Guidance and Control for Multi-stage Rendezvous and Docking Operations in the Presence of Uncertainty

Christopher Michael Jewison, David W. Miller

June 2017 SSL #03-17

Guidance and Control for Multi-stage Rendezvous and Docking Operations in the Presence of Uncertainty

Christopher Michael Jewison, David W. Miller

June 2017 SSL #03-17

This work is based on the unaltered text of the thesis by Christopher Michael Jewison submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of Doctor of Philosophy at the Massachusetts Institute of Technology.

Guidance and Control for Multi-stage Rendezvous and Docking Operations in the Presence of Uncertainty

by

Christopher Michael Jewison

Submitted to the Department of Aeronautics and Astronautics
on May 25, 2017, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics

Abstract

Rendezvous and docking missions have been a mainstay of space exploration from the Apollo program through present day operations with the International Space Station. There remains a growing interest in several mission types that not only rely on rendezvous and docking, but also rely on maneuvering spacecraft once docked. For example, there is active interest in orbital debris removal, on-orbit assembly, on-orbit refueling, and on-orbit servicing and repair missions. As these missions become more and more popular, the number of rendezvous and docking class operations will increase dramatically. Current methods focus on performing rendezvous and docking to very well-known targets and in very well-known conditions. Inherent to these new mission types, however, is an increasing element of uncertainty to which new guidance and control architectures will need to be robust.

As guidance and control techniques become more robust, a corresponding tradeoff in performance can typically be experienced. This thesis attempts to address the uncertainties in rendezvous and docking operations while maintaining a probabilistically optimal level of performance. There are two main focuses in the thesis: spacecraft trajectory optimization and reference-tracking controller selection. With respect to trajectory optimization, the goal is to find probabilistically optimal trajectories given large uncertainties in mission critical parameters, such as knowledge of an obstacle's position, while knowing that the trajectory is able to be replanned onboard the spacecraft when higher precision information is obtained. This baseline optimal trajectory and subsequently replanned trajectories are then followed by an optimally determined set of reference-tracking controllers. These controllers are selected and scheduled throughout the phases of the mission based on the probabilistically expected performance in the presence of noise and uncertain parameters.

This process is explored through its implementation on a generic problem setup for rendezvous, docking, and joint maneuvering. Results specific to this problem and associated analysis motivate the use of probabilistic planning in future space missions. Specifically, the thesis shows that fuel and tracking performance can be improved if multi-stage missions are planned continuously through phase transitions and without the use of waypoints. Furthermore, under the presence of large uncertainties, the techniques in this thesis produce better expected fuel and tracking performance than traditional trajectory planning and controller selection methods.

Thesis Supervisor: David W. Miller
Title: Jerome Hunsaker Professor of Aeronautics and Astronautics

Thesis Supervisor: Olivier L. de Weck
Title: Professor of Aeronautics and Astronautics and Engineering Systems

Thesis Supervisor: R. Scott Erwin
Title: Air Force Research Laboratory

Thesis Supervisor: Sertac Karaman
Title: Associate Professor of Aeronautics and Astronautics

Thesis Supervisor: Alvar Saenz-Otero
Title: Director of Space Systems Laboratory

Acknowledgements

The majority of this research was conducted with Government support under and awarded by the Department of Defense (DoD), Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. Additionally, the inspiration and basis of a large portion of this work was supported by two summer internships through the Air Force Research Laboratory Scholar Program sponsored by the Universities Space Research Association. The MIT Department of Aeronautics and Astronautics also provided supplemental support to the NDSEG Fellowship through the Donald W. Douglas Fellowship and the John and Irene M. Goldsmith Scholarship Fund and a semester of support for teaching assistance. The author gratefully thanks these sponsors for their generous support that enabled this research.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

AEKF	Augmented Extended Kalman Filter
AFRL	Air Force Research Laboratory
AIAA	American Institute of Aeronautics and Astronautics
ARE	Algebraic Riccati Equation
ASTRO	Admissible Subspace TRajectory Optimizer
ATV	Automated Transfer Vehicle
CAM	Collision Avoidance Maneuver
CDC	Conference on Decision and Control
CONOPS	Concept of Operations
CWH	Clohessy-Wiltshire-Hill
DARE	Discrete-time Algebraic Riccati Equation
DARPA	Defense Advanced Research Projects Agency
DART	Demonstration for Autonomous Rendezvous Technology
DEOS	Deutsche Orbitale Servicing Mission
DLR	Deutsches Zentrum für Luft und Raumfahrt
DOF	Degree of Freedom
ECI	Earth Centered Inertial
EKF	Extended Kalman Filter
ETS-VII	Engineering Test Satellite No. 7

FDI	Fault Detection and Isolation
FDIR	Fault Detection, Isolation and Recovery
FOV	Field Of View
GEO	Geostationary Earth Orbit
GLHS	Gaussian Latin Hypercube Sampling
GNC	Guidance, Navigation and Control
GPOPS-II	General Purpose OPtimal Control Software 2
GPS	Global Positioning System
HJB	Hamilton-Jacobi-Bellman
HTV	H-II Transfer Vehicle
IDVD	Inverse Dynamics in the Virtual Domain
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit
ISS	International Space Station
JSpOC	Joint Space Operations Center
KKT	Karush-Kuhn-Tucker conditions
LEO	Low Earth Orbit
LHS	Latin Hypercube Sampling
LOS	Line Of Sight
LQ	Linear-Quadratic
LQE	Linear-Quadratic Estimator
LQG	Linear-Quadratic-Gaussian
LQR	Linear-Quadratic Regulator
LTI	Linear Time Invariant
LVLH	Local Vertical Local Horizontal

MACE	Middeck Active Control Experiment
MCMC	Markov Chain Monte Carlo
MDP	Markov Decision Process
MERS	Model-Based Embedded and Robotic Systems
MEV	Mission Extension Vehicle
MILP	Mixed Integer Linear Program
MINP	Mixed Integer Nonlinear Program
MIT	Massachusetts Institute of Technology
MOGA	Multi-Objective Genetic Algorithm
MPC	Model Predictive Control
MRAC	Model Reference Adaptive Control
MTBF	Mean Time Between Failure
NASA	National Aeronautics and Space Association
NDSEG	National Defense Science and Engineering Graduate
OLEV	Orbital Life Extension Vehicle
PD	Proportional-Derivative
PDF	Probability Density Function
PID	Proportional-Integral-Derivative
PMP	Pontryagin's Maximum (or Minimum) Principle
POMDP	Partially Observable Markov Decision Process
PRISMA	PRecursore IperSpettrale della Missione Applicativa
RARC	Resource Aggregated Reconfigurable Control
RRT	Rapidly exploring Random Trees
RSGS	Robotic Servicing of Geosynchronous Satellites
SLAM	Simultaneous Localization and Mapping

SOCP	Second-Order Cone Programming
SPHERES	Synchronized Position Hold, Engage, Reorient, Experimental Satellites
SQP	Sequential Quadratic Programming
SRP	Solar Radiation Pressure
SSCO	Satellite Servicing Capabilities Office
SSL	Space Systems Laboratory
STS	Space Transportation System (i.e., Space Shuttle)
SUMO	Spacecraft for the Universal Modification of Orbits
SWLQE	Sensitivity Weighted Linear-Quadratic Estimator
SWLQG	Sensitivity Weighted Linear-Quadratic-Gaussian
SWLQR	Sensitivity Weighted Linear-Quadratic Regulator
Sys ID	System Identification
TLE	Two-Line Element
UKF	Unscented Kalman Filter
XSS-11	US Air Force Experimental Satellite System-11

Constants and Variables

$0_{n\times m}$	$n \times m$ matrix of zeros
$1_{n\times m}$	$n \times m$ matrix of ones
α	In-plane angle of the chaser with respect to the target
α_x	Scalar relative weighting of the importance of states in the $\hat{\imath}$-direction for Bryson's Rule
α_y	Scalar relative weighting of the importance of states in the $\hat{\jmath}$-direction for Bryson's Rule
α_z	Scalar relative weighting of the importance of states in the $\hat{k}$-direction for Bryson's Rule

β In-plane angle of the chaser with respect to the partner satellite

β_x Scalar relative weighting of the importance of control in the $\hat{\imath}$-direction for Bryson's Rule

β_y Scalar relative weighting of the importance of control in the $\hat{\jmath}$-direction for Bryson's Rule

β_z Scalar relative weighting of the importance of control in the $\hat{k}$-direction for Bryson's Rule

Δ_{max} Maximum allowable state deviance in the tube-constrained MPC problem

η Efficiency rating of electric with respect to chemical propulsion

γ_r Current angle the obstacle hyperplane constraint has rotated

$\gamma_{r_{tot}}$ Total angle the obstacle hyperplane constraint needs to rotate

λ_j Preference weighting of the importance of the j-th performance metric for the reference-tracking controller selection problem

$\mu,\ \mu_{\oplus}$ Standard gravitational parameter of the Earth

$\mu_{\odot}$ Standard gravitational parameter of the Sun

$\mu_{\bigcirc}$ Standard gravitational parameter of the Moon

Ω State propagation matrix used in trajectory optimization problem

$\omega_{_{CWH}}$ Rotation rate of the CWH frame with respect to the ECI frame

Ω_{dsc} Finer-discretized Ω matrix for constraint interpolation

Ω_i i-th block row of the Ω matrix

Ω_{vel} Shrunken Ω matrix to only include velocity states

ϕ Attitude of the target spacecraft about the $\hat{k}$-axis

π_k Preference weighting of the importance of the k-th uncertain parameter for univariate version of the reference-tracking controller selection problem

Ψ Control input propagation matrix used in trajectory optimization problem

Ψ_{dsc} Finer-discretized Ψ matrix for constraint interpolation

Ψ_i i-th block row of the Ψ matrix

Ψ_{vel}	Shrunken Ψ matrix to only include velocity states
$\boldsymbol{\rho}$	Vector of the relative distance from the target to the chaser
$\dot{\boldsymbol{\rho}}$	Vector of the relative velocity of the chaser with respect to the target
ρ	Magnitude of the relative distance from the target to the chaser
ρ_{atm}	Density of the atmosphere (a function of altitude)
ρ_d	Radial distance from the target at which docking phase begins and higher precision sensors are available
ρ_r	Radial distance from the target at which range measurements are available
Σ	Generic covariance matrix describing a Gaussian distribution
σ	Standard deviation of a univariate Gaussian distribution
$\theta,\ \theta_1,\ \theta_2$	LOS cone angle and two LOS pyramid angles
A	Continuous-time state (or system) matrix
$\mathbf{a}_{3rd}$	Acceleration vector acting on the spacecraft due to the effects of 3rd-body gravity from the Sun and Moon
A_c	Discrete-time state matrix for compensator
$\mathbf{a}_{ctrl}$	Acceleration vector executed by the spacecraft thrusters
$\mathcal{A}_d$	Cross-sectional area subjected to drag forces
A_d	Discrete-time state (or system) matrix
A_d^a	Augmented discrete-time state matrix for AEKF
A_{d_i}	Discrete-time state (or system) matrix, where i is the number of DOF
$\mathbf{a}_{drag}$	Acceleration vector acting on the spacecraft due to atmospheric drag forces
A_{dsc}	Finer-discretized A matrix for constraint interpolation
$\mathbf{a}_{GM}$	Acceleration vector due to Earth gravity experienced by the spacecraft
A_i	Continuous-time state (or system) matrix, where i is the number of DOF
$\mathbf{a}_{J2-6}$	Acceleration vector acting on the spacecraft due to the non-spherical perturbations of the Earth's gravity field
$\mathbf{a}_r$	Axis of rotation of the hyperplane for the linear, rotating obstacle constraint

$\mathbf{a}_{SRP}$	Acceleration vector on the spacecraft due to the solar radiation pressure
$\mathcal{A}_{\odot}$	Cross-sectional area facing the Sun
$\tilde{A}$	Linearized A matrix for EKF
$\mathbf{a}_{total}$	Total acceleration vector experienced by the spacecraft
b	Binary variable used for if-then constraints in integer programming
$B,\ B_u$	Continuous-time input matrix with respect to the control
B_w	Continuous-time input matrix with respect to the noise
B_c	Discrete-time input matrix for compensator
B_d	Discrete-time input matrix with respect to the control
B_{d_i}	Discrete-time input matrix with respect to the control, where i is the number of DOF
B_{dsc}	Finer-discretized B matrix for constraint interpolation
B_i	Continuous-time input matrix with respect to the control, where i is the number of DOF
$\boldsymbol{c}$	Vector specifying the direction the LOS zone constraint points
c	Represents a specific controller for the reference-tracking controller selection optimization problem
C_c	Discrete-time gain matrix for compensator
c_d	Coefficient of drag of the spacecraft
c_r	Coefficient of reflection of the of the spacecraft
$\tilde{C}$	Linearized measurement model matrix for EKF
C_y	Continuous-time measurement output matrix
C_z	Continuous-time performance output matrix
D_{yu}	Continuous-time feedthrough measurement matrix with respect to the input
D_{yw}	Continuous-time feedthrough measurement matrix with respect to the noise
D_{zu}	Continuous-time feedthrough performance matrix with respect to the input

D_{zw} Continuous-time feedthrough performance matrix with respect to the noise

$\boldsymbol{e}$ Tangent point where the hyperplane meets the ellipsoid for the linear rotating obstacle constraint

$\mathcal{E}$ Matrix of hyperplane tangent points for the linear, rotating obstacle constraint

e Out-of-plane angle, or elevation angle of the chaser with respect to the target

$\mathbf{f}(\cdot)$ Nonlinear state dynamics function for EKF

$f(\cdot)$ Probability density function

$\mathbf{f}^a(\cdot)$ Augmented nonlinear relative dynamics function for AEKF

$\mathbf{f}_{eq}(\cdot)$ Nonlinear equality constraints function

$\mathbf{f}_{ineq}(\cdot)$ Nonlinear inequality constraints function

$\mathbf{f}_{rm}(\cdot)$ Nonlinear relative dynamics propagation function for simulation

f_{thr} Force magnitude exerted by the thrusters (as a fraction of the nominally expected thrust)

f_{tot} Total propellant consumed

F_x Force executed by the thrusters in the $\hat{\imath}$ direction

F_y Force executed by the thrusters in the $\hat{\jmath}$ direction

F_z Force executed by the thrusters in the $\hat{k}$ direction

$\mathbf{g}$ Vector of ones used as summation in minimum fuel trajectory optimization

$\mathcal{H}$ Matrix of hyperplane normals for the linear, rotating obstacle constraint

h_n Nonlinear measurement model for the n-th mission phase

$\mathbf{h}_{target}$ Specific angular momentum vector of the target spacecraft's orbit

$\hat{\imath}$ Unit vector pointing in the direction radially out of the Earth in the CWH frame

$I_{n\times n}$ $n \times n$ identity matrix

$\hat{\jmath}$ Unit vector pointing in the in-track, orbital velocity direction of the target spacecraft

$J(\cdot)$ Standard LQG objective function for baseline controller

J_2, ..., J_6 Constants representing the weighting for the 2nd through 6th zonal harmonics of the Earth

$J_{init}(\cdot)$ Generic objective function for a full-mission trajectory optimization as a function of constraint parameters

$J_{nrg}(\cdot)$ Minimum energy (LQR) objective function for trajectory optimization

$J_{replan}(\cdot)$ Generic objective function for a replanned trajectory optimization paired with and in addition to the baseline $J_{init}(\cdot)$

$J_s(\cdot)$ Sensitivity Weighted LQG objective function for desensitized controller

$\hat{k}$ Unit vector pointing in the out-of-plane, cross-track direction

K Gain matrix for the LQR controller

L Gain matrix for the LQE estimator

m Mass

$M_1,\ M_2,\ M_3$ Arbitrarily large constants used for if-then constraints in integer programming

m_c Mass of the chaser spacecraft

m_j j-th performance metric in the reference-tracking controller selection problem

m_{kj} j-th performance metric as evaluated against the k-th univariate uncertain parameter

m_t Mass of the target spacecraft

$\boldsymbol{n}$ Normal vector of the hyperplane for the linear rotating obstacle constraint

n Mean motion of the target satellite's orbit

n_m Number of performance metrics in the reference-tracking controller selection problem

n_p Number of uncertain parameters in the reference-tracking controller selection problem

n_s Number of sensitivity variables in SWLQG

n_σ	Multiplier on the standard deviation of a Gaussian to represent the margin given to a path constraint
$\mathcal{P}$	Matrix representing the pyramid version of the LOS constraint
$\mathbf{P}$	Shape matrix of an ellipsoid
P	Solves the steady-state DARE for the LQR problem
p_k	Value of an instantiation of the k-th uncertain parameter in the reference-tracking controller selection problem
p_{sr}	Solar pressure exerted on the spacecraft
Q	Solves the steady-state DARE for the LQE problem
Q^a	Augmented covariance matrix for AEKF
$\mathbf{r}$	Reference vector the controller is attempting to follow
$\mathbf{r}_{partner}$	Radial distance vector from the partner satellite to chaser used for range measurement in Phase 3
r_0	Initial radius of the target's orbit or constant semi-major axis if orbit is circular
$R_{\alpha\alpha}$	Measurement noise variance with respect to the in-plane angle in Phases 1 and 2
$R_{\beta\beta}$	Measurement noise variance with respect to the in-plane angle in Phase 3
$r_\oplus$	Radius of the Earth
$R_{\rho\rho,a}$	Measurement noise variance with respect to the range measurement in Phase 1b
$\mathbf{r}_g$	Vector of relative distance from the obstacle to the goal location
R_n	Measurement noise covariance matrix for the n-th phase
$\mathbf{r}_s$	Vector of relative distance from the obstacle center to chaser spacecraft
$R_{\rho\rho,b}$	Measurement noise variance with respect to the range measurement in Phase 2
R_{rr}	Measurement noise variance with respect to the range measurement in Phase 3

R_{ss}	Weighting matrix of the sensitivity in the SWLQR optimization problem
R_{ss_i}	Weighting matrix of the i-th sensitivity parameter in the SWLQR optimization problem
$\mathbf{r}_{target}$	Position vector of the target spacecraft in ECI frame
R_{uu}	Relative weighting matrix to penalize the control terms in LQR
$R_{uu}^{'}$	Sensitivity modified relative weighting matrix to penalize the control terms in LQR
r_x	Distance from the partner satellite to the chaser in the $\hat{i}$ direction
R_{xu}	Relative weighting matrix to penalize the cross terms of state and control in LQR
$R_{xu}^{'}$	Sensitivity modified relative weighting matrix to penalize the cross terms of state and control in LQR
R_{xx}	Relative weighting matrix to penalize the state error terms in LQR
$R_{xx}^{'}$	Sensitivity modified relative weighting matrix to penalize the state terms in LQR
r_y	Distance from the partner satellite to the chaser in the $\hat{j}$ direction
r_z	Distance from the partner satellite to the chaser in the $\hat{k}$ direction
R_{zz}	Overall weighting matrix to penalize state error or control over each other
s	Sensitivity parameter
s_i	i-th sensitivity parameter
T	Discretization time step used for discrete control
t_0	Time at the start of the mission
T_{dsc}	Finer-discretized time step for constraint interpolation
t_e	Time that the eclipse occurs
t_{epoch}	Time elapsed since the epoch used to define the ECI frame
t_f	Requirement on the upper limit of the completion time of the mission
t_{nf}	Time at the end of the n-th phase

$\mathbf{u}$	Control input vector
$\bar{u}$	Maximum thruster force
$\mathbf{U}$	Stacked vector of control inputs for trajectory optimization problem
$\mathbf{U}^{-}$	Stacked vector of negative parts of control inputs for the minimum-fuel trajectory optimization problem
$\mathbf{U}^{+}$	Stacked vector of positive parts of control inputs for the minimum-fuel trajectory optimization problem
$\mathbf{U}^{'}$	Stacked vector of control inputs for the minimum-fuel trajectory optimization problem
u_x	Control input in the $\hat{\imath}$-direction
$(u_x)_{max}$	Maximum expected control input in the $\hat{\imath}$-direction for Bryson's Rule
u_y	Control input in the $\hat{\jmath}$-direction
$(u_y)_{max}$	Maximum expected control input in the $\hat{\jmath}$-direction for Bryson's Rule
u_z	Control input in the $\hat{k}$-direction
$(u_z)_{max}$	Maximum expected control input in the $\hat{k}$-direction for Bryson's Rule
$\mathbf{v}$	Gaussian, white measurement noise vector
$\bar{V}$	Maximum docking velocity
$\mathbf{V}_{eq}$	Stacked constraint matrix for equality constraints for trajectory optimization
$\mathbf{V}^{'}_{eq}$	Stacked constraint matrix for equality constraints modified for the minimum-fuel trajectory optimization problem
$\mathbf{V}_{ineq}$	Stacked constraint matrix for inequality constraints for trajectory optimization
$\mathbf{V}^{'}_{ineq}$	Stacked constraint matrix for inequality constraints modified for the minimum-fuel trajectory optimization problem
v_{sat}	Magnitude of the velocity of the spacecraft in the ECI frame
V_{ss}	Weighting matrix of the sensitivity in the SWLQE optimization problem

V_{ss_i} Weighting matrix of the i-th sensitivity parameter in the SWLQE optimization problem

$\mathbf{v}_{target}$ Velocity vector of the target spacecraft in ECI frame

V_{yy} Relative weighting matrix to penalize the measurement noise terms in LQE or EKF

V_{ww} Overall weighting matrix to penalize process or measurement noise over each other

V_{xy} Relative weighting matrix to penalize the cross terms of the process and measurement noise in LQE or EKF

V_{xx} Relative weighting matrix to penalize the process noise terms in LQE or EKF

V_{xx}^{a} Augmented process noise matrix for AEKF

$V_{xx}^{'}$ Sensitivity modified relative weighting matrix to penalize the process noise terms in LQE or EKF

$V_{xy}^{'}$ Sensitivity modified relative weighting matrix to penalize cross terms of the process and measurement noise in LQE or EKF

$V_{yy}^{'}$ Sensitivity modified relative weighting matrix to penalize the measurement noise terms in LQE or EKF

$\mathbf{w}$ Gaussian, white process noise

$\mathbf{W}_{eq}$ Stacked constraint matrix for equality constraints for trajectory optimization

$\mathbf{W}_{eq}^{'}$ Stacked constraint matrix for equality constraints modified for the minimum-fuel trajectory optimization problem

$w_{f_{thr}}$ Process noise for the random walk for the thruster force level estimation in AEKF

$\mathbf{W}_{ineq}$ Stacked constraint matrix for inequality constraints for trajectory optimization

$\mathbf{W}_{ineq}^{'}$ Stacked constraint matrix for inequality constraints modified for the minimum-fuel trajectory optimization problem

w_{r_0} Process noise for the random walk for the orbital radius estimation in AEKF

$\boldsymbol{\mathcal{X}}$	Matrix of relative positions between the spacecraft and center of obstacle
$\mathbf{X}$	Stacked vector of states for trajectory optimization problem
$\mathbf{x},\ \dot{\mathbf{x}}$	State vector consisting of position and velocity states and its time derivative
$x,\ \dot{x},\ \ddot{x}$	Position, velocity and acceleration scalars in the $\hat{\imath}$ direction
$\mathbf{x}^a$	Augmented state vector for AEKF
$\mathbf{x}_c$	Compensator state vector
$\mathbf{x}_{chaser}$	State vector of the chaser (position and velocity)
$\mathbf{x}_e$	Position vector of the center of the ellipsoid obstacle constraint
$\mathbf{x}_{docked}$	State vector of the docking location in the CWH frame
$\mathbf{X}_{dsc}$	Stacked vector of finer-discretized states for constraint interpolation
$\mathbf{x}_f$	Final desired state used in trajectory planning
$\hat{\mathbf{x}}$	Estimated state vector for EKF
$\hat{\mathbf{x}}^a$	Estimate of the augmented state vector for AEKF
$\mathbf{x}_i$	Desired waypoint used in trajectory planning
X_{LOS}	Set of states that satisfy the LOS zone constraint and are in the docking phase
$x_{max},\ \dot{x}_{max}$	Maximum expected states in the $\hat{\imath}$-direction for Bryson's Rule
$\mathbf{X}_{obst}$	Stacked vector of obstacle states for constraints in trajectory optimization problem
$\mathbf{X}_{obst_{dsc}}$	Stacked vector of finer-discretized obstacle states for constraint interpolation
$\mathbf{x}_{obst}$	State vector of an obstacle in the CWH frame
$\mathbf{x}_{rel}$	Full relative state vector of the chaser and target spacecraft
$\mathbf{x}_{relocation}$	State vector of the relocation spot in the CWH frame
$\mathbf{x}_{target}$	State vector of the target (position and velocity)
$\mathbf{y}$	Output or measurement vector
$y,\ \dot{y},\ \ddot{y}$	Position, velocity and acceleration scalars in the $\hat{\jmath}$ direction

$y_{max},\ \dot{y}_{max}$	Maximum expected states in the $\hat{\jmath}$-direction for Bryson's Rule
$\mathbf{y}_p$	Position state output for LQ trajectory optimization problem
$\mathbf{z}$	Performance output vector
$z,\ \dot{z},\ \ddot{z}$	Position, velocity and acceleration scalars in the $\hat{k}$ direction
$z_{max},\ \dot{z}_{max}$	Maximum expected states in the $\hat{k}$-direction for Bryson's Rule

Symbols and Operators

$\oplus$	Used in subscript to denote quantity regarding the Earth
$\bigcirc$	Used in subscript to denote quantity regarding the Moon
$\odot$	Used in subscript to denote quantity regarding the Sun
CWH$(\cdot)$	Shorthand for expressing the Clohessy-Wiltshire-Hill dynamics as a function of critical parameters
ΔV	Change in velocity
$diag\{\cdot\}$	Returns the elements from the diagonal of the inputted matrix in a column vector; or returns a matrix with the inputted vector as the diagonal
$\cdot$	Vector dot product
E$[\ \cdot\ ]$	Expectation operator
$[\ \cdot\ ]^{CWH}$	Vector is in the CWH frame
$[\ \cdot\ ]^{ECI}$	Vector is in the ECI frame
$\otimes$	Kronecker product
$[\ \cdot\ ](k\|k-1)$	Vector as known at time k only with information from time k-1
$\|\ \cdot\ \|$	Absolute value
$\|\|\ \cdot\ \|\|_1$	1-norm, sum of the absolute values
$\|\|\ \cdot\ \|\|,\ \|\|\ \cdot\ \|\|_2$	2-norm, vector magnitude
$\|\|\ \cdot\ \|\|_\infty$	Infinity norm, absolute value of maximum
P$[\ \cdot\ ]$	Probability operator

$\mathbb{R}^n$	Set of real numbers in n-dimensions
${}^{[A]}R^{[B]}$	Rotation matrix from frame A to frame B
$r_{[A][B]}$	Vector from point A to point B

Chapter 1

Introduction and Motivation

1.1 Historical Missions with Autonomous Rendezvous and Docking

From the start of the space race in the 1960's, rendezvous and docking has become an integral part of spaceflight. Contrasting the American and Russian approach to rendezvous and docking in the early years of spaceflight comes down to one word: autonomy. The Russians designed the Soyuz vehicle to dock through automatic control loops being fed data from an array of sensors, while the Americans let the astronaut command attitude and position corrections manually from Apollo through the Space Shuttle program. As of this writing, Soyuz and Shuttle are the only programs to build and fly spacecraft to transport crewmembers to the International Space Station (ISS) [1]. As the years have progressed, there has been more interest in unmanned operations and thus naturally more interest in automated rendezvous and docking.

There are also several unmanned vehicles used for cargo delivery to the ISS. The Orbital ATK Cygnus, SpaceX Dragon and Japanese HTV each rendezvous to the ISS, but are berthed through the Canadarm2 robotic arm controlled by a crewmember on the ISS.[1] The Soyuz/Progress vehicles are currently capable of autonomously performing rendezvous, docking, departure and re-entry maneuvers.[2] The European ATV has shown the capability to rendezvous and dock autonomously to the ISS through each of five missions [2].

Several demonstration missions have also been performed with varying success. In 1999,

[1] `http://www.nasa.gov/mission_pages/station/structure/launch/index.html`

[2] `http://www.nasa.gov/mission_pages/station/structure/elements/progress.html`

the Japanese ETS-VII mission demonstrated the first successful unmanned autonomous rendezvous, docking, and robotic arm maneuvers [3]. In 2005, the two spacecraft involved in NASA's DART mission unfortunately collided and failed to demonstrate autonomous docking [4]. Also in 2005, the Air Force Research Laboratory's XSS-11 micro-satellite was intended to perform autonomous rendezvous and inspection operations, although no technical reports with results have been issued since its launch.[3] In 2007, DARPA's Orbital Express mission successfully demonstrated rendezvous, robotic arm docking, servicing and refueling of a specially designed modular serviceable satellite through increasing levels of autonomy [5, 6]. In 2010, the Swedish Space Corporation's PRISMA mission demonstrated autonomous rendezvous, formation flight and collision avoidance maneuvers [7].

1.2 Proposed Missions and Applications

With the recent successes in unmanned rendezvous and docking technology, a variety of new mission types are becoming feasible. Some of these mission types will create larger demand for future missions (e.g., space station assembly) and others will serve to mitigate the effects of the growing number of missions (e.g., orbital debris removal). Regardless, with the continual trend toward the commercialization of space, there will become an overwhelming number of missions that rely on rendezvous and docking.

With the increase of orbital debris in Low Earth Orbit (LEO), a large number of papers have proposed methods for its removal through unmanned spacecraft rendezvous, capture and de-orbit [8–10]. Even a new company, Astroscale, has formed and is working toward a plan of removing space debris through adhesively docking disposable satellites to spent rocket bodies or defunct satellites.[4]

One of the major end-of-life causes for satellites on orbit is simply running out of fuel. A few missions have been proposed to refuel these depleted satellites on orbit. Orbital Express provided the first feasibility experiment on the topic, but it is difficult to refuel spacecraft that were not originally designed for refueling. Thus, another approach has been proposed for small spacecraft to dock to fuel-depleted satellites to act as ancillary propulsive modules. These missions are proposed both for satellite retirement into the graveyard orbit past Geostationary Earth Orbit (GEO) or for station-keeping and more benign modification

[3] http://www.kirtland.af.mil/shared/media/document/AFD-070404-108.pdf

[4] http://astroscale.com

of orbits such as DARPA's SUMO program [11] or DLR's OLEV program [12]. Some more grandiose plans involve fuel depots on orbit that satellites could maneuver to themselves or have fuel delivered to them.

The on-orbit servicing and repair of failed spacecraft that still have the potential for continued operations is also of interest in the community. This repair could be as simple as nudging a solar array that did not deploy correctly or as complicated as replacing a serviceable module such as a faulty flight computer or reaction wheel assembly. Several missions have been proposed for various levels of repair and serviceability such as DEOS from the DLR [12], RSGS from DARPA,[5] and Restore-L from NASA's SSCO.[6] On the commercial side there is ViviSat's MEV,[7] an Orbital ATK and US Space joint venture which recently seems to have signed a deal with Intelsat.[8]

On-orbit assembly is another area requiring rendezvous, docking, joint maneuvers and undocking [13]. Given launch vehicle constraints on payload size and mass, large space structures will require in-space assembly. These structures can range anywhere from space stations to large aperture telescopes [14]. Creating large space stations could pave the way for space tourism, which would require regularly scheduled transport involving rendezvous and docking.

The number of applications for rendezvous and docking is only increasing. Thus, the demand for more fuel and time efficient techniques will be at a premium, as will the need for safer and more robust algorithms to oversee these operations. Current methods are generally designed toward well-known, specific systems and require weeks of engineers' time for integration and testing. The current process of choosing appropriate guidance and control techniques for missions will need to be revamped to account for the inherently increased levels of uncertainty in these future missions.

1.3 Generalized Mission Phases

Although the applications of rendezvous and docking vary greatly in mission objective, each application can be split into a generalized set of up to four guidance and control phases

[5] http://www.darpa.mil/news-events/2016-03-25

[6] http://ssco.gsfc.nasa.gov/restore-L.html

[7] http://www.vivisat.com

[8] http://www.wsj.com/articles/orbital-atk-and-intelsat-set-to-sign-satellite-servicing-pact-1457444904

shown in Figure 1-1: (1) Rendezvous, (2) Docking, (3) Joint Maneuvers, and (4) Undocking.

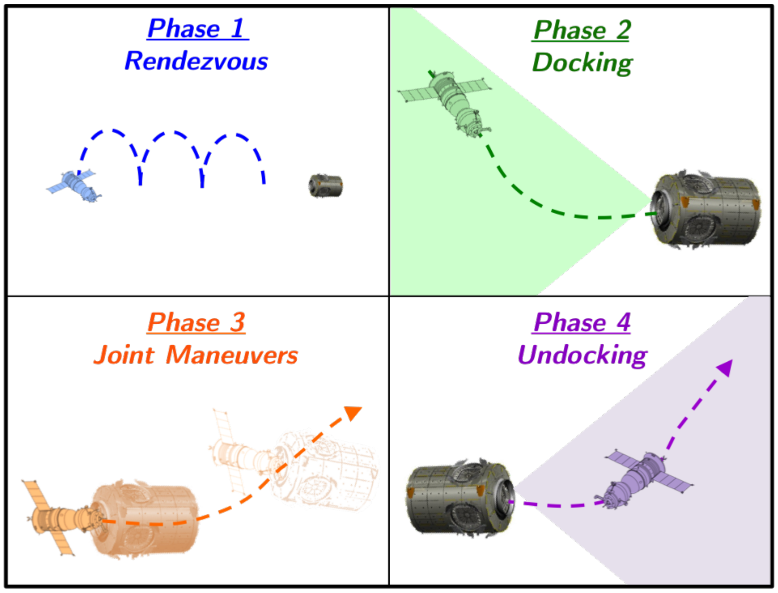


Figure 1-1: The four generalized mission phases

Each of these phases is associated with distinct objectives, constraints and uncertainties. Specific mission parameters will determine how long each of these phases will last. Some missions may not perform active control during the joint maneuver phase, others may not even have an undocking phase, and further still, some may loop over multiple phases to rendezvous and dock to numerous targets in a single mission. Nonetheless, this general framework should encapsulate the full range of rendezvous and docking mission phases from a guidance and control perspective. The following descriptions will highlight the differences between the phases and given more detail into the objectives and constraints present in each phase. For nomenclature's sake, note that the "chaser" spacecraft is the one approaching and docking to the "target" spacecraft.

