parameters in each distributed filter implementation. The distributed navigation algorithm aims to compute the deputy's relative states about a known chief's solution using the relative measurements (taken by the other agents in the formation). The proposed method is a stable-distributed nonlinear filter obtained by combining the UIF and CI algorithms. The resulting distributed filter is agnostic to the constellation's size and robust to the introduction/removal of agents at any time (i.e., an ad hoc network). The remainder of this chapter is organized as follows: In Section 5.2, the distributed estimator is derived. In Section 5.3, The estimator's performance is investigated with a five-spacecraft formation. We conclude the chapter in Section 5.4 with a summary of the presented work.

5.2 Distributed Navigation Algorithm

5.2.1 Fisher Information and Covariance Matrix

The Unscented Information Filter (UIF) is a variation of the Unscented Kalman Filter [205] which relies on unscented transforms to deal with nonlinearities in the dynamics and measurements. The UIF operates on the Fisher information and uses a nonlinear transform to approximate the resulting distribution as a Gaussian distribution. Let $\mathbf{x} \in \mathbb{R}^n$ be the Gaussian random vector to be estimated. Let's assume that all realizations of $\mathbf{x}$ can be the represented using a multivariate normal distribution $p(\mathbf{x}) \sim \mathscr{N}\left(\hat{\mathbf{x}}, \mathbf{P}\right)$ with mean $\hat{\mathbf{x}}$ and a $n \times n$ variance-covariance matrix $\mathbf{P}$. Consider the dynamics of $\mathbf{x}$ modeled as a discrete-time nonlinear process and measured by m sensors nodes:

$$\mathbf{x}_{k+1} = \mathbf{f}\left(\mathbf{x}_k, \mathbf{w}_k\right) \tag{5.1}$$

$$\mathbf{z}_k = \mathbf{h}\left(\mathbf{x}_k\right) + \mathbf{v}_k \tag{5.2}$$

where $\mathbf{z}_k \in \mathbb{R}^m$ is the measurement vector, $\mathbf{w}_k \in \mathbb{R}^n \sim \mathscr{N}\left(0, \mathbf{Q}_k\right)$ and $\mathbf{v}_k \in \mathbb{R}^m \sim \mathscr{N}\left(0, \mathbf{R}_k\right)$ represents zero-mean white Gaussian processes. During the time update step, the UIF uses $2n+1$ weighted sample points (sigma points) drawn from the normal distribution $\mathscr{N}\left(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}\right)$ and propagates them through the the nonlinear dynamics. At the end of the propagation, the sigma points resulting from the mapping are used for the computation of a marginal probability distribution $\mathscr{N}\left(\bar{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}\right)$, which is the a-priori distribution at time k. Once a measurement is available, it is used to update the a-priori state by calculating a conditional probability distribution which best explains that measurement. The UIF leverages the duality of a Gaussian distribution to operate on the information. The probability density function (PDF) of such a Gaussian distribution is:

$$\begin{aligned} p(\boldsymbol{x}) &= \mathscr{N}\left(\hat{\mathbf{x}}, \mathbf{P}\right) \\ &= \frac{1}{\sqrt{|2\pi\mathbf{P}|}} \exp\Big\{ -\frac{1}{2}\left(\mathbf{x}-\hat{\mathbf{x}}\right)^{\intercal}\mathbf{P}^{-1}\left(\mathbf{x}-\hat{\mathbf{x}}\right)\Big\} \\ &= \frac{e^{-\frac{1}{2}\mathbf{y}^{\intercal}\mathbf{Y}^{-1}\mathbf{y}}}{\sqrt{|2\pi\mathbf{Y}^{-1}|}} \exp\Big\{ -\frac{1}{2}\mathbf{x}^{\intercal}\mathbf{Y}\mathbf{x} + \mathbf{y}^{\intercal}\mathbf{x}\Big\} \\ &= \mathscr{N}^{-1}\left(\hat{\mathbf{y}}, \mathbf{Y}\right) \end{aligned} \tag{5.3}$$

where $\mathbf{Y}$ is the Fisher information matrix and $\mathbf{y}$ is the information vector. In Eq. (5.3), the information parametrization of the Gaussian distribution is equivalent to the moment (covariance) parametrization, and these two forms represent the duality of the Gaussian distribution. At every time step k, one can go back and forth between them using the following mapping:

$$\mathbf{Y}_{k|k} = \mathbf{P}_{k|k}^{-1} \tag{5.4}$$

$$\mathbf{y}_k = \mathbf{P}_{k|k}^{-1}\mathbf{x}_k = \mathbf{Y}_{k|k}\mathbf{x}_k \tag{5.5}$$

If one assumes that both the state and the measurement are Gaussian distributions

$$p(\boldsymbol{x}, \boldsymbol{z}) = \mathscr{N}\left(\begin{bmatrix}\hat{\boldsymbol{x}}_{\mathrm{x}} \\ \hat{\boldsymbol{x}}_{\mathrm{z}}\end{bmatrix}, \begin{bmatrix}\mathbf{P}_{\mathrm{x,x}} & \mathbf{P}_{\mathrm{x,z}} \\ \mathbf{P}_{\mathrm{z,x}} & \mathbf{P}_{\mathrm{z,z}}\end{bmatrix}\right) = \mathscr{N}^{-1}\left(\begin{bmatrix}\hat{\boldsymbol{y}}_{\mathrm{x}} \\ \hat{\boldsymbol{y}}_{\mathrm{z}}\end{bmatrix}, \begin{bmatrix}\mathbf{Y}_{\mathrm{x,x}} & \mathbf{Y}_{\mathrm{x,z}} \\ \mathbf{Y}_{\mathrm{z,x}} & \mathbf{Y}_{\mathrm{z,z}}\end{bmatrix}\right) \tag{5.6}$$

For both the moment and information forms, the time and measurement update steps can be represented in a compact form [72], as shown in Table 5.1. Here, it can be seen that the marginal probability distribution is easily computed in the moment form. On the other hand, the conditional probability distribution is readily implemented in the information form. Consequently, the proposed algorithm performs the time updates in the moment form and the measurement updates in the information form.

Table 5.1: Relationship between the conditional and marginal probabilities

Parameter Form	Marginal Probability	Conditional Probability
Moment parameter	$\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_{\mathrm{x}}$ $\mathbf{P} = \mathbf{P}_{\mathrm{x,x}}$	$\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_{\mathrm{x}} + \mathbf{P}_{\mathrm{x,z}}\mathbf{P}_{\mathrm{z,z}}^{-1}(\boldsymbol{Z} - \hat{\boldsymbol{x}}_{\mathrm{z}})$ $\mathbf{P} = \mathbf{P}_{\mathrm{x,x}} - \mathbf{P}_{\mathrm{x,z}}\mathbf{P}_{\mathrm{z,z}}^{-1}\mathbf{P}_{\mathrm{z,x}}$
Information parameter	$\hat{\boldsymbol{y}} = \hat{\boldsymbol{y}}_{\mathrm{x}} - \mathbf{Y}_{\mathrm{x,z}}\mathbf{Y}_{\mathrm{z,z}}^{-1}\hat{\boldsymbol{y}}_{\mathrm{z}}$ $\mathbf{Y} = \mathbf{Y}_{\mathrm{x,x}} - \mathbf{Y}_{\mathrm{x,z}}\mathbf{Y}_{\mathrm{z,z}}^{-1}\mathbf{Y}_{\mathrm{z,x}}$	$\hat{\boldsymbol{y}} = \mathbf{y}_{\mathrm{x}} - \mathbf{Y}_{\mathrm{x,z}}\boldsymbol{Z}$ $\mathbf{Y} = \mathbf{Y}_{\mathrm{x,x}}$

5.2.1.1 Filter Time Update

The $2n + 1$ sigma points are obtained from the columns of the following augmented matrix:

$$\mathcal{X}_k = \left[\ \hat{\mathbf{x}}_{k|k},\ \hat{\mathbf{x}}_{k|k} + \sqrt{(n+\lambda)\mathbf{P}_{k|k}},\ \hat{\mathbf{x}}_{k|k} - \sqrt{(n+\lambda)\mathbf{P}_{k|k}}\ \right] \tag{5.7}$$

where n is the number of states, $\lambda = \alpha^2(n+\kappa) - n$, $\kappa = 3 - n$, $\alpha < 1$. The operation $\hat{\mathbf{x}}_{k|k} + \sqrt{(n+\lambda)\mathbf{P}_{k|k}}$ returns an $n \times n$ matrix and is understood as the summation of the mean $\hat{\mathbf{x}}_{k|k}$ and each column of $\sqrt{(n+\lambda)P}$. The sigma points are then propagated forward in time through the system's non-linear dynamics, as shown below:

$$\mathcal{X}_{k|k-1} = \mathbf{f}\left(\mathcal{X}_{k-1}, \mathbf{w}_{k-1}\right) \tag{5.8}$$

The state estimate resulting from the time propagation step is calculated as a weighted average of the resulting sigma points. The covariance associated with this estimate is a measure of how much the sigma points deviate from that average value. Both the a-priori state estimate and the a-priori covariance expressed as

$$\bar{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n} \mathscr{W}_i^m \mathcal{X}_{i,k-1} \tag{5.9}$$

$$\mathbf{P}_{k|k-1} = \mathbf{Q}_k + \sum_{i=0}^{2n} \mathscr{W}_i^c \left(\mathcal{X}_{i,k-1} - \bar{\mathbf{x}}_{k|k-1}\right)\left(\mathcal{X}_{i,k-1} - \bar{\mathbf{x}}_{k|k-1}\right)^{\mathsf{T}} \tag{5.10}$$

where

$$\begin{cases} \mathscr{W}_0^m = \dfrac{\lambda}{n+\lambda} \\ \mathscr{W}_0^c = \dfrac{\lambda}{n+\lambda} + (1-\alpha^2+\beta) \\ \mathscr{W}_i^m = \mathscr{W}_i^m = \dfrac{\lambda}{2(n+\lambda)} \end{cases}$$

5.2.1.2 Filter Measurement Update

Once a measurement is received, the information it contains needs to be fused with any a-priori knowledge of the system's states to improve the quality of the estimates. Given a-priori estimate (i.e., $\bar{\mathbf{x}}_{k|k-1}$, $\mathbf{P}_{k|k-1}$), the $2n+1$ sigma points are computed at epoch $k|k-1$ as follows:

$$\mathcal{X}_{k|k-1} = \left[\ \bar{\mathbf{x}}_{k|k-1},\ \bar{\mathbf{x}}_{k|k-1} + \sqrt{(n+\lambda)\mathbf{P}_{k|k-1}},\ \bar{\mathbf{x}}_{k|k-1} - \sqrt{(n+\lambda)\mathbf{P}_{k|k-1}}\ \right] \tag{5.11}$$

where $\bar{\mathbf{x}}_{k|k-1} + \sqrt{(n+\lambda)\mathbf{P}_{k|k-1}}$ returns an $n \times n$ matrix and is undertood as the summation of the a-priori estimate $\bar{\mathbf{x}}_{k|k-1}$ and and each column of the matrix $\sqrt{(n+\lambda)\mathbf{P}_{k|k-1}}$

The expected measurement is computed as a weighted average of the measurements associated with each sigma point, as follows:

$$\mathcal{Z}_{k|k-1} = \mathbf{h}\left(\mathcal{X}_{k|k-1}\right) \tag{5.12}$$

$$\bar{\mathbf{z}}_k = \sum_{i=0}^{2n} \mathcal{W}_i^m \mathcal{Z}_{i,k|k-1} \tag{5.13}$$

The cross-correlation covariance between the a-priori state and the measurement is

$$\mathbf{P}_{xz} = \sum_{i=0}^{2n} \mathcal{W}_i^c \left(\mathcal{X}_{i,k-1} - \bar{\mathbf{x}}_{k|k-1}\right)\left(\mathcal{Z}_{i,k|k-1} - \bar{\mathbf{z}}_k\right)^{\intercal} \tag{5.14}$$

Assuming that m sensor measurements are received at epoch k, all the available information (both from the time update and the measurement update) can be added together into an information vector, and information matrix respectively expressed as

$$\mathbf{y}_k = \mathbf{y}_{k-1} + \mathbf{i}_k = \mathbf{P}_{k|k-1}^{-1}\mathbf{x}_{k|k-1} + \sum_{j=1}^{m} \mathcal{H}_{j,k}^{\intercal}\mathbf{R}_{j,k}^{-1}\left[\nu_{j,k} + \mathcal{H}_{j,k}\bar{\mathbf{x}}_{k|k-1}\right] \tag{5.15}$$

$$\mathbf{Y}_{k|k} = \mathbf{Y}_{k|k-1} + \mathbf{I}_k = \mathbf{P}_{xk|k-1}^{-1} + \sum_{j=1}^{m} \mathcal{H}_{j,k}^{\intercal}\mathbf{R}_{j,k}^{-1}\mathcal{H}_{j,k} \tag{5.16}$$

where $\mathcal{H}_{j,k} = \left(\mathbf{P}_{k|k-1}^{-1}\mathbf{P}_{j,xz}\right)^{\intercal}$ is the j^{th} sensor's measurement matrix,which is the sensitivity matrix mapping errors in the states to errors in the measurements. $\mathbf{i}_k$ and $\mathbf{I}_k$ are information contained in a new measurement at time k. The vector $\nu_{j,k} = \mathbf{z}_{j,k} - \bar{\mathbf{z}}_k$ is the error between the j^{th} sensor's measurement $\mathbf{z}_{j,k}$ and the expected measurement $\bar{\mathbf{z}}_k = \mathbf{h}\left(\bar{\boldsymbol{x}}_{k|k-1}\right)$ [117]. The time-update and measurement-update processes constitute the essence of the UIF. One of the major appeals of this navigation algorithm is capturing the nonlinearity of the SFF problem via the unscented transform. Another appeal in the UIF is embedded in the additivity of information. In a fully distributed estimation algorithm, this property allows any given node to quickly accumulate the relevant information from the messages communicated among nodes. It enables the information collected from different sensors to be accumulated and fused with the a-priori knowledge to generate a better estimate.

Ultimately, one would need to know the cross-correlation between sensors to generate the linear unbiased estimate (i.e., the estimate generated from a centralized filter). Tracking each spacecraft in a formation can be computationally expensive or intractable for large constellations. Therefore, it is desirable to opt for a fusing algorithm that provides a consistent estimate of every spacecraft's states. Covariance Intersection (CI) is used to combine the information matrices and vectors originating from the m sensor measurements in Eqs.(5.15) and (5.16). This approach allows spacecraft in the formation to only keep track of their state/covariances and update their state knowledge using relative measurements between them. The CI algorithm is further discussed in the subsequent section.

5.2.2 Covariance Intersection

In the study of cooperative localization, it is customary to deal with the problem of "*rumor propagation*"[16]. *Rumor propagation* occurs when correlated measurements between agents are treated as independent, causing information to be fictitiously created and added to the network. This phenomenon frequently occurs in dynamic networks where the connections between nodes are made ad hoc. If left unaddressed, it leads to the double-counting of information and inconsistent filter estimates [37]. A covariance intersection (CI) scheme was introduced in the construction of the algorithm to remedy this issue when the cross-correlation between filters' estimates is unknown. CI is a data fusion algorithm that combines two pieces of information with unknown cross-correlation while avoiding the problem of rumor propagation.

5.2.2.1 Data Fusion For Estimates With Known Cross-Correlation

Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be two pieces of information from two distinct sensors about a common phenomenon. Let $\bar{\boldsymbol{a}}$ and $\bar{\boldsymbol{b}}$ be estimates of $\boldsymbol{a}$ and $\boldsymbol{b}$, and let the statistics of these estimates be given by

$$\boldsymbol{E}[\tilde{a}] = 0, \boldsymbol{E}[\tilde{a}\tilde{a}^{\mathsf{T}}] = \bar{\boldsymbol{P}}_{aa} \tag{5.17}$$

$$\boldsymbol{E}[\tilde{b}] = 0, \boldsymbol{E}[\tilde{b}\tilde{b}^{\mathsf{T}}] = \bar{\boldsymbol{P}}_{bb} \tag{5.18}$$

where $\tilde{a} = \boldsymbol{a} - \bar{\boldsymbol{a}}$ and $\tilde{b} = \boldsymbol{b} - \bar{\boldsymbol{b}}$ are the errors in the estimates. Assume that the true value of the covariance matrices $\bar{\boldsymbol{P}}_{aa}$ and $\bar{\boldsymbol{P}}_{bb}$ are unknown, but there exists two known constant matrices $\boldsymbol{P}_{aa}$ and $\boldsymbol{P}_{bb}$ that bound $\bar{\boldsymbol{P}}_{aa}$ and $\bar{\boldsymbol{P}}_{bb}$ from above, as shown below

$$\exists \boldsymbol{P}_{aa} \mid \boldsymbol{P}_{aa} \geq \bar{\boldsymbol{P}}_{aa} \tag{5.19}$$

$$\exists \boldsymbol{P}_{bb} \mid \boldsymbol{P}_{bb} \geq \bar{\boldsymbol{P}}_{bb} \tag{5.20}$$

The inequalities in (5.19) and (5.20) are in the sense of positive definiteness of a matrix (i.e., given two $n \times n$ matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, $\boldsymbol{A} \geq \boldsymbol{B} \Leftrightarrow \forall \boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{x}^{\mathsf{T}} \left(\boldsymbol{A} - \boldsymbol{B}\right) \boldsymbol{x} \geq 0$).

The goal is to generate a new and consistent estimate $\boldsymbol{c}$ which fuses together the information contained in both $\boldsymbol{a}$ and $\boldsymbol{b}$. Let the cross-correlation between two estimates $\bar{\boldsymbol{a}}$ and $\bar{\boldsymbol{b}}$ be defined as

$$\boldsymbol{E}[\tilde{a}\tilde{b}^{\mathsf{T}}] = \bar{\boldsymbol{P}}_{ab} \tag{5.21}$$

and assume $\bar{\boldsymbol{P}}_{ab}$ to be known. One can construct a linear, unbiased estimator $\boldsymbol{c}$ as a convex combination of $\boldsymbol{a}$ and $\boldsymbol{b}$

$$\boldsymbol{c} = \boldsymbol{K}_1 \boldsymbol{a} + \boldsymbol{K}_2 \boldsymbol{b} \tag{5.22}$$

such that $\boldsymbol{K}_1 + \boldsymbol{K}_2 = \boldsymbol{I}$. The statistics of the error in the new estimate are given as [43, 150]

$$\boldsymbol{E}[\tilde{c}] = 0 \tag{5.23}$$

$$\bar{\boldsymbol{P}}_{cc} = \boldsymbol{E}[\tilde{c}\tilde{c}^\intercal] = \boldsymbol{K}_1\bar{\boldsymbol{P}}_{aa}\boldsymbol{K}_1^\intercal + \boldsymbol{K}_2\bar{\boldsymbol{P}}_{bb}\boldsymbol{K}_2^\intercal + \boldsymbol{K}_1\bar{\boldsymbol{P}}_{ab}\boldsymbol{K}_2^\intercal + \boldsymbol{K}_2\bar{\boldsymbol{P}}_{ba}\boldsymbol{K}_1^\intercal \tag{5.24}$$

In real-life applications, the statistics of the estimate may not be perfectly be known. To account for these additional uncertainties, the upper bound $\boldsymbol{P}_{cc}$ on the statics of the error in a new estimate, denoted $\bar{\boldsymbol{P}}_{cc}$, is computed as shown below:

$$\boldsymbol{P}_{cc} \geq \bar{\boldsymbol{P}}_{cc} \tag{5.25}$$

where $\tilde{c} = \boldsymbol{c} - \bar{\boldsymbol{c}}$ and $\boldsymbol{E}[\tilde{c}\tilde{c}^\intercal] = \bar{\boldsymbol{P}}_{cc}$. Geometrically, the covariance ellipsoid representing $\boldsymbol{P}_{cc}$ completely encloses the $\bar{\boldsymbol{P}}_{cc}$ ellipsoid (see Fig. 5.1). The upper bound matrix $\boldsymbol{P}_{cc}$ is expressed as:

$$\boldsymbol{P}_{cc} = \begin{bmatrix} \boldsymbol{K}_1 & \boldsymbol{K}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_{aa} & \bar{\boldsymbol{P}}_{ab} \\ \bar{\boldsymbol{P}}_{ab}^\intercal & \boldsymbol{P}_{bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{K}_1 \\ \boldsymbol{K}_2 \end{bmatrix} \tag{5.26}$$

where the coefficient matrices $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ are chosen to minimize the trace of the solution of $\boldsymbol{P}_{cc}$. $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ are the solution of the following problem:

$$K^* = \min_{\boldsymbol{K}} \boldsymbol{K}\boldsymbol{P}\boldsymbol{K}^\intercal$$

$$\text{where} \quad \boldsymbol{K}\begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{I} \end{bmatrix} = \boldsymbol{I}, \quad \boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_1 & \boldsymbol{K}_2 \end{bmatrix} \quad \text{and,} \quad \boldsymbol{P} = \begin{bmatrix} \boldsymbol{P}_{aa} & \bar{\boldsymbol{P}}_{ab} \\ \bar{\boldsymbol{P}}_{ab}^\intercal & \boldsymbol{P}_{bb} \end{bmatrix}$$

Fig. 5.1 shows the fusion of two estimates with known cross-correlation, in 2D. The blue ellipsoid represents the Gaussian distribution of the error in state estimate $\boldsymbol{a}$ (i.e., $\boldsymbol{P}_{aa}$) and the green ellipsoid represents that of the state estimate $\boldsymbol{b}$ (i.e., $\boldsymbol{P}_{bb}$). The cyan lines represent all realizations of the resulting distribution in the new estimate $\boldsymbol{c}$ (i.e., $\boldsymbol{P}_{cc}$) for different values of the known cross-correlation $\bar{\boldsymbol{P}}_{ab}$. Let the set $\mathbb{B}_{\boldsymbol{P}}(l) = \{\boldsymbol{x}^\intercal\boldsymbol{P}^{-1}\boldsymbol{x} < l\}$. Ref. [43] shows that:

$$\mathbb{B}_{\boldsymbol{P}_{cc}}(l) \subset \{\mathbb{B}_{\boldsymbol{P}_{aa}}(l) \cap \mathbb{B}_{\boldsymbol{P}_{bb}}(l)\} \tag{5.27}$$

In other words, all realizations of $\boldsymbol{P}_{cc}$ lie within the intersection of $\boldsymbol{P}_{aa}$ and $\boldsymbol{P}_{bb}$, which bounds $\bar{\boldsymbol{P}}_{cc}$.

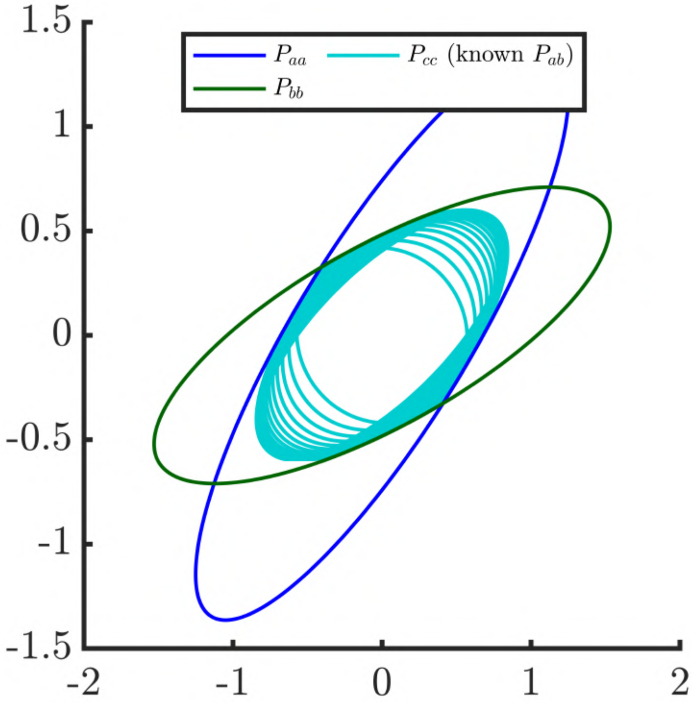


Figure 5.1: Fusion of two estimates with known cross-correlation.

5.2.2.2 Data Fusion For Estimates With Unknown Cross-Correlation

In the event that the cross-correlation $\bar{\boldsymbol{P}}_{ab}$ is unknown, care must be taken when computing the new estimate $\boldsymbol{c}$ and its associated covariance $\boldsymbol{P}_{cc}$. One can demonstrate that the computation for the $n \times n$ optimal matrices $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ can be reduced to a one dimensional optimization problem [43]. A consistent and conservative approximation for $\boldsymbol{P}_{cc}$ is defined as

$$\boldsymbol{P}_{cc}^{-1} = \omega^* \boldsymbol{P}_{aa}^{-1} + (1 - \omega^*) \boldsymbol{P}_{bb}^{-1} \tag{5.28}$$

where ω^* is the solution of the convex optimization problem, which aims at minimizing the size of the resulting matrix. One such metric for judging the size of a matrix is the determinant [68, 109], and ω can be chosen to be the solution to the following one-dimensional optimization problem.

$$\omega^* = \min_{\omega} \det\{\boldsymbol{P}_{cc}\} \tag{5.29}$$

$$\text{s.t:} \;\; 0 \leq \omega \leq 1$$

Other studies in the literature have used the trace as their metric [130, 220, 160], and the performance of one metric versus the other is case-specific. This analysis uses the determinant metric. Eq.(5.29) can be solved using a single value optimizer. Matlab's built-in *fminbnb* is the optimizer used in this work. It is an optimizer based on Golden Section search and parabolic interpolation and provides a local minimum of a continuous function within the prescribed bounds.

The resulting estimate $\boldsymbol{c}$ can also be written in terms of the optimal weight ω^* and initial estimates $\boldsymbol{a}$ and $\boldsymbol{b}$ as

$$\boldsymbol{P}_{cc}^{-1}\boldsymbol{c} = \omega^* \boldsymbol{P}_{aa}^{-1}\boldsymbol{a} + (1-\omega^*)\boldsymbol{P}_{bb}^{-1}\boldsymbol{b} \tag{5.30}$$

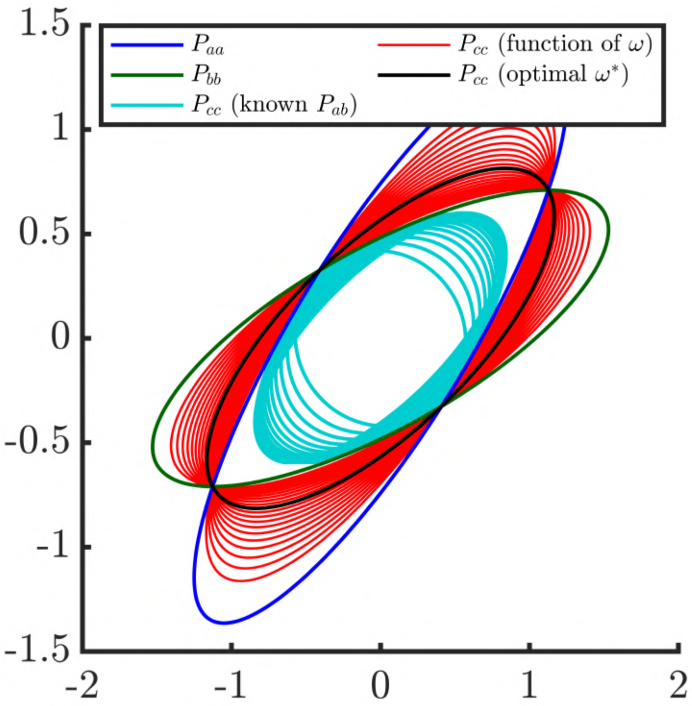


Figure 5.2: Fusion of two estimates with unknown cross-correlation

Eqs. (5.28) and (5.30) can be re-written in terms of the information matrix and information vector. Eqs. (5.31) and (5.32) are better suited for this analysis because they utilize the results from the UIF developed in Section 5.2

$$\boldsymbol{I}_c = \omega^* \boldsymbol{I}_a + (1 - \omega^*) \boldsymbol{I}_b \tag{5.31}$$

$$\boldsymbol{i}_c = \omega^* \boldsymbol{i}_a + (1 - \omega^*) \boldsymbol{i}_b \tag{5.32}$$

5.2.2.3 Distributed Estimator For SFF Cooperative Localization

In the setup for cooperative localization for the SFF problem, it is assumed that spacecraft can take measurements of each other. A sensor is defined as any spacecraft measuring another spacecraft, whereas a target is any spacecraft being measured by a sensor. Every agent in the formation is considered either a sensor or a target, depending on whether it takes a measurement or is being measured by a counterpart, respectively. The cross-correlation between the agents is assumed to be unknown, and each agent only tracks its state and the associated covariance. The communication range is further assumed to be larger than the sensor range; therefore, two agents will be considered neighbors if a measurement is taken between them (i.e., a sensor and a target are considered neighbors). Finally, let $\mathscr{O}_k^i$ be the set of agents/sensors that have taken a measurement of Agent$_i$ at epoch k. Ref [124] shows that, under mild conditions, the UKF converges to the truth, and the associated covariance is exponentially bounded despite the intermittent measurements from different sensors. To account for the sensor's uncertainty, the measurement covariance matrix associated with Agent$_j$, $j \in \mathscr{O}_i$, is inflated by the sensor uncertainty. The measurement update equation is reformulated to the following expression:

$$\begin{aligned} \mathbf{i}_k &= \mathcal{H}_{j,k}^\intercal \left(\mathbf{R}_j + \mathcal{H}_{j,k} \mathbf{P}_{j,k|k} \mathcal{H}_{j,k}^\intercal \right)^{-1} \left[\nu_{j,k} + \mathcal{H}_{j,k} \bar{\mathbf{x}}_{k|k-1} \right] \\ \mathbf{I}_k &= \mathcal{H}_{j,k}^\intercal \left(\mathbf{R}_j + \mathcal{H}_{j,k} \mathbf{P}_{j,k|k} \mathcal{H}_{j,k}^\intercal \right)^{-1} \mathcal{H}_{j,k} \end{aligned} \tag{5.33}$$

The distributed algorithm proposed in this work is briefly presented in Algorithm 3. It takes in a spacecraft's state, covariance, and the measurements collected about it and returns the state and covariance at the next filter epoch. Note the filter only estimates the state of a single agent because tracking all the sensors will require increasing the size of the state space, causing matrix inversion to be more expensive. Consequently, the state of the sensors is not estimated to minimize the computation performed by each filter, and N filters need to run in parallel to estimate a N spacecraft formation.

Algorithm 3: Distributed Estimator For SFF Cooperative Localization (for Agent$_i$)

```
Input: $\hat{\mathbf{x}}^i_{k-1|k-1},\ \mathbf{P}^i_{k-1|k-1},\ \mathbf{z}^j_k,\ \hat{\mathbf{x}}^j_{k|k},\ \mathbf{P}^j_{k|k}$ (with $j \in \mathscr{O}^i_k$)
// Time Update:
$\mathcal{X}^i_{k-1|k-1} = \left[\ \hat{\mathbf{x}}^i_{k-1|k-1},\ \hat{\mathbf{x}}^i_{k-1|k-1} + \sqrt{(n+\lambda)\mathbf{P}^i_{k-1|k-1}},\ \hat{\mathbf{x}}^i_{k-1|k-1} - \sqrt{(n+\lambda)\mathbf{P}^i_{k-1|k-1}}\ \right]$
$\bar{\mathbf{x}}^i_{k|k-1} = \sum_{l=0}^{2n} \mathscr{W}^m_l \mathcal{X}^i_{l,k|k-1}$   where   $\mathcal{X}^i_{k|k-1} = \mathbf{f}\left(\mathcal{X}^i_{k-1|k-1}, \mathbf{w}_k\right)$
$\mathbf{P}^i_{k|k-1} = \mathbf{Q}^i_k + \sum_{l=0}^{2n} \mathscr{W}^c_l \left(\mathcal{X}^i_{l,k-1} - \bar{\mathbf{x}}^i_{k|k-1}\right)\left(\mathcal{X}^i_{l,k-1} - \bar{\mathbf{x}}^i_{k|k-1}\right)^\intercal$
$\mathscr{W}^m_0 = \frac{\lambda}{n+\lambda},\ \mathscr{W}^c_0 = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta),\ \mathscr{W}^m_i = \mathscr{W}^m_i = \frac{\lambda}{2(n+\lambda)}$
// Measurement Update:
if  Agent$_i$ = Target then
    $\mathbf{i}^i_k = 0,\ \mathbf{P}^i_{k|k} = 0$
    for $j \in \mathscr{O}^i_k$ do
        $\mathbf{i}_{j,k} = \mathcal{H}^\intercal_{j,k}\left(\mathbf{R}_j + \mathcal{H}_{j,k}\mathbf{P}_{j,k|k}\mathcal{H}^\intercal_{j,k}\right)^{-1}\left[\nu_{j,k} + \mathcal{H}_{j,k}\bar{\mathbf{x}}_{k|k-1}\right]$
        $\mathbf{I}_{j,k} = \mathcal{H}^\intercal_{j,k}\left(\mathbf{R}_j + \mathcal{H}_{j,k}\mathbf{P}_{j,k|k}\mathcal{H}^\intercal_{j,k}\right)^{-1}\mathcal{H}_{j,k}$
        $\omega^*_j = \min_\omega \det\left(\boldsymbol{P}_{cc}\right)$
        $\mathbf{I}^i_k = \omega^*_j \mathbf{I}^i_k + (1 - \omega^*_j)\mathbf{I}_{j,k}$
        $\mathbf{i}^i_k = \omega^*_j \mathbf{i}^i_k + (1 - \omega^*_j)\mathbf{i}_{j,k}$
    $\mathbf{P}^i_{k|k} = \left(\mathbf{P}^{-1}_{i,k-1|k-1} + \mathbf{I}^i_k\right)^{-1}$
    $\hat{\mathbf{x}}^i_{k|k} = \left(\mathbf{P}^{-1}_{i,k-1|k-1} + \mathbf{I}^i_k\right)^{-1}\mathbf{i}^i_k$
else
    $\mathbf{P}^i_{k|k} = \mathbf{P}^i_{k-1|k-1}$
    $\hat{\mathbf{x}}^i_{k|k} = \bar{\mathbf{x}}^i_{k-1|k-1}$
Output: $\hat{\mathbf{x}}^i_{k|k}, \mathbf{P}^i_{k|k}$
```

At the beginning of each filtering cycle, the state and covariance are propagated forward in time using the unscented transform. The algorithm operates differently depending on whether the spacecraft is a sensor or the target. If the agent is a sensor, it will send the relative measurement and state/covariance to the target. On the other hand, if the agent is the target, the algorithm merges the filter estimate and the measurement using covariance intersection. If multiple sensors measure the same target, the information contained in those measurements is fused sequentially, which allows for optimizing one variable per measurement (as opposed to a multi-variable nonlinear optimization problem) [57].

5.3 Simulation, Results, and Discussion

The performance of the Distributed Estimator was analyzed for a set of five spacecraft flying in formation around an eccentric reference orbit. It is assumed that the chief's trajectory is known by each of the five deputies in the formation; therefore, no measurement from the reference trajectory is required. Due to the distributed nature of the algorithm, each deputy only estimates its relative position and velocity (i.e., the deputy i estimates $\mathbf{x}_i = [\boldsymbol{\rho}_i, \boldsymbol{\rho}_i'] = [x_i, y_i, z_i, x_i', y_i', z_i']$), and all the filters are run in parallel. The equations of motion of a deputy (as seen in the chief's Local-Vertical Local-Horizontal frame) are given by:

$$\frac{{}^{\mathscr{E}}d}{dt}(\mathbf{x}_i) = \mathbf{x}_i' = \begin{bmatrix} \boldsymbol{\rho}_i' \\ \delta\boldsymbol{a}_i + -\boldsymbol{\Omega}_i' \times \boldsymbol{\rho}_i - 2\boldsymbol{\Omega}_i \times \boldsymbol{\rho}_i' - \boldsymbol{\Omega}_i \times \boldsymbol{\Omega}_i \times \boldsymbol{\rho}_i \end{bmatrix} \tag{5.34}$$

where $\delta\boldsymbol{a}_i$ is the differential gravitational acceleration, and $\boldsymbol{\Omega}_i$ and $\boldsymbol{\Omega}_i'$ are the angular velocity and the angular acceleration of the chief's Local-Vertical Local-Horizontal frame, respectively. As the different agents fly around, they can either measure other spacecraft that pass through the lines-of-sight of their sensors or be measured if another agent's sensors record them. It is assumed that every deputy can measure range, range-rate, and relative orientation, and the measurements model is written as follows:

$$\mathbf{h}_{ij} = \begin{bmatrix} \alpha_{ij} \\ \gamma_{ij} \\ Az_{ij} \\ El_{ij} \end{bmatrix} = \begin{bmatrix} \|\boldsymbol{\rho}_i - \boldsymbol{\rho}_j\| \\ (\boldsymbol{\rho}_i - \boldsymbol{\rho}_j)^\intercal(\boldsymbol{\rho}_i' - \boldsymbol{\rho}_j')/\alpha_{ij} \\ \arcsin(z_{ij}/\alpha_{ij}) \\ \arctan(x_{ij}/y_{ij}) \end{bmatrix} \tag{5.35}$$

where $\alpha_{ij} = \sqrt{x_{ij}^2 + y_{ij}^2 + z_{ij}^2} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$ is the norm of the relative position of the sensor j taking the measurement of the target i.

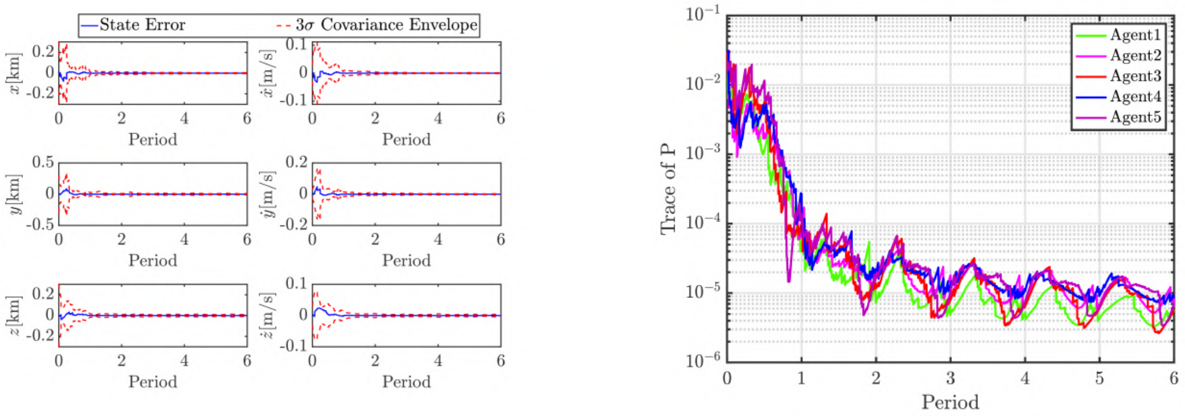
In the simulation discussed below, a measurement is taken once every minute. The target (i.e., the spacecraft being measured) is chosen at random. The number of measurements is random for every filter epoch, and their origin is also randomly chosen. That is, at each filter epoch, a random number q of randomly chosen spacecraft are chosen to be sensors, where $0 < q < N - 1$. Therefore, the target ID, the number of sensors, and their respective IDs are all randomly chosen quantities, every time a measurement is taken. The reference orbit is eccentric ($e = 0.2$), and the initial deviations, in [km] and [km/s], of the five deputies considered in this analysis are given in Table 5.2. The initial error covariance $\boldsymbol{P}_0 = \text{diag}\left([100\text{m}, 100\text{m}, 100\text{m}, 1\text{cm/s}, 1\text{cm/s}, 1\text{cm/s}]\right)$ is set to be the same for each deputy, and the sensor noise's covariance matrix is $\boldsymbol{R} = \text{diag}\left([1\text{E}{-3}\text{m}, 1\text{E}{-6}\text{m/s}, 1\text{E}{-3}\text{rad}, 1\text{E}{-3}\text{rad}]\right)$.