> **Rendezvous Phase:** For the purposes of this thesis, the orbital phasing maneuvers that involve large ΔV burns to change the orbits of the two spacecraft are left out of the problem formulation, and instead the rendezvous phase describes the period after the orbits are lined up and the spacecraft are separated by about 10 km to 100 m. The sensors available at these distances play the main role in defining the rendezvous phase, and thus, when compared to other phases, the foremost uncertainty in this phase is in the knowledge of the relative state information. Due to the larger distance traversed in this phase, the potential

for debris or other spacecraft to intersect a rendezvous trajectory is increased, and thus there is a requirement to adhere to obstacle path constraints. Due to the larger maneuvers in this phase, the control methodology used during the rendezvous phase is generally very concerned with reducing fuel consumption as much as possible. Additionally, the maneuvers tend to be executed over a much larger period of time to facilitate this fuel efficiency. In this respect, the rendezvous phase is suitable for guidance and control techniques that require longer computation time but guarantee better performance margins.

Docking Phase: The docking stage is concerned with closing the relative distance from 100 m until a rigid mating of the two satellites is accomplished. The sensors available in this phase reduce the uncertainty in the relative state significantly. However, due to the risk sensitive nature of operations this close to the target satellite, several additional constraints are placed on the chaser satellite. There are Line Of Sight (LOS) constraints to ensure the docking sensors or fiducial markers are in view. There are also constraints that limit the approach velocity, docking port tolerance, contact force, and plume impingement. Finally, there are safety constraints for Collision Avoidance Maneuvers (CAM), among others. Due to the constraints on the precision of the relative state, it is necessary to control and sense at a faster frequency than during the rendezvous phase, which was mainly concerned with fuel efficiency.

Joint Maneuver Phase: Once the spacecraft have rigidly mated at the end of the docking phase, the joint maneuver phase begins. As is fairly self-descriptive, the joint maneuver phase handles the guidance and control of the two combined spacecraft as one. Depending on the mission, the objective during this phase could be to station-keep or point a passive target, to transport the target to a new orbital location for assembly, or to deorbit a piece of debris, among others. This phase could involve either the aggregation and use of actuators and sensors from both spacecraft or the complete control by the chaser acting as a tug. Uncertainties in mass, inertia, actuator and sensor properties of the target have increased, and the dynamics of the combined system will now be different. The objective could force the need for high performance in either precision control or fuel efficiency depending on the specific system goals.

Undocking Phase: As it naturally falls in the progression of the phases, the undocking phase deals with releasing the docking port mechanism and safely separating the two spacecraft. The objectives in this phase can vary greatly depending on the mission application. For example, currently several resupply vehicles to the ISS simply deorbit themselves and burn up in the atmosphere, while in the future these types of vehicles may need to stay in orbit to perform another cargo delivery. In any case, a majority of the same constraints need to be met as described in the docking phase to ensure the safety of the target spacecraft. Once the separation is deemed far enough, the next orbital maneuver or rendezvous phase is able to begin, which is generally out of the scope of this thesis but can be used to link successive rendezvous and docking activities.

The main takeaway here is that there are several phases in the rendezvous and docking problem, each with their own objectives, constraints and uncertainties. Different guidance and control techniques are therefore better suited at different times in the mission as performance requirements change. The challenge is then to determine the best performing guidance and control techniques for each phase while taking into account the current level of uncertainty in the mission.

1.4 Uncertainty and Uncertain Events On Orbit

Over the course of the better part of a century, there is one thing that has remained a constant in spaceflight: uncertainty. Just the fact that spacecraft are extremely complicated machines designed and built by teams of specialized engineers makes it difficult to know exactly what is going on, let alone that fact that by definition these experts are not on the same planet during operation and have very limited means of diagnosis or verification. To make a terminology point here, *uncertainty* deals with limited knowledge in parameters of the spacecraft mission. *Uncertain events*, on the other hand, are incidents that could potentially happen once in space, but there is limited knowledge of when, how and if they will happen.

The sources of uncertainty in a spacecraft mission all stem from limited measurement precision and accuracy at some point in the process of manufacturing, assembling, launching, and on-orbit activities. Specifically, from assembly, there are uncertainties in the knowledge of both the chaser and target's physical properties such as mass, center of mass, inertia, actuator alignment and sensor alignment. Of course, there are reasonable estimates for these parameters, but there is always some measurement error and components often tend to shift during launch. A lot of these parameters can be calibrated once on orbit, but there still remains uncertainty resulting from those analyses. There are also uncertainties based upon unmodeled dynamics, noise in sensor readings, and noise in actuator performance. Essentially this amounts to biases, measurement noise and process noise that a state estimator will have to consider. Still though there is inherent uncertainty in the knowledge of a chaser's relative state with respect to a target (i.e., relative position, velocity, attitude and angular rate). Finally, for the orbital debris removal problem, there could be even larger uncertainties in the target's spin rate, spin axis, and inertia ratios, which all need to be estimated by the chaser spacecraft. Planning a mission relies heavily upon all of these mentioned uncertainties. In order to reduce risk, robust planning methods are needed that take all of this uncertainty into account.

Uncertain events cover a broad spectrum of anomalies, failures, mishaps and unexpected behavior that can happen during a mission. For reference, in the 20-year period from 1981 to 2000, there were 242 observed on-orbit failures of components of 2,431 payloads, 134 of those causing total failure [15]. So roughly ten percent of space payloads encounter a failure at some point during their lifetime, and rendezvous and docking missions are not immune. From 1990 to 2006, there were 64 incidents related to power-system failures on spacecraft, which for a docking mission would be disastrous if all control of the system were lost [16].

Some of these uncertain events have historical precedence of happening during rendezvous missions, and some are known to be possible but have not been encountered up to this point. In 1990, during the STS-32 mission the Shuttle experienced a radar anomaly during its rendezvous to the Long Duration Exposure Facility causing deviation from the nominal trajectory and extra fuel and time expenditures [17]. In 1997, the manually piloted Progress cargo vehicle collided with the Mir space station due to insufficient onboard sensors and poor visibility, damaging the solar arrays and puncturing a pressurized module.[9] In 1998, during the STS-91 mission, the Shuttle's final flight to the Mir station, a software error in processing GPS measurements caused inaccurate state propagation which again resulted in excess fuel and time [17]. In 1999, on the second rendezvous attempt of

[9] `http://history.nasa.gov/SP-4225/multimedia/progress-collision.htm`

ETS-VII, there were anomalous thruster firings on multiple final approach maneuvers and it took three additional weeks to complete the docking maneuver [3]. In 2005, NASA's DART spacecraft experienced a velocity bias from GPS measurements, causing it to unnecessarily consume all of its fuel and collide with the target satellite [4]. In 2007, during the otherwise very successful Orbital Express mission, a sensor computer failed causing the navigation to drop out and required two ground intervention steps to ensure the spacecraft's safety [18].

These events continue to happen today, even in the seemingly routine docking to the ISS. In September 2013, the Orbital Sciences Cygnus cargo spacecraft experienced a data mismatch in their GPS processing software that delayed reaching the ISS by a couple days until a software patch could be implemented.[10] In December 2015, the autonomous docking in the Soyuz mission had to be aborted and manually taken over after a thruster failure and subsequent misalignment.[11] In February 2017, a SpaceX Dragon approach to the ISS was aborted due to anomalous GPS data causing the navigation filter to diverge, delaying berthing by a day.[12]

So, from historical anomalies, we can categorize the problems into a few areas. First, failures in spacecraft subsystems and components are rampant on orbit due to the complexity and harsh environment. Second, estimator errors from measurement outliers, software mistakes and hardware failures have caused satellites to collide with each other or perform unnecessary maneuvers. Third, thrusters have failed through either fuel depletion or anomalous firings. All of these historical events are summarized in Table 1.1. However, an additional uncertain event, obstacle avoidance has not caused problems on orbit, because it is currently handled by extensive planning and meticulous tracking of space objects. As the number of objects in Earth orbit increases and rendezvous missions begin to go into orbits that are not as well tracked such as geostationary orbit, unexpected obstacle avoidance will need to be considered. Furthermore, as more universal, reusable, autonomous servicer or assembly spacecraft are used, the prospect of a mission changing or being redirected could also be considered an uncertain event that would potentially affect mission planning. For all of these uncertainties and uncertain events, robustness needs to be built into future systems

[10] https://www.nasa.gov/content/cygnus-rendezvous-postponed-to-no-earlier-than-saturday and https://www.fool.com/investing/general/2013/09/24/orbital-sciences-cygnus-craft-now-set-to-dock-with.aspx

[11] http://www.russianspaceweb.com/iss-soyuz-tma19m.html

[12] https://spaceflightnow.com/2017/02/22/spacex-waves-off-space-station-cargo-delivery-for-at-least-a-day/

or the consequences will be great.

Table 1.1: Historical examples of anomalies and mishaps on orbit

Date	Mission	Event	Effect
1981-2000	2,431 total missions	242 component failures	134 total mission failures
1990-2006	64 affected missions	Power-system failures	Reduced capability or mission failure
1990	STS-32	Space Shuttle radar anomaly	Deviation from trajectory, wasted fuel/time
1997	Progress to Mir	Spacecraft misguided under poor sensing	Collision with Mir Space Station
1998	STS-91	Software error in GPS estimates	Deviation from trajectory, wasted fuel/time
1999	ETS-VII	Anomalous thruster firings	3-week delay, wasted fuel
2005	DART	Velocity bias from GPS measurements	Propellant depletion, collision with target
2007	Orbital Express	Navigation system drop out	Ground intervention, wasted fuel/time
2013	Cygnus to ISS	GPS software error	2-day delay, wasted fuel/time
2015	Soyuz to ISS	Thruster failure and misalignment	Manual override, wasted fuel/time
2017	Dragon to ISS	Diverged estimation filter	1-day delay, wasted fuel/time

1.5 Summary of Thesis Motivation

It is clear that there is a substantial heritage for autonomous rendezvous and docking missions in Earth orbit. However, given the increasing demand for rendezvous, docking and joint maneuvers in the coming years, the potential for these missions to be in locations unsuitable for human control, and the drastic increase of uncertainties in these missions, a new method of mission planning is required. This new method must take into consideration the chances that trajectories will need to change due to the uncertainties present in the problem. Consequentially, at least some portion of the planning must be implemented onboard the spacecraft. With the increased number of missions, probability is no longer on our side. More failures and anomalies will begin to occur, and proactive mission plans need to be able to handle those uncertain events without significantly degrading performance. Additionally, there will inevitably need to be rendezvous and docking to a wide array of systems and in a wide variety of orbits and locations. Today, docking to the same, highly monitored, "designed-to-be-docked-to" spacecraft over and over on meticulously planned days and times makes for little uncertainty. Furthermore, the multi-stage nature of these future missions, with different objectives, uncertainties and constraints in each phase, suggests that appropriate control techniques may vary throughout the mission. As such, a scheduling problem arises of when and under what circumstances to use specific control algorithms.

Therefore, we are motivated to develop mission planning techniques for guidance and control algorithms under the presence of uncertainties and uncertain events. We want to make the best guidance and control architecture decisions based only upon the knowledge available before the mission starts. This knowledge for the most part includes the best available estimates of important mission parameters and the uncertainty distributions which describe how well we know those parameters. In the case that the baseline guidance and control architecture is not robust to the value of a parameter on orbit, there is a need to react. For example, onboard replanning of the trajectory or the type of controller currently being used may be necessary. A guidance and control framework that accounts for these types of uncertainties will be very beneficial in the future given the increasing activity in space.

As a final note of motivation, NASA's Technology Roadmap specifically calls out several relevant topics in Technology Area 4.5 (System-Level Autonomy) and 4.6 (Autonomous Rendezvous and Docking), including: 4.5.2, Activity Planning, Scheduling and Execution; 4.5.3, Autonomous Guidance and Control; 4.5.7, Path and Motion Planning with Uncertainty; 4.6.2, Guidance, Navigation and Control (GNC) Algorithms; and 4.6.4 Mission System Managers for Autonomy and Automation [19].

1.6 Thesis Objectives

The main goal of this thesis is to show that, in the realm of guidance and control for spacecraft proximity operations, there is a tradeoff between safety and performance, and that advanced planning can lead to probabilistically better performance under the presence of uncertainties and uncertain events. Thus, the thrust of this thesis is to develop guidance and control mission planning software that is capable of optimizing guidance and control techniques throughout the different phases of the generalized rendezvous and docking problem described in Section 1.3. In more detail, the following objectives are set:

1. To develop baseline optimal trajectories that will probabilistically outperform current techniques given large initial uncertainties in mission critical parameters and the possibility of uncertain events;

2. To schedule reference-tracking controllers throughout the different phases of the rendezvous and docking mission such that the controllers chosen have the probabilistically best performance given parameter uncertainty in the relative motion dynamics; and

3. To implement trajectory optimization and controller-scheduling algorithms in a form that incorporates all phases of the mission and is feasible to run onboard a spacecraft during rendezvous, docking and close-proximity operations.

1.7 Thesis Roadmap

Chapter 1 has attempted to introduce and motivate the thesis topic of planning guidance and control strategies for rendezvous and docking missions in the presence of uncertainties and uncertain events. Chapter 2 will place this thesis in the context of the literature and show the research gap it fills. Chapter 3 will describe the problem in more detail mathematically and set up the approach this thesis puts forth to meet these objectives. Chapter 4 will discuss trajectory optimization techniques and Chapter 5 will discuss the selection of reference-tracking controllers under uncertainty. Finally, Chapter 6 will summarize the thesis, list contributions, and suggest future work possible in this area. In a more digestible format than the table of contents, Figure 1-2 illustrates the structure that the thesis follows.

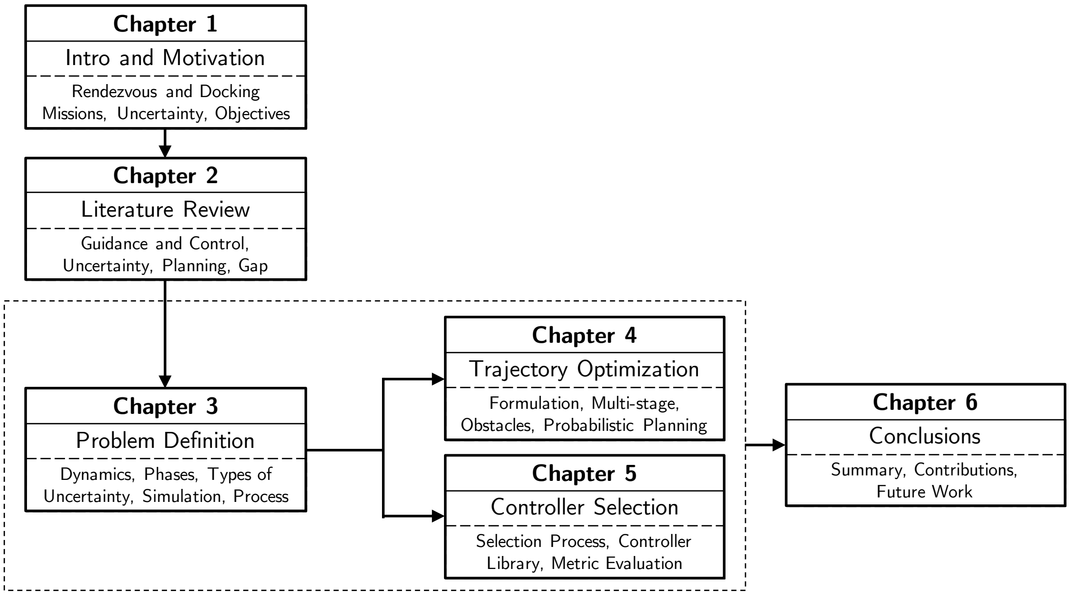

Chapter 1
Intro and Motivation
Rendezvous and Docking Missions, Uncertainty, Objectives
Chapter 2
Literature Review
Guidance and Control, Uncertainty, Planning, Gap
Chapter 3
Problem Definition
Dynamics, Phases, Types of Uncertainty, Simulation, Process
Chapter 4
Trajectory Optimization
Formulation, Multi-stage, Obstacles, Probabilistic Planning
Chapter 5
Controller Selection
Selection Process, Controller Library, Metric Evaluation
Chapter 6
Conclusions
Summary, Contributions, Future Work

Figure 1-2: The roadmap describing the contents and flow of the thesis chapters

Chapter 2

Literature Review

2.1 Approach to the Literature Survey

The literature survey presented in the next three sections focuses on the three main areas of critical interest to this thesis: (1) guidance and control algorithms throughout the previously described rendezvous, docking and joint-maneuver phases, (2) guidance and control algorithms capable of handling large uncertainty and unpredictable events, and (3) planning techniques that are capable of scheduling tasks throughout a mission. There are, of course, overlaps between these categories, which will also be discussed. However, there is very limited literature available when considering the overlap of all three areas. It is in this research gap that this thesis work falls and attempts to contribute to and advance the field. Although differences between the thesis work and the current literature will be discussed throughout Sections 2.2, 2.3, and 2.4, Section 2.5 will more formally summarize the research gap.

A major additional goal of this survey is to show the depth of research currently being performed solely on these individual topics and to identify the most promising techniques that could be adapted and used as stepping stones. Of particular interest, and where a large focus in this literature review has been placed, is identifying a spanning set of guidance and control techniques used for satellite proximity operations. An adapted subset of these algorithms will later be implemented as choices for the controller scheduler, so it is important to ensure the current state of the literature is explored. A similar endeavor is performed on the other two areas as well, attempting to find state of the art techniques to deal with uncertainties in guidance and control problems and to schedule and plan mission timelines in terms of guidance and control. In their current form, these techniques will not be perfectly applied to this thesis topic, and identifying ways in which the methods can be improved upon is another goal of the literature review.

Finally, the literature review should serve as a rich resource for others interested in the field of guidance and control in close-proximity spacecraft operations. The full survey covers almost two hundred papers and books, a large majority of which were published in the last twenty years. With this large compendium of literature on these topics, one should be able to get an excellent grasp of the state and progress of research in this field over the last couple of decades.

2.2 Guidance and Control for Rendezvous, Docking, and Joint Maneuvers

Guidance and control for spacecraft proximity operations including rendezvous and docking has been an actively studied topic from the beginnings of spaceflight and continues to be actively studied today. Joint spacecraft maneuvers have been less studied, but briefly gained popularity during the assembly of the International Space Station and have recently gained popularity due to the rise in interest in servicing and debris removal. Because of the great interest, researchers have applied a wide array of techniques to solve these problems.

Useful to read for general reference and an overall picture of the literature space, several historical surveys have been compiled on approaches for rendezvous and docking [20–22], on technologies used for rendezvous and docking [23, 24], on numerical methods for trajectory optimization [25], and on formation flying guidance and control [26, 27]. An additional resource helpful as a general reference is a 2003 textbook on automated rendezvous and docking by Fehse [1].

Guidance and control of rendezvous and docking to a stationary target in near circular orbit is probably the source of the largest amount of research, although topics range from 2DOF (Degree of Freedom) and 3DOF position-only rendezvous in circular orbits to 6DOF docking to tumbling targets in elliptical orbits. In optimal control problems, the objective tends to be either minimizing fuel or time or a combination of the two. Both real-time, online computations and off-line, *a priori* trajectory plans are explored, as well as formal proofs for more academic applications. All techniques have their own advantages and disadvantages and have been developed to different levels, however the following categorizes and details a variety of the approaches available in literature.

Glideslope, R-bar and V-bar: The more traditional approaches to rendezvous and docking found in the literature and used historically on orbit follow a glideslope trajectory from the radial or in-track directions. A glideslope is a calculated trajectory that prescribes an exponentially decreasing velocity as the chaser approaches the target. If the chaser is approaching along the radial direction, it is considered an R-bar approach, and if the chaser is approaching from the in-track or forward velocity direction, it is considered a V-bar approach. Traditionally, to conserve fuel and exploit the natural dynamics, the trajectory does not directly follow the glideslope, but instead makes hops with waypoints along the glideslope. Several tracking and control methods have been performed on orbit and proposed in literature [28–36].

Sliding Mode and Phase-Plane: Sliding mode controllers have proved useful in controlling nonlinear systems, and have been applied to the docking problem in both 3DOF position-only and full 6DOF space [37–41]. The sliding mode controllers have an n-DOF state feedback form and generally guarantee linear convergence to an artificially created "sliding surface" of order n-1 and then exponential convergence once the sliding surface is reached. Used in literature to solve the rendezvous problem with a state feedback form, phase-plane controllers change gains depending on an evaluation of the state [42, 43]. Although computationally quick and with guarantees on convergence, both of these methods do not explicitly accept path constraints and would need a probabilistic analysis in order to validate constraint adherence. Additionally, these methods are neither optimal in fuel nor time.

Artificial Potential Functions: Although succumbing to the same optimality pitfall as the sliding mode and phase-plane controllers, artificial potential function controllers are capable of accepting path constraints in rendezvous and docking problems [44–50]. The dynamics are essentially augmented with functions that create high potential in areas where constraints would be violated and low potential where the desired rendezvous location is. Again, a state feedback controller is able to guarantee convergence to the rendezvous location, however there are not hard guarantees on constraint adherence.

Genetic Algorithms: For the far-rendezvous problem, usually involving a small number of ΔV burns, genetic algorithms have been utilized to find fuel-optimal or time-optimal trajectories to rendezvous two spacecraft in vastly different orbits [51–55]. Genetic algorithms are particularly useful for finding global optima in non-convex problems with discrete variables and can be useful to find *a priori* trajectories when computation time is not a concern. However, for a docking problem, computational costs are too high to utilize genetic algorithms on a real-time system, as solution times do not scale well with the number and discretization level of variables required.

Search Methods: General path planning methods that involve exploring diverse areas to find globally optimum solutions have also been developed for the rendezvous and docking problem, with and without obstacles. The A* method, a best-first search method, has been used to plan both 3DOF position-only [56–58] and full 6DOF trajectories [59]. Rapidly exploring Random Trees (RRT*) and similar randomized search methods have been explored as well [56, 60–62]. Additionally, connectivity graph methods have been implemented such as control using positively invariant constraint admissible sets on a virtual net [63–65], dynamic programming [66], and generalized Voronoi graphs [67]. In general, these methods are good ways to search a non-convex space to find global minima, but are potentially computationally expensive and suffer from the curse of dimensionality.

Model Predictive Control: Optimality and constraint implementation are crucial for the rendezvous and docking problem, but the optimal control techniques described above are typically performed on the ground before a mission and a set of waypoints or a trajectory is sent to a spacecraft to track. This trajectory does not change if the chaser diverges from the path or if the dynamics are not properly modeled. To partially deal with this problem, Model Predictive Control (MPC) periodically re-optimizes the trajectory given the current state of the system. Typically, MPC only implements the first control of the generated optimal plan, then re-optimizes and implements the first step of the next plan, repeating until the objective function is minimized. This process allows for a pseudo state feedback behavior to the trajectory planning. Because of the ease of application to the rendezvous and docking problem with a variety of constraints, this method of control has been studied extensively in the literature [68–85] and has even been implemented and successfully employed on the PRISMA proximity operations flight system [7, 58, 86]. The curse of dimensionality is also an issue with MPC. Nevertheless, given appropriate horizon lengths and discretization, MPC can be run in real-time on satellites.

Miscellaneous Rendezvous and Docking Techniques: Other techniques found in the literature but that do not fit in the categories above include robust H_∞ control [87, 88]; analytical solutions to the Hamilton-Jacobi-Bellman (HJB) equations [89]; analytical solutions using Pontryagin's Minimum Principle (PMP) [90]; and Inverse Dynamics in the Virtual Domain (IDVD) with a sequential gradient-restoration algorithm, which has also been implemented in hardware [91–93]. Additionally, several other methods of optimization have been explored in the literature, such as Second Order Cone Programming (SOCP) [85, 94, 95], Sequential Quadratic Programming (SQP) [85, 96, 97], particle swarm optimization [98], Mixed Integer Linear Programming (MILP) [85, 99], and semi-definite programming [100]. Of particular note is the work on the ASTRO algorithm which uses slightly-suboptimal but fast optimization of Legendre polynomials to shape trajectories with constraint satisfaction, even including obstacle avoidance [101–104].

Docking to Tumbling Targets: Depending heavily on coupled attitude and position, the docking to a tumbling target problem can be considered in a different class than the traditional rendezvous and docking problem. Most of the work in this area involves analytical solutions or computationally intensive optimal control solutions that could not be run onboard a spacecraft. Nonetheless, several papers have been written on the topic, detailing the robotic arm capture of a non-cooperative tumbling target [105, 106], optimal control solutions of docking to tumbling targets using Gauss Pseudospectral Methods, Pontryagin's Minimum Principle, and Sequential Quadratic Programming respectively [107–

109], a heuristic method of docking to a tumbling target using pre-computed splines and Linear-Quadratic Regulator (LQR) tracking [110], and basic guidance to maintain docking port alignment and close distance [111]. Finally, a very recent paper implements a real-time nonlinear MPC solution to dock to a rotating target [112].

Joint Spacecraft Maneuvers: Also to be considered different than the rendezvous and docking problem is the control of two or more spacecraft after they have docked. This joint control problem has not been studied nearly as much in the literature. However, the topic is important for nearly all rendezvous missions. Specifically, there has been work on controlling the flexible modes in a robotically captured system [113], on reconfigurable control algorithms to change gains based on the current orientation and inertia properties of multiple docked spacecraft [114–117], and on attitude control of a defunct satellite by the chaser after it has docked [118]. Current literature does not explicitly discuss the transition of the control from a rendezvous and docking controller to a joint controller, other than suggesting a discrete transition from the individual to the joint estimators and controllers. The author of this thesis has also contributed some work to this topic in [119, 120], on Resource Aggregated Reconfigurable Control (RARC) covering the transition from single to dual spacecraft control.

Full Mission Hybrid Control Methods: In December 2016, the author of this thesis presented a benchmark problem for an invited session at the 2016 IEEE Conference on Decision and Control, challenging others in the session to solve the multi-stage rendezvous, docking and joint-maneuvering problem in the context of hybrid control and estimation [121]. This problem, with minor tweaks, is also presented in Chapter 3 of this thesis. The goal of the session was to have those outside the field of aerospace engineering study the problem with a heavier focus on formal methods and hybrid control. Several approaches were attempted with varying levels of success and coverage of the different phases. Of these approaches, Markov Decision Processes (MDPs) proved to be overly computationally complex and could not be applied to the 2DOF problem with four states, and only a 1DOF simplification was presented [122]. Another used differential flatness techniques in the far-field and MPC in the near-field, which

> proved promising [123]. In addition, reach-avoid sets were analyzed for a singular phase [124] and synthesis of adaptive control techniques were studied [125]. Excluding the baseline solution by the author, only one paper covered all of the phases. With different LQR and Proportional-Derivative (PD) controllers in each phase, they developed a hybrid supervisory controller that guaranteed robustness and constraint satisfaction through phase transitions [126]. With this source in particular, the author gained enough confidence in the applicability of hybrid techniques to provide robust transitions between phases that it is not explored any further in the thesis.

The literature review in this area (i.e., guidance and control for rendezvous, docking, and joint spacecraft maneuvers) serves three purposes. The first purpose is that a search of the current state of the art methods of trajectory planning and trajectory following serves as motivation for this thesis work. From this current state, it is clear that there are a variety of techniques, each with its own merits for particular problems, and that no approach has universally solved every aspect of the problem. For example, computationally fast methods with convergence guarantees have been developed for state feedback controllers following *a priori* defined trajectories, but are not able to adapt their trajectory to update path constraints and are neither optimal in fuel nor time. Methods that can update path constraints and are optimal in fuel and time are generally not computationally fast nor do they have convergence guarantees (with the notable exception of MPC). One aim of this thesis is to schedule these trajectory-tracking algorithms to allow for the most appropriate technique at different points throughout the mission. That way, for example, when computation time is not a major concern, higher fidelity, optimal methods can be used and when computation time is required to be very fast, state feedback controllers can be used. The second purpose of the literature review in this area is to investigate the best options of which to choose between to perform this controller scheduling. Several state-of-the-art methods will be adapted into case studies and directly used on the system as scheduled. The third purpose is to show that there are no current methods in the literature that are performing guidance and controller planning as described in this thesis nor are there methods in the literature discussing optimal guidance and control continuously through the full range of mission phases as described in Section 3.2.

2.3 Guidance and Control under Uncertainty

As discussed in the first chapter, uncertainty thrives in the space environment and autonomous methods are needed for spacecraft systems to be robust to these uncertainties. Most uncertainties experienced on orbit can be dealt with using a combination of current techniques developed for satellite proximity operations and those developed for control applications outside of the space realm. This literature search looks at both of these categories. The following describes guidance and control methods to address the uncertainties and uncertain events seen on orbit.

Measurement, Process, and Dynamic Model Uncertainty: Due to measurement noise, approximations in dynamics models, and process noise in actuators, spacecraft states are never known without some level of uncertainty associated with them. Several of the techniques described in Section 2.2 are not capable of handling these uncertainties and still guaranteeing convergence. For the rendezvous and docking problem, these uncertainties have been dealt with through robust control techniques such as chance constrained MPC [74, 119, 127], disturbance feedback MPC [80], Markov Chain Monte Carlo (MCMC) methods [128], and Sensitivity Weighted LQG (SWLQG) [129]; through adaptive control techniques such as adaptive output feedback control [130], Model Reference Adaptive Control (MRAC) [131], passivity-based methods with simple adaptive control [132, 133], and iterative learning control [134]; and through analysis techniques such as μ-analysis and describing functions [135]. For example, uncertainties in the dynamics model, such as an error in the eccentricity or semi-major axis of the orbit, would traditionally cause the spacecraft to diverge from the computed trajectory or consume excess fuel to counteract the limited knowledge. With these robust methods, this divergence and inefficiency is no longer a concern. However, there will be overhead in computation time and fuel efficiency when implementing these techniques, which should be taken into consideration when deciding whether the added robustness is necessary.

Mass and Inertia Property Uncertainty: Uncertainty in the mass and inertia properties of a spacecraft is also a concern, especially when docking to and controlling an unknown target object, such as a spent rocket body or other piece of space debris. One approach to tackle the identification of the mass and inertia properties of the spacecraft system is to run online mass ID algorithms with onboard sensors such as accelerometers and gyros [136]. Alternatively, adaptive controllers handle this type of uncertainty particularly well without the need to spend costly time performing system identification. Roughly, adaptive controllers either wrap feedback control and parameter estimation into one, or only adapt to output error, such that high fidelity information about the system is not needed *a priori*. Recently adaptive control has been proposed for space problems under mass and inertia property uncertainty [131–134, 137–139]. As discussed in the previous category, several of these adaptive techniques are also robust to measurement, process, and dynamic model uncertainties as well.

Angles-only Observability: In the angles-only navigation problem, where there is only a bearing angle measurement available to an estimator, there is an unobservable mode in the dynamics that can cause drift in the relative position estimation error. Woffinden, who originally discovered that the linear relative orbital dynamics were not observable, has developed optimal maneuvering to increase observability during an angles-only navigation phase [140]. Computing trajectories that will minimize the effects of this uncertainty during angles-only rendezvous has continued to be studied recently, however does require additional fuel when compared to nominal trajectories when range measurements are available [141, 142].

Actuator Failures: Reaction wheels and thrusters are critical components of any rendezvous and docking mission. If these actuators fail at any point in the mission, and the possibility of their failure has not been accounted for, a catastrophe will likely occur. For general spacecraft control, several techniques have been developed to adapt to known thruster failures [143], known reaction wheel failures [144], and unknown actuator failures [145]. Additionally, topics specific to rendezvous and docking have been explored [146–148]. Of particular interest is Breger and How's work guaranteeing the chaser is on a safe trajectory that will not intersect with the target satellite if thruster control is lost at any point [146]. The approach used during Cygnus missions is robust to missed and partial burns and is also always able to perform an abort [147]. Finally, a robust sliding-mode controller approach for the rendezvous and docking problem is used for partially failed or completely failed thrusters in [149].

Fault Detection, Isolation and Recovery: As a final category of uncertainty mitigation found in the literature, Fault Detection, Isolation and Recovery (FDIR) methods are essential to determining when failures occur and what the failures are. These methods first became popular in the aircraft industry in order to maintain control when engines fail. Typically, because the aircraft application requires fast correction for these faults, multiple dynamic models are run simultaneously and are switched between depending on the state of a failure mode estimator [150–152]. A compilation of FDIR techniques for spacecraft has been written by Wander [153], although commonly these techniques are only used to catch faults in a generic spacecraft bus. Work has also been performed on FDIR for relative navigation sensors during a docking mission [154]. Generally, this work is heavily weighted toward the FDI part, and the recovery part (topical to this thesis) is not covered as extensively. Although the topic of FDI seems to be rich, to scope the thesis, no FDI techniques will be investigated and the thesis work will assume FDI techniques developed elsewhere are providing fault feedback to the guidance and control software. Only the recovery algorithms or algorithms robust to these faults are discussed.

Again, the literature review of uncertainty mitigation techniques serves multiple purposes. First, state-of-the-art methods are found from which this thesis work can borrow ideas, particularly in the first four categories above. Second, it serves as motivation to address failure modes and uncertainties of which others in the field are currently interested. Third, it shows that a lacking area in the literature is planning and designing trajectories that are least sensitive to uncertainties and uncertain events. The idea to probabilistically plan optimal trajectories using an initially very large uncertainty in mission critical parameters and replan once the uncertainty resolves itself is not discussed in any of the literature explored in this review. This is one of the major gaps in the literature space that this thesis will help to fill. This gap lies in the overlap between Section 2.2, general guidance and control techniques for close-proximity operations, and Section 2.3, guidance and control in the presence of uncertainties.

2.4 Mission Planning and Scheduling

The third topic in this literature review, scheduling and planning methods, acts at a broader level than the previous two topics, which are directly focused on guidance and control techniques. Instead, this topic looks at the overarching task scheduling algorithms developed for aerospace systems concerned with resource allocation. The following categorizes the active areas of research into a few specialized areas and then discusses the overlap between this topic and those in the previous two sections.

> **Spacecraft Bus and Constellations:** In the late 1990's and early 2000's, the space community was interested in developing autonomous task scheduling for spacecraft. These algorithms would help plan spacecraft bus tasks such as payload operation periods, communications downlinks, and power management on several missions [155–157]. Recently, management of satellite constellations has become of increasing interest and several studies have investigated multiple-satellite and ground station scheduling problems [158, 159]. The state of the art commercial techniques that have been developed for the constellation problem include those from Orbit Logic Inc. and Planet Labs, Inc. respectively [160, 161].
>
> **Verification and Validation:** As the autonomous mission planning and scheduling methods became more popular, there was a need for proving that these techniques would work effectively under relevant circumstances. To gain the required confidence in the techniques, verification and validation methods have been developed to help test for use on orbit. While initial methods included scenario testing with model-based validation [162], this grew into more formal methods, such as those to verify mode consistency [163].
>
> **Multi-Objective Optimization Methods:** When performing mission planning and scheduling, we are generally concerned with optimization over a wide array of metrics in order to determine which substantiation of the plan or schedule is most preferred. Typically, optimization solvers only consider one metric. However, some methods have been developed to optimize and select designs under the multiple-metric framework. These are known as multi-objective optimization problems and come in many flavors. The concept of analyzing Pareto fronts is very important in this field as one can find non-dominated designs and

pick appropriate ones off the front. However, creating a Pareto front is time consuming if not done properly. The simplest method, known as a weighted sum technique, involves giving each metric a weighting that depends on the user's preference and adding all of the metrics to create a single objective value. By varying the weightings, a Pareto front can be generated by plotting the individual weighted-sum optimal designs as a function of the metrics as in [164]. A modification to the weighted-sum technique that chooses successive weightings to fully and evenly cover a Pareto front is the adaptive weighted sum technique [165]. Additionally, genetic algorithms can be used in a multi-objective form to create Pareto fronts [166]. As uncertainty is an issue in our problem, non-deterministic Pareto fronts have also been analyzed in [167]. If preferences are formatted in a ranking system where improvement in one metric should be weighted above any improvement in all other metrics, lexicographic methods can be used to choose designs [168, 169].

Orbit Design and Maneuvers: Autonomous mission planning has also been performed on orbit design for several problems. Orbital design for debris removal has been studied through optimization of particular orbits that will intersect with the largest number of debris pieces [170]. A traveling salesman problem has been explored for the on-orbit servicing and refueling of GEO spacecraft to autonomously determine which satellites to service at which times [171]. Additionally, work has been performed on autonomously determining and planning orbital maneuvers based on the spacecraft available resources, such as battery power, fuel and processing power available [172]. Hybrid optimal control has also been used as an approach to plan maneuvers for an asteroid rendezvous mission [173]. Generally, this type of mission planning ignores the controller selection part of this thesis and typically deals with large orbital maneuvers and ΔV burns.

Guidance and Control Selection: Specifically looking at the problem of optimally scheduling controllers over a spacecraft mission proves to be very sparsely covered in the literature. However, there are a few techniques from other fields that have looked into similar problems. For example, in the chemical processing field, MPC has been used to schedule Proportional-Integral-Derivative (PID) controller gains, effectively performing a receding horizon controller selection algorithm with a very limited set of controllers [174]. In the more mathematical controls sense, Multi-Objective Genetic Algorithms (MOGA) have been used to generate Pareto fronts useful in the design of sliding mode controllers for nonlinear systems, but do not look at varying the controllers throughout a mission [175]. Taking a step back to the space realm, attitude controllers have been developed to plan slews based on a number of nontraditional constraints and scheduling Lyapunov functions to guarantee convergence [176].

Planning under Uncertainty: As guidance and control in the real world requires consideration of uncertainty, so does mission planning. This paragraph covers the overlap between Sections 2.3 and 2.4. Along with the previously mentioned work on spacecraft bus task planning, the idea of iterative repair of autonomously generated plans has increased the responsiveness of re-planning due to unexpected events [177]. Additionally, around the same time, work on the New Millennium Deep Space One mission involved autonomy that would diagnose and recover from faults through re-planning and reallocating resources [178]. An unexplored architecture for autonomous mission planning for unmanned aircraft refueling missions under the presence of uncertainty has been described to use adaptive-neural inference and robust adaptive controllers with dynamic inversion [179]. Currently, the Model-based Embedded and Robotic Systems (MERS) group at MIT is actively studying a resilient spacecraft executive for Mars rover planning which deals with probabilistic risk assessment [180] and additional methods for planning under uncertainty involving partially-observable Markov decision processes (POMDPs) [181, 182]. The MACE II project dealt with a concept dubbed self-reliance which necessitates autonomous multi-level mission planning and control, fault detection, and controller reconfiguration through adaptive neural networks and system identification [183–185]. The main concern with MACE II was unknown physical parameters of a flexible spacecraft test article deployed in the space shuttle.

Current Approaches for Rendezvous and Docking: This paragraph covers the overlap between Sections 2.2 and 2.4 in terms of historical planning approaches that have been used on orbit during rendezvous and docking missions. The planning of sensor usage and controller implementation has been discussed in the literature for several of the already completed rendezvous and docking missions, including PRISMA [7, 58, 86], ATV [2], and the Space Shuttle [186]. Additionally, the methods from a few to-be-executed missions have been compared against previous missions, including a comparison of Space Shuttle and Orion plans [32] and a comparison of ATV, PRISMA and SIMBOL-X plans [187]. Although plans are often mission specific and depend on the sensors available at different distances for that particular mission, a commonality at this point is that all guidance and controller selection on these missions is being planned beforehand taking countless hours from teams of engineers on the ground. Additionally, if uncertain events occur during the mission, safe modes are usually initiated until ground control can intervene with appropriate action.

Proposed Approaches for Rendezvous and Docking: This paragraph covers the overlap between Sections 2.2 and 2.4 in terms of proposed planning methods for generic rendezvous and docking missions in the future. Several approaches deal with optimal planning of ΔV burns for rendezvous but do not touch on controllers used for those maneuvers nor on uncertainty [188–190]. Others cover the high-level planning and scheduling for the spacecraft bus during rendezvous and docking but still do not cover controllers or uncertainty [191, 192]. An expansive study was performed for a multiple satellite refueling mission that dealt with different phases of rendezvous and docking, each with different constraints and guidance approaches, but used the same method of control and did not perform autonomous planning for guidance and control algorithms [171]. Guglieri developed a method of maneuver planning for the rendezvous problem where a standard PID controller would track a maneuver selected by an autonomous maneuver scheduler [193]. Another interesting study was performed using Mixed Integer Nonlinear Programming (MINP) to plan sensor switching, the number and repetition of set maneuvers, and the time to docking, however did not detail any control methods used to execute these maneuvers [34]. Finally, and potentially most topical, is the autonomous mission management software built by the Charles Stark Draper Laboratory for rendezvous missions that selects from high and low thrust techniques and can handle thruster failure modes [194].

The purpose of including this topic in the literature review is similar to the previous two sections: to show the state of the art, which could potentially be used in this work, and to show how this thesis work differs from what is currently in the literature. As the overlaps between these sections were discussed in this section, the investigated papers begin to start looking closer to what is suggested here. However, in general, the literature shows that there is significant work in spacecraft mission planning, but most of the work focuses on higher level planning for the spacecraft bus and science objectives, and very little is focused on methods to perform planning for the rendezvous and docking control problem. Furthermore, there is no discussion of probabilistically planning the choices of controllers throughout different phases of any type of mission. The Draper work in [194] and the work in [34] come the closest to the thesis work, however they each fall short in a few areas. The methods do not consider the range of uncertainty required for fully autonomous rendezvous and docking, they do not perform any detailed guidance and control work other than computation of ΔV burns, and they do not cover joint maneuvers and undocking.

2.5 Summary of Research Gap

As the previous three sections detail, there are three main areas that are related to the thesis topic: guidance and control for spacecraft rendezvous, docking and joint maneuvers; guidance and control under uncertainty; and mission planning and scheduling. Each of these areas has been studied in the past, both in and out of the spacecraft control context; even the pairwise combinations of these areas have been studied. However, the true, full combination of all three of the areas has neither been investigated in theory nor in application. Thus, one of the main literature gaps to which this thesis will attempt to contribute is the topic of planning of spacecraft guidance and control over multiple stages in the presence of uncertainties. The contribution most applicable to this gap is the probabilistic (and subsequent real-time) planning of reference-tracking controllers throughout the phases of the mission depending on the current knowledge of uncertain parameters. Furthermore, although there are techniques that deal with trajectory planning for rendezvous and docking under uncertainty (covering the overlap of Section 2.2 and Section 2.3), there is still a gap in this area detailing the probabilistic planning of trajectories given the possibility of uncertain events later in the mission that would drastically affect the trajectory. No work has been completed on the topic aside from this thesis. Specifically, this thesis will be focusing on the full rendezvous, docking, and joint maneuver problem. This specific mission skeleton has not been investigated in the literature in the context of any of the three areas discussed above. So, another, smaller literature gap that this thesis will help to fill is that of guidance and control for this full mission profile.

Chapter 3

Problem Definition

This chapter is meant to fully describe and set up the problem for this thesis. This description includes the mathematical formulation of the dynamics in Section 3.1; both a conceptual and mathematical formulation of the mission phases in Section 3.2; a general overview and classification of different types of uncertainty in Section 3.3; a detailed presentation of the simulation created to test solutions to this problem in Section 3.4; and an overview of the approach and solution process used in the remainder of the thesis in Section 3.5.

As a special note, Section 3.1.1 and Section 3.2 combine to form a spacecraft benchmark problem meant for others in the fields of astrodynamics, guidance and control, hybrid systems and formal methods to solve and compare results against. The problem definition itself was developed for an invited session at the 2016 IEEE CDC titled, "A Spacecraft Benchmark Problem for Analysis and Control of Hybrid Systems." The papers from this session were included in the literature review in Section 2.2, but for convenience, the lead paper, [121], was written by the author of this thesis, with supporting papers of [122], [123], [124], [125], and [126]. As such, the respective sections will define options that are meant to make the problem interesting but not too difficult for the state of the art techniques in each of these fields. Not all of these option cases are solved for in this thesis, although a majority are.

3.1 Close-proximity Spacecraft Orbital Dynamics

This section discusses both a linear form of relative-motion orbital dynamics between a target and chaser in Section 3.1.1 and a nonlinear version computed to include perturbations and higher-order dynamics in Section 3.1.2. The linear form will be exclusively used for GNC execution, while the full nonlinear dynamics will be implemented in the simulation.

3.1.1 Linear Dynamics

Rendezvous and docking is often performed in a relative coordinate frame describing the difference in position and velocity between the chaser and target spacecraft. The Hill's frame as described in Figure 3-1, is similar to a Local Vertical Local Horizon (LVLH) frame centered on the target spacecraft. The $\hat{\imath}$-direction points radially outward away from the Earth, the $\hat{\jmath}$-direction points in the in-track, orbital velocity direction of the target satellite, and the $\hat{k}$-direction points in the cross-track direction out of the orbital plane to complete the triad. This coordinate system assumes a circular orbit of the target spacecraft to maintain orthogonality.

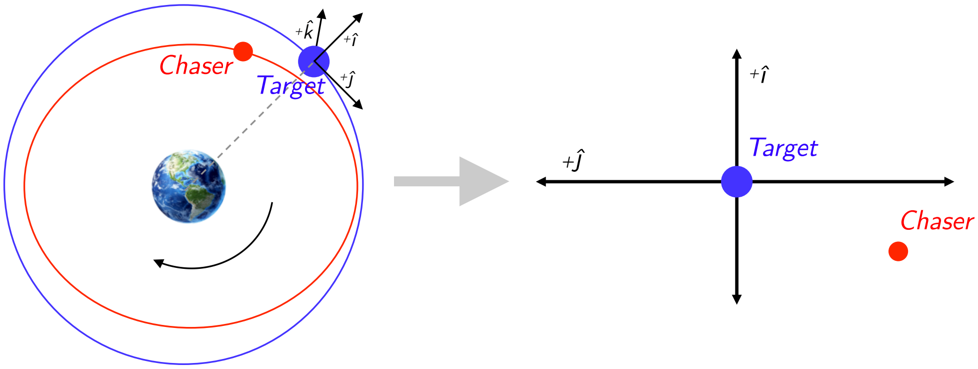


Figure 3-1: Hill's reference frame used for relative spacecraft state definition

In the Hill's frame, the relative translational motion of the chaser spacecraft with respect to the target can be defined by the Clohessy-Wiltshire-Hill (CWH) equations as developed in [195]. If the position vector is $x\hat{\imath} + y\hat{\jmath} + z\hat{k}$, the dynamics are represented as

$$\begin{aligned} \ddot{x} - 2n\dot{y} - 3n^2 x &= \frac{F_x}{m}, \\ \ddot{y} + 2n\dot{x} &= \frac{F_y}{m}, \text{ and} \\ \ddot{z} + n^2 z &= \frac{F_z}{m}, \end{aligned} \tag{3.1}$$

where m is the mass of the chaser spacecraft; F_x, F_y and F_z are the thrust forces applied by the chaser spacecraft; $n = \sqrt{\mu/r_0^3}$ is the mean motion of the target satellite; and r_0 is the radius of the target's circular orbit. These dynamics can be written in linear time-invariant (LTI), state-space form as

$$\dot{\mathbf{x}} = A_i \mathbf{x} + B_i \frac{\mathbf{u}}{m} = \text{CWH}(\mathbf{x}, \mathbf{u}, n, m), \tag{3.2}$$

where for the 2DOF case, $i = 2$, $\mathbf{x} = [\dot{x},\ \dot{y},\ \ddot{x},\ \ddot{y}]^T$ and $[F_x,\ F_y]^T = f_{thr}\mathbf{u}$, with f_{thr} being the thruster force magnitude which scales the control input. For the 3DOF case, $i = 3$, $\mathbf{x} = [\dot{x},\ \dot{y},\ \dot{z},\ \ddot{x},\ \ddot{y},\ \ddot{z}]^T$, and $[F_x,\ F_y,\ F_z]^T = f_{thr}\mathbf{u}$. A_3 and B_3 can easily be formed from (3.1) as

$$A_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix}, \quad \text{and} \quad B_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ f_{thr} & 0 & 0 \\ 0 & f_{thr} & 0 \\ 0 & 0 & f_{thr} \end{bmatrix}, \tag{3.3}$$

where normally for a controller, the B matrix would also be scaled by the inverse of the mass, $1/m$, as in (3.2). The chaser spacecraft is assumed to have mass, m_c, and the target spacecraft to have mass, m_t. The reason the mass is not included in this B matrix is such that the dynamics can be represented easily later as a function of mass as in (3.2).

Note that the motion in the $\hat{k}$-direction is decoupled from the $\hat{\imath}$ and $\hat{\jmath}$-directions, such that A_2 and B_2 can easily be separated from the full A_3 and B_3. In reality, this problem has 3DOF, although the 2DOF formulation is provided as it is the smallest state formulation that still is traceable to on-orbit missions. Many hybrid control techniques that may attempt to solve this problem are sensitive to state dimension and reducing the state vector by two elements could potentially bring their approaches into the realm of feasibility.

To match standard representation in relative spacecraft dynamics, the state vector $\mathbf{x}$ is made up of x, y and z positions and velocities and the measurement vector is $\mathbf{y}$. Bold notation represents vectors; scalars are italicized.

Additionally, since the controllers will eventually need to be implemented in discrete time and it is useful to be able to propagate the dynamics in a linear form, we obtain the discrete-time version of these dynamics as

$$\mathbf{x}(k+1) = A_{d_i}\mathbf{x}(k) + B_{d_i}\mathbf{u}(k), \tag{3.4}$$

where k is the current time-step,

$$\begin{aligned} A_{d_i} &= e^{A_i T}, \\ B_{d_i} &= \int_{t=0}^{T} e^{A_i t}\frac{B_i}{m}dt, \end{aligned} \tag{3.5}$$

and T is the discretization time, or time between successive steps k and $k+1$.

Fortunately, a closed form solution to A_{d_3} is available commonly in the literature as

$$A_{d_3} = \begin{bmatrix} 4-3\cos(nT) & 0 & 0 & \frac{1}{n}\sin(nT) & \frac{2}{n}\left(1-\cos(nT)\right) & 0 \\ 6\left(\sin(nT)-nT\right) & 1 & 0 & -\frac{2}{n}\left(1-\cos(nT)\right) & \frac{1}{n}\left(4\sin(nT)-3nT\right) & 0 \\ 0 & 0 & \cos(nT) & 0 & 0 & \frac{1}{n}\sin(nT) \\ 3n\sin(nT) & 0 & 0 & \cos(nT) & 2\sin(nT) & 0 \\ -6n\left(1-\cos(nT)\right) & 0 & 0 & -2\sin(nT) & 4\cos(nT)-3 & 0 \\ 0 & 0 & -n\sin(nT) & 0 & 0 & \cos(nT) \end{bmatrix}, \tag{3.6}$$

where again the 2DOF equations are easily separated. If implementing the control using an impulsive ΔV approach for control input, the B_{d_3} matrix can be instead represented as

$$B_{d_3} \quad = \quad e^{A_3 T} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.7}$$

because an instantaneous change in velocity is performed at each time step and that new velocity is then propagated forward in time with the A_{d_3} matrix. From here, we use the results from (3.5) and (3.6) to simplify the result to

$$B_{d_3} = \begin{bmatrix} \frac{1}{n}\sin(nT) & \frac{2}{n}\left(1-\cos(nT)\right) & 0 \\ -\frac{2}{n}\left(1-\cos(nT)\right) & \frac{1}{n}\left(4\sin(nT)-3nT\right) & 0 \\ 0 & 0 & \frac{1}{n}\sin(nT) \\ \cos(nT) & 2\sin(nT) & 0 \\ -2\sin(nT) & 4\cos(nT)-3 & 0 \\ 0 & 0 & \cos(nT) \end{bmatrix}. \tag{3.8}$$

Otherwise, when using continuous force commands, the full expression for B_{d_3} will need to be integrated or solved for numerically as shown in (3.5). This process results in

$$B_{d_3} = \frac{f_{thr}}{m} \begin{bmatrix} \frac{1}{n^2}\left(1-\cos(nT)\right) & \frac{2}{n^2}\left(nT-\sin(nT)\right) & 0 \\ -\frac{2}{n^2}\left(nT-\sin(nT)\right) & \frac{4}{n^2}\left(1-\cos(nT)\right)-\frac{3}{2}T^2 & 0 \\ 0 & 0 & \frac{1}{n^2}\left(1-\cos(nT)\right) \\ \frac{1}{n}\sin(nT) & \frac{2}{n}\left(1-\cos(nT)\right) & 0 \\ -\frac{2}{n}\left(1-\cos(nT)\right) & \frac{4}{n}\sin(nT)-3T & 0 \\ 0 & 0 & \frac{1}{n}\sin(nT) \end{bmatrix}. \tag{3.9}$$

Note that in Mission Phases 1 though 2, $m = m_c$ in (3.9) as the two spacecraft are not docked at this point, and in Mission Phase 3, $m = m_c + m_t$, because they are docked.

The linear formulation of these dynamics facilitates the use of the CWH equations in guidance and control algorithms for spacecraft rendezvous and docking missions. As such, most, if not all, close-proximity rendezvous and docking missions employ these equations today. It is worthwhile to note, however, that as the distance between the target and chaser spacecraft grows, the accuracy of the CWH equations diminishes. In the time frame of a rendezvous and docking mission in geostationary orbit, it is only relevant to use these equations into the tens of kilometers of separation. For lower orbits and longer missions, the separation distances for which the CWH equations are applicable can shrink significantly. For this mission in geostationary orbit, the thesis will use the linear dynamics in the controllers and estimators, as they maintain a very good approximation of the true dynamics throughout the mission.

3.1.2 Nonlinear Dynamics

The CWH equations described in the previous section are a linear approximation of the relative motion problem, where in reality the dynamics are nonlinear both with the inverse-square gravity formulation and with several orbital perturbations. To better represent the true dynamics seen on orbit, nonlinear modeling of these dynamics can be performed. This section deals with this nonlinear formulation of the relative-motion dynamics between the target and chaser. In most cases, the GNC laws will still use the linear formulation from the previous section, while the full nonlinear models will be used in the simulation as the "truth" dynamics.

The relative-motion nonlinear dynamics of the chaser with respect to the target in the CWH frame can be expressed as a function of the current state and control,

$$\dot{\mathbf{x}}^{CWH} = \mathbf{f}_{rm}\left(\mathbf{x}^{ECI}_{target},\ \mathbf{x}^{ECI}_{chaser},\ \mathbf{u}^{CWH},\ t_{epoch},\ m_c,\ m_t\right), \tag{3.10}$$

where $\mathbf{x}^{CWH}$ is the relative state in the CWH frame, $\mathbf{x}^{ECI}_{target}$ is the target's state in the Earth Centered Inertial (ECI) frame, $\mathbf{x}^{ECI}_{chaser}$ is the chaser's state in the ECI frame, $\mathbf{u}^{CWH}$ is the chaser's executed control force in the CWH frame, t_{epoch} is the time elapsed since the epoch which defines the ECI reference frame. What happens inside $\mathbf{f}_{rm}$ is quite complicated in some respects. The important thing to note here is that the dynamics will be described in the ECI frame and only converted to the CWH frame at the output. This is due to the fact that the nonlinear dynamics are much easier to represent in the ECI frame. Thus, the initial conditions must be known in the ECI frame to kick things off, and then the dynamics can be propagated in the ECI frame and output in the CWH frame.

Montenbruck and Gill describe the relative magnitudes and dominance of accelerations from different sources for a range of Earth orbits in a very useful chart in [196]. The dominant forces in GEO are as follows (in order of decreasing magnitude): J2, 3rd body from Moon, 3rd body from Sun, Solar Radiation Pressure (SRP), and J3 through J6. In LEO, the drag force rises to second in the list behind J2. In addition to the inverse-square gravity dynamics, each of these perturbations will be modeled as follows.

The acceleration for a single spacecraft in the ECI frame can be represented as

$$\mathbf{a}_{total} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}^{ECI} = \mathbf{a}_{GM} + \mathbf{a}_{J2-6} + \mathbf{a}_{3rd} + \mathbf{a}_{SRP} + \mathbf{a}_{drag} + \mathbf{a}_{ctrl}, \tag{3.11}$$

where $\mathbf{a}_{GM}$ is the gravitational acceleration from the Earth, $\mathbf{a}_{J2-6}$ is the acceleration due to the zonal harmonics of the Earth, $\mathbf{a}_{3rd}$ is the acceleration from the third body gravity interactions from the moon and sun, $\mathbf{a}_{SRP}$ is the acceleration from solar radiation pressure, $\mathbf{a}_{drag}$ is the acceleration due to atmospheric drag, and $\mathbf{a}_{ctrl}$ is the acceleration provided by the thrusters. Note that all $\mathbf{a}$ vectors are in the ECI frame. All of these individual components will be computed and propagated for the chaser and target independently. The following is a description of the components of acceleration due to each of the sources listed above as defined and modeled in [197].

The two-body, Earth-satellite, inverse-square acceleration is

$$\mathbf{a}_{GM} = -\frac{\mu}{r_{\oplus sat}^2} \cdot \frac{\mathbf{r}_{\oplus sat}}{\|\mathbf{r}_{\oplus sat}\|}, \tag{3.12}$$

where $r_{\oplus sat} = \|\mathbf{r}_{\oplus sat}\|$, the magnitude of the radius from the center of the Earth ($\oplus$) to the satellite in question, and the multiplicative unit vector points from the Earth to the satellite to orient the acceleration in the correct direction.

The inverse-square gravity law essentially treats the Earth as a point mass or a perfect sphere, yet in reality the Earth has oblateness and higher order spherical harmonics associated with a slightly distorted spheroid. Due to this property, there is an imbalanced mass distribution, and the gravitational field of the Earth varies depending on the location in orbit. J2 through J6 represent the second through sixth zonal spherical harmonics that help model the true shape of the Earth and the gravitational accelerations as a function of orbital location. Together, J2 through J6 affect the acceleration of a satellite in the ECI frame as

$$
\begin{aligned}
\ddot{x}_{J2-6} = -\frac{\mu x}{r_{\oplus sat}^3}\Bigg[&- \tfrac{3}{2}J_2\left(\frac{r_\oplus}{r_{\oplus sat}}\right)^2\left(5\frac{z^2}{r_{\oplus sat}^2} - 1\right) \\
&+ \tfrac{5}{2}J_3\left(\frac{r_\oplus}{r_{\oplus sat}}\right)^3\left(3\frac{z}{r_{\oplus sat}} - 7\frac{z^3}{r_{\oplus sat}^3}\right) \\
&- \tfrac{5}{8}J_4\left(\frac{r_\oplus}{r_{\oplus sat}}\right)^4\left(3 - 42\frac{z^2}{r_{\oplus sat}^2} + 63\frac{z^4}{r_{\oplus sat}^4}\right) \\
&- \tfrac{3}{8}J_5\left(\frac{r_\oplus}{r_{\oplus sat}}\right)^5\left(35\frac{z}{r_{\oplus sat}} - 210\frac{z^3}{r_{\oplus sat}^3} + 231\frac{z^5}{r_{\oplus sat}^5}\right) \\
&+ \tfrac{1}{16}J_6\left(\frac{r_\oplus}{r_{\oplus sat}}\right)^6\left(35 - 945\frac{z^2}{r_{\oplus sat}^2} + 3465\frac{z^4}{r_{\oplus sat}^4} - 3003\frac{z^6}{r_{\oplus sat}^6}\right)\Bigg], \\
\ddot{y}_{J2-6} = \frac{y}{x}\ddot{x}_{J2-6}, &\quad \text{and} \\
\ddot{z}_{J2-6} = -\frac{\mu z}{r_{\oplus sat}^3}\Bigg[&- \tfrac{3}{2}J_2\left(\frac{r_\oplus}{r_{\oplus sat}}\right)^2\left(5\frac{z^2}{r_{\oplus sat}^2} - 3\right) \\
&+ \tfrac{3}{2}J_3\left(\frac{r_\oplus}{r_{\oplus sat}}\right)^3\left(10\frac{z}{r_{\oplus sat}} - \frac{35}{3}\frac{z^3}{r_{\oplus sat}^3} - \frac{r_{\oplus sat}}{z}\right) \\
&- \tfrac{5}{8}J_4\left(\frac{r_\oplus}{r_{\oplus sat}}\right)^4\left(15 - 70\frac{z^2}{r_{\oplus sat}^2} + 63\frac{z^4}{r_{\oplus sat}^4}\right) \\
&- \tfrac{1}{8}J_5\left(\frac{r_\oplus}{r_{\oplus sat}}\right)^5\left(315\frac{z}{r_{\oplus sat}} - 945\frac{z^3}{r_{\oplus sat}^3} + 693\frac{z^5}{r_{\oplus sat}^5} - 15\frac{r_{\oplus sat}}{z}\right) \\
&+ \tfrac{1}{16}J_6\left(\frac{r_\oplus}{r_{\oplus sat}}\right)^6\left(245 - 2205\frac{z^2}{r_{\oplus sat}^2} + 4851\frac{z^4}{r_{\oplus sat}^4} - 3003\frac{z^6}{r_{\oplus sat}^6}\right)\Bigg],
\end{aligned}
\tag{3.13}
$$

where J_2 through J_6 are small coefficients defined in [197] and many other sources; $r_\oplus$ is the radius of the Earth; and x, y, and z are the satellite's position in the ECI frame. J_3 through J_6 are smaller effects and correspondingly are three orders of magnitude smaller than J_2. Note that this acceleration is mostly affected by the z-component of satellite's position, such that if the inclination of the orbit is zero degrees, the effects differ significantly.

While in geosynchronous orbit, the gravity from other celestial bodies actually exerts forces on the satellite and affect its orbit. The accelerations due to these forces are known as 3rd body effects. The Moon ($\bigcirc$) and Sun ($\odot$) cause the largest 3rd body acceleration in the geosynchronous orbit altitude. The ephemerides of the Moon and Sun with respect to the Earth are very important in this calculation. The SPICE toolbox within NASA's Orbital Determination Toolbox with the most recent ephemeris file is used to find these ephemerides [198]. The combined acceleration from both of these sources is computed in the ECI J2000 frame as

$$\mathbf{a}_{3rd} = \mu_{\odot}\left(\frac{\mathbf{r}_{sat\odot}}{\|\mathbf{r}_{sat\odot}\|^3} - \frac{\mathbf{r}_{\oplus\odot}}{\|\mathbf{r}_{\oplus\odot}\|^3}\right) + \mu_{\bigcirc}\left(\frac{\mathbf{r}_{sat\bigcirc}}{\|\mathbf{r}_{sat\bigcirc}\|^3} - \frac{\mathbf{r}_{\oplus\bigcirc}}{\|\mathbf{r}_{\oplus\bigcirc}\|^3}\right), \tag{3.14}$$

where $\mathbf{r}_{sat\odot}$ is the position vector from the satellite to the Sun, $\mathbf{r}_{\oplus\odot}$ is from the Earth to the Sun, $\mathbf{r}_{sat\bigcirc}$ is from the satellite to the Moon, $\mathbf{r}_{\oplus\bigcirc}$ is from the Earth to the Moon, and $\mu_{\odot}$ and $\mu_{\bigcirc}$ are the gravitational parameters of the Sun and Moon respectively.

Solar Radiation Pressure (SRP) is also included with additional importance on the ephemeris of the sun with respect to the Earth. This acceleration is due to the reflectance of the Sun's radiation. Therefore, in computing the acceleration, there is a required assumption on the size of the spacecraft (mass and sun-facing area, $\mathcal{A}_{\odot}$) as well as the reflective properties of the spacecraft. For this analysis, the effective sun-facing area is assumed to be 10 m^2 and the mass of the chaser and target are defined in Section 3.2.8. The acceleration due to SRP is

$$\mathbf{a}_{SRP} = -\frac{p_{sr}c_r\mathcal{A}_{\odot}}{m} \cdot \frac{\mathbf{r}_{sat\odot}}{\|\mathbf{r}_{sat\odot}\|}, \tag{3.15}$$

where $p_{sr} = 4.57 \times 10^{-6}$ N/m^2 is the solar pressure at Earth and $c_r = 1$ is the assumed coefficient of reflection of the spacecraft.

As the final disturbance, the acceleration due to atmospheric drag is included and depends greatly on the density of the atmosphere, ρ_{atm}, which is found for a specific altitude through interpolation of a standard atmospheric density data table. The drag coefficient, c_d, is assumed to be 2.2 as given by [199]. The cross-sectional area is assumed to be similar to the sun-facing area and is set to 10 m^2. The drag acceleration can be computed as

$$\mathbf{a}_{drag} = -\frac{1}{2}\rho_{atm}\mathcal{A}_d c_d v_{sat}^2, \tag{3.16}$$

where v_{sat} is the norm of the velocity of the satellite in the ECI frame.

Now that the orbital dynamics have been described, the acceleration commanded by the spacecraft controller needs to be considered. Since the controller works in the relative frame between the target and chaser, we need to transform the commanded thrust vector in the relative, CWH frame into the ECI frame. To perform this transformation, the instantaneous rotation matrix from the CWH to the ECI frame is required. This rotation matrix can be represented as

$$ {}^{CWH}R^{ECI}_{3\times 3} = \left[\begin{array}{c:c:c} | & | & | \\ \dfrac{\mathbf{r}_{\oplus target}}{\|\mathbf{r}_{\oplus target}\|} & \dfrac{\mathbf{r}_{\oplus target}}{\|\mathbf{r}_{\oplus target}\|} \times \dfrac{\mathbf{v}_{target}}{\|\mathbf{v}_{target}\|} \times \dfrac{\mathbf{r}_{\oplus target}}{\|\mathbf{r}_{\oplus target}\|} & \dfrac{\mathbf{r}_{\oplus target}}{\|\mathbf{r}_{\oplus target}\|} \times \dfrac{\mathbf{v}_{target}}{\|\mathbf{v}_{target}\|} \\ | & | & | \end{array}\right], \tag{3.17} $$

where $\mathbf{v}_{target}$ is the velocity of the target spacecraft in the ECI frame. Note that the double cross product is necessary in the second column as the velocity direction of the target will not form a perfect orthogonal triad due to the orbital perturbations bringing it off a perfectly circular orbit. The same rotation matrix is used for the chaser and the target, because the CWH coordinate frame is defined based on the target's position. This rotation matrix, ${}^{CWH}R^{ECI}_{3\times 3}$, should only be used to rotate a vector from one frame to another and does not describe the full state transformation from the CWH to ECI frame, because the CWH frame is a rotating reference frame. This is sufficient for the thruster acceleration transformation, however, which is computed as

$$ \mathbf{a}_{ctrl} = {}^{CWH}R^{ECI}_{3\times 3} \frac{f_{thr}}{m} \mathbf{u}^{CWH}, \tag{3.18} $$

where the mass, m, is specified for the appropriate satellite. Typically, this is only carried out for the chaser satellite or the docked combination of a chaser and target satellite. Depending on the control mode selected, the duration of this acceleration may differ from the control period and finer propagation times may be required.

All of the dynamics are now known and can be propagated as is. The only component left of the nonlinear function, $\mathbf{f}_{rm}$, is to transform the propagated target and chaser states into a relative state in the CWH frame. To start, the relative state in the ECI frame is calculated by taking the difference between the chaser and target states such that

$$ \mathbf{x}^{ECI}_{rel} = \mathbf{x}^{ECI}_{chaser} - \mathbf{x}^{ECI}_{target}. \tag{3.19} $$

Then the rotation matrix from the ECI frame to the CHW frame is just the transpose of the rotation matrix from the CWH to ECI frame computed earlier in (3.17), such that ${}^{ECI}R^{CWH}_{3\times 3} = \left({}^{CWH}R^{ECI}_{3\times 3}\right)^T$. This can be directly used to multiply the position states, however the velocity states will have an extra component due to the fact that the CWH frame is rotating with respect to the ECI frame at a rate of $\omega_{_{CWH}}$. The chaser's relative state to the target spacecraft in the CWH frame is

$$\mathbf{x}^{CWH} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}^{CWH} = \left[\begin{array}{c} | \\ {}^{ECI}R^{CWH}_{3\times 3}\mathbf{x}^{ECI}_{rel}(1:3) \\ | \\ \hdashline {}^{ECI}R^{CWH}_{3\times 3}\mathbf{x}^{ECI}_{rel}(4:6) - \begin{pmatrix} -\omega_{_{CWH}} y^{CWH} \\ \omega_{_{CWH}} x^{CWH} \\ 0 \end{pmatrix} \end{array}\right], \tag{3.20}$$

where

$$\omega_{_{CWH}} = \frac{\|\mathbf{h}_{target}\|}{\|\mathbf{r}^{CWH}_{\oplus target}\|^2} = \frac{\|\mathbf{r}^{CWH}_{\oplus target} \times \mathbf{v}^{CWH}_{target}\|}{\|\mathbf{r}^{CWH}_{\oplus target}\|^2}, \tag{3.21}$$

and $\mathbf{h}_{target}$ is the specific angular moment vector of the target spacecraft in its Earth orbit.