Table 5.2: Initial relative state in Cartesian coordinates

Initial ΔX	δx [km]	δy[km]	δz[km]	$\delta\dot{x}$[km/s]	$\delta\dot{y}$[km/s]	$\delta\dot{z}$[km/s]
Deputy1	0.6667	-0.1500	0.2004	1.468e-05	-6.430e-4	1.3114e-05
Deputy2	0.6667	0.3303	0.3720	2.939e-05	-6.430e-4	8.2704e-05
Deputy3	0.6667	-0.4501	0.6012	4.404e-05	-6.430e-4	3.9344e-05
Deputy4	0.6667	0.6605	0.7440	5.878e-05	-6.430e-4	1.654e-4
Deputy5	0.6667	-0.7501	1.0019	7.339e-05	-6.430e-4	6.558e-05

Fig. 5.3(a) shows the evolution of the state residuals and the associated covariance envelope for agent 1. The error in the estimates remains within the covariance bounds at all times and approaches zero as more relative measurements are processed. The covariance envelopes shrink accordingly, as well. The state residuals of the other agents exhibit similar behavior to that of agent 1. Fig. 5.4 shows the measurement residuals associated with agent 1. Fig 5.4(a) shows the post-fit residuals for the relative measurements from the other agents in the formation. At the beginning of the simulation, the uncertainty in the sensor location generates a faulty estimate, which in turn causes the measurement residuals to be outside the 3σ noise. As more measurements are gathered and processed, the distributed algorithm gradually constrains the state of all the agents, causing the post-fit residuals to fall below the sensor noise level. Fig. 5.4(b) shows the histogram of the post-fit residuals after one orbital period. These histograms show that measurement residuals are zero-mean Gaussian white noise with a spread equal to the 3σ confidence interval of the sensors. Finally, fig. 5.3(b) shows the time evolution of the trace of the covariance matrices associated with

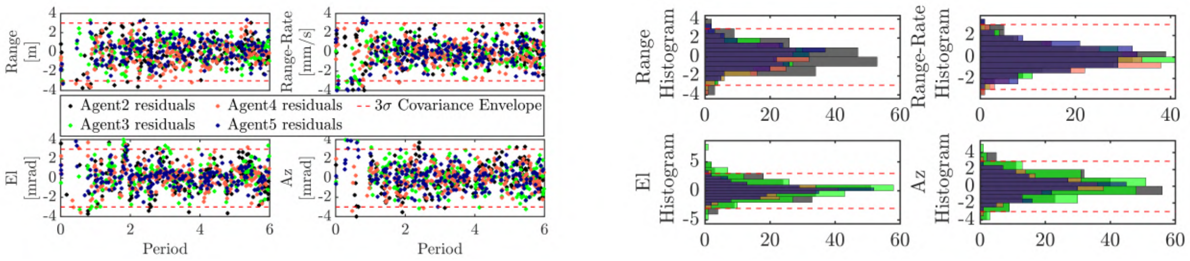
each agent. As the filter processes measurements, the location of the different agents gets more constrained, and the errors in each filter decrease. As shown in Eq. (5.33), the covariance of every agent is intertwined, causing the uncertainty of one agent to shrink and expand as the uncertainty of the other agents tightens and loosens, respectively.



(a) State-estimates error agent 1

(b) Trace of covariances of each filter

Figure 5.3: Distributed Estimator: state residuals and covariance envelopes

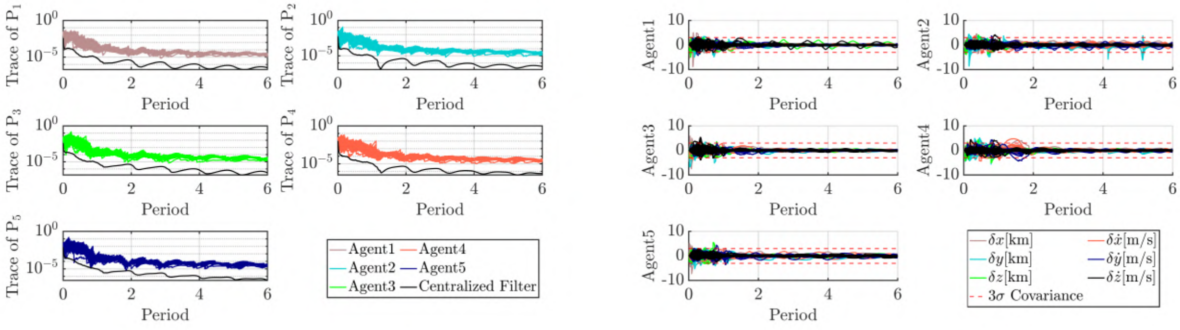


(a) Post-fit residuals

(b) Error histogram

Figure 5.4: Measurement residuals and error histogram over six orbital periods (agent 1)

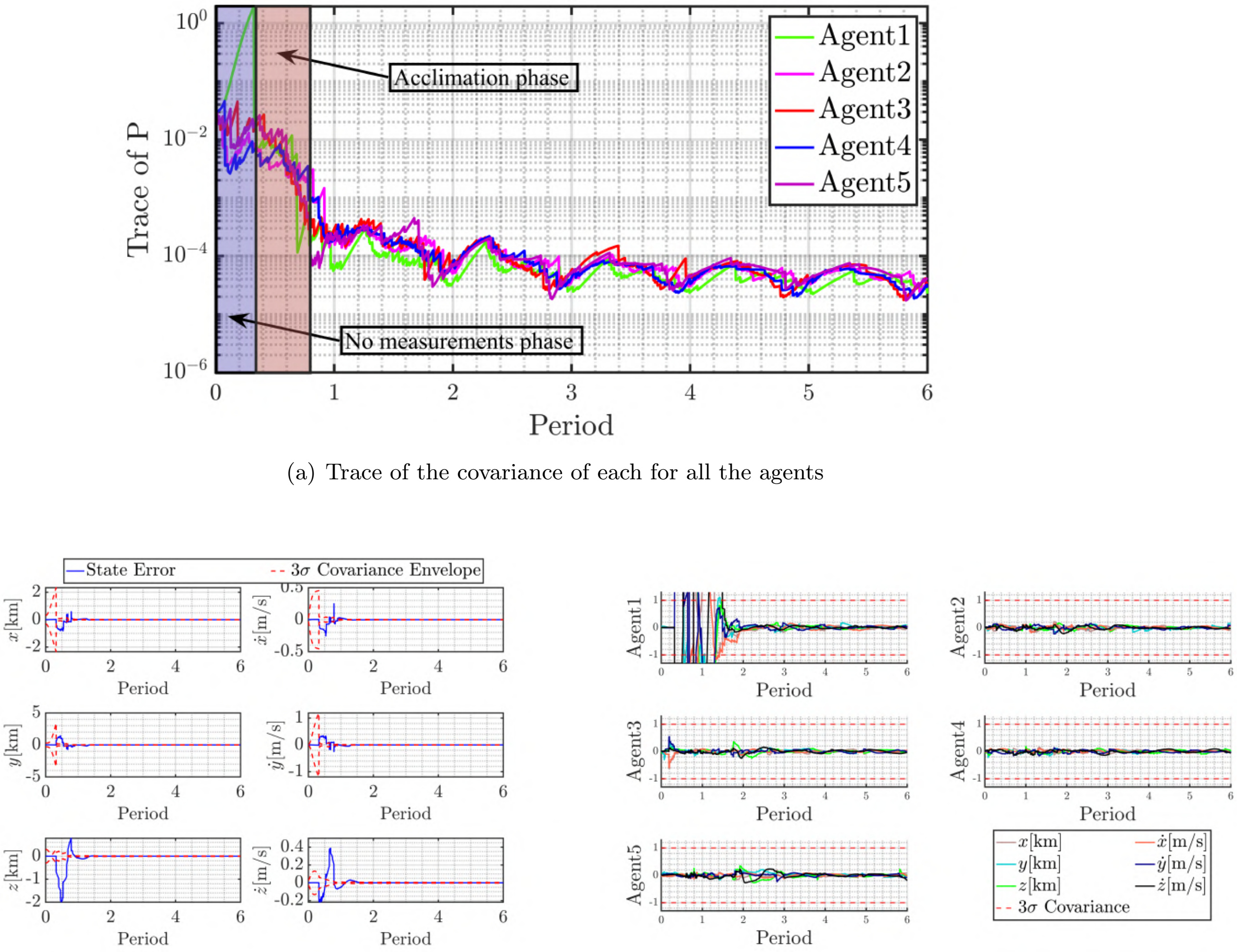
A thousand Monte-Carlo simulations were used to assess the robustness of the proposed algorithm. Fig. 5.5(a) shows the trace of the resulting covariances on a log scale. All the Monte-Carlo runs are consistent, and the trace of each covariance matrix exhibits the same behavior. The trace of the errors associated with the proposed solution is compared to a centralized filter, shown in black. In the centralized filter, the states of the agents are stacked in a 30 × 1 state vector. Then, a single filter processes all the measurements associated with the large state vector. This process considers all the cross-correlations between the agents. Consequently, all the information is extracted from the relative observations. In each subplot of Fig. 5.5(a), the corresponding block diagonal is extracted from the 30×30 centralized-covariance matrix, and its trace is plotted alongside the trace of the distributed estimator algorithm. The covariance associated with the distributed estimator algorithm is an upper bound to the true covariance retrieved from the centralized filter. The distributed filter's covariance does not converge to the centralized covariance for two reasons. First, the addition of the term $\mathcal{H}_{j,k}\mathbf{P}_{j,k|k}\mathcal{H}_{j,k}^{\intercal}$ in Eq. (5.33) inflates the error associated with the sensor j and reduces the amount of information available with measurement at epoch k. Second, the fused covariance resulting from Eq. (5.29) is an approximation of the real combined covariance. The covariances resulting from distributed estimator algorithm consistently converge to the same amplitude. Because this algorithm is completely distributed, it is easily scalable. The filter's fidelity is analyzed in fig. 5.5(b) where the state residuals of each agent are normalized by their respective covariance. Across all the Monte Carlo simulations, the state residuals of all the agents remain within their 3σ covariance bound. The proposed filter consistently generates a zero-mean estimate for all the agents across every Monte Carlo simulation.



(a) Distributed estimator and centralized filter comparison (b) Consistency Analysis for the distributed estimator

Figure 5.5: Results of a one thousand Monte Carlo simulations

Finally, the versatility of the algorithm was tested for scenarios where a deputy spacecraft is added to the filtering network at a later epoch. To be more specific, spacecraft 1 is introduced into the filtering loop half an orbit after the beginning of the simulation. Agent 1 starts with the same initial convariance as the other agent, but does not perform any measurement update for one half orbit, causing its covariance to grow. It is then introduced into the filtering loop (with a covariance envelope ten times larger than its initial value). This scenario would correspond to a spacecraft inserted into an already existing distributed navigation network. Fig. 5.6(a) shows the trace of the covariance of each agent. The uncertainty in agent 1 increases over the period of time it does not receive any measurements (blue box in fig. 5.6(a)). Then agent 1 is subject to an acclimation phase, where it uses the other agents' measurements to shrink its covariance and get a better estimate of its state. The acclimation phase is shown in dark orange in fig. 5.6(a). During this phase, agent 1 does not contribute to the distributed filter; agent 1 only gathers information from the relative measurements it receives. During the acclimation, the state residuals are outside the covariance bounds (see the first subfigure of fig. 5.6(c)). The covariance envelope shrinks drastically, and it takes a few filtering cycles for the navigation algorithm to adjust the state errors, as seen in fig. 5.6(b). During the acclimation phase, agent 1 was not allowed to introduce/add new information into the distributed network to stop its error from propagating across the network. Adding and removing agents at any time makes the proposed filter suitable for the navigation of heterogeneous constellations and for providing navigation data for collision avoidance maneuvers.



(a) Trace of the covariance of each for all the agents

(b) State-estimates error agent 1

(c) Normalized state residuals

Figure 5.5: Insertion of agent 1 into a pre-existing constellation

5.4 Conclusion

This chapter presented a novel algorithm for distributed spacecraft cooperative localization. The proposed algorithm elegantly blends the UIF and the CI algorithms to achieve exponential convergence under intermittent measurements. The UIF takes advantage of the duality of the Gaussian distribution to efficiently perform both the time and the measurement updates of the navigation filter. The measurements originating from neighboring agents are continuously fused using a CI scheme. The algorithm is designed to be agnostic to the number of spacecraft partaking in the navigation process, and every agent only tracks its local state and covariance. The particularity of the proposed approach is that it works well for both fixed topologies and ad hoc networks. The stability and consistency of the distributed algorithm were analyzed via Monte Carlo simulations. The proposed method allows arbitrarily large constellations to estimate their state trading relative information. Also, the fully distributed nature of the filter allows adding or removing new agents at any instant with no detrimental effect to the filter.

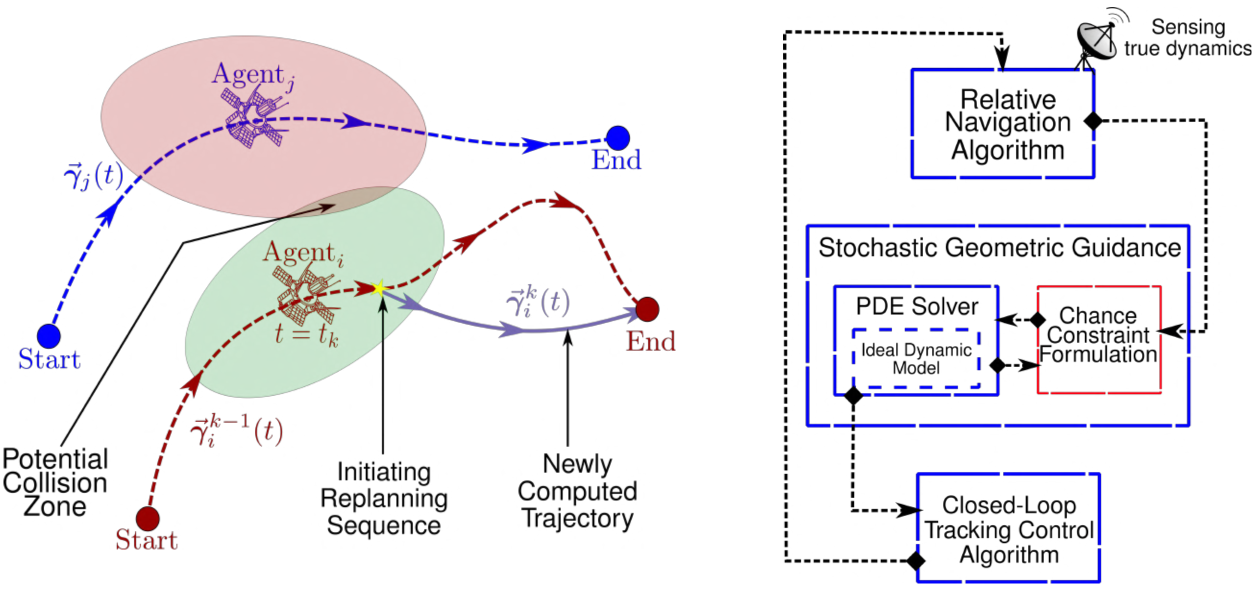
Chapter 6

Stochastic Distributed Motion Planning: Chance Constraint Implementation

Up to this point, it was assumed that the system was deterministic; that is there was complete knowledge of the state and dynamics of every spacecraft in the formation. However, real-world systems are uncertain and stochastic. Therefore, it is imperative to incorporate navigation information when generating safe trajectories in uncertainties environments. For the remainder of this study, it will be assumed that the system is subject to zero mean Gaussian noise due to state and measurement noises.

Fig. 6.1(a) represents the maneuvering spacecraft's behavior when an uncertain obstacle is detected. The trajectories are denoted by $\boldsymbol{\gamma}$, and a subscript (i.e., i or j) is used to assign a path to an agent. The superscript k marks the guidance epoch. Each path $\boldsymbol{\gamma}^k$ is computed using the distributed Geometric Guidance algorithm discussed in chapters 2 and 3. The focus of this study is to replace the avoidance sphere used in Chapter 4 by covariance envelopes provided by the distributed filter developed in Chapter 5. A chance constraint algorithm will be derived to integrate filter information in the existing Geometric Guidance.

To date, many methods for robust control have been utilized for trajectory planning for systems with uncertainties [21]. However, these methods are conservative and ill-suited for distributions with unbounded support (i.e., noises with Gaussian distribution). Within the realm of stochastic optimal control, unbounded uncertainties are handled with chance-constrained trajectory optimizations. This formulation guarantees, with a given level of confidence ϵ, that a constraint $g(\vec{\boldsymbol{x}}) \geq 0$ is


Agent$_j$
$\vec{\gamma}_j(t)$
End
Start
Agent$_i$
$t = t_k$
$\vec{\gamma}_i^k(t)$
End
$\vec{\gamma}_i^{k-1}(t)$
Potential Collision Zone
Start
Initiating Replanning Sequence
Newly Computed Trajectory
Sensing true dynamics
Relative Navigation Algorithm
Stochastic Geometric Guidance
PDE Solver
Ideal Dynamic Model
Chance Constraint Formulation
Closed-Loop Tracking Control Algorithm

(a) Chance Constraints

(b) Stochastic Geometric Planner

Figure 6.1: Concept Behind Proposed Algorithm

met for the random variable $\vec{\boldsymbol{x}}$. The constraint is written as the following probability:

$$\Pr\left(g(\vec{\boldsymbol{x}}) \geq 0\right) \geq \delta \tag{6.1}$$

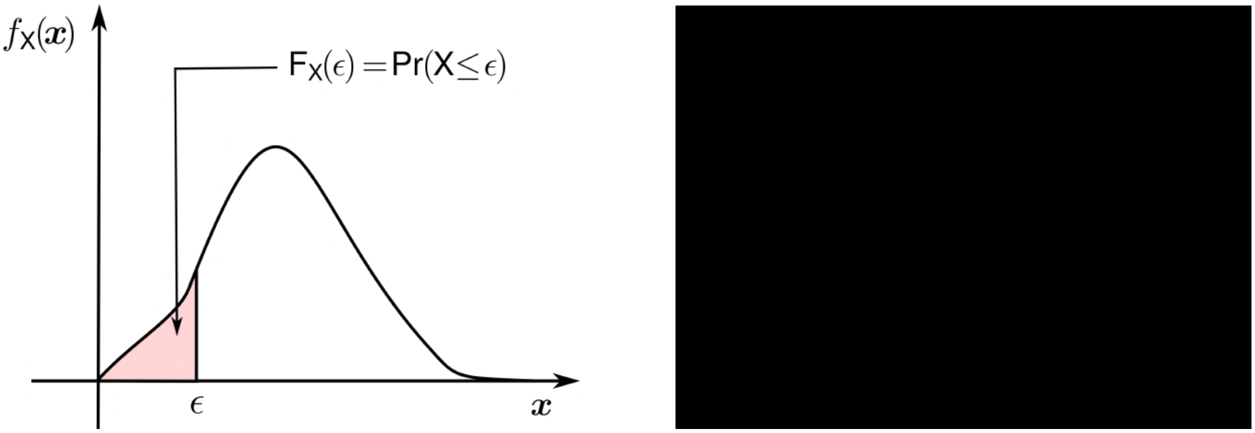
The probabilistic constraint presented in Eq. (6.1) must be reformulated as a deterministic constraint on the mean value of its distribution to be integrated into the geometric solver. It is essential to look back at some definitions from probability theory to better understand how probabilistic constraints are converted into deterministic constraints.