This relative state is needed by the controller and estimator because it is much easier to work in a relative reference frame for the rendezvous and docking problem. All of the measurements and trajectory states are known in the relative frame, so it makes sense to give the results of the nonlinear dynamics in that frame as well. As these dynamics are significantly more complicated than the CWH equations from (3.1), typically they will not be used in the loop for control, but only for analysis before the mission in simulation. For the mission duration and altitude of the orbit examined in this thesis, the nonlinear dynamics do not depart from the linear dynamics by a large enough margin to warrant use in on-board control algorithms.

3.2 Mission Phase Definitions

The rendezvous and docking problem in this paper is split into four phases. Each phase is defined by a separation distance between the chaser and target spacecraft, closing this distance from up to 10 km to 0 m, and then performing a maneuver once the satellites are docked. The relative displacement vector from the target to the chaser is defined in the Hill's frame as $\boldsymbol{\rho} = [x, y, z]^T$ in 3DOF and $\boldsymbol{\rho} = [x, y]^T$ in 2DOF. The magnitude of this displacement vector is the 2-norm and is represented as

$$\rho = \|\boldsymbol{\rho}\| = \sqrt{x^2 + y^2 + z^2}\ . \tag{3.22}$$

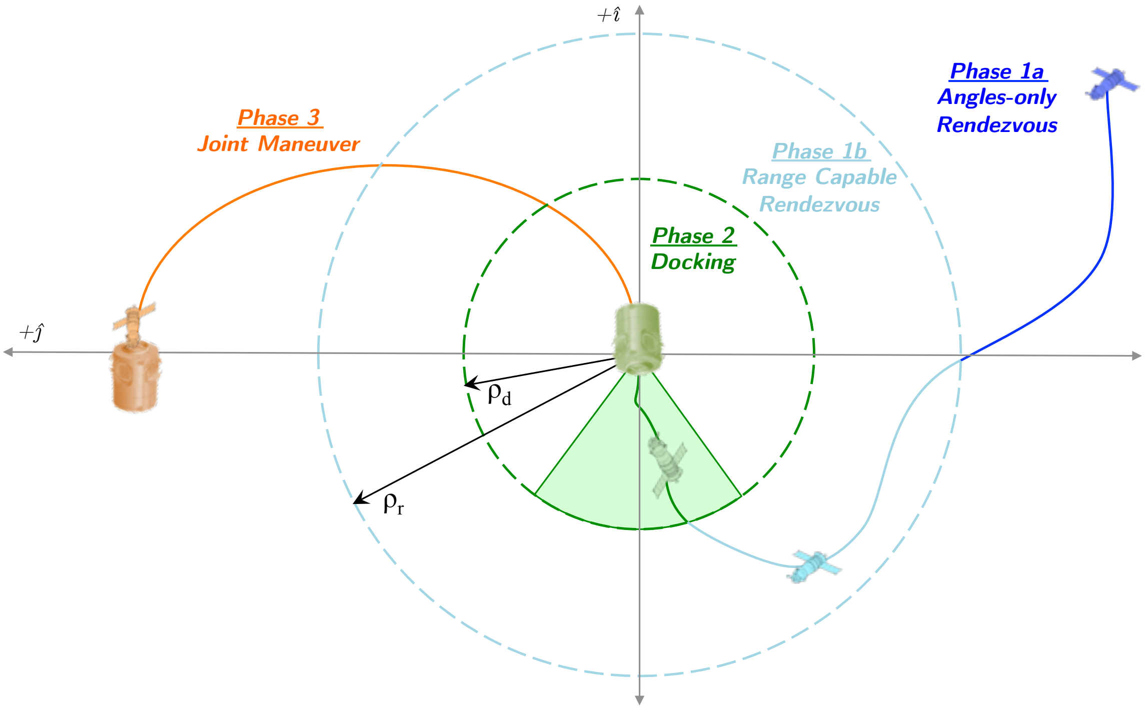


Figure 3-2: Description of the overall mission phases (not to scale)

As seen in Figure 3-2, the chaser spacecraft begins in Phase 1 while $\rho \geq \rho_r$, the separation distance at which ranging data is now available to the target spacecraft. In Phase 1a, the chaser attempts to rendezvous with the target but only has angular measurements available. While $\rho_d \leq \rho < \rho_r$, the mission moves into Phase 1b, where the chaser spacecraft now has a ranging measurement to the chaser spacecraft and must position itself for the Phase 2 docking. After the chaser moves such that $\rho < \rho_d$, the docking phase, Phase 2, is initiated, and additional docking port constraints are actived. Once the spacecraft dock (i.e., $\rho = 0$ m), both spacecraft move into Phase 3, where the joint assembly must move to the relocation position.

The problem can be generalized for any initial position and relocation position, however, this thesis will examine a specific instantiation of this problem. The position of the chaser satellite at the initial time, t_0, is 3 km behind the target spacecraft in the in-track direction. The distance at which ranging measurements are available, ρ_r, is 1 km. The docking phase begins at a ρ_d of 100 m. The relocation maneuver in Phase 3 takes the docked assembly to 5 km in the in-track direction.

For Phase n, the final time at which the chaser enters Phase $n+1$ is t_{nf} and is defined as the time elapsed since the initial time, t_0. The chaser is free to move between Phase 1a, 1b and 2 freely, but must move to Phase 3 before the eclipse occurs at time, t_e. Thus, t_{2f} is constrained to be less than t_e. Once in Phase 3, the spacecraft cannot move back to Phase 2. The full mission must be completed before t_f, meaning t_{3f} must be less than t_f.

The following section describes an example Concept of Operations (CONOPS), and the subsequent sections outline the dynamics, measurement model and constraints in each of these four phases, as well as offer additional discussion on the goals of each phase.

3.2.1 Example Mission Descriptions

The goal of this work is to be as mission-agnostic as possible. However, to allow precise definition of the problem for the remainder of the document, a specific conceptual mission will be outlined as the basis for the remaining development. The mission is to assemble a new space station that has been launched in separate modules. The space station is too large to fit in a current launch vehicle fairing as a single piece and thus requires on-orbit assembly from a number of separately launched modules. Given errors in orbit insertion after separation from the launch vehicle, these passive modules need to be transported to an assembly location. A tug spacecraft designated as the "chaser" spacecraft, is used to provide this transportation, and the current space station module is designated as the "target" spacecraft. Initially, the tug is too far away to obtain anything other than a passive image from its onboard telescope, providing a bearing-only measurement of the angles between the target and tug's body frame. Once the tug reaches a specified distance, the tug can make use of onboard ranging sensors such as LIDAR or RADAR, providing a direct measurement of range. The chaser must proceed to stage itself to dock with the target module, which requires approach within a specified angle of the target body frame in order to mate with a docking port on the target. Once staged, the chaser moves to dock to the target, closing the relative distance to zero, while staying within the approach angle and reducing the closing velocity as it approaches zero relative distance. When operating the chaser in this close proximity, a higher precision estimate of the relative state is available if the chaser remains in the line of sight of fiducial markings on the target spacecraft's docking port. Also, to maintain safe contact speeds under unforeseen circumstances, the velocity must be limited in this portion of the mission. Adding to the difficulty is the fact that the entire procedure must be completed before the tug and target module go into eclipse and the chaser loses valid data from its vision-based sensors. Once docked, the chaser must transport the space station module to the assembly location, where other modules have already been positioned. The chaser uses similar relative sensing information available with respect to the first module to approach the assembly location, however in this phase, the sensed state is with respect to a partner spacecraft located further away than the assembly location. The combined satellites must reach the assembly location before a predefined mission end time.

Alternatively, the dynamics and CONOPS can extend to several other mission types. In one such mission, a current orbital slot in geosynchronous orbit is occupied by a communications satellite that has unfortunately run out of fuel and could potentially collide with another satellite or exit the orbital slot. The operator would like to prevent this catastrophe as well as keep using the otherwise functional satellite by docking another spacecraft to it to act as a propulsion system. The same process of close approach, docking and joint maneuvering occurs, ending in both spacecraft attached at a new operational location in safety of other nearby spacecraft.

Another mission architecture could involve the removal of orbital debris, where instead of repositioning the debris into another location on orbit, the goal in the last phase of the mission would change to de-orbiting the debris. Yet another mission could be very similar to the debris removal except involving moving a defunct geosynchronous satellite to the graveyard orbit in the last phase of the mission. In addition, individual phases or subsets of phases could be explored such that only the rendezvous and docking portion remains, which brings in many more mission architecture such as cargo delivery to the ISS, spacecraft refueling, and crew transfer, among others.

3.2.2 Angles-only Rendezvous, Phase 1a

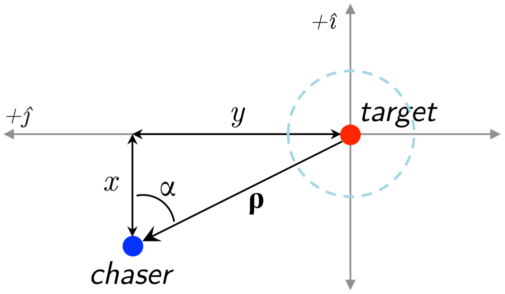


Figure 3-3: Diagram of states during Mission Phase 1a: Angles-only rendezvous

In Phase 1, the chaser spacecraft only has relative angular measurements to the target spacecraft and is attempting to move closer to the target spacecraft in order to gain additional sensing abilities. The dynamics are defined as

$$\dot{\mathbf{x}} = \mathrm{CWH}(\mathbf{x}, \mathbf{u}, n, m_c), \tag{3.23}$$

where the chaser is controlling itself using thrusters, and the target is stationary at the origin of the reference frame. The goal in this phase is to navigate and control well enough to reach the separation distance at which range measurements are available.

The measurement model is defined in 2DOF as

$$\mathbf{y} = h_{1a}(\mathbf{x}) = \arctan\left(\frac{y}{x}\right) + \mathrm{v} = \alpha + \mathrm{v} \ , \tag{3.24}$$

where the angle, α, is shown in Figure 3-3. In 3DOF, the model is

$$\mathbf{y} = h_{1a}(\mathbf{x}) = \begin{bmatrix} \arctan\left(\frac{y}{x}\right) \\ \arcsin\left(\frac{z}{\rho}\right) \end{bmatrix} + \mathbf{v} = \begin{bmatrix} \alpha \\ e \end{bmatrix} + \mathbf{v} \ , \tag{3.25}$$

where e is the elevation angle from the $+\hat{k}$-axis to $\boldsymbol{\rho}$. Please note that adding the measurement noise, $\mathbf{v}$, is optional in the benchmark problem.

In Phase 1a, the CWH equations are unobservable when there are only angle measurements, and thus the estimate of the chaser's state will develop error over time if not dealt with appropriately. In this thesis, the unobservability is left unattended, and there are often significant departures from the estimated position when entering Phase 1b, although never enough to be the sole cause of any instability or mission failure. Others in the literature attempt to account for the unobservability [141, 142].

3.2.3 Range-capable Rendezvous, Phase 1b

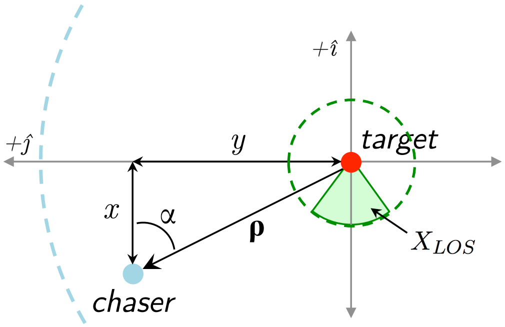


Figure 3-4: Diagram of states during Mission Phase 1b: Range-capable rendezvous

In Phase 1b, the chaser spacecraft is close enough such that a ranging measurement is now available. The chaser attempts to stage itself for docking to the target by closing the distance and entering the LOS of the enhanced sensors available in Phase 2. The dynamics remain as in (3.23). Adding a range measurement, the 2DOF measurement model expands to

$$\mathbf{y} = h_{1b}(\mathbf{x}) = \begin{bmatrix} \arctan\left(\dfrac{y}{x}\right) \\ \\ \sqrt{x^2+y^2} \end{bmatrix} + \mathbf{v} = \begin{bmatrix} \alpha \\ \\ \rho \end{bmatrix} + \mathbf{v} \ , \tag{3.26}$$

and the 3DOF model correspondingly expands to

$$\mathbf{y} = h_{1b}(\mathbf{x}) = \begin{bmatrix} \arctan\left(\dfrac{y}{x}\right) \\ \\ \arcsin\left(\dfrac{z}{\rho}\right) \\ \\ \sqrt{x^2+y^2+z^2} \end{bmatrix} + \mathbf{v} = \begin{bmatrix} \alpha \\ \\ e \\ \\ \rho \end{bmatrix} + \mathbf{v} \ . \tag{3.27}$$

In this phase, a couple of additional constraints arise such that the phase must end in the LOS region at a distance equal to ρ_d. These constraints are represented as

$$\mathbf{x}(t_{1bf}) \in X_{LOS}, \tag{3.28}$$

where t_{1bf} is free, and

$$\rho(t_{1bf}) = \rho_d \ . \tag{3.29}$$

The added range measurement in this phase allows for much more precise estimation and accordingly better control. Nonetheless, the error in the range measurements is large enough that the chaser would not be able to dock effectively and better estimates are needed. Additional cognizance must be placed on the fact that the chaser should not exit the range of the LIDAR or RADAR and transition back to Phase 1a. This could cause rapid switching in the control, and instabilities could result if not handled properly. This will be a concern at every phase transition.

3.2.4 Docking, Phase 2

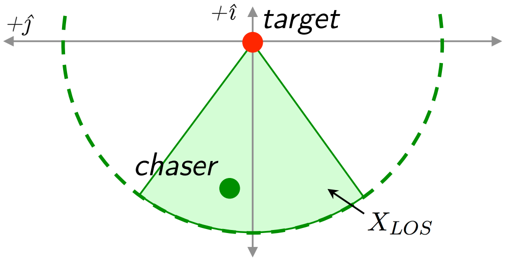


Figure 3-5: Diagram of states during Mission Phase 2: Docking

In Phase 2, the chaser spacecraft is attempting to dock to the target spacecraft (i.e., reduce ρ to zero) while remaining in the LOS region, X_{LOS}, shown in Figure 3-5 and maintaining a slow velocity as to reduce impact forces upon docking. The dynamics remain as in (3.23). The measurement model also remains the same as in Phase 1b, although the measurement noise is less.

$$\mathbf{y} = h_2(\mathbf{x}) = h_{1b}(\mathbf{x}) \; . \tag{3.30}$$

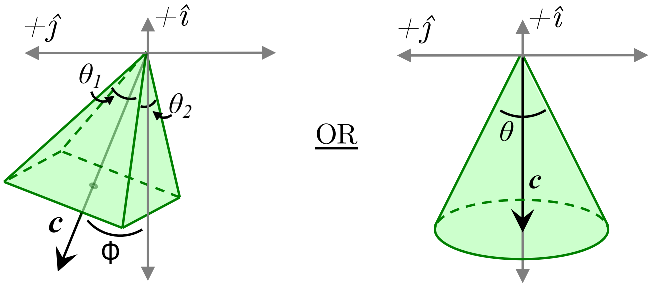


Figure 3-6: Description of the pyramid (left) and cone (right) line-of-sight constraint options

There is the option of setting the LOS constraint (see Figure 3-6) as either a nonlinear cone constraint defined as

$$\frac{\boldsymbol{\rho} \cdot \boldsymbol{c}}{\|\boldsymbol{\rho}\|\|\boldsymbol{c}\|} \geq \cos\left(\frac{\theta}{2}\right) \ , \tag{3.31}$$

or a linear pyramid constraint defined in 2DOF as,

$$\begin{bmatrix} \sin\left(\frac{\theta_1}{2} - \phi\right) & \cos\left(\frac{\theta_1}{2} - \phi\right) \\ \sin\left(\frac{\theta_1}{2} + \phi\right) & -\cos\left(\frac{\theta_1}{2} + \phi\right) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \ , \tag{3.32}$$

and in 3DOF as,

$$\begin{bmatrix} \sin\left(\frac{\theta_1}{2} - \phi\right) & \cos\left(\frac{\theta_1}{2} - \phi\right) & 0 \\ \sin\left(\frac{\theta_1}{2} + \phi\right) & -\cos\left(\frac{\theta_1}{2} + \phi\right) & 0 \\ \sin\frac{\theta_2}{2}\cos\phi & \sin\frac{\theta_2}{2}\sin\phi & \cos\frac{\theta_2}{2} \\ \sin\frac{\theta_2}{2}\cos\phi & \sin\frac{\theta_2}{2}\sin\phi & -\cos\frac{\theta_2}{2} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \ , \tag{3.33}$$

where the appropriate variables are defined graphically in Figure 3-6. Note that the rotation of the constraint in the linear form has only been implemented for rotations about the $\hat{k}$-axis by an angle, ϕ, while the nonlinear form works for all $\boldsymbol{c}$ vectors.

Additionally, the chaser's velocity must be kept under the specified value, with a choice similar to the LOS constraint of both a linear and nonlinear form. The nonlinear version of the velocity constraint is formulated as

$$-\dot{\boldsymbol{\rho}} \cdot \frac{\boldsymbol{\rho}}{\|\boldsymbol{\rho}\|} \leq \bar{V}, \tag{3.34}$$

where we are constraining the velocity in direction of the target satellite and effectively the impact velocity should something go wrong. The linear version of the velocity constraint involves the infinity norm constraints,

$$\begin{aligned} \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T &\leq \begin{bmatrix} \bar{V} & \bar{V} & \bar{V} \end{bmatrix}^T, \text{ and} \\ -\begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T &\leq \begin{bmatrix} \bar{V} & \bar{V} & \bar{V} \end{bmatrix}^T. \end{aligned} \tag{3.35}$$

The 3DOF versions are shown in (3.34) and (3.35), while the 2DOF versions only require removing the $\dot{z}$ component.

Finally, the chaser must dock to the target before the eclipse time, such that $t_{3f} \leq t_e$, and

$$\mathbf{x}(t_{2f}) = \mathbf{x}_{docked}. \tag{3.36}$$

Note that, if including measurement noise in the problem, the docking state constraint can be expanded into a couple bounding inequality constraints as opposed to an equality constraint. This bounding box for docking should represent the acceptable error in final docking state due to the docking port physical properties. For the ISS, this would be on the order of a meter, while for more precise applications could be on the order of centimeters.

3.2.5 Joint Maneuver, Phase 3

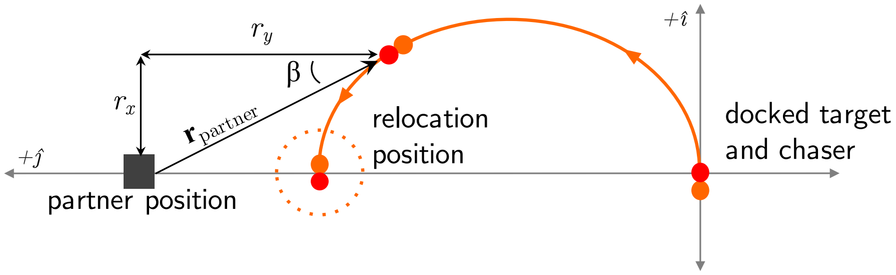


Figure 3-7: Diagram of states during Mission Phase 3: Joint Maneuver

In Phase 3, the spacecraft have successfully docked and now must maneuver together to the relocation spot using relative measurements from a partner located 5 km past the relocation spot. The mass of the docked pair has increased, and therefore the dynamics have changed to

$$\dot{\mathbf{x}} = \text{CWH}(\mathbf{x}, \mathbf{u}, n, m_c + m_t) \ . \tag{3.37}$$

As the angular and range measurements come from a different location, the measurement model has changed accordingly in 2DOF to

$$\mathbf{y} = h_3(\mathbf{x}) = \begin{bmatrix} \arctan\left(\dfrac{r_x}{r_y}\right) \\ \sqrt{r_x^2 + r_y^2} \end{bmatrix} + \mathbf{v} = \begin{bmatrix} \beta \\ \|\mathbf{r}_{partner}\| \end{bmatrix} + \mathbf{v} \ , \tag{3.38}$$

and in 3DOF to

$$\mathbf{y} = h_3(\mathbf{x}) = \begin{bmatrix} \arctan\left(\dfrac{r_x}{r_y}\right) \\ \arcsin\left(\dfrac{r_z}{\|\mathbf{r}_{partner}\|}\right) \\ \sqrt{r_x^2 + r_y^2 + r_z^2} \end{bmatrix} + \mathbf{v} = \begin{bmatrix} \beta \\ e \\ \|\mathbf{r}_{partner}\| \end{bmatrix} + \mathbf{v} \ , \tag{3.39}$$

where $\mathbf{r}_{partner} = [r_x, r_y, r_z]^T$, the relative position from the partner to the docked pair as shown in Figure 3-7, $\|\mathbf{r}_{partner}\| = \sqrt{r_x^2 + r_y^2 + r_z^2}$, β is the angle from the $+\hat{j}$-axis to $\mathbf{r}_{partner}$ in the chaser frame as shown in Figure 3-7, and e is now the angle from the $+\hat{k}$-axis to $\mathbf{r}_{partner}$ in the chaser frame.

There is one additional constraint during this phase that the final location must be the relocation spot, i.e.,

$$\mathbf{x}(t_{3f}) = \mathbf{x}_{relocation}, \tag{3.40}$$

where $t_{3f} \leq t_f$. Again, if including measurement noise in the problem, the final state constraint can be expanded into a couple bounding inequality constraints as opposed to an equality constraint.

This phase will take most of the mission time as it requires the largest maneuver with the largest mass. Only the thrusters from the chaser are in use, so the effective acceleration is much less. The dynamics can be thought of as if there were an imaginary target spacecraft stationary at the original location, and now the combined system is moving in the same, but now imaginary, CWH frame. This is interesting, because no longer are the errors in the CWH equations shrinking as the chaser closes distance to the target. Now the error between the CWH equations and reality grows as the mission moves forward.

3.2.6 Control Options

For all phases, the constraint on the maximum control input at any single point in time is

$$\|\mathbf{u}\|_\infty \leq \bar{u}, \tag{3.41}$$

where $\bar{u}$ is the upper limit on the thrust that can be produced in each of the axial directions. The problem offers the choice of three methods to implement control defined as follows.

On-Off Thrusters

The on-off thruster control scheme aims to mimic the behavior of spacecraft systems in which only the thruster pulse duration is controllable and the magnitude of the thrust is fixed, such as a cold-gas configuration. For the 3DOF problem, the spacecraft has a total of 6 thrusters to control in the positive and negative direction in each of the axes. Likewise, the 2DOF problem assumes the spacecraft to have a total of 4 thrusters.

There is, thus, the choice of operation in three modes for each axis of control. In mode 1, the positive-oriented thruster is turned on fully; in mode 2, the negative-oriented thruster is turned on fully; and in mode 3 both thrusters are turned off. Thus, the mode is individually chosen for each axis (F_x, F_y, F_z), and the control vector must be in the form $\mathbf{u} = [\{-\bar{u}, 0, \bar{u}\}, \{-\bar{u}, 0, \bar{u}\}, \{-\bar{u}, 0, \bar{u}\}]^T$ for the 3DOF case and $\mathbf{u} = [\{-\bar{u}, 0, \bar{u}\}, \{-\bar{u}, 0, \bar{u}\}]^T$ for the 2DOF case, where $\bar{u}$ is 10 N.

The fuel consumption for this control mode is tracked as

$$f_{tot} = \int_{t_0}^{t_f} \|\mathbf{u}\|_1 \; dt, \tag{3.42}$$

where f_{tot} is the total fuel consumption in the mission.

Continuously Variable Thrust

The continuously variable thrust control scheme is useful for the case in which the thrust magnitude from each thruster can be controlled directly. For the 3DOF problem, the spacecraft has a total of 6 thrusters to control in the positive and negative direction in each of the axes. Likewise, the 2DOF problem assumes the spacecraft to have a total of 4 thrusters. Each of these thrusters is capable of achieving a $\bar{u}$ of 10 N.

This formulation results in force values being individually chosen for each axis and the control vector in the form $\mathbf{u} = [F_x, F_y, F_z]^T$ where $\mathbf{u} \in \mathbb{R}^3$. Optionally, this method can be implemented with discrete intervals or continuous force selection. The fuel consumption for this control mode is again tracked as in (3.42).

Chemical & Electric Propulsion

The multiple-propulsion-system control scheme is similar to the continuously variable thrust example except with choice of a high thrust, less efficient chemical engine or a low thrust, very efficient electric engine. In this scenario, the spacecraft has two types of propulsion on board and has the choice of which engine to use at any point in time. This results in a choice between two modes of operation. This control option is meant to make the problem more interesting in terms of hybrid control mode switching and with it comes significant additional work in formal proofs of robustness and stability.

Mode 1 uses the high thrust, less efficient chemical propulsion where $\mathbf{u} = [F_x, F_y, F_z]^T$ where $\mathbf{u} \in \mathbb{R}^3$ and $\bar{u} = 10$ N. Similar to the previous control schemes, the fuel consumption for this control mode is tracked as in (3.42).

Mode 2 uses low thrust, very efficient electric propulsion, where $\mathbf{u} = [F_x, F_y, F_z]^T$, $\mathbf{u} \in \mathbb{R}^3$ and $\bar{u} = 0.1$ N. Contrary to the previous control schemes, the fuel consumption for this control mode is tracked as

$$f_{tot} = \int_{t_0}^{t_f} \frac{\|\mathbf{u}\|_1}{\eta} \, dt \tag{3.43}$$

where $\eta = 100$ is the efficiency gain from using this engine.

Note in (3.42) and (3.43), the units of fuel used are Newton-seconds (Ns). Dividing by the spacecraft mass at the current time will yield fuel in meters per second (m/s), a ΔV form.

3.2.7 Options Definitions

The options that can be specified for an instantiation of the benchmark problem are shown in Table 3.1. Note the alphanumeric code beside each option for easier communication of which options are chosen. The default options are 2BNY, which imply 2DOF (=2), continuous control (=B), nonlinear docking constraints (=N), and inclusion of measurement noise (=Y). The more difficult 3CNY option can be considered a reach goal. These options exist to tailor the problem as desired, although the closer to the default or reach goal the better. We realize certain techniques have different strengths and are trying to let the problem be equally difficult for all techniques while still upholding traceability to the same spacecraft rendezvous problem.

Table 3.1: Options to choose between when solving the benchmark problem

Problem Component	Options
State	3DOF (3) or 2DOF (2)
Control	On/Off (A) or Cont. (B) or Chem & Elec (C)
Constraints	Linear (L) or Nonlinear (N)
Measurement Noise	Yes (Y) or No (N)

The option choices of 3BNY and 3BLY are explored in this thesis, where the 2DOF and no-measurement-noise options are just subsets of these chosen options. The continuous control case is chosen, because this mission aims to mimic the Orbital ATK MEV, which is being designed and built to rendezvous, dock and maneuver a target satellite that has run out of fuel.[13] The eventual goal for the MEV will be to use high-power solar electric propulsion, which would enable and require continuous control methods. Additionally, in this thesis, although the controllers are designed to this problem's specification, they are also tested under greater uncertainty involving varying dynamics parameters, process noise through thruster force uncertainty, and full-nonlinear orbital dynamics.

3.2.8 Definition of Constants

Throughout the previous sections, numerous variables were used to represent constants in this problem. However, to fully define the problem, we define the values these constants take, such that the results from this thesis can be compared on a level playing field using the same parameter values. Table 3.2 sets these values.

[13] https://www.orbitalatk.com/space-systems/human-space-advanced-systems/mission-extension-services/default.aspx

Table 3.2: Values and definition of constants for the benchmark rendezvous, docking, and joint maneuvering problem

Variable	Value	Definition
t_e	4 hrs.	time of eclipse after t_0
t_f	8 hrs.	total mission duration
ρ_r	1 km	range measurement initial radius
ρ_d	100 m	docking phase initial radius
m_t	2000 kg	mass of target (defunct satellite)
m_c	500 kg	mass of chaser (tug spacecraft)
μ	3.986×10^{14} m^3/s^2	Earth's gravitational constant
r_0	42,164 km	semi-major axis of GEO
$\bar{u}$	10 N	maximum thruster force magnitude
$\bar{V}$	5 cm/s	maximum closing velocity if docking
$\theta = \theta_1 = \theta_2$	60°	Line Of Sight (LOS) angle
$\boldsymbol{c}$	$[-1, 0, 0]^T$	LOS cone direction
$\mathbf{x}_{docked}$	[(0, 0, 0) km, (0, 0, 0) km/s]^T	state of chaser required to dock
$\mathbf{x}_{target}$	[(0, 0, 0) km, (0, 0, 0) km/s]^T	state of target before docking
$\mathbf{x}_{relocation}$	[(0, 5, 0) km, (0, 0, 0) km/s]^T	state of relocation spot in Phase 3
$\mathbf{x}_{partner}$	[(0, 10, 0) km, (0, 0, 0) km/s]^T	state of partner spacecraft in Phase 3

In this problem, there is an option to include white Gaussian measurement noise to simulate sensor performance. Noise levels are time invariant, although because the sensors change between phases, the noise levels do as well. Let $R_{\alpha\alpha} = (0.001\text{rad})^2$, $R_{ee} = (0.001\text{rad})^2$, $R_{\beta\beta} = (0.001\text{rad})^2$, $R_{\rho\rho,a} = (10\text{m})^2$, $R_{\rho\rho,b} = (1\text{cm})^2$, and $R_{rr} = (10\text{m})^2$. For the n-th phase, $R_n = \text{E}\left[\mathbf{v}\mathbf{v}^T\right]$, where for the 2DOF case, $R_{1a} = R_{\alpha\alpha}$, $R_{1b} = \text{diag}\{R_{\alpha\alpha}, R_{\rho\rho,a}\}$, $R_2 = \text{diag}\{R_{\alpha\alpha}, R_{\rho\rho,b}\}$, and $R_3 = \text{diag}\{R_{\beta\beta}, R_{rr}\}$. For the 3DOF case, $R_{1a} = \text{diag}\{R_{\alpha\alpha}, R_{ee}\}$, $R_{1b} = \text{diag}\{R_{\alpha\alpha}, R_{ee}, R_{\rho\rho,a}\}$, $R_2 = \text{diag}\{R_{\alpha\alpha}, R_{ee}, R_{\rho\rho,b}\}$, and $R_3 = \text{diag}\{R_{\beta\beta}, R_{ee}, R_{rr}\}$.

3.3 Categorization of Uncertainty

In this section, we discuss the sources of uncertainty in the thesis problem. These sources are above and beyond what is proposed for the benchmark problem discussed in Section 3.1.1

and Section 3.2. Specifically, for this thesis, the uncertainty in this problem is being analyzed, and a few methods are proposed to handle this uncertainty. As such, Section 3.3.1 attempts to define and categorize the types of uncertainty present, and Section 3.3.2 suggests appropriate strategies to account for the different types.

3.3.1 Definitions of Uncertainty Categories

A major focus for this thesis is the ability to cope with uncertainty in the system as can be seen in the motivation and literature review in Sections 1.4 and 2.3. As stated before, there are two major divisions here: uncertainties and uncertain events.

Uncertainties

Uncertainties can be further divided into *epistemic* uncertainty and *aleatoric* uncertainty. Both of these types of uncertainty can be mitigated through robust controller design, however need to be handled differently. Epistemic uncertainties deal with parameters that can theoretically be known, but are not known exactly in practice. These are uncertainties like an error in mass or inertia properties, misalignment of sensors or actuators, unknown thruster force magnitude, or errors in orbital elements; they all cause systematic biases in the dynamics that could be removed if they were estimated or measured directly with high precision. For this thesis, the terms “uncertain parameters” and “epistemic uncertainties” will be used interchangeably.

Aleatoric uncertainties are those that cannot be known as the processes underlying them are beyond current application of scientific knowledge. These are uncertainties like sensor noise, actuator noise, and resulting noise in state estimates. Aleatoric uncertainties are very familiar to control system engineers and are for the most part reliably represented as Gaussian noise. They are arguably the main reason automatic feedback control exists. For the application in this thesis, we are also assuming higher-order nonlinear orbital dynamics and orbital perturbations to be aleatoric uncertainties with respect to our control systems. Although we do have the knowledge to model these uncertainties appropriately and do so in the simulation, the controllers are simplified to only use the linear form of the dynamics, as is very common in practice. With this exception in mind, it is the commonly the goal of certain adaptive controllers and system identification techniques to attempt to reduce the epistemic uncertainties to aleatoric uncertainties.

Uncertain Events

Uncertain events on the other hand involve a somewhat more nuanced discussion. Remember, rather than dealing with specific parameter uncertainty and noise, uncertain events describe discrete occurrences during a mission that affect the mission goals, constraints or performance in some manner, either positively or negatively. These can be faults or anomalies, but also events in which we gain better information about the mission. For example, an uncertain event could be estimator divergence or a critical subsystem fault which are definitively bad, but could also be getting a better estimate of the target spacecraft's attitude or position when sensors come in range. So, in a sense, an uncertain event must coincide with learning something new about the mission, yet when and where we learn the new fact and what exactly that new fact is can be uncertain.

There a few questions we can ask when defining an uncertain event. Do we know this event will happen with certainty at some point in the mission? Do we know when or where it will happen? Or at least, do we have a model of the uncertainty distribution that will define when or where it will happen? If so, does this model allow us to perform tradeoffs between risk and performance? And finally, what are the effects of the event? Are we able to replan in real-time to account for the event, or will the event result in a failed mission?

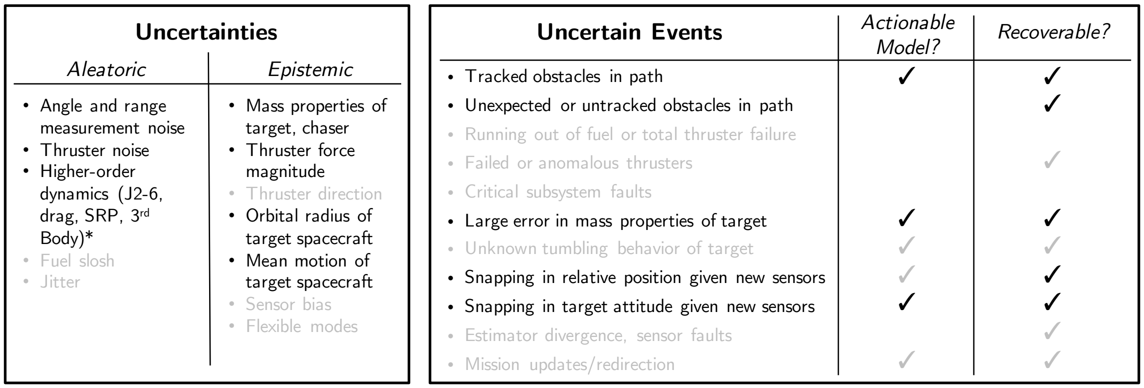
These questions bring up a couple key characteristics of an uncertain event. First, we can start with the consequences of the uncertain event. An uncertain event is deemed *recoverable*, if upon the event, the spacecraft can replan the mission with respect to the trajectory or selected controller. It is deemed *critical*, if upon the event, the spacecraft does not have the capability to finish the mission and must either passively or otherwise perform an abort maneuver. For example, a critical event would be something akin to complete thruster failure or running out of fuel, while a recoverable event would be something such as an unexpected deviance in mass of the target after docking or an avoidable obstacle crossing the planned trajectory.

Secondly, the uncertain event can be categorized by our ability to model the chances of it happening with some level of precision such that this model can be exploited to probabilistically plan the mission beforehand. In this case, we say that the event has an *actionable* model. Typically, this model involves more than just when and where this event will happen, but also information about what exactly will happen (uncertain or not).

For example, if we have a large initial uncertainty in a potential obstacle's position and will later find out if the obstacle intersects our trajectory or not, this initial uncertainty distribution would be an actionable model, and finding out the higher fidelity obstacle position later in the mission would be the uncertain event. In this case, we have uncertain information about when and where the event will happen based on our sensing capabilities and the initial uncertainty, but also have uncertain information about the possible locations the obstacle could be once we detect it based on the initial uncertainty. As another example, say we have a large epistemic uncertainty in the knowledge of the target spacecraft mass. For this example, we know exactly that the event will happen once we dock to the target satellite, but only have uncertain information about what will happen (i.e., what mass the target spacecraft will have). Both of these examples would have actionable models.

On the other hand, for things like subsystem failures or thruster failures, although we can approximate the probability of the event over time by Mean Time Between Failure (MTBF) models, the model is not significantly rich with information that allows us to proactively take advantage of it, and therefore is not actionable. The failures could happen with essentially equal probability over the entire trajectory, and thus we cannot exploit this information to tradeoff performance and risk, as we always have to be robust to these events. In the next section, it will become clearer why we distinguish between these types of models.

See Figure 3-8 for a list and categorization of uncertainties and uncertain events that could be explored for this problem and the specific uncertainties which will be investigated for this thesis. The next section will discuss strategies for handling these different categories of uncertainties and uncertain events.



Uncertainties	
Aleatoric	*Epistemic*
• Angle and range measurement noise • Thruster noise • Higher-order dynamics (J2-6, drag, SRP, 3rd Body)* • Fuel slosh • Jitter	• Mass properties of target, chaser • Thruster force magnitude • Thruster direction • Orbital radius of target spacecraft • Mean motion of target spacecraft • Sensor bias • Flexible modes

*Technically epistemic, but treated as noise in controllers

Uncertain Events	*Actionable Model?*	*Recoverable?*
• Tracked obstacles in path	✓	✓
• Unexpected or untracked obstacles in path		✓
• Running out of fuel or total thruster failure		
• Failed or anomalous thrusters		✓
• Critical subsystem faults		
• Large error in mass properties of target	✓	✓
• Unknown tumbling behavior of target	✓	✓
• Snapping in relative position given new sensors	✓	✓
• Snapping in target attitude given new sensors	✓	✓
• Estimator divergence, sensor faults		✓
• Mission updates/redirection	✓	✓

— Discussed but not handled directly in this thesis
— Results presented in the thesis

Figure 3-8: Categorization of uncertainties and examples in the scope of this problem (some examined in this thesis, others not)

3.3.2 Overview of Methods Used to Handle Uncertainty

First, there will be a brief discussion on techniques to handle aleatoric and epistemic uncertainties. These uncertainties are very common in the field of control and estimation, and thus there are many techniques in the literature both in and out of the aerospace field that should suit our purposes.

Aleatoric uncertainties, like sensor noise and actuator noise, can generally be handled well with a feedback control framework and estimator pairing. If an optimal control plan were to be executed burn by burn under aleatoric uncertainty without any feedback information on the state, it would almost certainly drift from the desired trajectory. Common choices for spacecraft control are fixed-gain PD or PID feedback controllers paired with a Kalman filter. Because the measurement models defined for our problem are nonlinear, instead we must use an Extended Kalman Filter (EKF) that linearizes the measurements at every time step. If the problem was sufficiently nonlinear, or the noise was not Gaussian, other estimators such as Unscented Kalman Filters (UKF) or particle filters could be implemented. Kalman filters are able to take noisy measurements from different sensors and combine them into a mathematically optimal estimate with consideration for the noise levels present. Additionally, if the controller requires guarantees of convergence and stability under higher levels of sensor and process noise, certain adaptive and robust control algorithms could be implemented to reduce the sensitivity of the controller to the noisy input. Many of these techniques will be explored in Chapter 5. In any case, aleatoric uncertainties should be handled with feedback control. As shown in Figure 3-8, aleatoric uncertainties explored in this thesis are angle and range measurement noise, noise in thruster firings and higher-order orbital dynamics. Additional uncertainties present on orbit that are not considered in this thesis could be jitter from reaction wheels or other vibrating equipment or fuel slosh in a propellant tank. Although not investigated here, these sources of aleatoric process noise should be able to be handled well with adaptive or robust feedback control design.

Epistemic uncertainties, or parameter uncertainties, manifest themselves as systematic biases in the spacecraft dynamics. Theoretically, they should be able to be reduced to the level of the aleatoric noise present in the system. But the question then is exactly how we eliminate these uncertainties. There are a few options. One option is to design a compensator that is robust to the uncertainty in the plant. In this manner, the goal is not to try to improve performance over time, just to increase the range of plants for which the compensator will maintain stability. Another option is to perform adaptive control that will attempt to learn information about the errors in the plant model over time and enact control changes based on this memory of past performance. There are a couple variants in adaptive control, both *directly* adapting the gains used in the controller, where we will not get an estimate of the true parameter values, or *indirectly* adapting other parameters, such as an estimate of the true plant, and recomputing gains based on those values. Finally, another technique is to perform system identification maneuvers by commanding rich control inputs to stress the system and enable precise determination of the uncertain parameters. After performing a system identification maneuver, the controller can be designed to the appropriately estimated, true plant dynamics. Examples from each of optimal, feedback, robust and adaptive control architectures are implemented and compared under different levels of epistemic uncertainty in Chapter 5. Specifically, the epistemic uncertainties this thesis covers are the mass properties of the target and chaser, the thruster force magnitude, and the orbital radius and consequentially mean motion of the target spacecraft. Other sources not covered here, but that could also be accounted for with similar techniques are sensor bias, thruster directions, and flexible spacecraft modes.

Uncertain events can be divided into four categories for the context of which strategy to employ. There are uncertain events with actionable models and those without. There are uncertain events that are recoverable and those that are critical. So, the four combinations of those two binary choices each involve a unique strategy. Table 3.3 shows the breakdown and the brief description of the appropriate strategy to use.

Table 3.3: Strategies for a given uncertain event category

	Recoverable Event	Critical Event
Uninformative Model or Absence of Model	Plan nominally or for sensitivity, then react	Plan for robustness, guaranteeing safety
Actionable Model	Plan for performance, then react	Plan for robustness, guaranteeing safety, while optimizing performance

As follows, Table 3.3 is deciphered, starting with uncertain events that do not have actionable models. In the case that there is an uninformative model or no model at all, it is difficult to adjust the mission plan to optimize performance, because we are unable to perform any obvious tradeoffs. Essentially, we have either the choice of adding a sensitivity term to our objective function or adding robustness and safety constraints to the optimization problem. If adding a sensitivity metric to the mission optimization, we would prefer that the possibility of this event happening at any time in our mission will minimally affect the performance. In the case that the event is recoverable, we can add this sensitivity factor if desired or just plan nominally. The important fact is that we are able to replan (i.e., *react*) in real-time after a recoverable event happens and still compete the mission. Reacting to an uncertain event can mean many things. It can mean replanning a trajectory due to new constraints in the problem, such as obstacle or a shifted LOS zone. It can mean enacting a more robust or adaptive controller to account for higher sensor or actuator faults. It can mean redesigning the gains of a controller to account for new information. Or it can mean changing the choice of a controller later in the mission due to a change in knowledge of future uncertainties. Several cases of events that require replanning or reacting are covered in the simulation as discussed in Section 3.4 and a few detailed results are presented in

Chapters 4 and 5, specifically an unexpected obstacle, snapping in relative state due to new sensors, and uncertain target spacecraft mass. In the case that it is a critical event, we are forced to add robustness constraints to the problem so that the event cannot cause worse consequences, such as crashing into the target spacecraft. These robustness constraints have been well handled in the literature, such as in [146], where trajectories are optimized with the constraint that losing all thrusters or propellant at any point during the mission will not result in collision with the target spacecraft. This category will not be investigated in the thesis, due to its coverage elsewhere.

The other half of the categories involve cases in which there is an actionable model of the uncertain event. If the uncertain event is mission critical, the planning is unfortunately very similar to the case in which we had no model whatsoever, because we still are required to guarantee safety. The only difference is that now we have a little more information about when or where this event could possibly occur, such that we can plan to optimize performance elsewhere and only enforce these constraints where required. This case will also not be discussed in the thesis due to the close similarities to the other non-recoverable category.

This leaves the most interesting category remaining, the case where there is an actionable model and a recoverable event. There is plenty of room here to develop performance optimizing algorithms that count on the fact the mission will be replanned after the event. The goal is to find, for example, a baseline optimal trajectory that will have the probabilistically best performance given the uncertainty model associated with the uncertain event. This is the topic of Section 4.4 and one of the major contributions of this thesis. A motivational example is discussed below and will be evaluated extensively in Section 4.4 along with the updated target attitude upon reaching the range of new sensors.

As a final point of illustration, Figure 3-9 depicts an uncertain event that has an actionable model. This involves the ground tracking of an obstacle (e.g., another spacecraft or piece of space debris) that has been detected and has some probability of intersecting the nominal trajectory based on the error covariance ellipsoid associated with the estimate of the ground tracking. The uncertainty here is Gaussian and represented by the red ellipses in the figure. There is thus a possibility that the obstacle will collide with the spacecraft trajectory, and this possibility should be accounted for in the mission trajectory planning. A very conservative method would be to take a path that avoids the full three-sigma uncertainty ellipsoid, however this may have a large fuel expense when compared to the optimal nominal trajectory. The uncertain event itself is defined in Figure 3-9b, where there is new knowledge at a later time based upon onboard sensors that the obstacle will actually be in a collision path. The correction required at this point to avoid the collision may be much larger for the nominal trajectory than the conservative one. By incorporating the chance that this correction will need to be implemented or other corrections for other possible obstacle locations, one could find the probabilistically optimal trajectory that would decrease the fuel consumption on average. This is an easy to illustrate version, although one could imagine this translating to the other uncertain events with actionable models described above, such as a snap in the target state given new sensor information or determining the uncertain mass of the target satellite.

These uncertainties and uncertain events are dealt with in different ways. Controllers and estimators used to handle uncertainties will be covered in Chapter 5, along with analysis and comparison of the implementation of several techniques, and scheduling of these controllers over the different mission phases. For the most part, Chapter 4 will cover the handling of uncertain events with respect to trajectory planning and show results for several instances. Chapter 5 will also very briefly cover the uncertain event of only determining

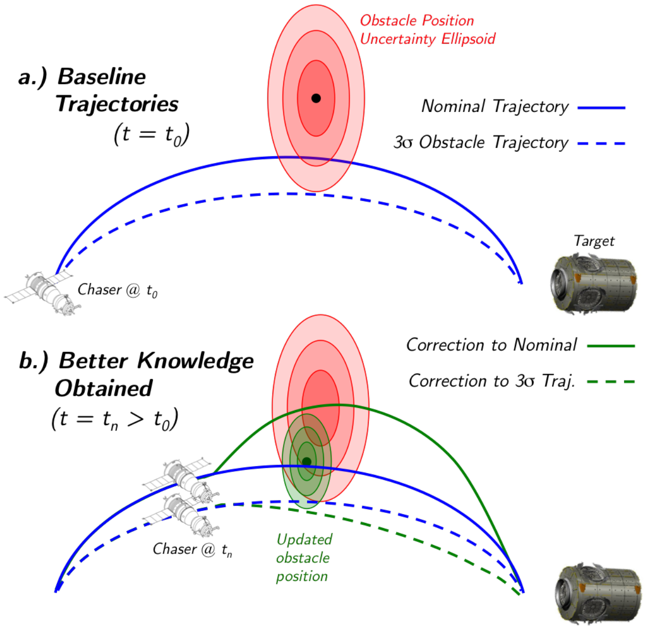


Figure 3-9: Depiction of uncertain event in the form of obstacle avoidance and the deviation required from different baseline trajectories

the target spacecraft's mass after docking. These are interesting cases as they involve the combination of uncertainties and uncertain events. The uncertain event can be thought of as the discovery of a better estimate of the epistemic uncertainty.

3.4 Simulation Environment

In developing the simulation for the relative motion dynamics of a chaser and target spacecraft, it is desired to model the orbital dynamics of the system at a higher fidelity than the plant model in the controllers. Simulating algorithms in more realistic environments is critical to assessing their traceability to actual missions. To this end, the simulation propagates the states of both spacecraft individually in the ECI J2000 frame with two-body, inverse-square gravity; J2 through J6 zonal harmonics to approximate the non-spherical Earth; 3rd body gravitational effects from the Sun and Moon; solar radiation pressure and atmospheric drag. These particular disturbances were chosen and modeled as discussed in Section 3.1.2. The simulation is implemented primarily using MATLAB, however, the SPICE Toolkit developed at NASA's Navigation and Ancillary Information Facility is used for ephemeris querying of the Earth-Sun-Moon system [198].

Typically, a spacecraft uses the data from onboard sensors to estimate its own state. This estimate of the state has uncertainty associated with it, so the spacecraft does not have exact knowledge of where it currently is. The goal is thus to simulate a real-world environment, where an EKF is fed a state that has inherent error associated with it. The simulation injects a random error drawn from a Gaussian distribution to the true measurements before the measurements are input to the EKF and the EKF then sends its estimated relative state to the controller. See Figure 3-10 for the exact location of this injected noise. Additionally, there is also error associated with the executed versus commanded thrust. Thrusters are not perfect and cannot deliver exactly the desired force commanded to them all times. To simulate this behavior, a random error drawn from a Gaussian distribution is injected to the simulated thrust each time the control algorithm commands a thrust. The goal of adding these stochastic aspects to the simulation is to show that the method is robust to typical levels of noise seen on orbit. Thus, the nonlinear orbital dynamics, orbital perturbations, and noises discussed here are the aleatoric uncertainties that are present in the simulation.

Another area of uncertainty present in the simulation is the epistemic uncertainty, or parameter uncertainty. The simulation is capable of separating guidance and control knowledge from the true simulation propagation. In this manner, the simulation can address epistemic uncertainty due to the guidance and control methods not knowing the specific values of uncertain parameters. The true values of these parameters and the estimated values for the guidance and control can each be varied through an input vector such that repetitive trials can be run to explore the effects of a mismatch between assumed and true parameter values.

The simulation specifically addresses the knowledge of the exact orbital parameters assumed to be known by the trajectory optimization and controller plant model in (3.1). In fact, the CWH equations assume that the eccentricity of the target satellite's orbit is zero (i.e., that the orbit is circular). The mean motion, n, depends directly upon the semi-major axis, or radius, of the orbit and is also hopefully known very well. Unfortunately, once on orbit, it is difficult to guarantee that the desired orbit is achieved. There can be significant errors in the values assumed in the plant model. Thus, it is necessary to test the algorithm's robustness to this uncertainty. Upon initialization of the simulation, the true orbital parameters can be set separately than the orbital parameters used in the plant model to facilitate this scenario. Additionally, in a similar vein, other parameters can be varied such as biases in knowledge of spacecraft mass properties, thruster force magnitude, noise magnitudes, and target spacecraft state.

Finally, depending on the mission scenario, uncertain events can be programmed into the simulation to occur anytime during the mission. The full realm of uncertain events modeled in the simulation is quite large and there is a capability to turn on and off the possibility that any one of them will occur during a mission, as well as tune the probabilities and effects as desired. Table 3.4 lists the types and variations of uncertain events present in the simulation. In general, the guidance, navigation and control software is given notification that a specific uncertain event has happened and is given the opportunity to respond to the uncertain event. In some situations, this will involve replanning a trajectory, in others it will mean selecting a new controller or redesigning a controller based on the new parameter identified. Other uncertain events are put in the simulation as a means of tracking the algorithms robustness to strange happenings on orbit, such as measurement drops, estimator biases, jumps in the estimate, and single or multiple thruster failures for only one pulse. The intent is that the simulation would be run several times, potentially in a Monte Carlo

Table 3.4: Uncertain events modeled in the simulation

Uncertain Event Subcategory	Option Description	Tunable Variables
Thruster Failures	Single random failure for one pulse	probability, percent thrust reduction
Thruster Failures	Multiple random failure for one pulse	probability, number, percent reduction
Thruster Failures	Single random failure forever	probability, percent reduction
Thruster Failures	Multiple random failure forever	probability, number, percent reduction
Thruster Failures	Custom/preset mode	specific time, thrusters, percent reduction
Propellant Depletion	Random at any time	probability
Propellant Depletion	Custom/preset mode	specific time, probability, etc.
Measurement Bias	Random bias added in new phase	covariance of random bias, phase, duration
Measurement Bias	Custom/preset mode	specific bias, phase, duration
Measurement Dropped	Single measurement drop	probability
Measurement Dropped	Custom/preset mode	specific time range
Estimator Bias	Random bias added in new phase	covariance of random bias, phase, duration
Estimator Bias	Custom/preset mode	specific bias, phase, duration
Estimator Drop Out	Single estimate dropped	probability
Estimator Drop Out	Custom/preset mode	specific time range
Obstacles	Random obstacle	covariance of random obstacle state, time
Obstacles	Single custom/preset obstacle	obstacle initial state, phase
Obstacles	Multiple custom/preset obstacles	obstacle initial states, phases
Updated Target State	Random jump in state	probability, covariance
Updated Target State	Random jump at specific distance	covariance
Updated Target State	Custom/preset jump	specific jump, time, distance
Updated Target LOS	Random jump in target attitude	probability, covariance
Updated Target LOS	Random jump at specific distance	covariance
Updated Target LOS	Custom/preset jump in target attitude	specific jump, time, distance
Subsystem Fault	Random failure	probability
Subsystem Fault	Custom/preset failure	time, distance
Updated Target Mass	Random mass bias added at docking	covariance
Updated Target Mass	Specific mass bias added at docking	specific mass
Change in Mission	Random new goal on V-bar, anytime	probability, covariance, mean
Change in Mission	Random new goal anywhere, anytime	probability, covariance, mean
Change in Mission	Custom/preset goal, anytime	probability, specific goal state
Change in Mission	Custom/preset goal at specific time	specific time, specific goal state

sense, such that the events would not happen deterministically every run. The chosen probabilities would hopefully then be able to accurately represent those in a real mission. Of course, for debugging purposes, there are also custom preset modes for each uncertain event when the user can define when and what uncertain events occur in a particular simulation run.

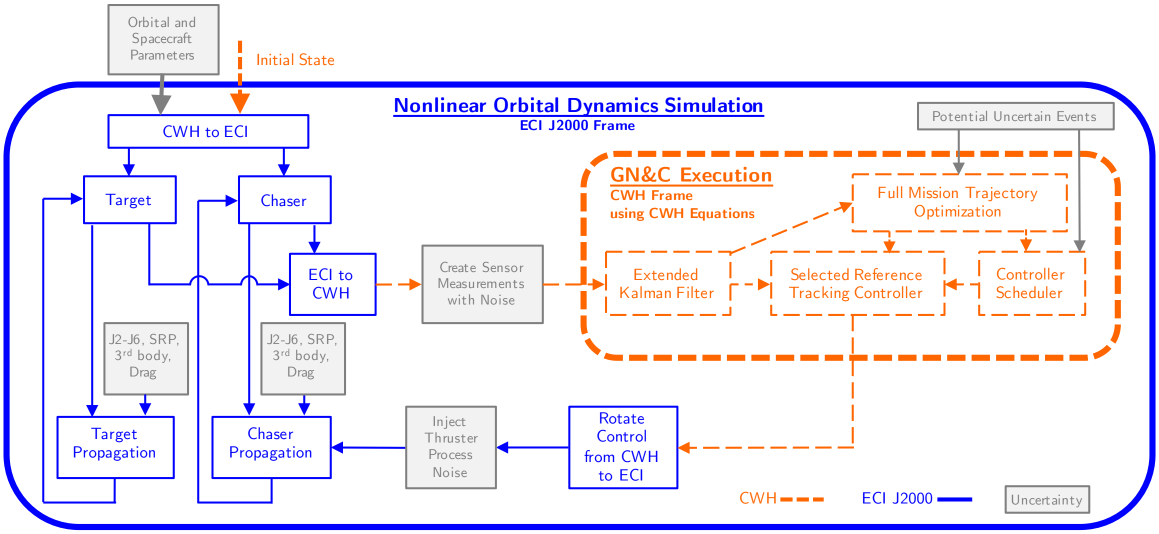

Orbital and Spacecraft Parameters
Initial State
Nonlinear Orbital Dynamics Simulation
ECI J2000 Frame
Potential Uncertain Events
CWH to ECI
Target
Chaser
GN&C Execution
CWH Frame
using CWH Equations
Full Mission Trajectory Optimization
ECI to CWH
Create Sensor Measurements with Noise
Extended Kalman Filter
Selected Reference Tracking Controller
Controller Scheduler
J2-J6, SRP, 3rd body, Drag
J2-J6, SRP, 3rd body, Drag
Target Propagation
Chaser Propagation
Inject Thruster Process Noise
Rotate Control from CWH to ECI
CWH
ECI J2000
Uncertainty

Figure 3-10: Block diagram depicting the high-fidelity simulation developed to test the autonomous guidance and control software

The diagram in Figure 3-10 shows how the simulation is structured and shows where the conversions from frame to frame happen, as well as the flow of the chaser and target state propagation. Starting from the beginning, the target orbit is defined in orbital elements, and the chaser position is reported in the relative CWH frame. These initial states are immediately converted to the ECI frame for the propagation. The CWH frame relative state is computed by differencing the target and chaser's ECI states and rotating appropriately. True measurements are computed based on the relative state information and the measurement model for the current phase. Noise is then added to the measurements to simulate uncertainty from a spacecraft's sensors. These noisy measurements are sent to the EKF estimator, which calculates the relative state estimate and then relays this estimated state to the trajectory optimization and controller. The reference-tracking controller currently selected by the controller scheduler then executes the appropriate control. The autonomous trajectory planning and controller scheduling is performed at the first instance in the simulation and then each time an uncertain event occurs as appropriate. The appropriately selected reference-tracking controller is run on the chaser spacecraft only, which results in a commanded control in the CWH frame. This control command is rotated to the ECI frame and then immediately added to the ECI state before input into the propagator. The nonlinear perturbations are then added to the dynamics, and the state is propagated in the ECI frame using MATLAB's `ode45`. On the next step in the trajectory, the ECI states of the target and the chaser are converted to the target's CWH relative frame and the same procedure is repeated until the mission has timed out, completed successfully or a constraint has been violated. Mission phase transitions are not shown directly in this diagram, however that information is given to the controller scheduler every simulation loop such that the appropriate controller is put to use as desired. As uncertainty is involved in this simulation, bounds are set on docking and relocation maneuver finishing positions, to avoid transitioning issues. Also, not shown in the diagram, during the joint maneuver phases, both the target and chaser spacecraft are propagated as a single spacecraft to be represented as the chaser spacecraft for the purpose of the diagram.

Finally, as a representation of the types of results that can be output from the simulation, Figure 3-11 shows a run of the simulation in terms of spacecraft estimated state, truth state, and reference trajectory in the orbital plane. The simulation also outputs representative results and plots that help analyze specific simulation results, such as control input over time, fuel over time, LQR cost over time, estimated states through the mission, and covariance on the estimated state throughout the mission.

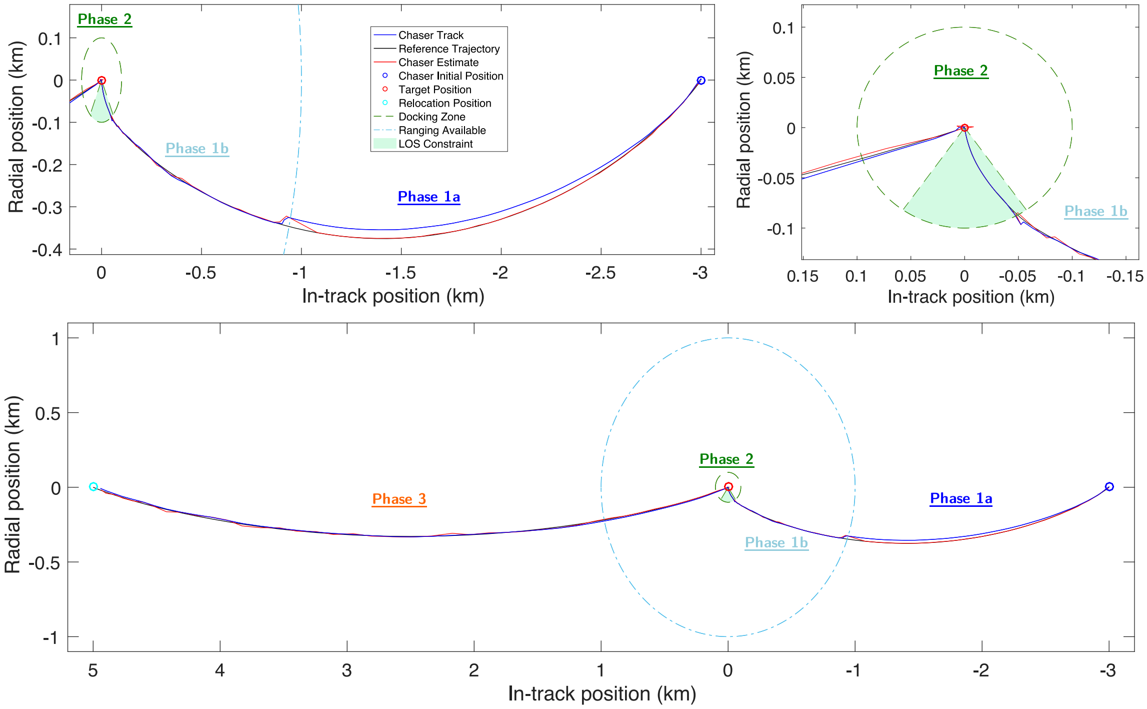


Figure 3-11: Sample results from a LQR reference-tracking controller with full nonlinear model simulation and aleatoric uncertainties (neither uncertain events nor epistemic uncertainty present)

The simulation also has a batch mode and a Monte Carlo mode. In the batch mode, different controllers and instantiations of uncertain parameters can be tested on the simulation with the same random seed to maintain a true comparison. In the Monte Carlo mode, the simulation can be run several hundred or thousands of times with different random seeds driving the noise characteristics and is typically used to evaluate the probabilistic performance of a guidance and control technique. Results from these modes are fairly complicated to explain here, so will instead be left for the results sections later in Chapters 4 and 5.

3.5 Overview of Guidance and Control Planning Process

At a high level, the approach to this thesis is divided into two major components as shown in Figure 3-12: trajectory optimization (in green, on top) and controller selection (in blue, on bottom). In the context of the problem set up to this point, there has been no distinction between the two except for in the description of the simulation. In both the thesis objectives in Section 1.6, and the literature review in Chapter 2, it is made clear however that we wish to separate these two components. Roughly, this is splitting guidance from control, and is typically how these concepts are handled for real applications in space. It is important to make the distinction.

The overall process is also split up into the pre-mission planning and onboard implementation phases. The dashed line in Figure 3-12 depicts this divide between the proactive

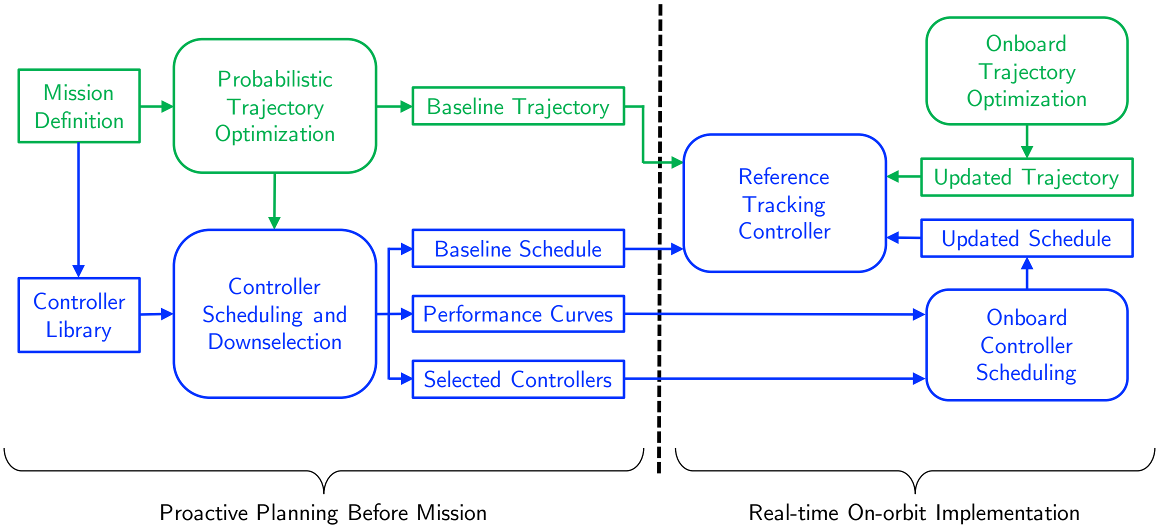

Mission Definition
Probabilistic Trajectory Optimization
Baseline Trajectory
Onboard Trajectory Optimization
Reference Tracking Controller
Updated Trajectory
Updated Schedule
Baseline Schedule
Controller Library
Controller Scheduling and Downselection
Performance Curves
Selected Controllers
Onboard Controller Scheduling
Proactive Planning Before Mission
Real-time On-orbit Implementation

Figure 3-12: High level description of the overall process of guidance and controller planning both before and during a mission

planning performed prior to the mission and the real-time replanning and rescheduling required during the mission. Alliteration aside, the goal is to create computationally intensive, probabilistically optimal guidance and control plans before the mission starts and then implement simplified versions on the spacecraft in real-time.

In more detail, the trajectory optimization process will function as follows. The execution must occur individually for a specific mission, so the first step is to define all the mission objectives, goal states, constraints, performance metrics, potential uncertainties, and uncertain events. This mission definition is then used to construct and plan a baseline optimal trajectory for the mission. If recoverable uncertain events with actionable models exist in the mission definition, then the trajectory will be planned to optimize performance probabilistically over this uncertainty model. This can involve significant computational power to solve and thus must be done on ground computers before the mission commences. Once a baseline optimal trajectory is computed, it is sent to the spacecraft to track with a reference-tracking controller. At a certain point in the mission, an uncertain event may occur such that replanning of the trajectory is required. In this scenario, the onboard version of the trajectory optimization software is implemented such that a quick updated trajectory is available to the reference-tracking controller. All of the theory and results pertaining to the trajectory optimization are presented in Chapter 4.

On the other side, the controller selection process starts with the creation of a controller library. This controller library is a set of hand-designed controllers that are implemented specifically for the problem defined in this chapter. The makeup of the library will contain a heterogeneous mix of control architectures from standard PD controllers to optimal controllers to adaptive controllers. These controllers would hopefully have varying performance with respect to the uncertainties present in the mission definition. Therefore, before the mission, we have the job of probabilistically assessing the performance of each of these controllers as the uncertainties change. The goal of this process is to find a non-dominated set of controllers that each perform the best under some set of circumstances and schedule these controllers across the different phases given our initial understanding of the uncertainty. Additionally, from the process, performance curves are created that express controller performance as a function of uncertainty. From here, the baseline controller schedule is passed to the spacecraft such that the appropriate reference-tracking controller can run. Again, at a certain point in the mission, an uncertain event may occur and the spacecraft will be required to replan the appropriate controller schedule. This online controller scheduler receives the performance curves for each of the selected controllers and evaluates these curves over the current knowledge of uncertainty to see if the current plan needs to be updated. If so, a new updated schedule is relayed such that the appropriate reference-tracking controller can be executed. All of the theory and results pertaining to the controller selection are presented in Chapter 5.

The goal of this section is to clearly set up the top-level structure into which the thesis work fits. It is meant to show high-level interactions between the different thesis components and clarify the assumptions made about what information is available where. As such, the diagram in Figure 3-12 may be referenced elsewhere in the thesis when detailing specific components. For a breakdown of which blocks of this diagram fall in which sections of the thesis, see Table 3.5.

Table 3.5: Mapping of the components of the guidance and control planning process to sections in the thesis

Topic	Locations
Mission Definition	Section 3.2
Probabilistic Trajectory Optimization	Section 4.4
Baseline Trajectory Results	Section 4.4.3, Section 4.4.4
Onboard Trajectory Optimization	Section 4.1, Section 4.2, Section 4.3
Updated Trajectory Results	Section 4.2.2, Section 4.3.3
Controller Library	Section 5.2
Controller Scheduling and Selection	Section 5.1.1, Section 5.1.2, Section 5.3.3
Baseline Schedule Results	Section 5.4
Performance Curves	Section 5.3.1, Section 5.3.2
Onboard Controller Scheduling	Section 5.1.4
Selected/Scheduled Controller Results	Section 5.4

Chapter 4

Trajectory Optimization

This chapter discusses the formulation and results of the trajectory optimization and path planning research completed for this thesis. Section 4.1 details two objective functions, a minimum energy and a minimum fuel form; both linear and nonlinear constraints to which each of these optimization problems are subjected; and a short description of solution techniques. Section 4.2 explains why it is beneficial to optimize trajectories continuously between phases instead of individually for each phase. Section 4.3 adds obstacle avoidance constraints to the problem, and defines and compares two methods used to account for these obstacles. Finally, using all of the previous work and results in this chapter, Section 4.4 provides one of the major contributions of this thesis: probabilistic trajectory planning in the presence of uncertain events. Specifically, the process is described in detail and then two case studies are explored with results.