6.1 Review of Probability Theory

A random variable X can intuitively be understood as a variable whose value cannot be known with certainty until it is sampled. However, it can also be understood as a mapping from an experiment to a real number. As such, one can define the cumulative distribution function $\mathrm{F}_{\mathrm{X}}(\boldsymbol{\epsilon})$, the most fundamental property of a random variable X, as follows:

$$\mathrm{F}_{\mathrm{X}}(\boldsymbol{\epsilon}) = \Pr\left(\mathrm{X} \leq \boldsymbol{\epsilon}\right) = \int_{-\infty}^{\boldsymbol{\epsilon}} f_{\mathrm{X}}\left(\boldsymbol{x}\right) \mathrm{d}x \tag{6.2}$$

where f_{X} is the probability density function[1] (pdf). Equation (6.2) relates a random variable to a deterministic expression. The CDF can be determined analytically or imperially (data-driven). The meaning of the probability of the random variable is shown in Fig. 6.2.



(a) Probability Density Function

(b) Cumulative Distribution Function

Figure 6.2: Meaning of Probability $\Pr\left(\mathrm{X} \leq \boldsymbol{\epsilon}\right)$

The mean $\boldsymbol{\mu}$ of a probabilistic distribution is the first moment about the origin, and the standard deviation $\boldsymbol{\sigma}$ is the (positive) square root of the second moment about the mean. They are both given by the following expressions:

$$\boldsymbol{\mu} = \mathrm{E}\left[\boldsymbol{x}\right] = \int_{-\infty}^{\infty} \boldsymbol{x} f_{\mathrm{X}}\left(x\right) \mathrm{d}x \tag{6.3}$$

$$\boldsymbol{\sigma}^2 = \mathrm{E}\left[\boldsymbol{x} - \boldsymbol{\mu}\right] = \int_{-\infty}^{\infty} \left(\boldsymbol{x} - \boldsymbol{\mu}\right)^2 f_{\mathrm{X}}\left(x\right) \mathrm{d}x \tag{6.4}$$

where $\mathrm{E}[\cdot]$ symbolizes the expectation operator. The expected value of the image of the random variable $\boldsymbol{x}$ through any generic functional g is given by:

$$\mathrm{E}\left[g\left(\boldsymbol{x}\right)\right] = \int_{-\infty}^{\infty} g\left(\boldsymbol{x}\right) f_{\mathrm{X}}\left(x\right) \mathrm{d}x \tag{6.5}$$

[1] The pdf is defined such that $f_{\mathrm{X}}(\boldsymbol{\epsilon}) \geq 0$ and $\int_{-\infty}^{\infty} f_{\mathrm{X}}(x) dx = 1$

6.2 Stochastic Guidance With Constant Covariance

Let's return to constrained trajectory optimization. Let $\mathbf{a}, \boldsymbol{x} \in \mathbb{R}^n$, $c \in \mathbb{R}$, and $\epsilon \in [0.5, 1]$. If it is assumed that the decision variable $\boldsymbol{x} \sim \mathscr{N}\left(\bar{\boldsymbol{x}}, \boldsymbol{\Sigma}\right)$, then (see Appendix C)

$$\Pr\left(\mathbf{a}^\intercal \boldsymbol{x} \le c\right) \ge \epsilon \iff \mathbf{a}^\intercal \bar{\boldsymbol{x}} + \mathrm{F}_\mathrm{X}^{-1}(\epsilon)\sqrt{\mathbf{a}^\intercal \boldsymbol{\Sigma} \mathbf{a}} \le c \tag{6.6}$$

where $\mathrm{F}_\mathrm{X}(\epsilon)$ is the cumulative distribution function for a standard Gaussian variable (i.e., zero mean and unit standard deviation) [35]. Conceptually, chance constraints can be thought of as a confidence set ellipsoid where the probability that the random variable $\vec{\boldsymbol{x}} \sim \mathscr{N}\left(\vec{\boldsymbol{\mu}}, \boldsymbol{\Sigma}\right)$ belongs to a feasible set $\mathcal{B}(\vec{\boldsymbol{\mu}}, \boldsymbol{\Sigma})$ is larger than ϵ (i.e., $\Pr(\vec{\boldsymbol{x}} \in \mathcal{B}(\vec{\boldsymbol{\mu}}, \boldsymbol{\Sigma})) \ge \epsilon$). Here,the ellipsoid $\mathcal{B}$ is defined as:

$$\mathcal{B}(\vec{\boldsymbol{\mu}}, \boldsymbol{\Sigma}) = \{\vec{\boldsymbol{x}} | (\vec{\boldsymbol{\mu}} - \vec{\boldsymbol{x}})^\intercal \boldsymbol{\Sigma} (\vec{\boldsymbol{\mu}} - \vec{\boldsymbol{x}}) \le 1\} \tag{6.7}$$

The parameter ϵ controls how far away from the ellipsoid the point $\vec{\boldsymbol{x}}$ is. The diagonal entries of $\boldsymbol{\Sigma}$ are the standard deviations that control how far the surface of the ellipsoid is from the mean $\vec{\boldsymbol{\mu}}$.

Assume for instance that a maneuvering spacecraft Agent_i and a uncooperative obstacle Agent_j are on a collision course. The collision avoidance constraint between agents i and j can be expressed as $\delta\rho_{ij} \ge r_i + r_j$, where $\delta\rho_{ij} = \|\vec{\boldsymbol{\rho}}_j(t) - \vec{\boldsymbol{\rho}}_i(t)\|$ is the relative distance between two agents, and r_i, r_j are the radius of the sphere enclosing agent i and j, respectively. Now, assume that both agents are subject to zero-mean Gaussian white noises (i.e., $\mathbf{x}_i \sim \mathcal{N}\left(\hat{\mathbf{x}}_i, \boldsymbol{\Sigma}_i\right)$ and $\mathbf{x}_j \sim \mathcal{N}\left(\hat{\mathbf{x}}_j, \boldsymbol{\Sigma}_j\right)$). If the noises are independent (i.e., $\mathbf{E}[\hat{\mathbf{x}}_i^\intercal \hat{\mathbf{x}}_j] = \mathbf{0}_{n\times n}$ where n is the dimension of the state space), the collision probability between agents can be approximated by the following equation [194]:

$$\begin{aligned} P(c) &= V_i \int_{\mathbf{x}_i} p\left(\mathbf{x}_j = \mathbf{x}_i | \mathbf{x}_i\right) p\left(\mathbf{x}_i\right) \mathrm{d}\mathbf{x}_i \\ &= \frac{V_i}{\sqrt{\det\left(2\pi\Sigma_c\right)}} \exp\left[-\frac{1}{2}\left(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\right)^\intercal \Sigma_\mathrm{c}^{-1}\left(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\right)\right] \end{aligned} \tag{6.8}$$

where $\Sigma_c = \Sigma_i + \Sigma_j$ is the combined-state covariance, and V_i is the volume of the sphere enclosing the spacecraft.

For uncertain states, the collision avoidance constraint in Eq. 6.1 can be converted into a deterministic contraint on the mean relative position between the two agents as shown bellow:

$$l\left(\delta\rho_{ij})\right) \triangleq \left(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\right)^\intercal \Sigma_{\mathrm{c}}^{-1} \left(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\right) \geq -2\ln\left(\frac{1-\epsilon}{V_{\mathrm{r}}}\sqrt{\det\left(2\pi\Sigma_{\mathrm{c}}\right)}\right) \tag{6.9}$$

Now, let the following barrier function:

$$\psi_j(\vec{\boldsymbol{\rho}}_i, \vec{\boldsymbol{\rho}}_j) = K_j \exp\left(-\left(\frac{l(\delta\rho_{ij})}{P_{ij}}\right)^2\right) \tag{6.10}$$

Eq. (6.10) is a repelling function that causes the optimized cost function to increase as the intersecting volume between the covariance matrices increases. The chance constraint can seamlessly be integrated in the Geometric Guidance algorithm by multiplying $\psi(\vec{\boldsymbol{x}})$ with the Riemannian metric:

$$G_i(\vec{\boldsymbol{x}}) = \psi(\vec{\boldsymbol{x}})\left(\bar{\mathbf{F}}^{-\intercal}(\vec{\boldsymbol{x}})\mathbf{D}\bar{\mathbf{F}}^{-1}(\vec{\boldsymbol{x}})\right) \tag{6.11}$$

6.2.1 Application to a Two-Agent Formation

In the following example, two deputy spacecraft are traveling between two sets of boundary conditions. The orbital elements of the chief's trajectory and the relative orbital element of each deputy spacecraft are given in Tables 6.1 and 6.2, respectively.

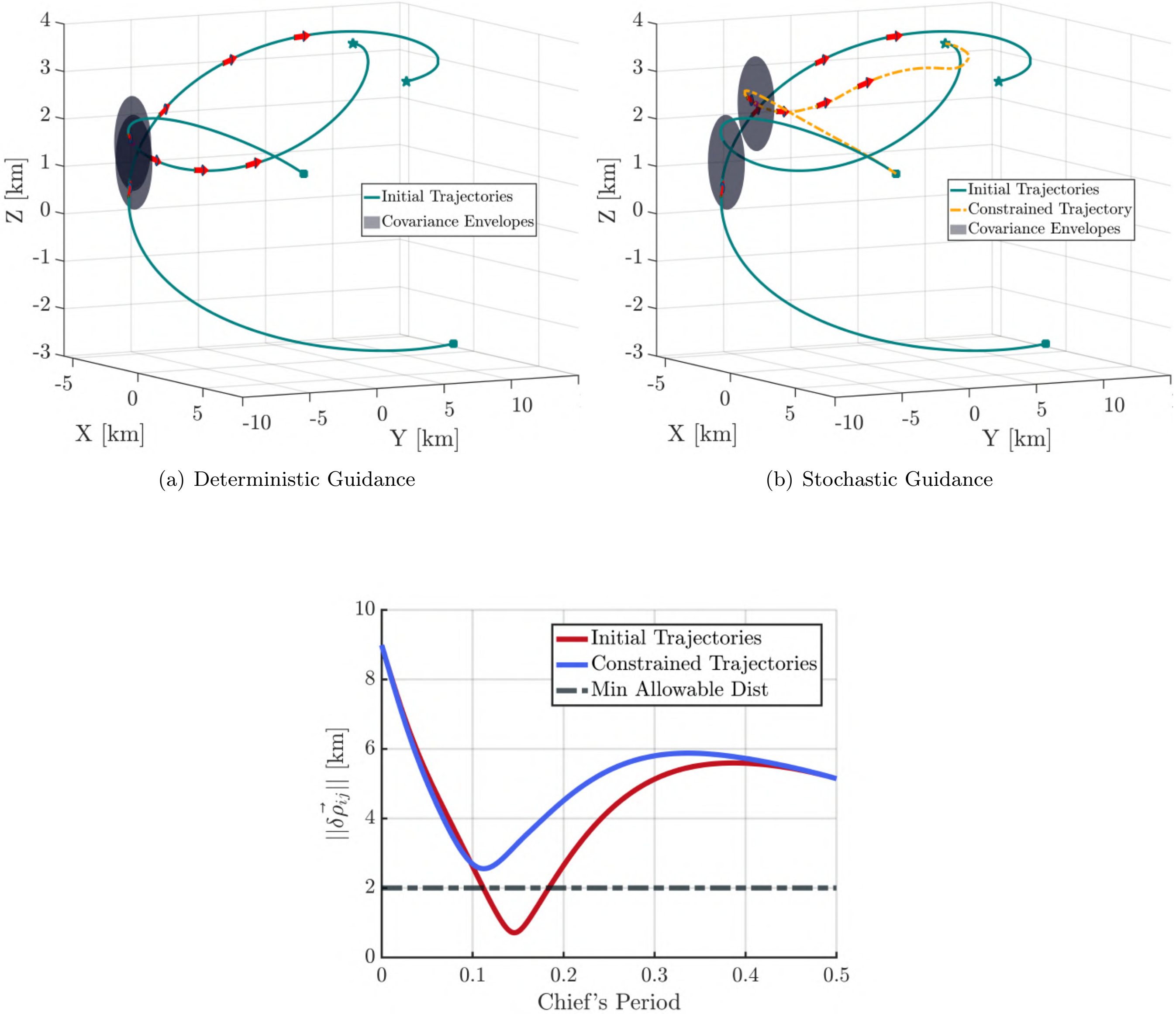
Table 6.1: Reference Orbit in Orbital Elements

Initial OE	a [km]	e	i [rad]	Ω [rad]	ω [rad]	f [rad]
Reference	1.42e04	0.5	50	10	10	0.0

Table 6.2: Initial Differential Orbital Elements

ΔOE	δa [km]	δe	δi [rad]	$\delta\Omega$ [rad]	$\delta\omega$ [rad]	δf [rad]
Deputy1 (initial)	0.0	$1/(6a)$	$-1/(3a)$	0.0	$-2\pi * 1e-5$	$\pi * 1e-6$
Deputy2 (initial)	0.0	$1/(8a)$	$2/(10a)$	$-\pi * 1e-5$	$\pi * 1e-5$	$\pi * 1e-6$
Deputy1 (final)	0.0	$1/(6a)$	$1/(2a)$	0.0	0.0	0.0
Deputy2 (final)	0.0	$-1/(8a)$	$-1/(20a)$	$\pi * 1e-6$	$-5 * \pi * 1e-7$	0.0

It is assumed that each agent's state is corrupted by Gaussian random noise. The noise level is kept constant throughout the simulation, and the position covariance is arbitrarily set to $\mathbf{\Sigma} = \text{diag}\left([1,1,1]\right)$ km for each spacecraft. The scenario presented in Fig. 6.3 is an orbital transfer where two spacecraft fly very close to one another. The duration of the transfers is half of the chief's orbit. In both Figs. 6.3(a) and 6.3(b), the two error covariances are plotted, as grey ellipsoids, at the distance of the closest approach. Initially, the two spacecraft travel on paths that violated the probabilistic constraints. The intersecting covariance envelopes represented in Fig. 6.3(a) indicates a high likelihood of collision. After constructing the barrier function presented in Eq. 6.10, a stochastic-aware replanning sequence is initiated onboard the maneuvering agent. The newly computed path, shown in Fig. 6.3(b), ensures the covariance envelopes of the two spacecraft do not to intersect. Fig. 6.3(c) shows the norms of the relative position between the two agents, for both the initial paths (in red) and the constrained paths (in blue). The distance of closest approach is less than one kilometer in the deterministic guidance case. By imposing the chance constraint, the stochastic-aware guidance is able to increase that distance to more than two kilometers (between the distributions' mean values).



(a) Deterministic Guidance

(b) Stochastic Guidance

(c) Separation Distance

Figure 6.3: Chance Constraint Probabilistic Maneuvering

6.3 Stochastic Guidance with Variable Covariance

Assume that an optimal path $\vec{\boldsymbol{\gamma}}^*(t)$ between two boundary conditions has been computed, as shown in Fig. 6.4. Assume that a feedback control $\vec{\boldsymbol{u}}^*(t)$ strategy has been developed to track the optimal trajectory and zero out the initial deviation (i.e., $\vec{\boldsymbol{u}}^*(t) \implies \lim_{t\rightarrow\infty} \delta\vec{\boldsymbol{r}}(t_0) = \mathbf{0}$). If the initial deviation is corrupted by the normally distributed random disturbance $\vec{\boldsymbol{\epsilon}}(t) \sim \mathscr{N}(0, \mathbf{P})$, the system's dynamics will cause the error associated with the random variable to grow over time regardless of the feedback control strategy (i.e., $\mathrm{P}(t_0) \leq \mathrm{P}(t_1) \leq \cdots \leq \mathrm{P}(t_f)$), assuming no measurements.

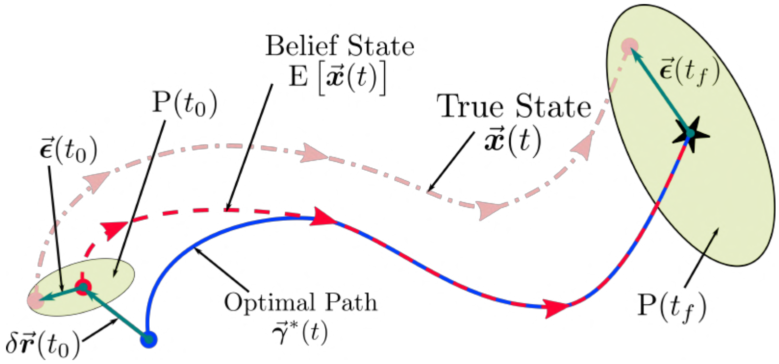

Belief State
$\mathrm{E}\left[\vec{\boldsymbol{x}}(t)\right]$
$\mathrm{P}(t_0)$
$\vec{\boldsymbol{\epsilon}}(t_0)$
True State
$\vec{\boldsymbol{x}}(t)$
$\vec{\boldsymbol{\epsilon}}(t_f)$
$\delta\vec{\boldsymbol{r}}(t_0)$
Optimal Path
$\vec{\gamma}^*(t)$
$\mathrm{P}(t_f)$

Figure 6.4: Dynamic Evolution of the Covariance

For multi-agent systems, the error associated with each agent will grow over time. This phenomena, if not dealt with, will induce collisions for trajectories whose optimal paths are not intersecting, as shown in Fig. 6.5. In the event where a boundary condition of an agent lies in the error zone of another agent, there does not exist a solution that solves the constrained optimization problem. It then becomes imperative to manage the growth of the covariance uncertainty when doing guidance on stochastic systems. The size of the covariance matrix is shrunk by integrating measurements information via a navigation filter.