4.1 Trajectory Optimization Formulation

The goal of the trajectory optimization problem is to find both the best path for the spacecraft to reach its goal locations and the control input required to attain that path. In the problem described in Chapter 3, the path should take the spacecraft from the initial state through Phases 1a and 1b to stage itself for docking within the required sensor LOS zone, complete docking to the satellite in Phase 2, and then transition to control of the chaser and target spacecraft together in Phase 3, such that they reach the desired relocation orbital position. Thus, we want to solve for the path and control input that takes the spacecraft through all of these phases and completes the mission for the lowest possible cost subject to all of the constraints along the way. This cost can be thought of in multiple forms. In this thesis, we only consider the minimum linear-quadratic cost that penalizes both control input and state error and the minimum fuel cost that purely penalizes total control input. Solutions to both of these optimization formulations are discussed in the following sections.

4.1.1 Minimum Energy

For this section, the minimum energy formulation of the trajectory optimization problem will be discussed. This formulation is a minimum energy formulation, because it involves the square of the control input rather than the control input alone. Optimal solutions will differ from the pure minimum fuel or minimum control input formulation. To match a common linear-quadratic cost function in optimal control, the state error is also included, however the weighting on the control penalty is very high when compared to the weighting on the state error. Typically, a cost function like this will be used in MPC as described in Section 5.2.4, and there is substantial literature available on the topic.

The optimization format used in this thesis will be that using a discretized mission horizon. The desired length of the mission is divided into discrete time steps at which the control input is able to change. The solver then attempts to choose the optimal control input for each of these time steps throughout the mission horizon, such that at the final time step, the spacecraft has reached the desired goal location.

The minimum energy objective function to be optimized over the mission is denoted as J_{nrg}, where

$$J_{nrg}(\mathbf{U}, \mathbf{x}(0)) = \sum_{i=0}^{N-1} \Big(\mathbf{y}_p(i)^T R_{xx} \mathbf{y}_p(i) + \mathbf{u}(i)^T R_{uu} \mathbf{u}(i) \Big), \tag{4.1}$$

in which $\mathbf{U} = \Big[\mathbf{u}(0)^T, \quad \mathbf{u}(1)^T, \quad \ldots, \quad \mathbf{u}(N-1)^T \Big]^T$ is the $3N \times 1$ stacked vector of control input, $\mathbf{x}(0)$ is the chaser's initial state, $\mathbf{y}_p(i)$ is the error between the chaser's planned position at the i-th step of the N-step horizon and the final goal location, $R_{xx} \geq 0$ is the state error cost weighting matrix, and $R_{uu} > 0$ is the control cost weighting matrix. Further details on this objective function and a similar formulation for MPC can be found in [83] and [200]. In (4.1), there is a penalty on each step of the trajectory. For all but the last step in the trajectory, the penalty is on a combination of reference tracking error and applied control. For the last step, there is a constraint that the spacecraft reaches the final location.

With the previously described formulation, the objective function is minimized by varying the control input across each step of the planned trajectory. Adding appropriate dynamics, maximum thrust, and endpoint constraints, it is desired to find

$$
\begin{aligned}
& \min_{\mathbf{U}} \ J_{nrg}(\mathbf{U}, \mathbf{x}(0)), \\
\text{subject to} \quad \mathbf{x}(i+1) &= A_d\mathbf{x}(i) + B_d\mathbf{u}(i), && \forall i = 0, \ \dots \ , \ N-1, \\
\mathbf{x}(N) &= \mathbf{x}_f, \\
\mathbf{y}_p(i) &= [I_{3\times3} \ \ 0]\left(\mathbf{x}(i) - \mathbf{x}_f\right), && \forall i = 0, \ \dots \ , \ N, \quad \text{and} \\
\|\mathbf{u}(i)\|_\infty &\leq \bar{u}, && \forall i = 0, \ \dots \ , \ N-1,
\end{aligned}
\tag{4.2}
$$

where $\bar{u}$ is the maximum thrust that can be applied in each direction as specified in Section 3.2.8, $\mathbf{x}_f$ is the desired final state, and A_d and B_d are the discretized linear CWH matrices described in Section 3.1.1. For a typical rendezvous and docking problem, $\mathbf{x}_f$ will be the center of the CWH frame as a vector of zeros. Alternately it may be an offset distance from that location to account for a docking port or it may be a different waypoint or state in the CWH frame as desired. For consistency throughout, only the 3DOF system will be represented in the analysis as the 2DOF system is only a decoupled subset and can be derived easily in all expressions. Note that the specific constraints detailed in Section 3.2 for each phase of the mission have not been implemented at this point and will be discussed later in Section 4.1.3.

The problem stated in (4.2) can be reformulated into a form that is more suitable for optimization solvers. The states, $\mathbf{x}(i)$, for each time step in the horizon are stacked into a long column vector, which is defined by the $6N \times 1$ vector, $\mathbf{X} = [\mathbf{x}(1)^T, \mathbf{x}(2)^T, \dots, \mathbf{x}(N)^T]^T$, and the control inputs, $\mathbf{u}(i)$, are stacked into $\mathbf{U}$ as defined before. In this form, the discrete dynamics propagation becomes

$$\mathbf{X} = \Psi\mathbf{x}(0) + \Omega\mathbf{U}, \tag{4.3}$$

where

$$\Psi = \begin{bmatrix} A_d \\ A_d^2 \\ \vdots \\ A_d^N \end{bmatrix} \quad \text{and} \tag{4.4}$$

$$\Omega = \begin{bmatrix} B_d & 0_{6\times3} & 0_{6\times3} & 0_{6\times3} & \dots & 0_{6\times3} \\ A_dB_d & B_d & 0_{6\times3} & 0_{6\times3} & \dots & 0_{6\times3} \\ A_d^2B_d & A_dB_d & B_d & 0_{6\times3} & \dots & 0_{6\times3} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ A_d^{N-2}B_d & A_d^{N-3}B_d & \dots & A_dB_d & B_d & 0_{6\times3} \\ A_d^{N-1}B_d & A_d^{N-2}B_d & \dots & A_d^2B_d & A_dB_d & B_d \end{bmatrix}. \tag{4.5}$$

The optimal control problem without the additional path constraints added, as posed in (4.2), can then be reworked to the form of

$$\begin{gathered} \min_{\mathbf{U}} \; \tfrac{1}{2}\mathbf{U}^T\mathbf{S}\mathbf{U} + \mathbf{H}^T\mathbf{U}, \\ \text{subject to} \\ \mathbf{V}_{ineq}\mathbf{U} \leq \mathbf{W}_{ineq}, \text{ and} \\ \mathbf{V}_{eq}\mathbf{U} = \mathbf{W}_{eq}, \end{gathered} \tag{4.6}$$

where $\mathbf{S}$, $\mathbf{H}$, $\mathbf{V}_{ineq}$, $\mathbf{W}_{ineq}$, $\mathbf{V}_{eq}$, and $\mathbf{W}_{eq}$ are left undefined here simply to show the structure of the optimal control problem, but are formulated as discussed in [200]. $\mathbf{S}$ and $\mathbf{H}$ are constructed such that the discrete dynamics constraints and output constraints are enforced in the cost function rather than as separate equality constraints. Now the form of the optimization problem is much simpler for a solver to accept as it is in quadratic form with linear constraints. The $\mathbf{V}$ and $\mathbf{W}$ matrices are defined in full detail in Section 4.1.3 to include all constraints in (4.2) and additional linear path constraints added into this quadratic program for different phases.

Minimum energy solutions usually involve continuously active control input, rather than the bang-coast-bang manner for which minimum fuel problems are solved. The goal states must change between the origin of the CWH frame in the docking problem to the relocation position in the joint-maneuvering problem. Thus, the optimization is attempting to minimize the total state error from these goal states at the same time as minimizing the control effort to reach the goal state. This tradeoff creates the continuous control input signal. And because the control penalty is much larger than the state error penalty, the solution will always take the full mission horizon to reach the goal state.

4.1.2 Minimum Fuel

The minimum fuel formulation of the trajectory optimization problem seeks to find the path that minimizes the total control input required through the mission. A very similar optimization formulation to the previous section will be used here as well, although this time the objective function will be the sum of the propellant required at each time step of the discretized mission horizon, represented in a nonlinear form as

$$\begin{gathered}\min_{\mathbf{U}} \sum_{i=0}^{N-1} \|\mathbf{u}(i)\|_1 \\ \text{subject to } \; \mathbf{V}_{ineq}\mathbf{U} \leq \mathbf{W}_{ineq} \text{ and } \mathbf{V}_{eq}\mathbf{U} = \mathbf{W}_{eq},\end{gathered} \tag{4.7}$$

where the $\mathbf{V}$'s, $\mathbf{W}$'s, $\mathbf{U}$, $\mathbf{u}(i)$, i, and N are described for the same mission horizon as in Section 4.1.1, and the $\|\cdot\|_1$ operator represents the 1-norm or sum of absolute values of the components of the vector on which it acts. Note that the $\mathbf{V}$ and $\mathbf{W}$ matrices include the constraints on the final endpoint condition, control saturation and dynamics and are described in full in Section 4.1.3.

Unfortunately, this function is nonlinear due to the absolute value function present in the 1-norm. Nonlinear solvers have difficulty with minimum fuel problems due to the bang-coast-bang nature of the control creating discontinuities in the solution. Fortunately, however, this can be transformed into linear form in a couple ways as found in [201].

The first method is to reform (4.7) as

$$\begin{aligned} &\min_{\mathbf{U},\,\mathbf{Z}} \sum_{i=0}^{N-1} \|\mathbf{z}(i)\|_1 \\ \text{subject to } \quad \mathbf{V}_{ineq}\mathbf{U} &\leq \mathbf{W}_{ineq}, \\ \mathbf{V}_{eq}\mathbf{U} &= \mathbf{W}_{eq}, \\ \mathbf{u}(i) &\leq \mathbf{z}(i) \qquad \forall\, i = 0,\ \ldots,\ N-1, \text{ and} \\ -\mathbf{u}(i) &\leq \mathbf{z}(i) \qquad \forall\, i = 0,\ \ldots,\ N-1, \end{aligned} \tag{4.8}$$

where $\mathbf{Z}$ and $\mathbf{z}(i)$ mimic $\mathbf{U}$ and $\mathbf{u}(i)$ although are required to be positive and equal to the absolute values of the components of $\mathbf{U}$ at the optimal solution. However, this doubles the number of optimization variables and adds $6N$ additional constraints. There is a better option that only doubles the number of optimization variables and maintains the same number of the constraints. Instead, we can split $\mathbf{U}$ into positive and negative components such that $\mathbf{U} = \mathbf{U}^+ - \mathbf{U}^-$ and reform (4.7) as

$$\begin{aligned} \min_{\mathbf{U}^+,\, \mathbf{U}^-} \; & \sum_{j=1}^{3N} \Big(\mathbf{U}(j)^+ + \mathbf{U}(j)^- \Big) \\ \text{subject to} \quad \mathbf{V}_{ineq}\mathbf{U}^+ - \mathbf{V}_{ineq}\mathbf{U}^- & \leq \mathbf{W}_{ineq}, \\ \mathbf{V}_{eq}\mathbf{U}^+ - \mathbf{V}_{eq}\mathbf{U}^- & = \mathbf{W}_{eq}, \text{ and} \\ \mathbf{U}^+, \mathbf{U}^- & \geq 0. \end{aligned} \tag{4.9}$$

After one final reorganization, we can see that this is in standard linear form as

$$\begin{aligned} \min_{\mathbf{U}'} \; & \mathbf{g}^T\mathbf{U}' \\ \text{subject to} \quad \mathbf{V}'_{ineq}\mathbf{U}' & \leq \mathbf{W}'_{ineq}, \\ \mathbf{V}'_{eq}\mathbf{U}' & = \mathbf{W}'_{eq}, \text{ and} \\ \mathbf{U}' & \geq 0, \end{aligned} \tag{4.10}$$

where $\mathbf{U}' = \Big[(\mathbf{U}^+)^T \quad (\mathbf{U}^-)^T \Big]^T$, the $\mathbf{V}'$ and $\mathbf{W}'$ matrices have changed to enforce the same constraints, and $\mathbf{g}$ is simply a vector of ones.

4.1.3 Constraint Overview

The constraints for the minimum energy and minimum fuel problem are identical and as defined throughout Section 3.2, however need to be enacted slightly differently due to the formulation of the optimization variables in each problem. First, the constraints will be developed for the minimum energy problem as this is more intuitive. Then the constraints will be modified slightly to work for the minimum fuel problem. Nonetheless, they are so similar that the description for both can fit in one section here.

A standard optimization format for most solvers is to separate linear equality and inequality constraints such that they can each be represented by two matrices. In this thesis, we represent these constraints as

$$\begin{aligned} \mathbf{V}_{ineq}\mathbf{U} & \leq \mathbf{W}_{ineq} \text{ and} \\ \mathbf{V}_{eq}\mathbf{U} & = \mathbf{W}_{eq}, \end{aligned} \tag{4.11}$$

where $\mathbf{U}$ is the stacked vector of control inputs as described earlier. Additionally, we separate nonlinear constraints into the functions

$$\begin{aligned} \mathbf{f}_{ineq}(\mathbf{U}) &\leq 0 \text{ and} \\ \mathbf{f}_{eq}(\mathbf{U}) &= 0. \end{aligned} \tag{4.12}$$

The following section will discuss the creation of these constraint matrices and functions to match this form. In general, throughout this section, individual constraints will be represented in the form shown above and all of the constraints will be stacked into the final **V** and **W** matrices and functions as appropriate. Thus, the number of rows in the matrices and the number of rows in the output of the functions will change, but the number of columns in the matrices and input into the functions will remain constant.

First, we will discuss the linear constraints present in all phases. These constraints are control saturation and potential phase or mission endpoints or waypoints. These constraints would not benefit from a nonlinear formulation, and thus will always be represented as linear constraints.

The control saturation constraints are the simplest, since it is only a bound on the optimization variables such that $\|\mathbf{u}\|_\infty \leq \bar{u}$. This is implemented in the minimum energy problem as

$$I_{3N\times 3N}\mathbf{U} \leq \begin{bmatrix} \bar{u} \\ \vdots \\ \bar{u} \end{bmatrix}_{3N\times 1} \quad \text{and} \quad -I_{3N\times 3N}\mathbf{U} \leq \begin{bmatrix} \bar{u} \\ \vdots \\ \bar{u} \end{bmatrix}_{3N\times 1}, \tag{4.13}$$

where $\bar{u}$ is defined in Section 3.2.8 and noting that we impose the constraint on both negative and positive thrusts as desired from the infinity-norm. This can be represented in the minimum fuel form as

$$\begin{bmatrix} I_{3N\times 3N} & 0_{3N\times 3N} \\ 0_{3N\times 3N} & I_{3N\times 3N} \end{bmatrix} \begin{bmatrix} \mathbf{U}^+ \\ \mathbf{U}^- \end{bmatrix} \leq \begin{bmatrix} \bar{u} \\ \vdots \\ \bar{u} \end{bmatrix}_{6N\times 1}, \tag{4.14}$$

where we also must remember that

$$\begin{bmatrix} -I_{3N\times 3N} & 0_{3N\times 3N} \\ 0_{3N\times 3N} & -I_{3N\times 3N} \end{bmatrix} \begin{bmatrix} \mathbf{U}^{+} \\ \mathbf{U}^{-} \end{bmatrix} \leq \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}_{6N\times 1} \tag{4.15}$$

to enforce that the split control input vector, $\mathbf{U}^{'}$, is always positive as described in Section 4.1.2.

Next, we will discuss the endpoint and waypoint constraint formulation. This can be used to create waypoints that must be met within a phase or a stopping point at which to switch over to the next phase. Alternatively, only endpoint constraints may exist such as a docking state constraint and a final relocation position constraint. In any of these conditions, the formulation of the constraints follows the same pattern. For the minimum energy form, we can use the same large propagation matrices, Ω and Ψ, created to simplify the objective function. By propagating the spacecraft state to the appropriate time step with all of the control inputs selected by the optimizer and constraining that state, we have

$$\Omega_i \mathbf{U} = \left(\mathbf{x}_i - \Psi_i \mathbf{x}(0)\right), \tag{4.16}$$

where Ψ_i and Ω_i are the i-th block row of the block matrices described in (4.4) and (4.5), and $\mathbf{x}_i$ is the desired waypoint or endpoint at the i-th step of the N-step horizon. For example, an endpoint constraint on the final step in the horizon would look in expanded form as

$$\begin{bmatrix} A_d^{N-1}B_d & A_d^{N-2}B_d & \dots & A_d^2 B_d & A_d B_d & B_d \end{bmatrix} \mathbf{U} = \left(\mathbf{x}_f - A_d^N \mathbf{x}(0)\right), \tag{4.17}$$

where the $\mathbf{x}_f$ is the desired final point or endpoint. For the minimum fuel form, this constraint becomes

$$\begin{bmatrix} \Omega_i & -\Omega_i \end{bmatrix} \begin{bmatrix} \mathbf{U}^{+} \\ \mathbf{U}^{-} \end{bmatrix} = \left(\mathbf{x}_i - \Psi_i \mathbf{x}(0)\right). \tag{4.18}$$

Interestingly, for the minimum fuel case, this is the only appearance of the dynamics in the optimization problem, because the dynamics constraints were absorbed in the cost function for the minimum energy case but were not in the minimum fuel case. Nonetheless, as long as an endpoint constraint is always specified for the minimum fuel problem, which should always be the case, there will be no issues as all of the dynamics are included inherently in this constraint.

Now the phase specific constraints will be explored. The docking component of the mission, Phase 2 is the only phase that has special constraints. Each of these constraints can take either a linear or nonlinear form as desired. Described as a function of the spacecraft state in Section 3.2.4, there are two extra constraints in Phase 2: a velocity constraint and a LOS zone constraint to maintain sensor coverage.

The velocity constraint is an infinity-norm constraint in the linear form, which, for the minimum energy formulation, translates to

$$\Omega_{vel}\mathbf{U} \leq \left(\begin{bmatrix} \bar{V} \\ \vdots \\ \bar{V} \end{bmatrix}_{3N\times 1} - \Psi_{vel}\mathbf{x}(0) \right), \tag{4.19}$$

where Ω_{vel} and Ψ_{vel} are the shrunken Ω and Ψ matrices to only include the velocity states from A_d and B_d, and $\bar{V}$ is the maximum velocity permitted in the docking phase as defined in Section 3.2.8. Ω_{vel} and Ψ_{vel} are $3N \times 3N$ and $3N \times 6$ respectively and just contain the rows pertaining to velocity states. In the minimum fuel form, this constraint becomes

$$\begin{bmatrix} \Omega_{vel} & -\Omega_{vel} \end{bmatrix} \begin{bmatrix} \mathbf{U}^{+} \\ \mathbf{U}^{-} \end{bmatrix} \leq \left(\begin{bmatrix} \bar{V} \\ \vdots \\ \bar{V} \end{bmatrix}_{3N\times 1} - \Psi_{vel}\mathbf{x}(0) \right). \tag{4.20}$$

A more lenient constraint, that does not penalize velocity in directions pointing away from the target, is the dot product constraint mentioned in Section 3.2.4. This must be implemented in nonlinear form. For the minimum energy case, all that needs to be done is convert the optimization variables $\mathbf{U}$ into states $\mathbf{X}$ through (4.3), and then loop through each step of the horizon, extracting $\boldsymbol{\rho}$ and $\dot{\boldsymbol{\rho}}$ and setting

$$\mathbf{f}_{ineq}(\mathbf{U}) = \begin{bmatrix} -\dot{\boldsymbol{\rho}}(1) \cdot \dfrac{\boldsymbol{\rho}(1)}{\|\boldsymbol{\rho}(1)\|} \\ \vdots \\ -\dot{\boldsymbol{\rho}}(i) \cdot \dfrac{\boldsymbol{\rho}(i)}{\|\boldsymbol{\rho}(i)\|} \\ \vdots \\ -\dot{\boldsymbol{\rho}}(N) \cdot \dfrac{\boldsymbol{\rho}(N)}{\|\boldsymbol{\rho}(N)\|} \end{bmatrix} - \bar{V}, \tag{4.21}$$

where $\boldsymbol{\rho}(i)$ and $\dot{\boldsymbol{\rho}}(i)$ are the position and velocity in the i-th step of the N-step horizon as defined in Section 3.2.4. For the minimum fuel case, all that needs to be done is transform the $\mathbf{U}'$ vector into a $\mathbf{U}$ vector by setting $\mathbf{U} = \mathbf{U}^{+} - \mathbf{U}^{-}$ before implementing the above procedure.

The LOS zone constraint can also be implemented in linear form through a pyramid constraint and nonlinear form through a cone constraint as described in Section 3.2.4. The linear form for the minimum energy formulation is

$$\left(I_{N\times N} \otimes \mathcal{P}\right) \Omega \mathbf{U} \leq -\left(I_{N\times N} \otimes \mathcal{P}\right) \Psi \mathbf{x}(0), \tag{4.22}$$

where $\otimes$ is the Kronecker product and

$$\mathcal{P} = \begin{bmatrix} \sin\left(\frac{\theta_1}{2} - \phi\right) & \cos\left(\frac{\theta_1}{2} - \phi\right) & 0 & 0 & 0 & 0 \\ \sin\left(\frac{\theta_1}{2} + \phi\right) & -\cos\left(\frac{\theta_1}{2} + \phi\right) & 0 & 0 & 0 & 0 \\ \sin\frac{\theta_2}{2}\cos\phi & \sin\frac{\theta_2}{2}\sin\phi & \cos\frac{\theta_2}{2} & 0 & 0 & 0 \\ \sin\frac{\theta_2}{2}\cos\phi & \sin\frac{\theta_2}{2}\sin\phi & -\cos\frac{\theta_2}{2} & 0 & 0 & 0 \end{bmatrix}. \tag{4.23}$$

For the minimum fuel problem, this becomes,

$$\begin{bmatrix} \left(I_{N\times N} \otimes \mathcal{P}\right) \Omega & -\left(I_{N\times N} \otimes \mathcal{P}\right) \Omega \end{bmatrix} \begin{bmatrix} \mathbf{U}^{+} \\ \mathbf{U}^{-} \end{bmatrix} \leq -\left(I_{N\times N} \otimes \mathcal{P}\right) \Psi \mathbf{x}(0). \tag{4.24}$$

When considering a sensor on the target spacecraft aiding in the navigation of the chaser, the pyramid constraint would mimic the field of view of a typical rectangular sensor.

When the target has fiducial markings and no onboard sensors, the chaser's line of sight to the fiducial markings would trace a cone instead of a pyramid, thus sometimes the nonlinear cone constraint maybe worthwhile to use instead. Finding this nonlinear version of the LOS constraint follows a similar procedure to the nonlinear velocity constraint. First, we convert the optimization variables $\mathbf{U}$ into states $\mathbf{X}$ through (4.3), and then loop through each step of the horizon, extracting $\boldsymbol{\rho}$ and setting

$$\mathbf{f}_{ineq}(\mathbf{U}) = \begin{bmatrix} \dfrac{-\boldsymbol{\rho}(1) \cdot \boldsymbol{c}}{\|\boldsymbol{\rho}(1)\|\|\boldsymbol{c}\|} \\ \vdots \\ \dfrac{-\boldsymbol{\rho}(i) \cdot \boldsymbol{c}}{\|\boldsymbol{\rho}(i)\|\|\boldsymbol{c}\|} \\ \vdots \\ \dfrac{-\boldsymbol{\rho}(N) \cdot \boldsymbol{c}}{\|\boldsymbol{\rho}(N)\|\|\boldsymbol{c}\|} \end{bmatrix} + \cos\left(\frac{\theta}{2}\right), \tag{4.25}$$

where $\boldsymbol{\rho}(i)$ are the position vectors in the i-th step of the N-step horizon as defined in Section 3.2.4. For the minimum fuel case, all that needs to be done is transform the $\mathbf{U}'$ vector into a $\mathbf{U}$ vector by setting $\mathbf{U} = \mathbf{U}^{+} - \mathbf{U}^{-}$ before implementing the above procedure.

That finishes the description of all the constraints present in the benchmark problem discussed in Section 3.2. Of course, more constraints could be added to the problem following a similar process to that described above. Once the individual constraints are in the correct form, simply concatenating them to the end of the existent $\mathbf{V}$ and $\mathbf{W}$ matrices and output vectors of the nonlinear functions is sufficient for implementation. We will add additional obstacle avoidance constraints in Section 4.3

4.1.4 Solution Techniques

From Section 4.1.1 through Section 4.1.3, both the minimum energy and minimum fuel optimization problems have been structured into forms that are amenable to solvers. This section discusses the solvers used to complete the work in this thesis and high level descriptions of how those solvers work.

> **Minimum Energy**
>
> At this point, the minimum energy optimization problem is in quadratic form with options of both linear and nonlinear constraints. The linear constraint formulation is convex, and thus can be solved in a very efficient manner with any generic quadratic programming solver. The work in this thesis uses MAT-

LAB's quadprog function, which uses an interior point method combined with Lagrangian duality methods and guarantees a global minimum if formulated feasibly. There are many other quadratic solvers available, both open source and for purchase. Given that the Karush-Kuhn-Tucker (KKT) conditions are checked and are valid, the globally optimal solution can be found due to the convex nature of the minimum energy formulation.

If using nonlinear constraints, the problem is no longer entirely quadratic, so alternate optimization techniques need to be used. Sequential Quadratic Programming (SQP) is an efficient approach and is implementable with a little effort through MATLAB's fmincon function, but also has been written for flight system execution in C/C++ [202]. SQP essentially is Newton's method of optimization with constraints added through Lagrange multipliers. SQP, of course, takes significantly longer to run than a quadratic solver. The work in this thesis with nonlinear constraints was all implemented through fmincon. There are tricks one can use to get the most out of the solver including variable scaling to achieve a Hessian matrix with order of about one on the diagonal. This will allow gradient based solvers to better be able to search when the objective function is very flat or very steep in the areas near the optimum. Additionally, sensitivity studies can be performed to see the appropriate tolerances and exit criterion such that effective optimization performance is achieved for the quickest time. These analyses are left out of this thesis but discussed more in [97] if interested. Due to the nonlinear constraint formulation, the problem has the potential to become non-convex. If non-convex, the KKT conditions will not guarantee a globally optimal solution, only a locally optimal solution. Note that significant effort should be placed in approximating non-convex constraints with convex constraints, however, if the constraints need to be non-convex, the user must accept that possibly sub-optimal local minima may be found. Nonetheless, the feasibility of the solution comes first, and this possible performance degradation is acceptable given that a feasible trajectory is still found.

Minimum Fuel

The linear formulation of the minimum fuel problem with linear constraints allows solution through any generic linear solver, which will enable even higher efficiency than the quadratic solvers. Again, for the work in this thesis, MATLAB's `linprog` function is used. This implementation of linear solver uses dual simplex methods to quickly solve large order problems with guaranteed globally optimal solutions.

Unfortunately, there is not a great option openly available to solve the minimum fuel linear objective function with nonlinear constraints. One could implement the problem using MATLAB's `fmincon`, and the author has done just this, however the solutions are not extremely accurate and the run times are extremely long. The challenge is that for a minimum fuel problem, the solution will have a discontinuous bang-coast-bang solution for the control input. Gradient based methods such as `fmincon` will have a lot of trouble with these discontinuities. The solution this author uses with `fmincon` is to remove the ability of the optimizer to select nonzero control at intermediate times and only allow control near the beginning and end of the mission. This significantly reduces the number of optimization variables and although it still takes a long time to run, will give suboptimal, feasible solutions. The author does not advise using this method and has not placed any results from the minimum fuel problem with nonlinear constraints in this thesis due to the method's inherent fragility. Again, given possibly non-convex constraints applied to the linear problem, there is no guarantee for a globally optimal solution.

4.2 Multi-stage Optimization

Traditionally, trajectories and missions are designed with extreme safety and risk aversion in mind. Artificial waypoints or stopping points are built into the mission plan such that ground crews can issue abort maneuvers every so often or choose to wait at these gates if deemed necessary. Some of these waypoints are created to ensure that sensing systems are active and receiving correct measurements before continuing onto the next maneuver. This is analogous to the transition to the LOS zone in Phase 2 of our problem, where the higher precision docking sensors are available. New constraints are necessary in this phase of the mission to maintain sensor lock and ensure safe docking, and it is typically easier to separate this phase of the trajectory analysis such that the constraints are only dealt with where necessary. Unfortunately, stopping the trajectories between these phases creates an unnecessary constraint that decreases performance. Freeing up this waypoint constraint could therefore improve performance if the same level of safety can still be achieved.

Additionally, trajectories are designed using common knowledge of ways to exploit the orbital dynamics in the problem. Thus, simplified trajectories are created to ensure the rendezvous and docking is safe at every step. A common technique involves hopping through several waypoints on the way to dock, such that if thrusters fail, collision with the target spacecraft will be avoided for an arbitrarily large number of orbital periods. When the spacecraft misses a burn during these hopping maneuvers, the natural orbital dynamics will cause the spacecraft to drift away from the target. Unfortunately, however, designing a mission in this manner leads to overly safe trajectories and a corresponding loss of performance. Techniques have been examined, however, to avoid these waypoint requirements and still guarantee collision avoidance if thrusters fail [146].

Finally, the fact that these trajectory optimization problems can be solved in real-time allows us to examine higher performing methods that have smoother transitions between phases and an absence of waypoints. If something else were to go wrong in a transition between phases, the spacecraft itself could plan a new trajectory to safely maneuver. The risk that these failures will happen is fairly low, and if the same level of safety can be attained by reacting to them, there is the potential to probabilistically gain better performance in the long run.

Therefore, the next two sections will discuss the process of trajectory optimization continuously through multiple phases that each have their own constraints. This will be known as full-mission optimization while the traditional approach would be phase-independent optimization.

4.2.1 Linking Phases

Given that waypoints between phases will overconstrain the problem, we require a method to free up the transition point from this fixed constraint. By splitting up phases and performing phase-independent optimization, the engineer is forced to either select a specific position and velocity for a final fixed endpoint constraint or can develop a set of states that are possible for the endpoint. Even if the endpoint constraint consists of a larger set, there is still no guarantee that the optimally selected endpoint of the individual phase will be the optimal transition state between the phases. There is also no guarantee that constraints in the next phases will not be violated if a high velocity exists at that transition point. By setting the transition point as a free parameter (in time, position, and velocity), and only implementing appropriate constraints for each particular phase, we can avoid these pitfalls of the phase-independent optimization approach. Note that in some cases, like the end of the docking phase, there is a required fixed position with zero velocity. In these cases, with no free variables at the phase transition, phase-independent optimization will produce the same results as the full-mission optimization. Thus, in these cases, computation times will significantly improve if the phases are separated. Note that this is only true if all variables are fixed at the phase transition. If any state variable is free, however, there is the possibility that the phase-independent optimization will produce suboptimal results.

The specific transitions from Phase 1a to 1b and from Phase 1b to 2 will be explored in this section, where the transition from Phase 2 to 3 is considered suitable for phase-independent optimization. Therefore, we solve for two trajectories: one with a goal location of $\mathbf{x}_{docked}$ that covers Phases 1a through 2 and another with a goal location of $\mathbf{x}_{relocation}$ covering Phase 3. This section therefore discusses the process of linking the different phases in the optimization problem. Additionally, a reactive technique will be discussed in the case that during the actual mission there is large reference-tracking error when transitioning from phase to phase. In this case, replanning an optimal trajectory at this point will result in better overall performance metrics for the mission.

In general, this problem of having multiple mission phases and free transition variables fits very well into the GPOPS-II optimal control solver framework. Unfortunately, the GPOPS-II solver was deemed to be too slow to execute trajectory planning in real-time and has not been developed for implementation on a spacecraft. Thus, alternate techniques were investigated. Specifically, Mixed Integer Linear Programming (MILP) and Mixed Integer Nonlinear Programming (MINP) serve our purposes very well. The benefit with integer programming is that if-then constraints can be implemented quite elegantly. These if-then constraints are necessary for our problem between Phase 1b and Phase 2 because there are constraints specific to Phase 2 that should not be enforced in previous phases. Thus, these constraints should only be applied if the state meets the conditions for Phase 2. In this manner, we do not need to set a transition waypoint as the constraints will be implemented automatically as the state varies.

For MINP, the if-then constraint for Phase 2 can be implemented as follows. One integer, binary variable is added to the optimization variable vector for every point in the trajectory. If this point is within the radius that defines Phase 2, the variable will automatically be set to 0, while if it lies outside the Phase 2 radius, the variable will automatically be set to 1. The docking velocity constraint in (3.34) can be reformulated as

$$-\dot{\boldsymbol{\rho}} \cdot \frac{\boldsymbol{\rho}}{\|\boldsymbol{\rho}\|} \leq \bar{V} + M_1 b, \tag{4.26}$$

where M_1 is an arbitrarily large number and $b \in \{0, 1\}$. The LOS zone constraint from (3.31) can be reformulated as

$$\frac{\boldsymbol{\rho} \cdot \boldsymbol{c}}{\|\boldsymbol{\rho}\|\|\boldsymbol{c}\|} + M_2 b \geq \cos\left(\frac{\theta}{2}\right), \tag{4.27}$$

where M_2 is another arbitrarily large number. In conjunction with these constraints, an additional constraint is required to force the value of b to be either 0 or 1 depending if the spacecraft is inside the radius of Phase 2 or outside. This constraint is that

$$\rho_d \leq \|\mathbf{x}_{pos}\| + M_3(1 - b), \tag{4.28}$$

where M_3 is another arbitrarily large number. To meet each of these three constraints at the same time, b will be forced to be 1 in Phases 1a and 1b when $\|\mathbf{x}_{pos}\| > \rho_d$ and will be forced to be 0 in Phase 2 when $\|\mathbf{x}_{pos}\| \leq \rho_d$. The same process can be used with the linear versions of these constraints to form the problem for the MILP solver, although the radius for Phase 2 will need to be converted to a conservative, linear plane constraint tangent to the circle at that radius. Note that in the optimization framework described in the previous sections, one b-variable will need to be implemented for each point in the trajectory, meaning that the optimization variable vector will grow by another factor of the horizon length, N. The full formulation is left out here, but follows the general form of the discussion of formulating these types of constraints in Section 4.1.3. After these constraints are implemented with one of these solvers, the full mission optimization can be performed as the docking phase constraints will only be implemented where necessary and the transition state will be free between the phases.

Alternately, if a MILP or MINP solver is not available, it is more difficult to handle this phase transition. Since these solvers are not as widely available to the general public as linear or quadratic solvers are, this thesis will present all results using an approximation of the above integer constraints suitable for linear and quadratic solvers. This involves implementing a constraint interpolation and search process to find the optimal phase transition time. The basic process is to only enforce the docking phase constraints for the particular time steps relevant for the docking phase. Finding the appropriate time steps to enforce the constraints is the goal of this process. Typically, this can be done as finding an offset from the docking time (i.e., the end of Phase 2 and final point of our fixed horizon).

Initially, a conservative guess is made requiring the docking constraints to be met much earlier than truly required. This is illustrated in Figure 4-1. Results from the optimization are recorded, the resulting trajectory is interpolated at a finer discretization level using methods from Section 4.1.3 and analyzed against the true nonlinear constraint. If the trajectory is feasible, the offset length from the docking time is shrunk by a predetermined amount and the optimization is run again. This is repeated until either the recorded performance cost from the optimization reaches a minimum and then starts to rise, or if the trajectory is deemed infeasible. The search for this minimum can be performed as desired. The results in this thesis are obtained from a standard bisection search algorithm. Although this increases the time required to solve the problem, the search process usually produces

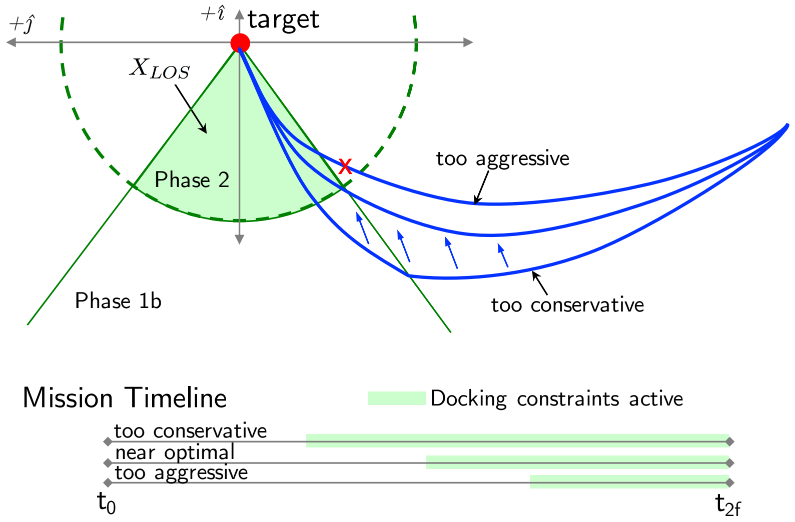


Figure 4-1: Brief graphic describing the goals of the constraint interpolation process for finding the optimal transition time between Phases 1b and 2

a solution in less than ten iterations.

Although phase-independent optimization can be used to split Phases 2 and 3 without performance loss, there are still some differences when planning a trajectory for Phase 3 when compared to the other phases. The constant reference point, $\mathbf{x}_f$, now changes from $\mathbf{x}_{docked}$, a vector of zeros, to $\mathbf{x}_{relocation}$, which is located 5km in the in-track direction. Importantly, this location is on the $\hat{\jmath}$-axis and experiences no relative motion with respect to the center of the CWH frame at the target satellite's initial location. From the perspective of trajectory planning, the pair of satellites is now maneuvering in an imaginary CWH frame centered where the original frame was; the frame does not follow the target satellite as it is maneuvered by the chaser. Additionally, from Phase 2 to 3, the dynamics change when the target docks with the chaser, and this needs to be incorporated at the correct time in the Ω matrix, by computing a new discretized B_d matrix as discussed in Section 3.1.1.

The goal is to perform all optimization computations fast enough that the process can be performed in real-time onboard a spacecraft. The results section will discuss the running time in more detail, but for the purpose of this discussion it suffices to say that it is fast enough. Given that the multi-stage trajectory optimization is able to be implemented in real-time, the next topic discusses performance improvements that can be achieved upon a specific uncertain event. The uncertain event is where there is a large tracking-error uncovered when entering a new phase due to the higher precision sensors available in that new phase. Because this is topical to linking phases together, the reactive techniques to such an uncertain event will be discussed here.

During Phase 1a, there is an unobservability in the system due to only having angular measurements. This can cause the spacecraft to drift off of the reference-trajectory while thinking that it is tracking perfectly. Fortunately, when the spacecraft gains a range measurement upon entering Phase 1b, the error in the position of the spacecraft can be quickly reduced significantly. Because the spacecraft has gone well off-track, now there will be substantial control effort required to return to the originally planned reference trajectory. The idea here is that the spacecraft can plan a new trajectory onboard after this uncertain event which would return a new optimal trajectory from that state onward. This new trajectory likely will be lower performing when compared to the original trajectory, however is the best path to follow going forward given this large current offset. Figure 4-2 depicts this scenario. This behavior can also happen during other phase transitions, yet will be most

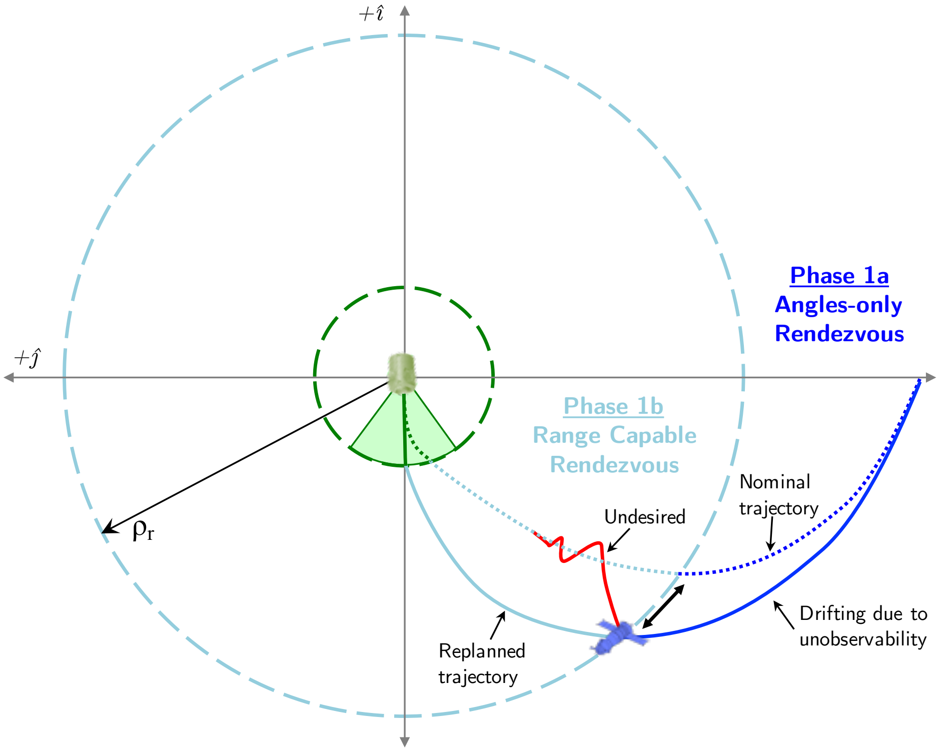


Figure 4-2: Description of the trajectory optimization recovery from the uncertain event of a large error in tracking due to the unobservability in Phase 1a getting reduced when entering Phase 1b

pronounced during the transition from Phase 1a to 1b. This is a reactionary technique to an uncertain event as classified in Section 3.3.2.

4.2.2 Results

The optimal trajectory computation is performed on the mission described in Section 3.2 with the minimum energy method described in Section 4.1.1 and Section 4.2.1. Three methods are compared to show the benefits of optimizing the full mission as opposed to individual phases. Figure 4-3 shows the full mission trajectories for the three methods.

The first method, named the traditional hopping method is similar to the method that the Space Shuttle used to dock to the ISS. A fly around is performed from the V-bar to the R-bar such that the Shuttle would directly align with the ISS docking port in the radial direction. The Shuttle would then perform an approximation of a straight line trajectory to close the distance on the R-bar, because of the orbital dynamics and safety requirements for stopping points along the path, this behavior looks like it is hopping in the approach.

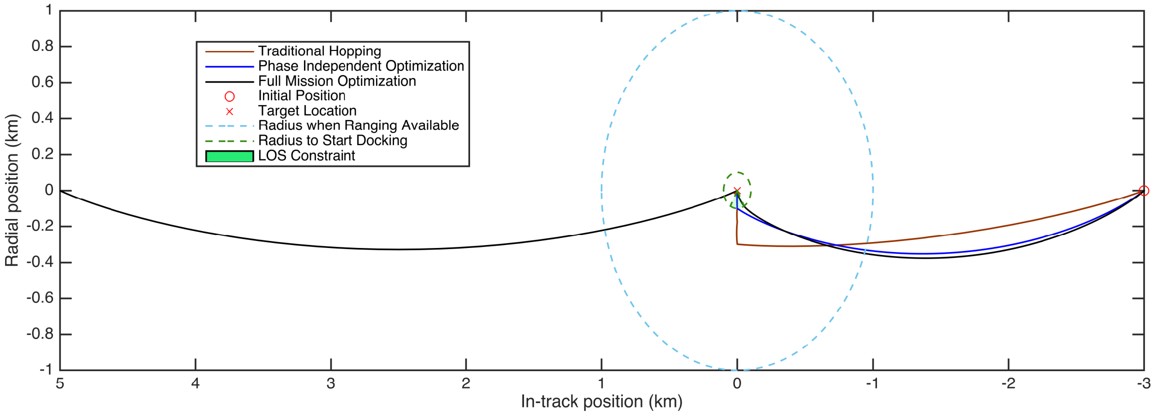


Figure 4-3: Comparison of trajectories generated following a traditional hopping R-bar approach, a phase independent approach and a full mission length optimization approach

Figure 4-5 shows this hopping behavior in more detail in Phases 1b and 2. The second approach shown is an optimization of Phase 1 and Phase 2 individually and is called phase-independent optimization. When optimizing phases separately and there is no knowledge of where along the phase transition line to control nor what velocity to control during the transition, the engineer must choose a position and velocity; normally a centered position and zero velocity are chosen. The third approach, and the approach posited by this thesis is to perform the trajectory planning through the full mission and thus avoid these transitioning problems associated with the previous two methods. With this method, constraints

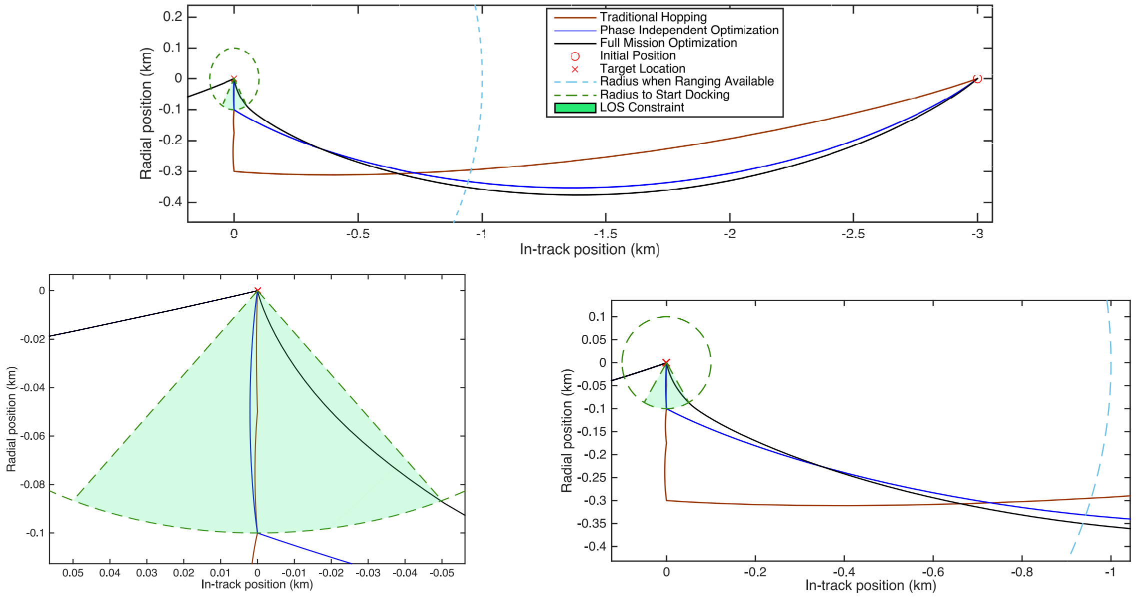


Figure 4-4: Comparison of trajectories generated following a traditional hopping R-bar approach, a phase independent approach and a full mission length optimization approach; shown phase by phase

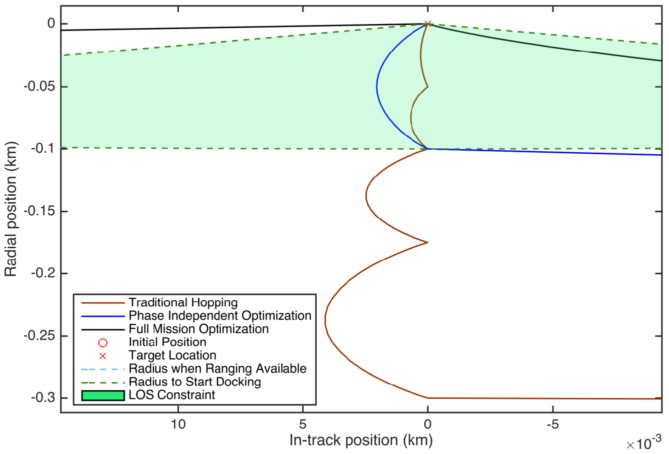


Figure 4-5: More detailed view of the hopping approach

are only enforced while the spacecraft is in a particular phase, and thus the full problem can be considered. Each of the trajectory optimization methods exhibits the same planning for the joint maneuver phase, as the transition is fixed in time and space. A more detailed, phase by phase comparison of the methods is described in Figure 4-4.

The performance for the three methods can be compared in terms of fuel consumption along the trajectory and computation time required to solve. The time to mission completion is set to be constant with all three methods. The hopping method was designed for the LEO docking application of Shuttle operations, while this mission is a GEO docking in 4 hours. Because of these constraints, the hopping trajectory, which would naturally try to exploit the orbital dynamics and use less fuel but take longer to dock, does not perform nearly as close in fuel consumption and it would not be a fair comparison. It is mainly shown here as a comparison to historical docking methods.

The full-mission optimization method performs the mission with about 15% less fuel required than the phase-independent mission, however uses about an order of magnitude more computation time as the problem becomes more complex to solve. Since Phase 3 results are identical across all techniques, we will ignore those results and focus on the propellant consumed in Phase 1 and 2. The phase-independent optimization uses 1.326 m/s while the full-mission optimization uses 1.125 m/s. The true objective for this optimization problem, the LQR cost, also shows markedly better results with full-mission optimization. The full-mission optimization minimum energy was 45% less than that for the phase-independent optimization. When running in MATLAB on a late-2013 MacBook Pro with a 2 GHz Intel Core i7 processor and 16 GB of 1600 MHz DDR3 memory, the average solution time for the phase-independent optimization is about 0.8 seconds and the average solution time for the full-mission optimization is about 5.5 seconds. Nonetheless, the average computation time is still on the order of a few seconds for this problem when considering a discretization of 150 total mission steps and implementing the iterative search for the optimal phase transition time with a quadratic solver. This time scale is very feasible for implementation on orbit, where the entire mission would last a few hours.

The reason the full mission optimization technique uses less fuel is because it does not require the pit stop when entering Phase 2 and instead cuts the corner of the LOS region. Note that the result will not always be that the corner is cut like this and depends on the initial location and relative velocity of the chaser spacecraft. As the spacecraft moves closer to the target in the V-bar direction, the optimal trajectory tends to move away from the corner. Thus, it is difficult to predict an exact transition point and time ahead of the optimization process. The full-mission optimization is providing utility here by finding this optimal transition point and time that could not be determined simply otherwise.

As an important side note, this analysis was performed with no uncertainty considered, so the corner cutting by the full mission technique in Figure 4-4 would in reality be very risky. The target attitude may be slightly different than expected causing the LOS zone of the sensor to be shifted away from our initial conception. As discussed in Section 4.4.4, this uncertainty will be incorporated into the optimization such that appropriate margins are maintained when skirting constraint boundaries. With the appropriate risk bound, performance still improves (albeit slightly less) with comparison to the phase-independent optimization.

Finally, to wrap up the results discussion for multi-stage trajectory optimization, the reactive trajectory planning to uncertain events will be discussed. Because we have proven that the trajectory optimization process only takes a few seconds and is feasible to run onboard a spacecraft, we have opened up the possibility that trajectories can be replanned when more information about the system is obtained. A specific point where this can happen is at the transition from Phase 1a to 1b as depicted and discussed around Figure 4-2. Given that a large error is experienced upon a state estimator convergence on the real spacecraft position in Phase 1b, we can run this trajectory optimization process from this point in the orbit rather than from the initial position. This involves very little modification of the original problem, only a change in initial position and appropriate reduction in length of the horizon depending on the current time.

To analyze this problem, we must simulate a spacecraft following one of these computed trajectories. Using the simulation environment described in Section 3.4, the nominal behavior can be seen in Figure 4-6b, where the spacecraft is acting on the estimated position (red line) and trying to track the trajectory (black line). The truth position of the satellite (blue line) drifts from the estimated position during Phase 1a, as the covariance in the angles-only measurement error is too high and there is an unobservability in the dynamics model. Upon entry into Phase 1b, we can see the estimator snap onto the correct position when a range estimate is gained. For a static trajectory, as in Figure 4-6b, the spacecraft must use significant control effort to return to the trajectory. When replanning is considered in Figure 4-6a, the minimum energy trajectory from the spacecraft's current state is computed as it enters Phase 1b. This reaction allows the resulting trajectory to be smoother and significantly increase overall performance in terms of fuel consumption and LQR cost.

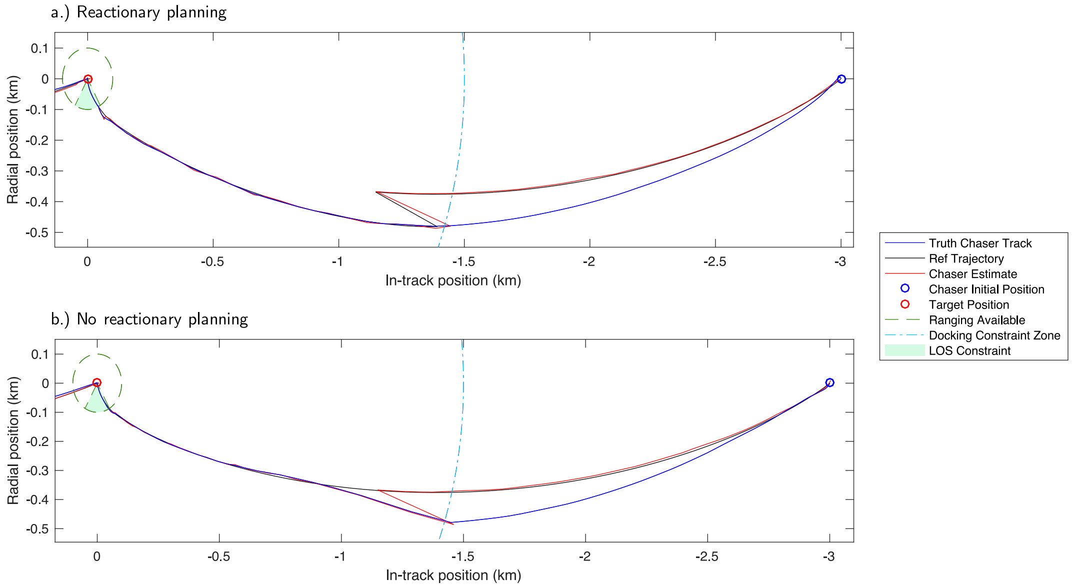


Figure 4-6: Example results from simulation of a reactionary trajectory replanning approach (a) compared to no replanning (b) to account for the uncertain event of entering Phase 1b with a large error

To accentuate the error in the tracking performance in Figure 4-6, an additional measure was taken to reduce the thruster force level to 80% of nominal. With high covariance measurements in Phase 1a, the false dynamics model is trusted more, and the resulting tracking error is even more pronounced than in the case with only the unobservability present. Additionally, the distance at which range measurements are available is increased to 1.5 km as opposed to the 1 km used in the baseline mission design described in Section 3.2.8. In this case however, we can see an average improvement over ten trials of approximately 52% in fuel consumption when implementing the replanning compared to no replanning. This example highlights the significant size of the possible benefits of replanning the trajectory given large uncovered errors. The cost of correcting for this error immediately is very expensive, and the smoother trajectories allow the spacecraft to use its current velocity to its advantage rather than trying to correct for it unnecessarily. Other missions and instantiations of this mission can also see benefits from this technique, although this example highlights the benefits particularly well.

Hopefully this section gives motivation to the importance of phase transitions in multistage rendezvous and docking operations. Two specific methods have been implemented to successfully improve performance across these phase boundaries: full-mission optimization and reactionary planning. The remainder of the thesis will use the techniques developed and discussed here to plan trajectories and react to some uncertain events.

4.3 Obstacle Avoidance Techniques

Until this point in the chapter, trajectory planning has only been performed on a nominally executed mission from start to end. This section deals with the possibility that a piece of orbital debris or another satellite is predicted to move in the way of the previously optimal mission trajectory. This obstacle is likely being tracked by ground assets, such that the chaser spacecraft has knowledge of the obstacle's current state estimate and knowledge of the uncertainty present in that state estimate. The goal is of course to avoid collision with the obstacle for the minimum possible cost and still complete the mission. For consistency, the same cost functions and constraints discussed in Section 4.1 are used here, the only difference is that an obstacle avoidance constraint is added to the problem. This section will discuss and compare two methods to structure the collision avoidance constraints: a nonlinear ellipsoid constraint and a linear, rotating hyperplane constraint.

4.3.1 Ellipsoid Method

The first constraint explored here will be the ellipsoid constraint, which bounds the obstacle's position by a 3D ellipsoidal keep out zone. Thus, if the planned trajectory intersects this ellipsoid, the constraint will be violated. There are two applications in which this constraint could serve useful. One is when the obstacle to avoid is large and has a very little uncertainty in its state estimate, as shown in Figure 4-7a. Here, the main concern is to avoid the extent of the object's structure as motivated by avoiding the ISS, so the bounding ellipsoid circumscribes the physical size of the obstacle with a predetermined amount of safety margin. The other is when there is high uncertainty in the obstacle's state estimate or when the obstacle is small, as shown in Figure 4-7b. Uncertainty is almost always quantified in terms of Gaussian noise and associated covariance when it comes to state estimation in aerospace engineering. The main driver of choosing an ellipsoid in the large uncertainty case is that this commonly available covariance matrix actually represents an uncertainty ellipsoid. Therefore, this covariance ellipsoid can be used directly as the obstacle to avoid and the size of the ellipsoid directly relates to the probability the true obstacle location is inside of it. In Figure 4-7b, the three-sigma ellipsoid is illustrated, which corresponds to a roughly 97% chance the obstacle is contained within. This formulation is thus quite convenient for analysis and constraint design.

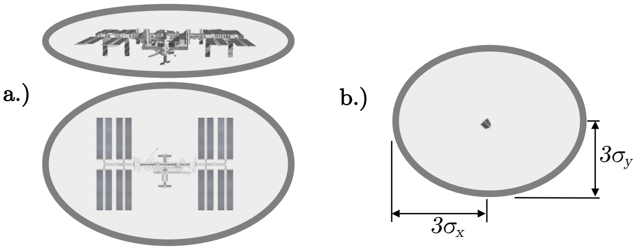


Figure 4-7: Ellipsoidal obstacle bound in cases of (a) low and (b) high uncertainty. Shown in 2DOF here although it extends to 3DOF.

When considering the bounding of spacecraft structure by ellipsoid keep-out-zones, an even higher fidelity modeling of structures can be completed through the use of multiple ellipsoids approximating components of a structure or mapping of structures using 3D SLAM techniques as in [103]. In any case, the scenarios investigated in this thesis are more concerned with very high levels of uncertainty where the precise modeling of the spacecraft structures is not required.

There is a helpful property that allows easy evaluation of whether a point lies inside or outside an ellipsoid. The quadratic inequality,

$$(\mathbf{x}_{pos} - \mathbf{x}_e)^T \mathbf{P} (\mathbf{x}_{pos} - \mathbf{x}_e) \geq 1 \ , \tag{4.29}$$

where $\mathbf{x}_{pos}$ is the position of the point in question, $\mathbf{x}_e$ is the position of the centroid of the ellipsoid, and $\mathbf{P}$ is the shape matrix of the ellipsoid, is true when the point lies outside of the ellipsoid and false when the point lies within the ellipsoid. For the covariance formulation, $\mathbf{P}$ is the inverse of covariance matrix multiplied by n^2 for an n-sigma ellipsoid, $\left(n^2\Sigma\right)^{-1}$. During trajectory optimization, this constraint needs to be evaluated against all points in the horizon (i.e., every point along the planned path cannot intersect the ellipsoid). The inequality in (4.29) is therefore reformulated to compute a vector of constraint evaluations for each point in the horizon. Formatting this constraint in the required structure follows a similar procedure to that discussed in Section 4.1.3. First, we convert the optimization variables $\mathbf{U}$ into states $\mathbf{X}$ through (4.3), and then loop through each step of the horizon, extracting $\mathbf{x}_{pos}$ and setting

$$\mathbf{f}_{ineq}(\mathbf{U}) = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{N\times 1} - \mathit{diag}\left\{\boldsymbol{\mathcal{X}}^T \mathbf{P} \boldsymbol{\mathcal{X}}\right\} \leq \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}_{N\times 1} \quad , \tag{4.30}$$

where $\boldsymbol{\mathcal{X}} = \begin{bmatrix} \mathbf{x}_{pos}(1) - \mathbf{x}_e, & \cdots, & \mathbf{x}_{pos}(i) - \mathbf{x}_e, & \cdots, & \mathbf{x}_{pos}(N) - \mathbf{x}_e \end{bmatrix}$, a $3\times N$ matrix; $\mathit{diag}\{\cdot\}$ places the diagonal elements of the square matrix into a column vector; and $\mathbf{x}_{pos}(i)$ are the spacecraft position states (x, y, and z) at step i of the N-step horizon.

For multiple obstacles, the nonlinear constraint can be repeated for each bounding ellipsoid and the corresponding vectors can be stacked into one large vector to represent the nonlinear inequality constraint. For moving obstacles, the centroid of the ellipsoid can be propagated according to the CWH equations and each step of the horizon can be compared to the expected location of the obstacle according to its propagation based on a given initial position and velocity. Again, the structure of this propagation (assuming no control input by the obstacle) is

$$\mathbf{X}_{obst} = \begin{bmatrix} \mathbf{x}_{obst}(1) \\ \vdots \\ \mathbf{x}_{obst}(N) \end{bmatrix}_{6N \times 1} = \Psi \mathbf{x}_{obst}(0), \tag{4.31}$$

where $\mathbf{x}_{obst}(i)$ is the full state of the obstacle at the i-th step of the N-step horizon. Noting that $\mathbf{x}_e(i)$ is the extracted position state for the i-th time step, the constraint in (4.30) can be used except this time the only change is that

$$\boldsymbol{\mathcal{X}} = \Big[\mathbf{x}_{pos}(1) - \mathbf{x}_e(1), \quad \cdots, \quad \mathbf{x}_{pos}(i) - \mathbf{x}_e(i), \quad \cdots, \quad \mathbf{x}_{pos}(N) - \mathbf{x}_e(N)\Big]. \tag{4.32}$$

Finally, if desired, one can propagate the covariance in the obstacle's estimated velocity to the size of the obstacle ellipsoid to simulate the growth of the uncertainty of this ellipsoid. This additional consideration is not difficult to implement as it only involves using a different $\mathbf{P}$ for each step of the trajectory and slightly reformulating (4.30) to account for this. The results in this thesis assume a small velocity uncertainty, as it is fairly common to have much smaller error in velocity than position. Thus, the obstacle ellipsoid will not grow significantly over the fairly short mission durations discussed in this thesis, and a static size assumption is made. Unless the velocity uncertainty is large, this assumption should not affect results or applicability of the methods discussed here.

Of course, all of these formulations of the ellipsoid obstacle constraint are nonlinear, and thus nominal quadratic and linear solvers are unable to solve the problem. Instead, an SQP approach is used as described in Section 4.1.4.

At this point, it should be noted that the nonlinear ellipsoidal constraint is non-convex. This characteristic is not desired in optimization problems for a variety of reasons. For path-planning problems, the largest issue is that successive points in the path can effectively "jump" obstacles without technically violating the constraint as seen in Figure 4-8. If the problem is well posed, and the distance between horizon points is smaller than the minimum characteristic length of the obstacle, this will not affect the feasibility of the solution. However, if the problem is ill-posed, action needs to be taken to avoid returning mathematically feasible, but practically infeasible solutions.

Along this line, as a mitigation step, an interpolation scheme was developed. This scheme discretizes the dynamics of the system at a finer sampling time resolution than

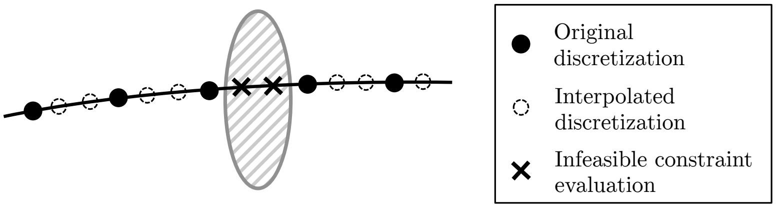


Figure 4-8: Illustration of the non-convexity of the ellipsoid constraint and a finer resolution discretization

given for the trajectory optimization horizon. These finer discretized A and B matrices can be used in the formulation of new Ω and Ψ matrices in the constraint propagation only. The objective functions will not be executed at this fine resolution, because the control input cannot be applied at every one of these smaller segments. This requirement makes Ω, the stacked, block propagation matrix defined in (4.5), invalid. Note that Ψ remains of the same form defined in (4.4). For a simplified case with two interpolation points, three steps in the horizon, and the continuous form of the control input the new Ψ_{dsc} is

$$\Psi_{dsc} = \begin{bmatrix} A_{dsc} \\ A_{dsc}^2 \\ \vdots \\ A_{dsc}^9 \end{bmatrix}, \tag{4.33}$$

and the modified Ω_{dsc} is

$$\Omega_{dsc} = \begin{bmatrix} B_{dsc} & 0_{6\times3} & 0_{6\times3} \\ A_{dsc}B_{dsc} + B_{dsc} & 0_{6\times3} & 0_{6\times3} \\ A_{dsc}^2 B_{dsc} + A_{dsc}B_{dsc} + B_{dsc} & 0_{6\times3} & 0_{6\times3} \\ A_{dsc}^3 B_{dsc} + A_{dsc}^2 B_{dsc} + A_{dsc}B_{dsc} & B_{dsc} & 0_{6\times3} \\ A_{dsc}^4 B_{dsc} + A_{dsc}^3 B_{dsc} + A_{dsc}^2 B_{dsc} & A_{dsc}B_{dsc} + B_{dsc} & 0_{6\times3} \\ A_{dsc}^5 B_{dsc} + A_{dsc}^4 B_{dsc} + A_{dsc}^3 B_{dsc} & A_{dsc}^2 B_{dsc} + A_{dsc}B_{dsc} + B_{dsc} & 0_{6\times3} \\ A_{dsc}^6 B_{dsc} + A_{dsc}^5 B_{dsc} + A_{dsc}^4 B_{dsc} & A_{dsc}^3 B_{dsc} + A_{dsc}^2 B_{dsc} + A_{dsc}B_{dsc} & B_{dsc} \\ A_{dsc}^7 B_{dsc} + A_{dsc}^6 B_{dsc} + A_{dsc}^5 B_{dsc} & A_{dsc}^4 B_{dsc} + A_{dsc}^3 B_{dsc} + A_{dsc}^2 B_{dsc} & A_{dsc}B_{dsc} + B_{dsc} \\ A_{dsc}^8 B_{dsc} + A_{dsc}^7 B_{dsc} + A_{dsc}^6 B_{dsc} & A_{dsc}^5 B_{dsc} + A_{dsc}^4 B_{dsc} + A_{dsc}^3 B_{dsc} & A_{dsc}^2 B_{dsc} + A_{dsc}B_{dsc} + B_{dsc} \end{bmatrix}, \tag{4.34}$$

where A_{dsc} and B_{dsc} are the finer discretized matrices with discretization time $T_{dsc} = 1/3T$. These increased discretization matrices would then be used in the state propagation of the nonlinear constraint functions. The form is slightly different for the ΔV control approach, as

$$\Omega_{dsc} = \begin{bmatrix} B_{dsc} & 0_{6\times3} & 0_{6\times3} \\ A_{dsc}B_{dsc} & 0_{6\times3} & 0_{6\times3} \\ A^2_{dsc}B_{dsc} & 0_{6\times3} & 0_{6\times3} \\ A^3_{dsc}B_{dsc} & B_{dsc} & 0_{6\times3} \\ A^4_{dsc}B_{dsc} & A_{dsc}B_{dsc} & 0_{6\times3} \\ A^5_{dsc}B_{dsc} & A^2_{dsc}B_{dsc} & 0_{6\times3} \\ A^6_{dsc}B_{dsc} & A^3_{dsc}B_{dsc} & B_{dsc} \\ A^7_{dsc}B_{dsc} & A^4_{dsc}B_{dsc} & A_{dsc}B_{dsc} \\ A^8_{dsc}B_{dsc} & A^5_{dsc}B_{dsc} & A^2_{dsc}B_{dsc} \end{bmatrix}, \tag{4.35}$$

because the control is implemented instantaneously rather than continuously through each discretized sub-step. For the general case, it is difficult to fit on a single page, but the examples shown should be enough to infer the pattern and extrapolate into the desired number of interpolation points and length of horizon. In similar form to (4.3), remember we have

$$\mathbf{X}_{dsc} = \Psi_{dsc}\mathbf{x}(0) + \Omega_{dsc}\mathbf{U}, \tag{4.36}$$

for spacecraft state propagation and

$$\mathbf{X}_{obst_{dsc}} = \Psi_{dsc}\mathbf{x}_{obst}(0), \tag{4.37}$$

for obstacle state propagation.