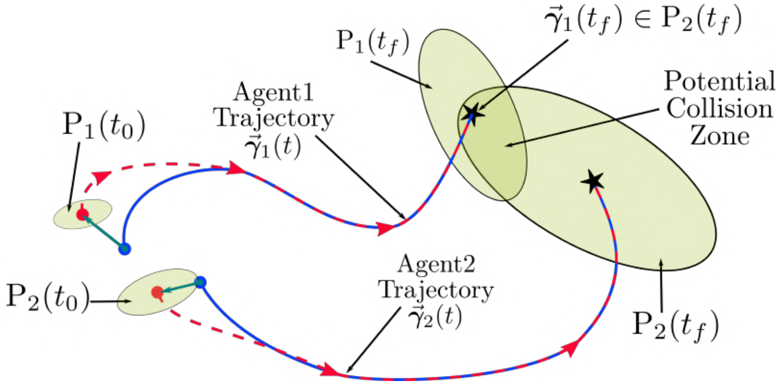


Figure 6.5: Collision Induced by Covariance Growth

The covariance propagation is done through an unscented transform because of the nonlinear nature of the dynamical system studied [205]. Let $\mathbf{f}\left(\mathbf{x}_i\right)$ and $\mathbf{h}\left(\mathbf{x}_i\right)$the dynamic and measurement model for each agent. The navigation algorithm's process can be discretized as follows:

$$\begin{aligned} \mathbf{x}_{i,k+1} &= \mathbf{f}\left(\mathbf{x}_{i,k}\right) + \boldsymbol{w}_{i,k} \\ \mathbf{z}_{i,k} &= \mathbf{h}\left(\mathbf{x}_{i,k}\right) + \boldsymbol{v}_{i,k} \end{aligned} \tag{6.12}$$

where $\boldsymbol{w}_{i,k} \sim \mathscr{N}\left(0, \mathbf{Q}_{i,k}\right)$ and $\boldsymbol{v}_{i,k} \sim \mathscr{N}\left(0, \mathbf{R}_{i,k}\right)$ represent zero-mean white Gaussian sequences in the dynamics and measurements, respectively. Let n designate the size of the solution manifold. To propagate the uncertainty associated with agent i, $2n+1$ sigma points are sampled from the distribution $\mathscr{N}\left(\hat{\mathbf{x}}_{i,k-1|k-1}, \mathbf{P}_{i,k-1|k-1}\right)$ and propagate through the nonlinear dynamics in Eq. (6.12). The weighted samples are defined below:

$$\mathcal{X}_{k-1|k-1} = \left[\ \hat{\mathbf{x}}_{k-1|k-1},\ \hat{\mathbf{x}}_{k-1|k-1k} + \sqrt{(n+\lambda)\mathbf{P}_{k-1|k-1}},\ \hat{\mathbf{x}}_{k-1|k-1} - \sqrt{(n+\lambda)\mathbf{P}_{k-1|k-1}}\ \right] \tag{6.13}$$

where $\lambda = \alpha^2(n+\kappa) - n$, $\kappa = 3-n$, $\alpha < 1$. Let $\mathcal{X}_{i,k|k-1} = \mathbf{f}\left(\mathcal{X}_{i,k-1|k-1}\right)$ represent the images of the sigma points resulting from the nonlinear propagation, and let $\mathcal{Z}_{i,k|k-1} = \mathbf{h}\left(\mathcal{X}_{i,k|k-1}\right)$ be their images. The expected value of the state after the propagation can be approximated by the following:

$$\begin{gathered} \bar{\mathbf{x}}_{k|k-1} = \mathrm{E}\left[\mathbf{x}_{k|k-1}\right] = \sum_{l=0}^{2n} \mathscr{W}_l^m \mathcal{X}_{l,k|k-1} \\ \text{where} \quad \mathscr{W}_0^m = \frac{\lambda}{n+\lambda},\ \mathscr{W}_0^c = \frac{\lambda}{n+\lambda} + (1-\alpha^2+\beta),\ \mathscr{W}_l^m = \mathscr{W}_l^m = \frac{\lambda}{2(n+\lambda)} \end{gathered} \tag{6.14}$$

Likewise, the expected measurement after the nonlinear propagation is given by:

$$\bar{\mathbf{z}}_{k|k-1} = \mathrm{E}\left[\mathbf{h}\left(\mathbf{x}_{k|k-1}\right)\right] = \sum_{l=0}^{2n} \mathscr{W}_l^m \mathcal{Z}_{l,k|k-1} \tag{6.15}$$

To ensure that the uncertainties remain bounded (i.e., the covariance envelope does not grow infinitely large) during the planning horizon, a constant frequency of measurement will be assumed. The guidance algorithm is run in parallel with the navigation algorithm. When a guidance sequence is initiated, the covariance is propagated forward in time without any measurement update (i.e., the covariance is always increasing as far as the guidance algorithm is concerned). It is therefore imperative to run the navigation algorithm at a higher frequency than the guidance to start the planning sequence with a small error and avoid the scenario depicted in in Fig. 6.5. The collected measurements are the relative states between the agents. If m measurements are collected at filter epoch k the covariance $\mathbf{P}_{k|k}^{-1}$ is given by the expression below:

$$\begin{aligned} \mathbf{P}_{k|k}^{-1} &= \mathbf{P}_{xx}^{-1} + \sum_{j=1}^{m} \left(\mathbf{P}_{xx}^{-1}\mathbf{P}_{xz}\right) \mathbf{R}^{-1} \left(\mathbf{P}_{xx}^{-1}\mathbf{P}_{xz}\right)^{\intercal} \\ \text{where} \quad \mathbf{P}_{xx} &= \mathbf{Q}_k + \sum_{l=0}^{2n} \mathscr{W}_l^c \left(\mathcal{X}_{l,k|k-1} - \bar{\mathbf{x}}_{k|k-1}\right) \left(\mathcal{X}_{l,k-1} - \bar{\mathbf{x}}_{k|k-1}\right)^{\intercal} \\ \mathbf{P}_{xx} &= \sum_{l=0}^{2n} \mathscr{W}_l^c \left(\mathcal{X}_{l,k|k-1} - \bar{\mathbf{x}}_{k|k-1}\right) \left(\mathcal{Z}_{l,k-1} - \bar{\mathbf{z}}_{k|k-1}\right)^{\intercal} \end{aligned} \tag{6.16}$$

Because real measurements are not generated in this analysis, the state distribution's mean values will be left unchanged. Although conservative (because the mean values are not updated), the resulting recursive guidance considers the uncertainties of all the agents when generating trajectories.

6.3.1 Application to a Six-Agent Formation

In the following example, a six-agent formation is considered, and the initial error of each agent is sampled from the following normal distribution:

$$\mathrm{P}_i = \operatorname{diag}\left([0.5\text{km}, 0.5\text{km}, 0.5\text{km}, 0.1\text{m/s}, 0.1\text{m/s}, 0.1\text{m/s}]\right), i \in \{1,2,3,4,5,6\}$$

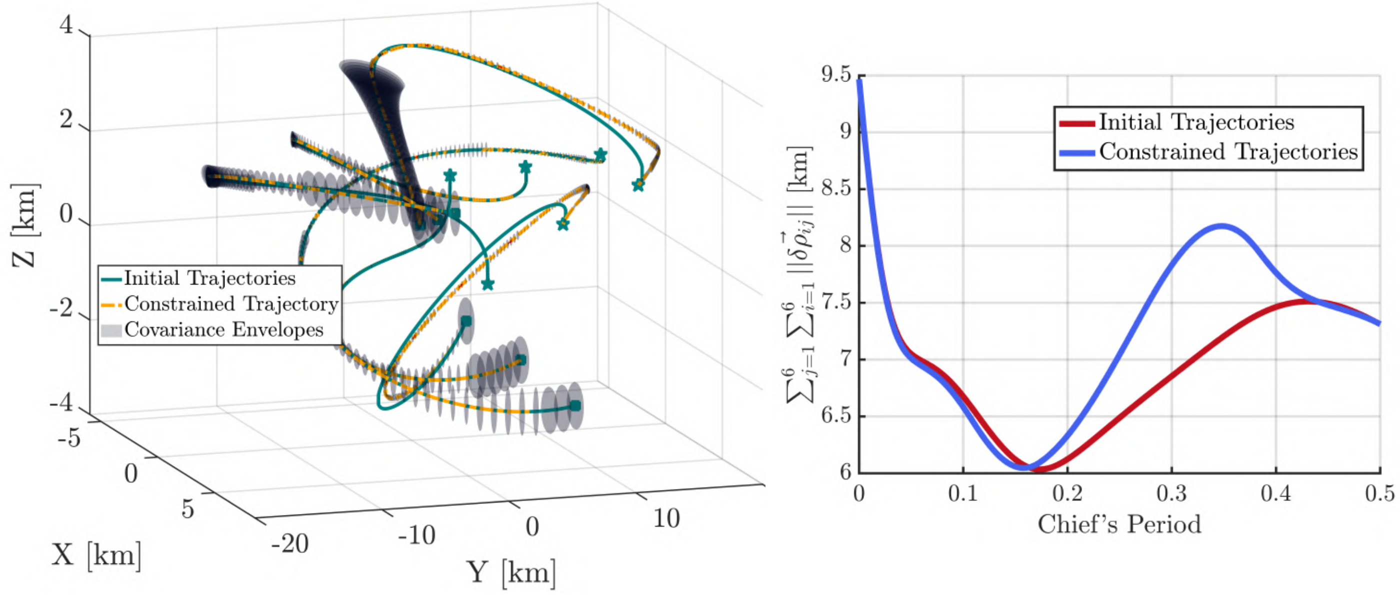
Table 6.3: Deputies's Orbit in Relative Orbital Elements ($k \in \{1,2,3,4,5,6\}$)

ΔOE	δa [km]	δe	δi [rad]	$\delta\Omega$ [rad]	$\delta\omega$ [rad]	δf [rad]
Deputy k (initial)	0.0	$-\frac{4k}{3a}$	$-\frac{10}{a}(-1)^k$	$-2\pi\text{E}{-5}(-1)^k$	0.0	$\pi\text{E}{-4}(-1)^k$
Deputy k (final)	0.0	$\frac{k}{2a}$	$\frac{10}{a}(-1)^k$	$\frac{k\pi}{3}\text{E}{-3}$	$\frac{k\pi}{15}\text{E}{-3}$	$\pi\text{E}{-6}$

The chief's parameters are the same as in Table 6.1, and the deputy's parameters are given in Table 6.3. The navigation algorithm received measurements once per minute, and the agent being updated is chosen at random. This implies that the covariance of all the agents do not shrink at the same rate, and some agents may have better knowledge of their whereabouts compared to their counterparts. The guidance update is performed every fifteen minutes. The filter's high frequency helps minimize the error at the beginning of each guidance sequence. Fig. 6.6(a) shows the trajectory of each agent before and after the replanning sequences. The dark ellipsoids along the paths are a plot of a the covariance matrices along each path at different filter epochs. As noted above, not all agents are updated simultaneously. This causes agent 1's covariance to increase during the initial portion its transfer. On the other hand, in the example studied here, other agents receive measurement more regularly causing their covariance envelopes to become very small (appearing as black dots along the constrained paths). Fig. 6.6(b) shows the average relative distance between the agents for the initial and the constrained trajectories.

Considering the error associated with the agent causes the constrained trajectories to move further apart from each other, especially later in the propagation where the uncertainty is the largest during the guidance update (i.e., earlier guidance updates only propagate the covariance with no measurement updates). This causes Agent 2 to deviate from Agent 3 even though their respective covariances are relatively small throughout the transfer, as shown in Fig. 6.7(a). In this scenario, Agent 2 identifies Agent 3 as an obstacles earlier in their transfers. This fact is confirmed by Fig. 6.7(b) where Agent 2's constrained trajectory moves away from Agent 3 relatively early in the simulation.

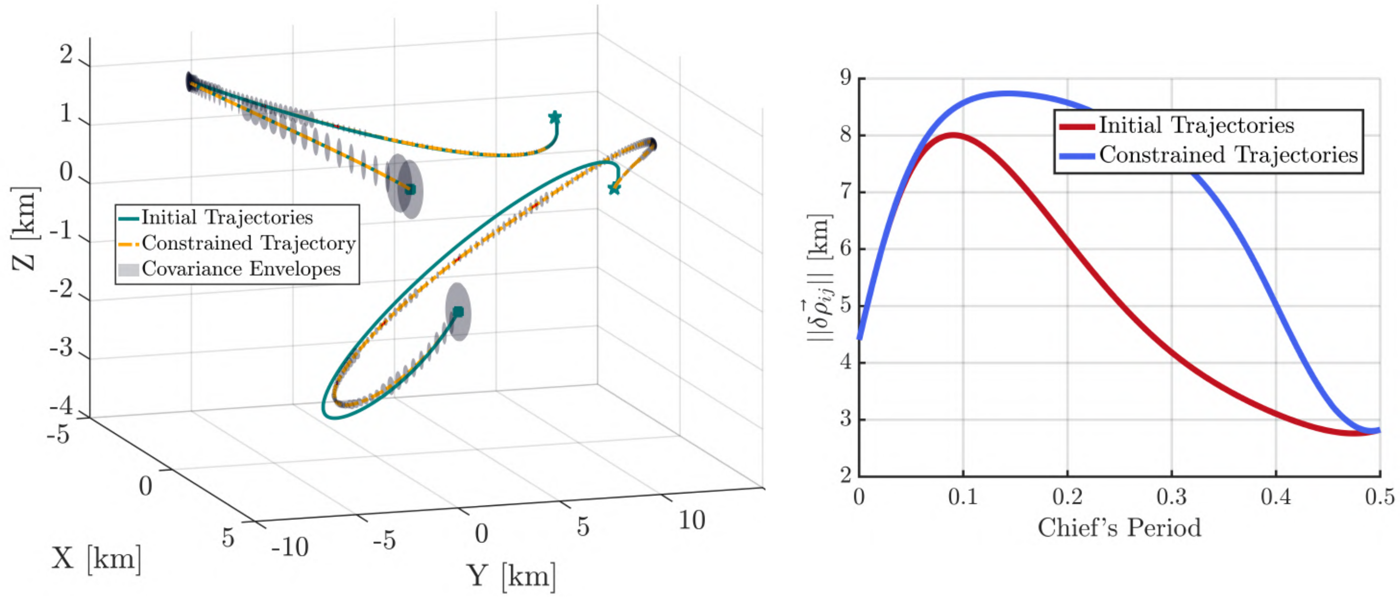
One last interesting case is that of Agents 3 and 6, where the initial and constrained paths are relatively unchanged. Analogous to the observations made above, Agent 6 does not enter the collision cone of Agent 3 until later in the simulation. By then, their covariances are small and the time remaining in the simulation is small. The time propagated covariances $\mathrm{P}_3(t_f)$ and $\mathrm{P}_6(t_f)$ do not intersect, implying that no course correction is needed.



(a) Trajectories And Covariance Envelopes Throughout Orbital Transfer

(b) Average Relative Separation

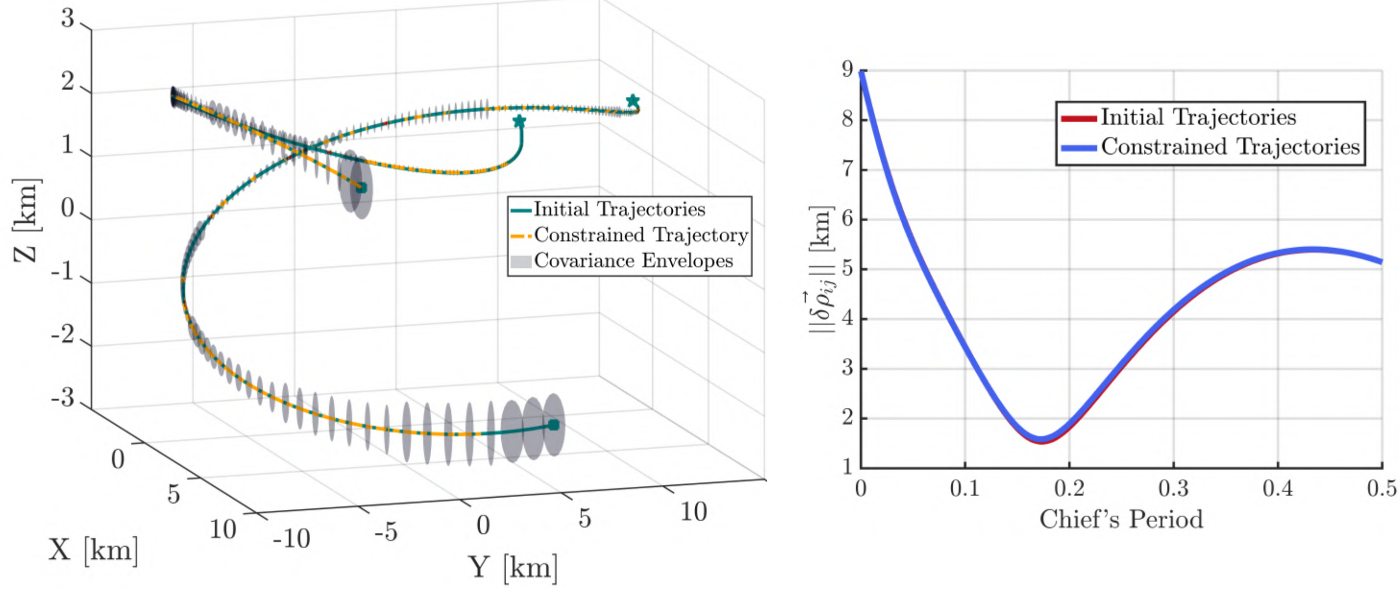
Figure 6.6: Stochastic Guidance for all the Agents



(a) Trajectories And Covariance Envelopes Throughout Orbital Transfer

(b) Relative Separation

Figure 6.7: Stochastic Guidance for Agents 2 and 3



(a) Trajectories And Covariance Envelopes Throughout Orbital Transfer

(b) Relative Separation

Figure 6.8: Stochastic Guidance for Agents 3 and 6

6.4 Conclusion

This chapter demonstrated how uncertainty could be integrated into geometric guidance. The probabilistic constraints are transformed into a deterministic condition on the mean of the distribution using a chance constraints formulation. All the system states were modeled as Gaussian distributions, the ellipsoid envelope associated with each covariance was imposed as a keep-out zone. A navigation filter was run in parallel with the Geometric Guidance to bound the growth of the normal distributions over time. And the navigation algorithm has to run at a higher frequency than the guidance algorithm to ensure the existence of feasible paths. Simulations show that the proposed algorithm can be implemented for distributed formation flying guidance. A replanning is initiated only when the covariance envelopes intersect, and the mean separation between the agents increases to accommodate for the uncertainty in the system.

Chapter 7

Conclusion and Future Work

7.1 Conclusion on Completed Work

Coming at the end of this work, we have developed a GNC algorithm that addresses all the problems we set out to solve in chapter 1. Specifically, a geometric guidance algorithm was developed using insights from nonlinear geometric control. The fuel-optimal problem was cast as a curve length minimization (geodesic) on a Riemannian manifold. It was shown how to impose constraints through the Riemannian metric. Also, the guidance was suitable for large formation control if the computation is distributed across every agent in a formation. The distributed guidance is a receding horizon algorithm that iteratively replans spacecraft's paths as they sense approaching obstacles. The guidance algorithm was extended to incorporate stochasticity using a chance constraints formulation. The stochastic-aware distributed guidance is shown to generate feasible paths in the presence of dynamics uncertainties and errors in the relative measurements between the agents. In the last chapter, a distributed relative navigation is presented to generate covariance information to be fed to the stochastic guidance. The aim of this dissertation was to introduce the novel frame work for distributed motion planning in a stochastic environment. Many possible extensions can stem from analysis conducted in the earlier chapters, and some of them are briefly discussed in the next section.

7.2 Extensions and Future Work

The work presented in this dissertation offers many opportunities for extensions, and some of them are discussed below:

(1) **Automate the tuning of the Geometric Guidance Algorithm:** The tuning parameters $\lambda > 0$ in the Riemmanian metric affect the convergence time as well as the quality of the solution (see Eq. (2.15)). A considerable amount of time was spent carefully selecting these terms for each of the studies conducted in this dissertation. In fact, the relative size between λ dictates how important some constraints are compared to others. But for systems in a cascade where some states depend on others (i.e., in SRP 6DoF plate, the derivative of the velocity depends on the attitude), the relative importance of constraints is not straightforward. The difficulty associated with the algorithm's tuning increases with the dimension of the problem. For example, the algorithm is less sensitive to the choice of penalties for a three degree of freedom of trajectory optimization versus a six degree of freedom. A logical extension to this research would be to automate the selection of these parameters to make the algorithm more autonomous. A preliminary research investigation literature review suggests that Reinforcement Learning techniques could provide an exciting avenue to address this problem. Namely, stochastic gradient descent (SGD) with momentum would offer an efficient way to select the penalty weights, especially for higher-dimensional problems [108, 132, 190]. If training data is available for different dynamical regimes, an SGD algorithm can make Geometric guidance more versatile to drastic changes in dynamical conditions. For an Earth-moon transfer, for example, the spacecraft will transition from an Earth-centric dynamic to a Restricted three-body problem and then to a moon-centric dynamics to the moon.

(2) **Analyze the reachable set of the SRP based guidance:** In Chapter 4, we use Lie brackets theory to argue the controllability of the relative motion of a flat plate deputy around a cannonball chief. It was shown that the system is controllable if differential

SRP is used to generate the propulsion capability between the two agents. However, Drag cannot arbitrarily change the relative states because it dissipates energy in the system. Let's assume that the area of the deputy is constant. As the two crafts dive closer to the surface, the atmospheric density, and consequently the magnitude of the drag force, increases. The value of SRP is constant at Earth (see Eq. (4.5)). There exists a critical altitude at which the magnitude of the differential SRP can not overcome the dissipative effect of Drag. The guidance algorithm will fail to converge onto a fuel-free trajectory at that altitude because such a path does not exist. Therefore, it is imperative to map out the reachable set (minimum height) achievable using SRP and Drag. Two potential ways to conduct this analysis are analyzing the eigenvalues of the accessibility matrix (defined in Eq. (4.14)) and performing a modal decomposition of the system dynamics [30, 164].

(3) **Extend the Stochastic guidance to hardware in the loop demonstration:** The Stochastic Distributed Motion Planning presented in Chapter 6 assumes that measurements are received at a constant frequency; Therefore, the covariance matrix, which is continuously updated, remains bounded. Although realistic, this assumption is used in place of hardware-in-the-loop implementation, where the collected measurements reflect changes in the system. Sampling actual measurements will allow for updating the mean of the state distributions and further constrain the states of the system. One can anticipate seeing less conservative solutions in the hardware-in-the-loop implementation. We also proposed in Chapter 5 a relative filter that could be used to generate the state estimates. It would be interesting to explore the applicability of such a filter in real-life scenarios and the adaptability of the guidance algorithm to other navigation algorithms.

Bibliography

[1] Study on maintaining formations during satellite formation flying based on sdre and lqr. Open Physics, 15(1):394–399, 2017.

[2] Muhammad Ahsan and Sarah Farrukh. A new type of shooting method for nonlinear boundary value problems. Alexandria Engineering Journal, (4):801–805, 2013.

[3] Salvatore Alfano and Daniel Oltrogge. Probability of collision: Valuation, variability, visualization, and validity. Acta Astronautica, 148:301–316, 2018.

[4] Kyle Alfriend, Srinivas Rao Vadali, Pini Gurfil, Jonathan How, and Louis Breger. Spacecraft formation flying: Dynamics, control and navigation, volume 2. Elsevier, 2009.

[5] Kyle T Alfriend and Hanspeter Schaub. Dynamic and Control of Spacecraft Formations: Challenges and Some Solutions. The Journal of the Astronautical Sciences, 48(2):249–267, 2000.

[6] Kyle T. Alfriend, Hanspeter Schaub, and Dong-Woo Gim. Gravitational perturbations, nonlinearity and circular orbit assumption effect on formation flying control strategies. In Proceedings of the Annual AAS Rocky Mountain Conference, pages 139–158, Breckenridge, CO, Feb. 2–6 2000.

[7] Roberto Alonso, Ju-Young Du, Declan Hughes, John L Junkins, and John L Crassidis. Relative navigation for formation flying of spacecraft. 2001.

[8] G. Antonelli, F. Arrichiello, S. Chiaverini, and G. S. Sukhatme. Observability analysis of relative localization for auvs based on ranging and depth measurements. In 2010 IEEE International Conference on Robotics and Automation, pages 4276–4281, 2010.

[9] J.-S. Ardaens and D. Fischer. Tandem-x autonomous formation flying system: Flight results. IFAC Proceedings Volumes, 44(1):709–714, 2011.

[10] F. Arrichiello, G. Antonelli, A. P. Aguiar, and A. Pascoal. Observability metric for the relative localization of auvs based on range and depth measurements: Theory and experiments. In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3166–3171, 2011.

[11] Peerawat Artitthang, Mingpei Lin, and Ming Xu. Optimal control for deployment of charged spacecraft formation flying based lqr. In 2019 5th International Conference on Control Science and Systems Engineering (ICCSSE), pages 35–39, 2019.

[12] Andrea Bacciotti and Gianna Stefani. On the relationship between global and local controllability. Mathematical systems theory, 16(1):79–91, 1983.

[13] Alexander Bahr, John J Leonard, and Maurice F Fallon. Cooperative localization for autonomous underwater vehicles. The International Journal of Robotics Research, 28(6):714–728, 2009.

[14] Alexander Bahr, Matthew R Walter, and John J Leonard. Consistent cooperative localization. In 2009 IEEE International Conference on Robotics and Automation, pages 3415–3422. IEEE, 2009.

[15] Lionel Baize, Martial Vanhove, Pascale Flagel, and Alberto Novelli. The ATV "Jules Verne" SuppLies the ISS.

[16] Muhammad Abu Bakr and Sukhan Lee. Distributed Multisensor Data Fusion under Unknown Correlation and Data Inconsistency. Sensors (Basel, Switzerland), 17(11):2472, oct 2017.

[17] Yaakov Bar-Shalom and Leon Campo. The effect of the common process noise on the two-sensor fused-track covariance. IEEE Transactions on Aerospace and Electronic Systems, (6):803–805, 1986.

[18] Richard H Battin and Robin M Vaughan. An elegant lambert algorithm. Journal of Guidance, Control, and Dynamics, 7(6):662–670, 1984.

[19] Charles A Beichman, NJ Woolf, and CA Lindensmith. The Terrestrial Planet Finder (TPF): a NASA Origins Program to search for habitable planets, volume 99. National Aeronautics and Space Administration, 1999.

[20] Kelly Holley-Bockelmann Jillian Bellovary, P. Bender, E. Berti, Warren Brown, R. Caldwell, N. Cornish, J. Darling, M. Digman, M. Eracleous, K. Gultekin, Z. Haiman, K. Holley-Bockelmann, J. Key, Shane Larson, Xin Liu, S. Mcwilliams, P. Natarajan, D. Shoemaker, K. Smith, M. Soares-Santos, and R. Stebbins. Getting ready for lisa: The data, support and preparation needed to maximize us participation in space-based gravitational wave science. arXiv: Instrumentation and Methods for Astrophysics, 2020.

[21] Alberto Bemporad and Manfred Morari. Robust model predictive control: A survey BT - Robustness in identification and control. pages 207–226, London, 1999. Springer London.

[22] Anthony M Bloch. Nonholonomic mechanics. In Nonholonomic mechanics and control, pages 207–276. Springer, 2003.

[23] Ugo Boscain and Mario Sigalotti. Introduction to controllability of non-linear systems. In Serena Dipierro, editor, Contemporary Research in Elliptic PDEs and Related Topics. Springer, 2019.

[24] Arnaud Boutonnet, Bénédicte Escudier, and Vincent Martinot. Direct Approach for Spacecraft Formation-Flying Positioning in Quasicircular Low Earth Orbit.

[25] Owen Brown and Paul Eremenko. Fractionated space architectures: A vision for responsive space. 4th Responsive Space Conference, AIAA, Reston, VA, 2006.

[26] Netsai Buaphim, Kawin Onsaard, Primporn So-ngoen, and Thitarie Rungratgasame. Some reviews on ranks of upper triangular block matrices over a skew field. In International Mathematical Forum, volume 13, pages 323–335, 2018.

[27] F. Bullo. Lectures on Network Systems. Kindle Direct Publishing, 1.3 edition, July 2019. With contributions by J. Cortes, F. Dörfler, and S. Martinez.

[28] Francesco Bullo and Andrew D. Lewis. Geometric Control of Mechanical Systems, volume 49 of Texts in AppLied Mathematics. Springer Verlag, New York-Heidelberg-Berlin, 2004.

[29] Francesco Bullo and Andrew D Lewis. Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems, volume 49. Springer, 2005.

[30] Ethan Burnett and Hanspeter Schaub. Modal Decomposition of Spacecraft Relative Motion in Quasi-Periodic Orbits. Aug. 2020. Paper AAS 20-506.

[31] Ethan R Burnett and Hanspeter Schaub. Spacecraft formation and orbit control using differential attitude-dependent solar radiation pressure. Advances in Space Research, 67(11):3396–3408, 2021.

[32] Ethan. R Burnett and Hanspeter Schaub. Study of highly perturbed spacecraft formation dynamics via approximation. Advances in Space Research, 67(11):3381–3395, 2021.

[33] Ethan R. Burnett and Hanspeter Schaub. Geometric perspectives on fundamental solutions in the linearized satellite relative motion problem. Acta Astronautica, 190:48–61, 2022.

[34] Luke Bury and Jay W McMahon. Landing Trajectories to Moons from the Unstable Invariant Manifolds of Periodic Libration Point Orbits. In AIAA Scitech 2020 Forum, AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, jan 2020.

[35] G C Calafiore and L El Ghaoui. On Distributionally Robust Chance-Constrained Linear Programs. Journal of Optimization Theory and Applications, 130(1):1–22, 2006.

[36] Mark E. Campbell. Planning algorithm for multiple satellite clusters. Journal of Guidance, Control, and Dynamics, 26(5):770–780, 2003.

[37] Mark E Campbell and Nisar R Ahmed. Distributed data fusion: Neighbors, rumors, and the art of collective knowledge. IEEE Control Systems Magazine, 36(4):83–109, 2016.

[38] Luis C Carrillo-Arce, Esha D Nerurkar, José L Gordillo, and Stergios I Roumeliotis. distributed multi-robot cooperative localization using covariance intersection. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1412–1417. IEEE, 2013.

[39] Thomas E. Carter. State transition matrices for terminal rendezvous studies: Brief survey and new example. Journal of Guidance, Control, and Dynamics, 21(1):148–155, 1998.

[40] Webster C. Cash. Maxim: micro-arcsecond x-ray imaging mission. In Michael Shao, editor, Interferometry in Space, volume 4852, pages 196 – 209. International Society for Optics and Photonics, SPIE, 2003.

[41] R Chai, A Savvaris, A Tsourdos, S Chai, and Y Xia. Stochastic Spacecraft Trajectory Optimization With the Consideration of Chance Constraints. IEEE Transactions on Control Systems Technology, 28(4):1550–1559, 2020.

[42] F Kenneth Chan et al. Spacecraft collision probability. Aerospace Press El Segundo, CA, 2012.

[43] Lingji Chen, Pablo O Arambel, and Raman K Mehra. Fusion under unknown correlation-covariance intersection as a special case. In Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002.(IEEE Cat. No. 02EX5997), volume 2, pages 905–912. IEEE, 2002.

[44] Zhiyong Chen and Jie Huang. Attitude tracking and disturbance rejection of rigid spacecraft by adaptive control. IEEE Transactions on Automatic Control, 54(3):600–605, 2009.

[45] Michelle Chernick. Optimal Impulsive Control of Spacecraft Relative Motion. PhD thesis, Ann Arbor, 2021.

[46] David Chichka. Dynamics of clustered satellites via orbital elements. 103:147–161, 01 2000.

[47] W. H. Clohessy and R. S. Wiltshire. Terminal guidance system for satellite rendezvous. Journal of the Aerospace Sciences, 27(9):653–658, 1960.

[48] Bruce A Conway and Stephen W Paris. Spacecraft Trajectory Optimization Using Direct Transcription and Nonlinear Programming. In Bruce A Conway, editor, Spacecraft Trajectory Optimization, Cambridge Aerospace Series, pages 37–78. Cambridge University Press, Cambridge, 2010.

[49] Bruce A Conway and John E Prussing. Orbital Mechanics. Oxford University Press, 1993.

[50] S. D'Amico and J. R. Carpenter. Satellite formation flying and rendezvous. In Position, Navigation, and Timing Technologies in the 21st Century: Integrated Satellite Navigation, Sensor Systems, and Civil Applications, pages 1921–1946, 2021.

[51] Simone D'Amico. Autonomous Formation Flying in Low Earth Orbit. PhD thesis, TU Delft, 2010.

[52] Luigi d'Arcio, Anders Karlsson, Marcos Bavdaz, and Thomas Jagemann. Search for extraterrestrial planets: the DARWIN mission. In Errico Armandillo, Josiane Costeraste, and Nikos Karafolas, editors, International Conference on Space Optics — ICSO 2006, volume 10567, pages 110 – 116. International Society for Optics and Photonics, SPIE, 2017.

[53] Ginalber Luiz de Oliveira Serra. Kalman filters-theory for advanced applications. 2018.

[54] Oscar De Silva, George K I Mann, and Raymond G Gosine. Observability Analysis of Relative Localization Filters Subjected to Platform Velocity Constraints. Journal of Dynamic Systems, Measurement, and Control, 139(5), mar 2017.

[55] W. D. Deininger, J. Bladt, W. Davis, A. Jarski, C. O'Quinn, B. S. Porter, E. H. Sholes, G. M. Brown, and K. C. Patel. Starlight spacecraft concept descriptions. In Proceedings, IEEE Aerospace Conference, volume 1, pages 1–254 vol.1, 2002.

[56] Elad Denenberg. Satellite closest approach calculation through chebyshev proxy polynomials. Acta Astronautica, 170:55–65, 2020.

[57] Zili Deng, Peng Zhang, Wenjuan Qi, Jinfang Liu, and Yuan Gao. Sequential covariance intersection fusion kalman filter. Information Sciences, 189:293–309, 2012.

[58] G. Di Mauro, M. Lawn, and R. Bevilacqua. Survey on guidance navigation and control requirements for spacecraft formation-flying missions. Journal of Guidance, Control, and Dynamics, 41(3):581–602, 2018.

[59] Zhe Dong and Qin Zhou. Robust kalman filtering for relative state estimation of spacecraft-formation maintenance. In International Conference on Space Information Technology, volume 5985, page 59854K. International Society for Optics and Photonics, 2006.

[60] Ian Elliott, Christopher Sullivan, Natasha Bosanac, Jeffrey R Stuart, and Farah Alibay. Designing Low-Thrust Trajectories for a SmallSat Mission to Sun–Earth L5. Journal of Guidance, Control, and Dynamics, 43(10):1854–1864, jul 2020.

[61] ESA. Tdx (tandem-x) - eoportal directory - satellite missions. https://directory.eoportal.org/web/eoportal/satellite-missions/t/tandem-x, Sep. 2006.

[62] Ariadna Farrés. Transfer orbits to L4 with a solar sail in the Earth-Sun system. Acta Astronautica, 137:78–90, 2017.

[63] Ariadna Farrés, Dave Folta, and Cassandra Webster. Using spherical harmonics to model solar radiation pressure accelerations. In AAS/AIAA Astrodynamics Specialist Conference, 2017.

[64] Ariadna Farrés and Àngel Jorba. Station Keeping Strategies for a Solar Sail in the Solar System BT - Recent Advances in Celestial and Space Mechanics. pages 83–115. Springer International Publishing, Cham, 2016.

[65] Ariadna Farres, Cassandra Webster, and David Folta. High fidelity modeling of SRP and its effect on the relative motion of Starshade and WFIRST. 2018 Space Flight Mechanics Meeting, pages 1–15, 2018.

[66] Stephen C Felter. An overview of distributed kalman filter techniques. In IEEE Technical Conference on Southern Tier, pages 79–87. IEEE, 1990.

[67] Wikimedia Foundation. Magnetospheric multiscale mission. https://en.wikipedia.org/wiki/Magnetospheric_Multiscale_Mission, Dec. 2021.

[68] Dietrich Franken and Andreas Hupper. Improved fast covariance intersection for distributed data fusion. In 2005 7th International Conference on Information Fusion, volume 1, pages 7–pp. IEEE, 2005.

[69] G Gaias and S D'Amico. Impulsive Maneuvers for Formation Reconfiguration Using Relative Orbital Elements. Journal of Guidance, Control, and Dynamics, 38(6):1036–1049, apr 2014.

[70] G. Gaias and S. D'Amico. Impulsive maneuvers for formation reconfiguration using relative orbital elements. Journal of Guidance, Control, and Dynamics, 38(6):1036–1049, 2015.

[71] Francisco Gavilan, Rafael Vazquez, and Eduardo F Camacho. Chance-constrained model predictive control for spacecraft rendezvous with disturbance estimation. Control Engineering Practice, 20(2):111–122, 2012.

[72] Yan Guo, Meiping Wu, Kanghua Tang, and Lu Zhang. Square-root unscented information filter and its application in sins/dvl integrated navigation. Sensors, 18(7), 2018.

[73] P. Gurfil, Jacob Herscovitz, and M. Pariente. The samson project – cluster flight and geolocation with three autonomous nano-satellites. 2012.

[74] Jeffery Hall. An Overview of the ST-7 Aerocapture Flight Test Experiment. In AIAA Atmospheric Flight Mechanics Conference and Exhibit, Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, aug 2002.

[75] Jeffery L Hall, Muriel A Noca, and Robert W Bailey. Cost-Benefit Analysis of the Aerocapture Mission Set. Journal of Spacecraft and Rockets, 42(2):309–320, mar 2005.

[76] S. J. Hall, B. Jorns, A. Gallimore, H. Kamhawi, T. Haag, Jonathan A. Mackey, J. Gilland, P. Y. Peterson, and M. J. Baird. High-power performance of a 100-kw class nested hall thruster. 2017.

[77] Ahmed M Hassan and Haithem E Taha. Geometric control formulation and nonlinear controllability of airplane flight dynamics. Nonlinear Dynamics, 88(4):2651–2669, 2017.

[78] Russell L. Herman. Euler equation and geodesics. http://people.uncw.edu/hermanr/GRcosmo/euler-equation-geodesics.pdf.

[79] R. Hermann and A. Krener. Nonlinear controllability and observability. IEEE Transactions on Automatic Control, 22(5):728–740, 1977.

[80] Erik A. Hogan and Hanspeter Schaub. Relative motion control for two-spacecraft electrostatic orbit corrections. Advances in the Astronautical Sciences, 142(1):967–986, 2012.

[81] S. P. Hughes and L. M. Mailhe. A preliminary formation flying orbit dynamics analysis for leonardo-brdf. In 2001 IEEE Aerospace Conference Proceedings (Cat. No.01TH8542), volume 2, pages 2/579–2/596 vol.2, 2001.

[82] Steven Hughes, , Steven P. Hughes, and Christopher D. Hall. Mission performancemeasures for spacecraft formation flying. In in Flight Mechanics Symposium, 1999.

[83] Steven P Hughes and Christopher D Hall. Optimal Configurations for Rotating Spacecraft Formations. The Journal of the Astronautical Sciences, 48(2):225–247, 2000.

[84] A. Imani, M. Bahrami, and B. Ebrahimi. Optimal sliding mode control for spacecraft formation flying. In The 2nd International Conference on Control, Instrumentation and Automation, pages 38–43, 2011.

[85] Alberto Isidori. Nonlinear Control Systems: An Introduction (2nd Ed.). Springer-Verlag, Berlin, Heidelberg, 1985.

[86] Dario Izzo and Lorenzo Pettazzi. Autonomous and Distributed Motion Planning for Satellite Swarm. Journal of Guidance, Control, and Dynamics, 30(2):449–459, mar 2007.

[87] B Jakubczyk. Introduction to geometric nonlinear control; Controllability and Lie bracket. Technical report, International Atomic Energy Agency (IAEA), 2002.

[88] Frédéric Jean. Control of nonholonomic systems: from sub-Riemannian geometry to motion planning. Springer, 2014.

[89] Jürgen Jost and Jèurgen Jost. Riemannian geometry and geometric analysis, volume 42005. Springer, 2008.

[90] Velimir Jurdjevic. Geometric Control Theory. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1996.

[91] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, 82(1):35–45, 03 1960.

[92] R. E. Kalman and R. S. Bucy. New Results in Linear Filtering and Prediction Theory. Journal of Basic Engineering, 83(1):95–108, 03 1961.

[93] Aniketh Kalur, Kavyashree Shivakumar, Matthias Schmid, and John L. Crassidis. Adaptive control for spacecraft formation flying with solar radiation pressure and reduction of secular drift. Proceedings of the American Control Conference, 2016-July:5830–5835, 2016.

[94] B. Kang, F. Y. Hadaegh, and D. Scharf. On the validity of the double integrator approximation in deep space formation flying. 2002.

[95] Hermann Kaptui Sipowa and Jay McMahon. First order approximation of the effects of solar radiation pressure on relative motion using a linearized representation of relative orbital elements. In AAS/AIAA Astrodynamics Specialist Conference, volume 167, Aug. 2019. Paper AAS 18-474.

[96] Hermann Kaptui Sipowa and Jay McMahon. Analysis of srp-disturbed relative motion using geometric nonlinear control theory. In AAS/AIAA Astrodynamics Specialist Conference, volume 175, Aug. 2020. Paper AAS 20-474.

[97] Hermann Kaptui Sipowa and Jay McMahon. 6dof nonlinear guidance for spacecraft formation flying. In AAS/AIAA Astrodynamics Specialist Conference, Aug. 2021. Paper AAS 21-768.

[98] Hermann Kaptui Sipowa and Jay McMahon. Fuel-Optimal Geometric Guidance for Spacecraft Formation Flying. Journal of Guidance, Control, and Dynamics, 2021. Manuscript submitted for publication.

[99] Hermann Kaptui Sipowa and Jay McMahon. Distributed and stochastic guidance for multi-agent systems. Journal of Guidance, Control, and Dynamics, 2022. Manuscript in preparation.

[100] Hermann Kaptui Sipowa and Jay McMahon. Distributed Estimator for Spacecraft Cooperative Localization. Journal of Guidance, Control, and Dynamics, pages 1–6, feb 2022.

[101] Hermann Kaptui Sipowa and Jay McMahon. Fuel-efficient distributed path planning for spacecraft formation flying. In IEEE Aerospace Conference, Mar. 2022. Paper 2393.

[102] Hermann Kaptui Sipowa and Jay McMahon. Fuel-optimal geometric path planning algorithm for spacecraft formation flying. Journal of Guidance, Control, and Dynamics, 2022. Manuscript submitted for publication.

[103] Hermann Kaptui Sipowa and Jay McMahon. Stochastic distributed geometric motion planner for spacecraft formation flying. In International Workshop on Satellite Constellations & Formation Flying, Jun. 2022. (Submitted).

[104] Evan Kaufman, T Alan Lovell, and Taeyoung Lee. Nonlinear Observability for Relative Orbit Determination with Angles-Only Measurements. The Journal of the Astronautical Sciences, 63(1):60–80, 2016.

[105] Usman A Khan and José MF Moura. Distributing the kalman filter for large-scale systems. IEEE Transactions on Signal Processing, 56(10):4919–4935, 2008.

[106] Amir Khodabandeh, Peter J.G. Teunissen, and Safoora Zaminpardaz. Consensus-based distributed filtering for gnss. In Ginalber Luiz de Oliveira Serra, editor, Kalman Filters, chapter 14. IntechOpen, Rijeka, 2018.

[107] Son-Goo Kim, John L Crassidis, Yang Cheng, Adam M Fosbury, and John L Junkins. Kalman filtering for relative spacecraft attitude and position estimation. Journal of Guidance, Control, and Dynamics, 30(1):133–143, 2007.

[108] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[109] John Klingner, Nisar Ahmed, and Nikolaus Correll. Fault-tolerant covariance intersection for localizing robot swarms. In Distributed Autonomous Robotic Systems, pages 485–497. Springer, 2019.

[110] Adam W. Koenig and Simone D'Amico. Robust and safe n-spacecraft swarming in perturbed near-circular orbits. Journal of Guidance, Control, and Dynamics, 41(8):1643–1662, 2018.

[111] Edmund M C Kong, David W Miller, and Raymond J Sedwick. Exploiting Orbital Dynamics for Aperture Synthesis Using Distributed Satellite Systems: Applications to a Visible Earth Imager System. The Journal of the Astronautical Sciences, 47(1):53–75, 1999.

[112] Holger Krag, Tim Flohrer, and Stijn Lemmens. Consideration of Space Debris Mitigation Requirements in the Operation of LEO Missions, pages 413–429.

[113] Ryo Kurazume, Shigemi Nagata, and Shigeo Hirose. Cooperative positioning with multiple robots. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation, pages 1250–1257. IEEE, 1994.

[114] Steven M. LaValle. Planning algorithms. Cambridge university press, 2006.

[115] Steven M LaValle. Planning algorithms. Cambridge university press, 2009.

[116] Steven M. LaValle. Nonholonomic system theory. http://planning.cs.uiuc.edu/node824.html, 2012. accessed July 4, 2019.

[117] Deok-Jin Lee. Unscented information filtering for distributed estimation and multiple sensor fusion. In AIAA Guidance, Navigation and Control Conference and Exhibit, page 7426, 2008.

[118] John M Lee. Introduction to Riemannian manifolds. Springer, 2018.

[119] T. Lee, M. Leok, and N. H. McClamroch. Geometric tracking control of a quadrotor uav on se(3). In 49th IEEE Conference on Decision and Control (CDC), pages 5420–5425, 2010.

[120] T. Lee, N. H. McClamroch, and M. Leok. Optimal control of a rigid body using geometrically exact computations on se(3). In Proceedings of the 45th IEEE Conference on Decision and Control, pages 2710–2715, 2006.

[121] Fátima Silva Leite. Introduction to geometric control theory - controllability and lie bracket. http://www.mat.uc.pt/ mmc/courses/PhD_Seminar/Fatima_Leite.pdf, 2010. accessed July 4, 2019.

[122] G Leitmann. Guaranteed avoidance strategies. Journal of Optimization Theory and Applications, 32(4):569–576, 1980.

[123] Andrew D Lewis. A brief on controllability of nonlinear systems. http://www.princeton.edu/ biswadip/Teaching/RAssign_NL_Cont__ALewis.pdf, 2001. accessed July 4, 2019.

[124] Li Li and Yuanqing Xia. Stochastic stability of the unscented Kalman filter with intermittent observations. Automatica, 48(5):978–981, 2012.

[125] Daniel Liberzon. Calculus of variations and optimal control theory, a concise introduction. http://liberzon.csl.illinois.edu/teaching/cvoc/node87.html, 2010. accessed July 4, 2019.

[126] J.-C. Liou and N.L. Johnson. Instability of the present leo satellite populations. Advances in Space Research, 41(7):1046–1053, 2008.

[127] G. Liu and S. Zhang. A survey on formation control of small satellites. Proceedings of the IEEE, 106(3):440–457, 2018.

[128] Shenyu Liu, Yinai Fan, and Mohamed-Ali Belabbas. Affine geometric heat flow and motion planning for dynamic systems. IFAC-PapersOnLine, 52(16):168–173, 2019.

[129] Mary Kae Lockwood. Titan Aerocapture Systems Analysis. In 39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Joint Propulsion Conferences. American Institute of Aeronautics and Astronautics, jul 2003.

[130] Ian Loefgren, Nisar Ahmed, Eric Frew, Christoffer Heckman, and Sean Humbert. Scalable event-triggered data fusion for autonomous cooperative swarm localization. In 2019 22th International Conference on Information Fusion (FUSION), pages 1–8. IEEE, 2019.

[131] Timothy J. Logue and Joseph N. Pelton. Overview of Commercial Small Satellite Systems in the "New Space" Age, pages 69–86. Springer International Publishing, Cham, 2020.

[132] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[133] Daan Maessen and Eberhard Gill. Relative state estimation and observability analysis for formation flying satellites. Journal of Guidance, Control, and Dynamics, 35(1):321–326, 2012.

[134] Daan Maessen and Eberhard Gill. Relative state estimation and observability analysis for formation flying satellites. Journal of Guidance, Control, and Dynamics, 35(1):321–326, 2012.

[135] Damián Marelli, Mohsen Zamani, Minyue Fu, and Brett Ninness. Distributed kalman filter in a network of linear systems. Systems & Control Letters, 116:71–77, 2018.

[136] Marin I Marin, Ravi P Agarwal, and S R Mahmoud. Nonsimple material problems addressed by the Lagrange's identity. Boundary Value Problems, 2013(1):135, 2013.

[137] Jerrold Marsden, Wang Koon, Richard Murray, and Josep Masdemont. J2 dynamics and formation flight.

[138] Jerrold E Marsden and Tudor S Ratiu. Introduction to mechanics and symmetry. Physics Today, 48(12):65, 1995.

[139] A. Martinelli and R. Siegwart. Observability analysis for mobile robot localization. In 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1471–1476, 2005.

[140] C. Mathieu and A. Weigel. Assessing the flexibility provided by fractionated spacecraft. Space, 2005.

[141] Kai Matsuka, Aaron O. Feldman, Elena S. Lupu, Soon-Jo Chung, and Fred Y. Hadaegh. distributed formation pose estimation for spacecraft swarms. Advances in Space Research, 2020.

[142] Leonel Mazal and Pini Gurfil. Closed-loop distance-keeping for long-term satellite cluster flight. Acta Astronautica, 94(1):73–82, 2014.

[143] Jonathan C. McDowell. The low earth orbit satellite population and impacts of the SpaceX starlink constellation. The Astrophysical Journal, 892(2):L36, apr 2020.

[144] Jay W McMahon and Daniel J Scheeres. Linearized lambert's problem solution. Journal of Guidance, Control, and Dynamics, pages 2205–2218, 2016.

[145] Daniel Morgan, Soon-Jo Chung, Lars Blackmore, Behcet Acikmese, David Bayard, and Fred Y. Hadaegh. Swarm-keeping strategies for spacecraft under j2 and atmospheric drag perturbations. Journal of Guidance, Control, and Dynamics, 35(5):1492–1506, 2012.

[146] Rafail Murtazin and Nikolay Petrov. Short profile for the human spacecraft soyuz-tma rendezvous mission to the iss. Acta Astronautica, 77:77–82, 2012.

[147] NASA. Gravity recovery and climate experiment. https://www.jpl.nasa.gov/missions/gravity-recovery-and-climate-experiment-grace, Mar. 2002.

[148] NASA. Magnetospheric multiscale mission. https://www.nasa.gov/feature/new-simplex-mission-to-send-smallsats-on-longest-deep-space-journey-to-date, Sep. 2020.

[149] National Geophysical Data Center. U.S. standard atmosphere (1976). Planetary and Space Science, 40(4):553–554, 1992.

[150] Wolfgang Niehsen. Information fusion based on fast covariance intersection filtering. In Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002.(IEEE Cat. No. 02EX5997), volume 2, pages 901–904. IEEE, 2002.

[151] Henk Nijmeijer and Arjan Van der Schaft. Nonlinear dynamical control systems, volume 175. Springer, 1990.

[152] Jorge Nocedal and Stephen Wright. Numerical optimization. Springer Science & Business Media, 2006.

[153] K Oguri, M Ono, and J W McMahon. Convex Optimization over Sequential Linear Feedback Policies with Continuous-time Chance Constraints. In 2019 IEEE 58th Conference on Decision and Control (CDC), pages 6325–6331, 2019.

[154] Kenshiro Oguri, Gregory Lantoine, and Jay W McMahon. Solar Sailing Primer Vector Theory: Indirect Trajectory Optimization with Practical Mission Considerations. Journal of Guidance, Control, and Dynamics, pages 1–9, sep 2021.

[155] Kenshiro Oguri and Jay W McMahon. Solar Radiation Pressure–Based Orbit Control with Application to Small-Body Landing. Journal of Guidance, Control, and Dynamics, 43(2):195–211, nov 2019.

[156] Yoshiaki Ohkami and Isao Kawano. Autonomous rendezvous and docking by engineering test satellite vii: a challenge of japan in guidance, navigation and control—breakwell memorial lecture. Acta Astronautica, 53(1):1–8, 2003.

[157] Reza Olfati-Saber. Distributed kalman filter with embedded consensus filters. In Proceedings of the 44th IEEE Conference on Decision and Control, pages 8179–8184. IEEE, 2005.

[158] Reza Olfati-Saber. Distributed kalman filtering for sensor networks. In 2007 46th IEEE Conference on Decision and Control, pages 5492–5498. IEEE, 2007.

[159] OneWeb. One world. https://www.oneweb.world/. Accessed: 03/09/2021.

[160] Michael Ouimet, Nisar Ahmed, and Sonia Martínez. Event-based cooperative localization using implicit and explicit measurements. In 2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pages 246–251. IEEE, 2015.

[161] Michael Ouimet, David Iglesias, Nisar Ahmed, and Sonia Martínez. Cooperative robot localization using event-triggered estimation. Journal of Aerospace Information Systems, 15(7):427–449, 2018.

[162] Gianfranco Parlangeli, Paola Pedone, and Giovanni Indiveri. Relative pose observability analysis for 3d nonholonomic vehicles based on range measurements only. IFAC Proceedings Volumes, 45(27):182–187, 2012.

[163] Russell P. Patera. Satellite collision probability for nonlinear relative motion. Journal of Guidance, Control, and Dynamics, 26(5):728–733, 2003.

[164] Marielle Pellegrino and Daniel J Scheeres. Reachability of a Passive Solar Sail in Earth Orbit. Journal of Guidance, Control, and Dynamics, 44(2):360–369, nov 2020.

[165] G. Purcell, D. Kuang, S. Lichten, S. C. Wu, and L. Young. Autonomous Formation Flyer (AFF) Sensor Technology Development. Telecommunications and Mission Operations Progress Report, 134:1–21, April 1998.

[166] Tong Qin, Dong Qiao, and Malcolm Macdonald. Relative orbit determination using only intersatellite range measurements. Journal of Guidance, Control, and Dynamics, 42(3):703–710, 2019.

[167] Stephane Rousseau, Etienne Perot, Claude Graves, James Masciarelli, and Eric Queen. Aerocapture Guidance Algorithm Comparison Campaign. In AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, aug 2002.

[168] Timothy E. Rumford. Demonstration of autonomous rendezvous technology (DART) project summary. In Peter Tchoryk Jr. and James Shoemaker, editors, Space Systems Technology and Operations, volume 5088, pages 10 – 19. International Society for Optics and Photonics, SPIE, 2003.

[169] R D Russell and L F Shampine. A collocation method for boundary value problems. Numerische Mathematik, 19(1):1–28, 1972.

[170] Chris Sabol, Rich Burns, and Craig A. McLaughlin. Satellite formation flying design and evolution. Journal of Spacecraft and Rockets, 38(2):270–278, 2001.

[171] A. Saccon, A. P. Aguiar, A. J. Häusler, J. Hauser, and A. M. Pascoal. Constrained motion planning for multiple vehicles on se(3). In 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), pages 5637–5642, 2012.

[172] Shankar Sastry. Nonlinear systems: analysis, stability, and control, volume 10. Springer Science & Business Media, 2013.

[173] D. P. Scharf, F. Y. Hadaegh, and S. R. Ploen. A survey of spacecraft formation flying guidance and control. part ii: control. In Proceedings of the 2004 American Control Conference, volume 4, pages 2976–2985 vol.4, 2004.

[174] Hanspeter Schaub. Relative orbit geometry through classical orbit element differences. Journal of Guidance, Control and Dynamics, 27(5):839–848, Sept.–Oct. 2004.

[175] Hanspeter Schaub and Kyle T. Alfriend. Impulsive feedback control to establish specific mean orbit elements of spacecraft formations. Journal of Guidance, Control, and Dynamics, 24(4):739–745, 2001.

[176] Hanspeter Schaub and Kyle T. Alfriend. j_2 invariant relative orbits for spacecraft formations. Celestial Mechanics and Dynamical Astronomy, 79(2):77–95, 2001.

[177] Hanspeter Schaub and John L. Junkins. Analytical Mechanics of Space Systems. AIAA Education Series, Reston, VA, 4th edition, 2018.

[178] C. Schiff, D. Rohrbaugh, and J. Bristow. Formation flying in elliptical orbits. In 2000 IEEE Aerospace Conference. Proceedings (Cat. No.00TH8484), volume 7, pages 37–47 vol.7, 2000.

[179] Markus Schlotterer, Eviatar Edlerman, F. Fumenti, Pini Gurfil, Stephan Theil, and Hao Zhang. On-ground testing of autonomous guidance for multiple satellites in a cluster. In 8th International Workshop on Satellite Constellations and Formation Flying, June 2015.

[180] Markus Schlotterer and Sergej Novoschilov. On-ground path planning experiments for multiple satellites. In 23rd International Symposium on Space Flight Dynamics 2012, October 2012.

[181] Romain Serra, Denis Arzelier, Mioara Joldes, Jean-Bernard Lasserre, Aude Rondepierre, and Bruno Salvy. Fast and accurate computation of orbital collision probability for short-term encounters. Journal of Guidance, Control, and Dynamics, 39(5):1009–1021, 2016.

[182] Kamran Shahid and Pini Gurfil. Top-Level Control of Disaggregated Satellites: Cluster Maintenance and Scatter/Re-gather Maneuvers.

[183] G Singh and F Hadaegh. Collision avoidance guidance for formation-flying applications. In AIAA Guidance, Navigation, and Control Conference and Exhibit, Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, aug 2001.

[184] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, and S.S. Sastry. Kalman filtering with intermittent observations. IEEE Transactions on Automatic Control, 49(9):1453–1464, 2004.

[185] Hermann Kaptui Sipowa, Jay W. McMahon, and Taralicin Deka. Distributed unscented-information kalman filter (uikf) for cooperative localization in spacecraft formation flying. In AIAA Scitech 2020 Forum. Paper AIAA-2020-1917.

[186] Stefania Soldini, Camilla Colombo, and Scott J I Walker. Solar Radiation Pressure Hamiltonian Feedback Control for Unstable Libration-Point Orbits. Journal of Guidance, Control, and Dynamics, 40(6):1374–1389, mar 2017.

[187] Joern Spurmann and Simone D'Amico. Proximity operations of on-orbit servicing spacecraft using an eccentricity/inclination vector separation. In 22nd International Symposium on Spaceflight Dynamics, 2011.

[188] H. Sussmann. A general theorem on local controllability. Siam Journal on Control and Optimization, 25:158–194, 1987.

[189] Hector J Sussmann. A general theorem on local controllability. SIAM Journal on Control and Optimization, 25(1):158–194, 1987.

[190] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML'13. JMLR.org, 2013.

[191] Brian D. Swinburne, Martin Davidson, and Stephen Kemble. Control of close formations in low earth orbit. Acta Astronautica, 51(1):427–438, 2002.

[192] B. D. Tapley, S. Bettadpur, M. Watkins, and C. Reigber. The gravity recovery and climate experiment: Mission overview and early results. Geophysical Research Letters, 31(9), 2004.

[193] N E Du Toit and J W Burdick. Probabilistic Collision Checking With Chance Constraints. IEEE Transactions on Robotics, 27(4):809–815, 2011.

[194] N E Du Toit and J W Burdick. Robot Motion Planning in Dynamic, Uncertain Environments. IEEE Transactions on Robotics, 28(1):101–115, 2012.

[195] J Tschauner and P Hempel. Rendezvous zu einem in elliptischer bahn umlaufenden ziel. Astronautica Acta, 11(2):104–109, 1965.

[196] John Tsitsiklis, Dimitri Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. IEEE transactions on automatic control, 31(9):803–812, 1986.

[197] Thomas Uriot, Dario Izzo, Luís F Simões, Rasit Abay, Nils Einecke, Sven Rebhan, Jose Martinez-Heras, Francesca Letizia, Jan Siminski, and Klaus Merz. Spacecraft collision avoidance challenge: Design and results of a machine learning competition. Astrodynamics, 2021.

[198] S. Vadali, H. Schaub, and K. Alfriend. Initial conditions and fuel-optimal control for formation flying of satellites.

[199] David A Vallado. Fundamentals of astrodynamics and applications, volume 12. Springer Science & Business Media, 2001.

[200] David A. Vallado and Wayne D. McClain. Fundamentals of astrodynamics and applications. Microcosm Press, Hawthorne, CA, 4th edition, 2013.

[201] Jozef C. Van Der Ha and Vaios J. Lappas. Long-term attitude drift of spinning spacecraft under solar radiation torques. Advances in the Astronautical Sciences, 123 II(5):1123–1142, 2006.

[202] Steven van Leeuwen. A Chance Constraint Predictive Control and Estimation Framework for Spacecraft Descent with Field Of View Constraints. In AIAA Scitech 2021 Forum, AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, jan 2021.

[203] Matthew A. Vavrina, C. Eugene Skelton, Keith D. Deweese, Bo J. Naasz, David E. Gaylor, and Christopher D'souza. Safe rendezvous trajectory design for the restore-l mission. In Spaceflight Mechanics 2019, Advances in the Astronautical Sciences, pages 3649–3668. Univelt Inc., January 2019.

[204] GERALD WALBERG. Aerocapture for manned Mars missions - Status and challenges. In 18th Atmospheric Flight Mechanics Conference, Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, aug 1991.

[205] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373), pages 153–158. Ieee, 2000.

[206] Thumeera R Wanasinghe, George K I. Mann, and Raymond G Gosine. Relative Localization Approach for Combined Aerial and Ground Robotic System. Journal of Intelligent & Robotic Systems, 77(1):113–133, 2015.

[207] Chieh-Chih Wang, Charles Thorpe, Sebastian Thrun, Martial Hebert, and Hugh Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. The International Journal of Robotics Research, 26(9):889–916, 2007.

[208] Shuquan Wang and Hanspeter Schaub. Spacecraft Collision Avoidance Using Coulomb Forces with Separation Distance and Rate Feedback. Journal of Guidance, Control, and Dynamics, 31(3):740–750, may 2008.

[209] Xiaogang Wang, Wutao Qin, Yuliang Bai, and Naigang Cui. A novel distributed relative navigation algorithm for spacecraft formation flying. Aerospace Science and Technology, 48:28–36, 2016.

[210] Yu Wang, Xiaogang Wang, and Naigang Cui. Robust decentralised state estimation for formation flying spacecraft. IET Radar, Sonar & Navigation, 13(5):814–823, 2019.

[211] M. Watterson, T. Smith, and V. Kumar. Smooth trajectory generation on se(3) for a free flying space robot. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5459–5466, 2016.

[212] T. Weismuller and M. Leinz. Gn&c technology demonstrated by the orbital express autonomous rendezvous and capture sensor system. 2006.

[213] P Wercinski, W Henline, H Tran, F Milos, P Papadopoulos, Y.-K. Chen, E Venkatapathy, M Tauber, P Wercinski, W Henline, H Tran, F Milos, P Papadopoulos, Y.-K. Chen, E Venkatapathy, and M Tauber. Trajectory, aerothermal conditions, and thermal protection system mass for the mars 2001 aerocapture mission. In 35th Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics, jan 1997.

[214] Luke B. Winternitz, William A. Bamford, Samuel R. Price, J. Russell Carpenter, Anne C. Long, and Mitra Farahmand. Global positioning system navigation above 76,000 km for nasa's magnetospheric multiscale mission. NAVIGATION, 64(2):289–300, 2017.

[215] Yun Hua Wu, Xi Bin Cao, Yan Jun Xing, Peng Fei Zheng, and Shi Jie Zhang. Relative motion coupled control for formation flying spacecraft via convex optimization. Aerospace Science and Technology, 14(6):415–428, 2010.

[216] Ming Xu, Yuying Liang, and Xiaoyu Fu. Formation flying on quasi-halo orbits in restricted Sun–Earth/Moon system. Aerospace Science and Technology, 67:118–125, 2017.

[217] Dan Xue, Xibin Cao, and Yunhua Wu. distributed determination of relative orbit for formation flying satellite. In 2006 1st International Symposium on Systems and Control in Aerospace and Astronautics, pages 6–pp. IEEE, 2006.

[218] Koji Yamanaka and Finn Ankersen. New state transition matrix for relative motion on an arbitrary elliptical orbit. Journal of Guidance, Control, and Dynamics, 25(1):60–66, 2002.

[219] H. Zhang and P. Gurfil. Cooperative orbital control of multiple satellites via consensus. IEEE Transactions on Aerospace and Electronic Systems, 54(5):2171–2188, 2018.

[220] Peng Zhang, Wenjuan Qi, and Zili Deng. Covariance intersection fusion kalman smoother for systems with colored measurement noises. Procedia Engineering, 29:616–622, 2012.

Appendix A

Equivalence Between Geodesic and Euler-Lagrange Equation

The Euler-Lagrange equations for an n-dimesional manifold are written as

$$\frac{d}{dt}\left(\frac{L(\vec{\boldsymbol{x}},\dot{\vec{\boldsymbol{x}}})}{\partial \dot{x}^i}\right) = \frac{\partial L(\vec{\boldsymbol{x}},\dot{\vec{\boldsymbol{x}}})}{\partial x^i} \tag{A.1}$$

Here, the Einstein summation convention applies and summation is implied over repeated indices that occur as a subscript and superscript pair. x^i and $\dot{x}^i$ are the i^{th} local coordinate of representation of $\vec{\boldsymbol{x}}$ and $\dot{\vec{\boldsymbol{x}}}$, respectively. The Lagrangian of our problem is the length of the curve $\vec{\boldsymbol{\gamma}}$ on the Riemannian manifold $\mathscr{M}$ and is written in terms of local coordinates as

$$L(\vec{\boldsymbol{x}},\dot{\vec{\boldsymbol{x}}}) = \frac{1}{2} g_{ik}\dot{x}^i\dot{x}^k \tag{A.2}$$

where g_{ik} are the components of the Riemannian metric $\mathbf{G}$. Plugging Eq. (A.2) into Eq. (A.1) yields the following expressions:

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial(\frac{1}{2}g_{mk}\dot{x}^m\dot{x}^k)}{\partial \dot{x}^i}\right) &= \frac{\partial(\frac{1}{2}g_{mk}\dot{x}^m\dot{x}^k)}{\partial x^i}\\
\implies g_{ik}\ddot{x}^k + \frac{\partial g_{ik}}{\partial x^m}\dot{x}^m\dot{x}^k &= \frac{1}{2}\frac{\partial g_{mk}}{\partial x^i}\dot{x}^m\dot{x}^k\\
\implies g_{ik}\ddot{x}^k + \frac{1}{2}\left(\frac{\partial g_{ik}}{\partial x^m} + \frac{\partial g_{im}}{\partial x^k} - \frac{\partial g_{mk}}{\partial x^i}\right)\dot{x}^m\dot{x}^k &= 0\\
\implies \ddot{x}^i + \frac{1}{2}g^{ij}\left(\frac{\partial g_{jk}}{\partial x^m} + \frac{\partial g_{jm}}{\partial x^k} - \frac{\partial g_{mk}}{\partial x^j}\right)\dot{x}^m\dot{x}^k &= 0\\
\implies \frac{d^2x^i}{dt^2} + \frac{dx^m}{dt}\Gamma^i_{mk}\frac{dx^k}{dt} &= \nabla_{\dot{\vec{\boldsymbol{x}}}}\dot{\vec{\boldsymbol{x}}} = 0
\end{aligned}$$

Here $\Gamma^i_{mk} = \frac{1}{2}g^{ij}\left(\frac{\partial g_{jk}}{\partial x^m} + \frac{\partial g_{jm}}{\partial x^k} - \frac{\partial g_{mk}}{\partial x^j}\right)$ and the scalars g^{ij} are the components of the inverse Riemannian metric $\mathbf{G}^{-1}$ (i.e., $g^{ij}g_{jk} = \delta_{ik}$, where $\delta_{ik} = 1$ if $i = k$ and $\delta_{ik} = 0$ if $i \neq k$).

Appendix B

Lyapunov Feedback Tracking Control

Let the $\delta\vec{\boldsymbol{X}} = [\vec{\delta\boldsymbol{r}}\ \ \vec{\delta\boldsymbol{v}}]^\intercal$ be the tracking error between the deputy's states and the reference state to be tracked. Let $\vec{\delta\boldsymbol{a}}$ denote the acceleration error. The candidate Lypunov function can be defined as follows:

$$\mathrm{V}(\vec{\delta\boldsymbol{r}}, \vec{\delta\boldsymbol{v}}) = \frac{1}{2}\vec{\delta\boldsymbol{v}}^\intercal\vec{\delta\boldsymbol{v}} + \frac{1}{2}\vec{\delta\boldsymbol{r}}^\intercal[\mathbf{k}]\vec{\delta\boldsymbol{r}} \tag{B.1}$$

where $\vec{\delta\boldsymbol{r}}(t) = \vec{\boldsymbol{r}}_i(t) - \vec{\boldsymbol{r}}^*(t, s_{max})$ is the difference between the instantaneous position of the deputy $\vec{\boldsymbol{r}}_i(t)$ and the desired position $\vec{\boldsymbol{r}}^*(t, s_{max})$. Likewise, $\vec{\delta\boldsymbol{v}}(t) = \vec{\boldsymbol{v}}_i(t) - \vec{\boldsymbol{v}}^*(t, s_{max})$ is deviation between the instantaneous velocity $\vec{\boldsymbol{v}}_i(t)$and the desired velocity $\vec{\boldsymbol{v}}^*(t, s_{max})$. The derivative of Eq. (B.1) is given by the following expression:

$$\dot{\mathrm{V}} = \vec{\delta\boldsymbol{v}}^\intercal\left(\vec{\delta\boldsymbol{a}} + [\mathbf{K}]\vec{\delta\boldsymbol{r}}\right) \tag{B.2}$$

where $\vec{\delta\boldsymbol{a}}$ is the difference between the instantaneous acceleration of the deputy and the acceleration along the optimal path. By setting the time derivative of the Lyapunov function to $\frac{d}{dt}V(\vec{\delta\boldsymbol{r}}, \vec{\delta\boldsymbol{v}}) = -\vec{\delta\boldsymbol{v}}^\intercal[\boldsymbol{P}_1]\vec{\delta\boldsymbol{v}}$, one obtains the following expression:

$$\dot{\mathrm{V}} = -\vec{\delta\boldsymbol{v}}^\intercal[\mathbf{P}]\vec{\delta\boldsymbol{v}} \tag{B.3}$$

$$\Longrightarrow \vec{\boldsymbol{u}} = -\vec{\delta\boldsymbol{a}} - [\mathbf{P}]\vec{\delta\boldsymbol{v}} - [\mathbf{K}]\vec{\delta\boldsymbol{r}} \tag{B.4}$$

The control matrices $[\boldsymbol{K}_1]$ and $[\boldsymbol{P}_1]$ are the sensitivity of the controller to deviation in position and velocity, respectively.

Appendix C

Meaning of Chance Constraints

let $\mathbf{a}, \boldsymbol{x} \in \mathbb{R}^n$, $c \in \mathbb{R}$, and $\epsilon \in [0.5, 1]$. If $\boldsymbol{x} \sim \mathscr{N}\left(\bar{\boldsymbol{x}}, \boldsymbol{\Sigma}\right)$, then $\gamma = \boldsymbol{a}^\intercal \boldsymbol{x} \sim \mathscr{N}\left(\bar{\gamma}, \Gamma\right)$ where $\bar{\gamma} = \boldsymbol{a}^\intercal \bar{\boldsymbol{x}}$, and $\Gamma = \boldsymbol{a}^\intercal \boldsymbol{\Sigma} \boldsymbol{a}$. The probabilistic expression

$$\Pr\left(\gamma \leq c\right) = \frac{1}{\sqrt{2\pi}\sqrt{\Gamma}} \int_{-\infty}^{c} \exp\left(-\frac{\left(\gamma - \bar{\gamma}\right)^\intercal \Gamma^{-1} \left(\gamma - \bar{\gamma}\right)}{2}\right) d\boldsymbol{x} \tag{C.1}$$

Let $\gamma = \Gamma^{\frac{1}{2}} z + \bar{\gamma}$. It is trivial to show that $z \sim \mathscr{N}\left(0, 1\right)$. Substituting every value γ in Eq.(C.1) of by the corresponding z yields:

$$\begin{aligned}
\Pr\left(\gamma \leq c\right) &= \frac{1}{\sqrt{\Gamma}} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\Gamma^{-\frac{1}{2}}\left(c - \bar{\gamma}\right)} \exp\left(-\frac{\left(\Gamma^{\frac{1}{2}} z + \bar{\gamma} - \bar{\gamma}\right)^\intercal \Gamma^{-1} \left(\Gamma^{\frac{1}{2}} z + \bar{\gamma} - \bar{\gamma}\right)}{2}\right) \Gamma^{\frac{1}{2}} dz \\
&= \frac{1}{\sqrt{\Gamma}} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\Gamma^{-\frac{1}{2}}\left(c - \bar{\gamma}\right)} \exp\left(-\frac{z^\intercal z}{2}\right) \Gamma^{\frac{1}{2}} dz \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\Gamma^{-\frac{1}{2}}\left(c - \bar{\gamma}\right)} \exp\left(-\frac{z^\intercal z}{2}\right) dz \\
&= \mathrm{F}_X\left(\Gamma^{-\frac{1}{2}}\left(c - \bar{\gamma}\right)\right)
\end{aligned} \tag{C.2}$$

Note that the CDF of the standard Gaussian distribution $\mathrm{F}_X(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} \exp\left(-\frac{z^\intercal z}{2}\right) dz$ is invertible. The probabilistic constraint

$$\begin{aligned}
\Pr\left(\mathbf{a}^\intercal \boldsymbol{x} \leq c\right) \geq \epsilon &\iff \mathrm{F}_X\left(\Gamma^{-\frac{1}{2}}\left(c - \bar{\gamma}\right)\right) \geq \epsilon \\
&\iff c - \bar{\gamma} \geq \mathrm{F}_X^{-1}\left(\epsilon\right) \sqrt{\Gamma} \\
&\iff \mathbf{a}^\intercal \bar{\boldsymbol{x}} + \mathrm{F}_{\mathrm{X}}^{-1}(\epsilon) \sqrt{\mathbf{a}^\intercal \boldsymbol{\Sigma} \mathbf{a}} \leq c
\end{aligned} \tag{C.3}$$

Appendix D

Proof of the Lie Rank Condition for 6DoF Fuel-Free Motion Planning

(1) **Submatrix Δ_{21}:**

$$\Delta_{21} = \begin{bmatrix} \boldsymbol{i}_1 & \boldsymbol{i}_2 & \boldsymbol{i}_3 \end{bmatrix} = \begin{bmatrix} \mathrm{i}_1 & 0 & 0 \\ 0 & \mathrm{i}_2 & 0 \\ 0 & 0 & \mathrm{i}_3 \end{bmatrix} \tag{D.1}$$

where i_1, i_2, and i_1 are the principal axis moments of inertia. The determinant Δ_{21} is

$$det(\Delta_{21}) = \mathrm{i}_1\mathrm{i}_2\mathrm{i}_3 \neq 0 \implies rank(\Delta_{21}) = 3$$

(2) **Submatrix Δ_{12}:**

$$\Delta_{12} = [\mathrm{I}]\left[\frac{\partial\dot{\vec{\boldsymbol{\sigma}}}}{\partial\vec{\boldsymbol{\omega}}}\right] = \frac{1}{4}[\mathbf{I}]\boldsymbol{B}(\vec{\boldsymbol{\sigma}}) = \frac{1}{4}[\mathbf{I}]\left((1-\vec{\boldsymbol{\sigma}}^{\intercal}\vec{\boldsymbol{\sigma}})[\mathbb{I}_{3\times3}] + 2[\tilde{\boldsymbol{\sigma}}] + 2\vec{\boldsymbol{\sigma}}\vec{\boldsymbol{\sigma}}^{\intercal}\right) \tag{D.2}$$

The determinant of Δ_{12} is

$$\det(\Delta_{12}) = \frac{\mathrm{i}_1\mathrm{i}_2\mathrm{i}_3\left(\sigma_1^2+\sigma_2^2+\sigma_3^2+1\right)^3}{64} \neq 0 \;\forall\vec{\boldsymbol{\sigma}} \implies rank(\Delta_{12}) = 3$$

(3) **Submatrix Δ_{43}:**

$$\Delta_{43} = \left[\frac{\partial\delta\vec{\boldsymbol{a}}^{s}}{\partial\vec{\boldsymbol{\sigma}}}\right]\Delta_{12} \tag{D.3}$$

$$\left[\frac{\partial\vec{\boldsymbol{a}}^{s}}{\partial\vec{\boldsymbol{\sigma}}}\right] = -\frac{P_s}{m_d}A_d\Big[\left((1-\alpha_s)\,\hat{\boldsymbol{r}} + 2\left(\alpha_s\hat{\boldsymbol{n}}_3^{\intercal}\hat{\boldsymbol{r}} + \frac{\alpha_d}{3}\right)\hat{\boldsymbol{n}}_3\right)\hat{\boldsymbol{r}}^{\intercal} + \\ \hat{\boldsymbol{n}}_3^{\intercal}\hat{\boldsymbol{r}}\left(2\left(\alpha_s\hat{\boldsymbol{n}}_3^{\intercal}\hat{\boldsymbol{r}} + \frac{\alpha_d}{3}\right)\left[\mathbb{I}_{3\times3}\right] + 2\alpha_s\hat{\boldsymbol{n}}_3\hat{\boldsymbol{r}}^{\intercal}\right)\Big]\left[\frac{\partial\hat{\boldsymbol{n}}_3}{\partial\vec{\boldsymbol{\sigma}}}\right] \tag{D.4}$$

The matrix $\left[\frac{\partial \hat{\boldsymbol{n}}_3}{\partial \vec{\boldsymbol{\sigma}}}\right]$ is of rank 2, and its right null space is $\hat{\boldsymbol{n}}_3$. The matrix $\left[\frac{\partial \vec{\boldsymbol{a}}^s}{\partial \vec{\boldsymbol{\sigma}}}\right]$ is full rank if the position of deputy relative to the sun has a component in the plate normal direction (i.e., $\hat{\boldsymbol{n}}_3^\intercal \hat{\boldsymbol{r}} \neq 0 \Leftrightarrow \vec{\boldsymbol{a}}^s_B \neq \mathbf{0}$). Finally, $\text{rank}\left(\left[\frac{\partial \vec{\boldsymbol{a}}^s}{\partial \vec{\boldsymbol{\sigma}}}\right]\right) = 3 \implies \text{rank}\left(\Delta_{43}\right) = 3$