Usually, the choice of how many constraint interpolation points to use can be decided *a priori*, based on expected velocities and a desirable minimum distance between points. Alternatively, the level of discretization can be adjusted in real time based on the velocity at that specific point in the trajectory to ensure that the distance between points of constraint evaluation is always below a certain characteristic width of the obstacle.

Additionally, although not as critical, non-convexity can prevent a nonlinear solver from finding an optimal solution, even if the solution is feasible. Because of the non-convex nature of the obstacle, there becomes multiple paths that the spacecraft can take around the obstacle. The choice of which side to take can cause local minimums to appear in the problem and may prevent the solver from finding the globally optimum solution. As this does not affect the feasibility of the solution, it is not as large of a problem, although does prohibit any immediate claims that the true global optimum has been found outright. In this case, different initial solutions can be given to the solver and the optimization problem can be solved multiple times with sufficient coverage that the optimal solution is confidently found. This process becomes more difficult when multiple obstacles are considered because the number of initial guesses will increase unboundedly with the number of obstacles and defining such initial solutions becomes very challenging. For the purpose of this thesis, the number of obstacles is kept small such that the number of initial guesses is manageable. Other search methods, such as RRT* and dynamic programming, are better able to handle these situations with correspondingly increased computational power.

Convexity will become a problem when obstacles are significantly small or fast or when the chaser is moving fast. In these cases, the ellipsoid constraint method may not be able to find a solution and it is suggested to use the convex hyperplane method described in the next section to find a feasible solution for use as an initial guess. Note that besides convexity issues, there still is the possibility that the obstacle has made the problem infeasible within the time constraints of the problem. In this case, the mission objective must change to either use more time to finish the mission or to back off and start the mission over at a later time. The design of this abort procedure is not discussed in this thesis. In the worst-case scenario, if the obstacle is large enough or fast enough, it may be impossible for the spacecraft to move out of the way at all due to control saturation, and therefore collision may be unavoidable by any method of trajectory design.

4.3.2 Rotating Hyperplane Method

Due to the nonlinear and non-convex form of the ellipsoid constraint presented in the previous section, a convex and linear obstacle avoidance constraint is desired. This proposed linear constraint would speed up the optimization solver for use in real-time or in Monte Carlo simulation, but also would guarantee feasible results due to its convex nature. However, approximating the ellipsoid constraint will lead to some loss in accuracy from the intended constraint and thus inefficiencies in the optimal solutions to this new problem.

The rotating hyperplane method is the proposed approximation to the ellipsoid constraint. At a high level, this method involves using half-space constraints that lay tangent to the ellipsoid of interest and rotate as the spacecraft proceeds through the mission. Thus, as the spacecraft approaches the obstacle, it is constrained by the half-plane not to enter the ellipsoid individually at each time step. Initially, even the goal state is in the infeasible region, but as the hyperplane rotates, the goal state comes into view. Figure 4-9 illustrates this behavior. Additional details and the inspiration for this method can also be found in [70] and [73].

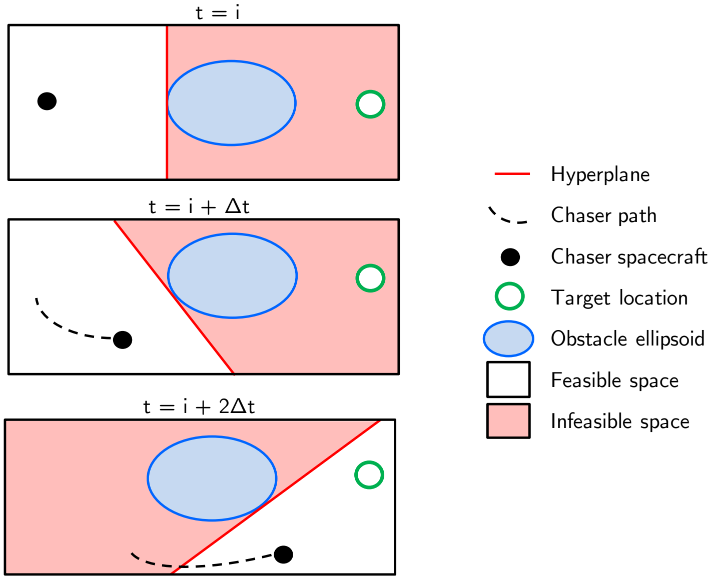


Figure 4-9: Illustration of the rotating hyperplane constraint through three time steps during the mission

The approach from [70] and [73] is slightly different from what is implemented in this thesis. These papers assume a stationary obstacle, where this thesis uses a moving obstacle formulation. Additionally, they rotate the hyperplane throughout the entire horizon, which creates inefficiencies, because it causes the spacecraft to be pushed around by false hyperplane constraints due to too slow or too fast of an arbitrarily set rotation speed.

The procedure for creating the linear, rotating hyperplane constraint is as follows, assuming we know the obstacle's position, $\mathbf{X}_{obst}$, for all time steps throughout the mission and we know the initial spacecraft location and final goal location. First, we attempt to optimize the problem without the obstacle constraint and evaluate the ellipsoid constraint from Section 4.3.1 for every time step in the horizon. If the trajectory does intersect the ellipsoid, we find the time step at which the constraint is violated by the largest magnitude and call this the collision time. If it does not intersect, we ignore the obstacle constraint and use this unconstrained solution.

Differing from the previous literature, to determine at which time steps we wish the hyperplane constraint to be rotated, we take the ratio of the largest dimension of the ellipsoid to the straight-line distance required for the spacecraft to reach the goal. We double this and multiply by the duration of the full horizon such that the hyperplane will be rotated roughly for double the time the spacecraft would spend passing the obstacle. From the collision time, we center the rotation window and apply the rotation of the constraint only within this window, scaling the speed of rotation to span the full angle required. Outside the rotation window, we use the adjacent beginning and final hyperplane constraints as a non-rotating hyperplane.

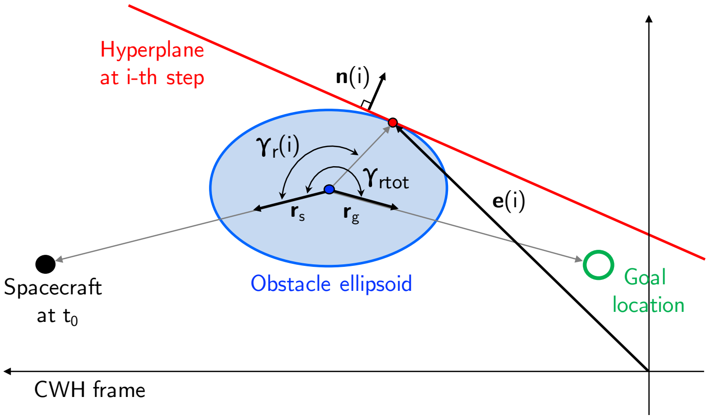


Figure 4-10: Diagram of the vectors used to formulate the hyperplane constraints

From the position of the obstacle at the collision time, we compute the unit vectors from the obstacle ellipsoid center to the goal, $\mathbf{r}_g$, and to the spacecraft, $\mathbf{r}_s$, through vector addition and normalization. Figure 4-10 shows each of these and the following vectors in a 2DOF diagram. The cross product of these vectors, $\mathbf{r}_g \times \mathbf{r}_s$, produces the axis of rotation for the hyperplane, $\mathbf{a}_r$, and the angle between the two vectors, $\gamma_{r_{tot}}$, tells us the total amount of rotation we expect the hyperplane to make. We can then determine an angle of rotation, $\gamma_r(i)$, from the vector pointing toward the spacecraft for each time step, i, in the rotation window. With this angle of rotation at each time step we can compute the point, $\boldsymbol{e}(i)$, at which the rotated $\mathbf{r}_s$ vector intersects the ellipsoid surface and can compute the surface normal of the ellipsoid at this point, $\boldsymbol{n}(i)$. Since the obstacle is moving between each time step, the $\boldsymbol{e}(i)$ vector must be updated using the current center of the ellipsoid, $\mathbf{x}_e(i)$. For the entire mission horizon, we use $\boldsymbol{n}(i)$ and $\boldsymbol{e}(i)$ to form the linear constraints for the minimum energy problem as

$$-\mathcal{H}\Omega\mathbf{U} \leq -\mathcal{H}\mathcal{E} + \mathcal{H}\Psi\mathbf{x}(0), \tag{4.38}$$

where the $\mathcal{H}$ and $\mathcal{E}$ matrices are defined as

$$\mathcal{H} = \begin{bmatrix} \boldsymbol{n}(1)^T & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & \dots & 0_{1\times3} & 0_{1\times3} \\ 0_{1\times3} & 0_{1\times3} & \boldsymbol{n}(2)^T & 0_{1\times3} & \dots & 0_{1\times3} & 0_{1\times3} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & \dots & \boldsymbol{n}(N)^T & 0_{1\times3} \end{bmatrix}_{N\times6N} \quad \text{and} \tag{4.39}$$

$$\mathcal{E} = \begin{bmatrix} \boldsymbol{e}(1) \\ 0_{3\times1} \\ \boldsymbol{e}(2) \\ 0_{3\times1} \\ \vdots \\ \boldsymbol{e}(N) \\ 0_{3\times1} \end{bmatrix}_{6N\times1} , \tag{4.40}$$

where $\boldsymbol{n}(i)$ is the surface normal of the hyperplane corresponding to the ellipsoid's surface normal at point $\boldsymbol{e}(i)$ on the surface of the ellipsoid, each for the i-th step of the N-step horizon. Remember that if k is the first step of the rotation window and j is the last step,

$$\begin{gathered} \Big\{\boldsymbol{n}(1),\ \dots\ ,\ \boldsymbol{n}(k-1)\Big\} = \boldsymbol{n}(k), \\ \Big\{\boldsymbol{e}(1),\ \dots\ ,\ \boldsymbol{e}(k-1)\Big\} = \Big\{\boldsymbol{e}(k) - \mathbf{x}_e(k) + \mathbf{x}_e(1),\ \dots\ ,\ \boldsymbol{e}(k) - \mathbf{x}_e(k) + \mathbf{x}_e(k-1)\Big\}, \\ \Big\{\boldsymbol{n}(j+1),\ \dots\ ,\ \boldsymbol{n}(N)\Big\} = \boldsymbol{n}(j), \text{ and} \\ \Big\{\boldsymbol{e}(j+1),\ \dots\ ,\ \boldsymbol{e}(N)\Big\} = \Big\{\boldsymbol{e}(j) - \mathbf{x}_e(j) + \mathbf{x}_e(j+1),\ \dots\ ,\ \boldsymbol{e}(j) - \mathbf{x}_e(j) + \mathbf{x}_e(N)\Big\}. \end{gathered} \tag{4.41}$$

This enforces the fact that the hyperplane only rotates during the rotation window from k to j, and stays stationary other than sticking to the edge of the obstacle as it moves.

Finally, for the minimum fuel problem, we have that

$$\begin{bmatrix} -\mathcal{H}\Omega & \mathcal{H}\Omega \end{bmatrix} \begin{bmatrix} \mathbf{U}^+ \\ \mathbf{U}^- \end{bmatrix} \leq -\mathcal{H}\mathcal{E} + \mathcal{H}\Psi\mathbf{x}(0). \tag{4.42}$$

When choosing the angle, $\gamma_{r_{tot}}$, we inherently choose one direction for the constraint to rotate. Therefore, we are not entirely sure if we are circumnavigating the obstacle in the correct direction. This is again due to the non-convex nature of the original constraint.

To alleviate any concerns that the spacecraft will go around the obstacle in the incorrect direction, we run the optimization once again, this time choosing to rotate the hyperplane in the opposite direction. In 2DOF, this ensures we cover both ways the spacecraft could avoid the obstacle, but for the 3DOF case there could potentially be other local minima, such as above and below the obstacle. In 3DOF, we can run the optimization for several rotation axes until the ellipsoid is well covered. For multiple obstacles, this becomes a fairly complex combinatorial optimization problem such that each direction of each hyperplane rotation must be chosen correctly to find the true optimal solution. The multiple obstacle case is not explored in this thesis while using the hyperplane approach for this reason, although a brute force method could be implemented using the techniques here and going through all combinations of the hyperplane rotation directions.

An additional note of caution is that this method was derived to heuristically work with obstacle of the size, location and speed considered in this thesis. The method is sensitive to the relative size of the rotation window and the time during the mission at which the obstacle will intersect the nominal trajectory. The method of choosing parameters as discussed in this section seemed to work for the widest range of possible obstacles, however it is likely that the specific parameter choices here will be fragile in the general case. Nonetheless, it works very well for this problem and produces great results for problems similar to those seen in this thesis.

4.3.3 Results

The obstacle avoidance techniques discussed in the previous two sections are implemented in the MATLAB simulation discussed in Section 3.4. This section will present results and compare the performance of the methods for different obstacle avoidance problems using the minimum energy objective function formulation from Section 4.1.1.

The same optimization setup is used as described in Section 4.1, although nonlinear, non-convex, ellipsoid obstacle path constraints are added in one case and a linear rotating-hyperplane constraint is added in the other two cases. An MPC approach is also investigated in this results section in which the same optimization problem is solved in a receding horizon format such that at every time step the same optimization problem is solved, and the first step of the solution is implemented. The full description of the MPC method is described in [97], however it follows very closely to the methodology presented above, only that it is implemented for MPC. Specifically, this form is used such that comparisons can be made against the linear rotating-hyperplane avoidance technique developed for MPC in [83]. After these comparisons, the method from [83] is modified slightly to work with moving obstacles and in a fixed horizon problem as is desired for this thesis.

First, three examples are examined in which the nonlinear ellipsoid method from this thesis is compared to the linear rotating-hyperplane avoidance technique in [83], both in the MPC form. The first example, in Figure 4-11, is a mission scenario where the optimal non-obstructed path to the target just narrowly intersects the obstacle; the case aims to show what happens to each of the algorithms when the optimal path is only slightly perturbed. The second example, in Figure 4-12, is a mission scenario where the optimal non-obstructed path to the target is completely blocked by a close, large obstacle; the case compares the two methods' reactions when a large change in path is required. The third example, in Figure 4-13 is a mission scenario where the obstacle is very thin in the cross-track direction, such that the penalty for traveling in the cross-track direction would be smaller than the penalty for avoiding the obstacle in-plane.

One can see from Figure 4-11 and Figure 4-12 that the nonlinear ellipsoid method is better at planning a path to narrowly avoid the obstacle, thus conserving fuel and saving time. The rotating hyperplane method from [83] tends to keep a larger buffer between the true obstacle constraint and the spacecraft. The computation time of the ellipsoid method on the other hand is more than an order of magnitude higher that the hyperplane method. Figure 4-13 simply shows that the hyperplane method from [83] does not account for possible

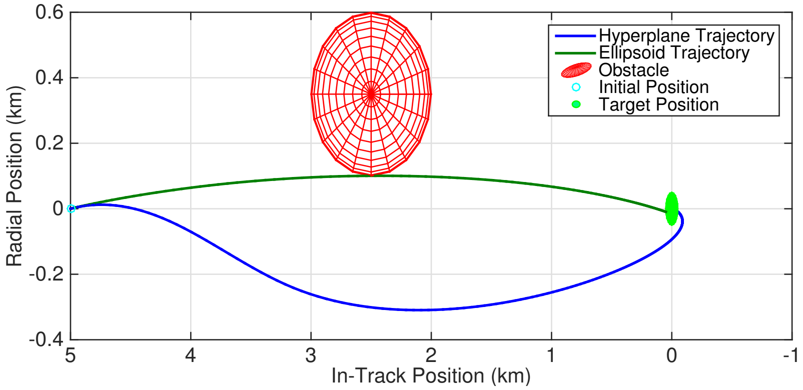


Figure 4-11: Obstacle avoidance trajectory comparison between the MPC formulation with a nonlinear ellipsoid constraint and the hyperplane method from [83] – Obstacle narrowly intersects nominal trajectory

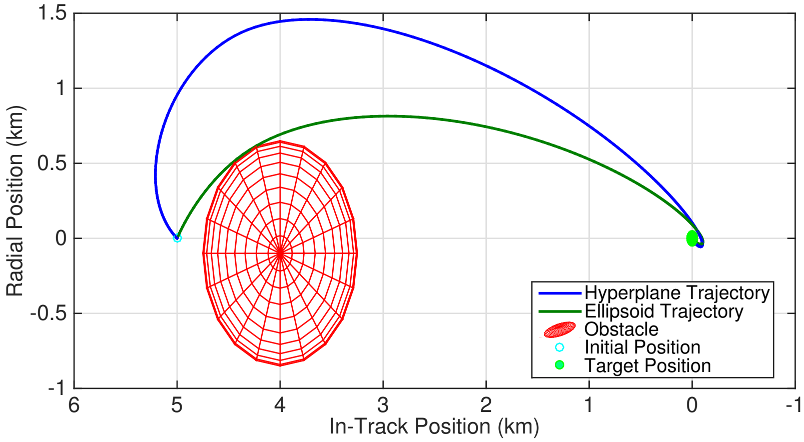


Figure 4-12: Obstacle avoidance trajectory comparison between the MPC formulation with a nonlinear ellipsoid constraint and the hyperplane method from [83] – Obstacle close and directly in path

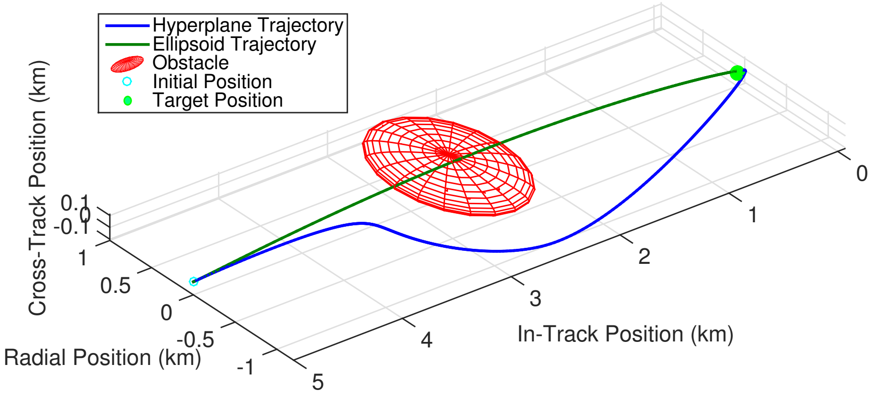


Figure 4-13: Obstacle avoidance trajectory comparison between the MPC formulation with a nonlinear ellipsoid constraint and the hyperplane method from [83] – Obstacle is very thin in the cross-track direction

motion in the cross-track direction, while the nonlinear ellipsoid formulation can account for it. Quantitative comparison can be found in [97], but for this thesis it is only important to show that the hyperplane method tends to "push" the trajectory around the obstacle depending on the arbitrary rotation rate set for the mission, while the nonlinear ellipsoid method avoids this behavior.

The analysis from this point forward is performed on a modified hyperplane method from [83] to maintain consistency with other parts of this thesis. This modified method solves the fixed-horizon, terminally constrained problem as opposed to the open-ended, receding horizon MPC problem for which it was originally developed. Additionally, the improved hyperplane method from this thesis as discussed in Section 4.3.2, which only

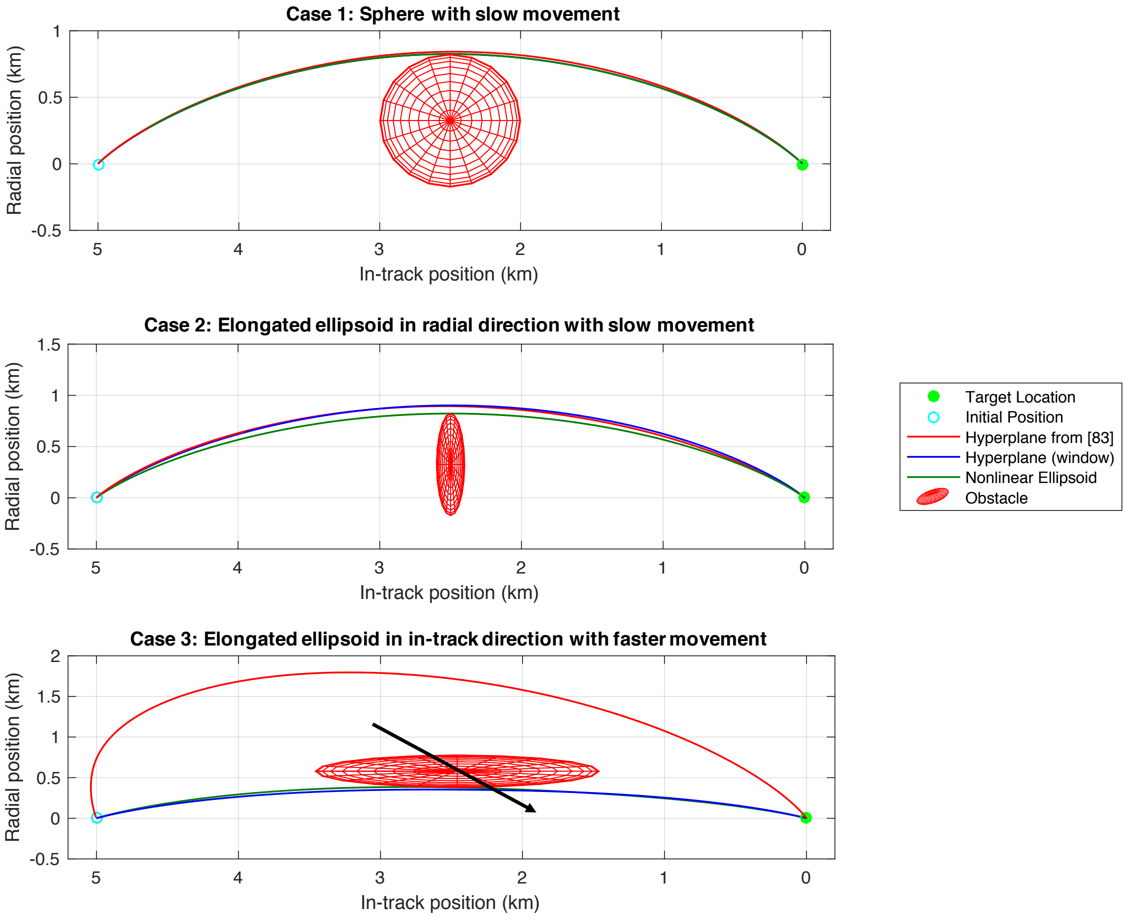


Figure 4-14: Three cases for comparison of the avoidance trajectories computed from each of three methods: the nonlinear ellipsoid and hyperplane window method developed for this thesis and the hyperplane method from [83]

rotates the hyperplane in a specified window around the obstacle instead of for the whole mission timeline is implemented to compare results to the nonlinear ellipsoid method and the previous hyperplane method from the literature. Figure 4-14 compares these three methods for three different cases.

Case 1 is a fairly simple and easy case in which the obstacle is spherical and moving very slowly. The spherical nature of the constraint allows for a constant rotation throughout the mission, while more elliptical constraints will cause the hyperplane to move in translation faster at certain points due to the requirement it is tangent to the ellipsoid. All of the methods perform well in Case 1 as shown in Figure 4-14 and in the data in Table 4.1. This thesis's hyperplane method matches the performance of the nonlinear ellipsoid constraint very well (within 1%) and only uses a tenth of the computation time. The hyperplane method from [83] performs about 20% worse in terms of LQR cost, but matches the other results from this thesis well.

Table 4.1: Comparison of avoidance trajectory optimization results for the two options presented in this thesis and one found in literature for Case 1, a spherical and slow obstacle

	Ellipsoid	Hyperplane (window)	Hyperplane from [83]
LQR Cost	1,787	1,800	2,210
Fuel (m/s)	1.5198	1.5197	1.5195
Computation Time (s)	3.5	0.32	0.31

Case 2 adds an elongation of the obstacle in the radial direction, such that the hyperplane methods will have a more difficult time due to the rotation causing the translational motion of the hyperplane to move very quickly at points. In this example, Figure 4-14 also shows fairly close trajectories, although the data in Table 4.2, shows a slightly different story. Due to this elongation, both hyperplane methods suffer in LQR cost, but still have fuel consumption within 2% and significantly reduce the computation time over an order of magnitude.

Table 4.2: Comparison of avoidance trajectory optimization results for the two options presented in this thesis and one found in literature for Case 2, a radially elongated obstacle

	Ellipsoid	Hyperplane (window)	Hyperplane from [83]
LQR Cost	1,740	3,862	5,682
Fuel (m/s)	1.5200	1.5457	1.5412
Computation Time (s)	3.8	0.26	0.29

Case 3 goes a step further and both elongates the ellipsoid in the in-track direction and adds velocity such that the obstacle is moving faster with respect to the spacecraft. This wreaks havoc on the hyperplane method from [83], but the hyperplane method presented here actually performs very closely to the true optimum from the nonlinear ellipsoid constraint. Figure 4-14 and Table 4.1 show how much the previous hyperplane method suffers from needing to rotate over the full mission timeline. This extended rotation window and the fact that the obstacle travels under the target location before the end of the mission force the trajectory that ducks under the obstacle to be infeasible, and thus the much more costly path of going above the obstacle is required. Since the rotation for the hyperplane in this thesis only occurs in a range around the obstacle instead of for the full timeline, it is able to find the much better and truly feasible path to duck under the obstacle as the nonlinear ellipsoid method found.

Table 4.3: Comparison of avoidance trajectory optimization results for the two options presented in this thesis and one found in literature for Case 3, a fast and in-track elongated obstacle

	Ellipsoid	Hyperplane (window)	Hyperplane from [83]
LQR Cost	5,501	5,978	258,880
Fuel (m/s)	1.6374	1.6388	2.3265
Computation Time (s)	3.9	0.26	0.29

Next, the ellipsoid method is run on a few cases that the hyperplane methods are not. Specifically, these cases are for multiple obstacles and ones requiring significant 3D motion. The nonlinear nature of the constraints allowed while using SQP permit easier development for this algorithm and thus faster advancement to the multiple, moving obstacle case. The following plots in Figures 4-15 through 4-17 show some of the more interesting cases that were run for the ellipsoid method only. The set of plots in Figure 4-16 show frames taken

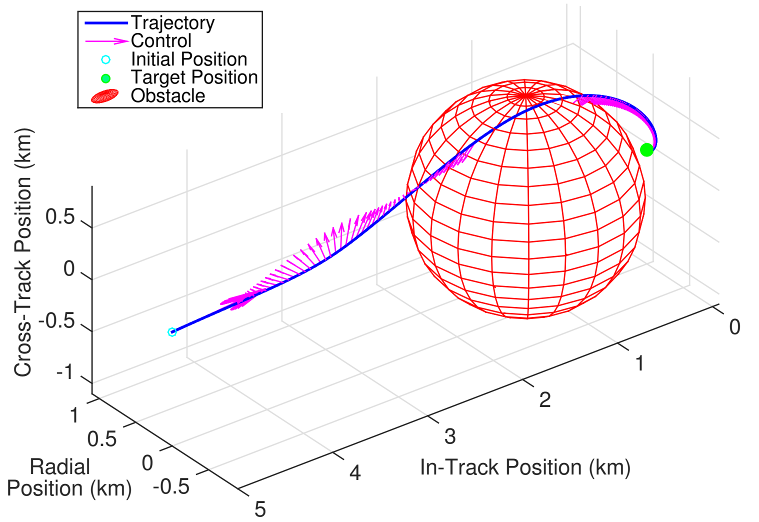


Figure 4-15: The executed trajectory and control in the case of a large obstacle shifted slightly out of plane and positioned close to the target spacecraft

at specific periods in time throughout the trajectory in an attempt to show the movement of the obstacles and the successful avoidance each obstacle.

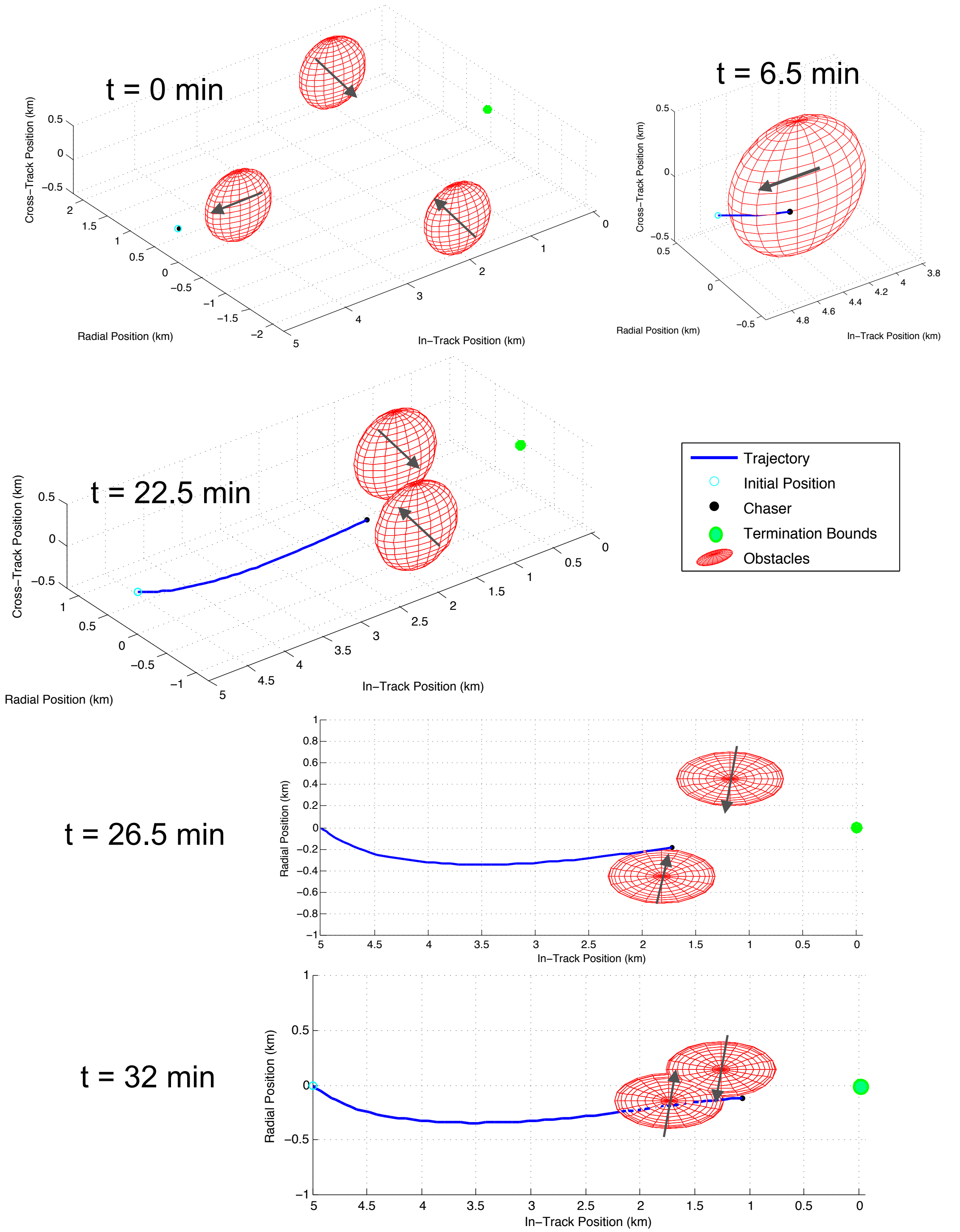


Figure 4-16: The executed trajectory in the case of three moving obstacles, each in the path of the spacecraft, forcing the optimal path between the obstacles. Each plot is shown as a step in time to capture the movement of the obstacles.

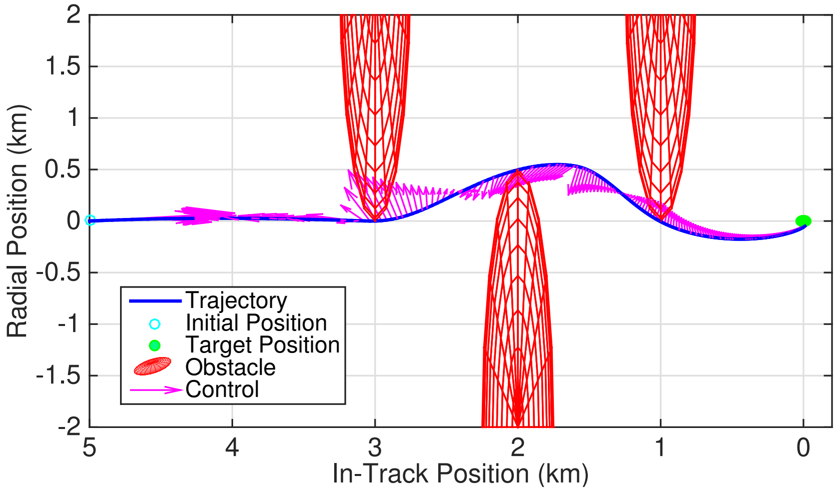


Figure 4-17: The executed trajectory and control in the case of three obstacles, each in the path of the spacecraft, forcing the optimal path between the obstacles

Finally, we have shown that these trajectories can be computed in a time scale that permits real-time usage onboard a spacecraft. These computation times of roughly 4 seconds for the nonlinear ellipsoid method and 0.3 seconds for the hyperplane methods were found using MATLAB on a late-2013 MacBook Pro with a 2 GHz Intel Core i7 processor and 16 GB of 1600 MHz DDR3 memory. 100 steps are used for the horizon length. Since the timescales on orbit are significantly longer than a few seconds, these methods can be used at will throughout the mission. Specifically, they could be implemented in the case of an unknown obstacle appearing on the onboard sensors or from ground tracking while the mission is already in action. This replanning is shown in Figure 4-18, where the obstacle is detected when the spacecraft reaches 500 meters of colliding with it. Regardless, the spacecraft is able to replan the trajectory and avoid the obstacle in time.

The obstacle avoidance problem has and can also be explored for the minimum fuel case; however this analysis is left out of the thesis. In short, the nonlinear optimization solver has trouble solving minimum fuel problems due to the discrete changes in control required, and thus, there would be no comparison available in this respect to validate results. Additionally, the previous literature used for comparison here has only looked at minimum energy forms for these constraints. Overall, the trajectories computed are very similar, however, and a more thorough discussion can be had for the minimum energy case.

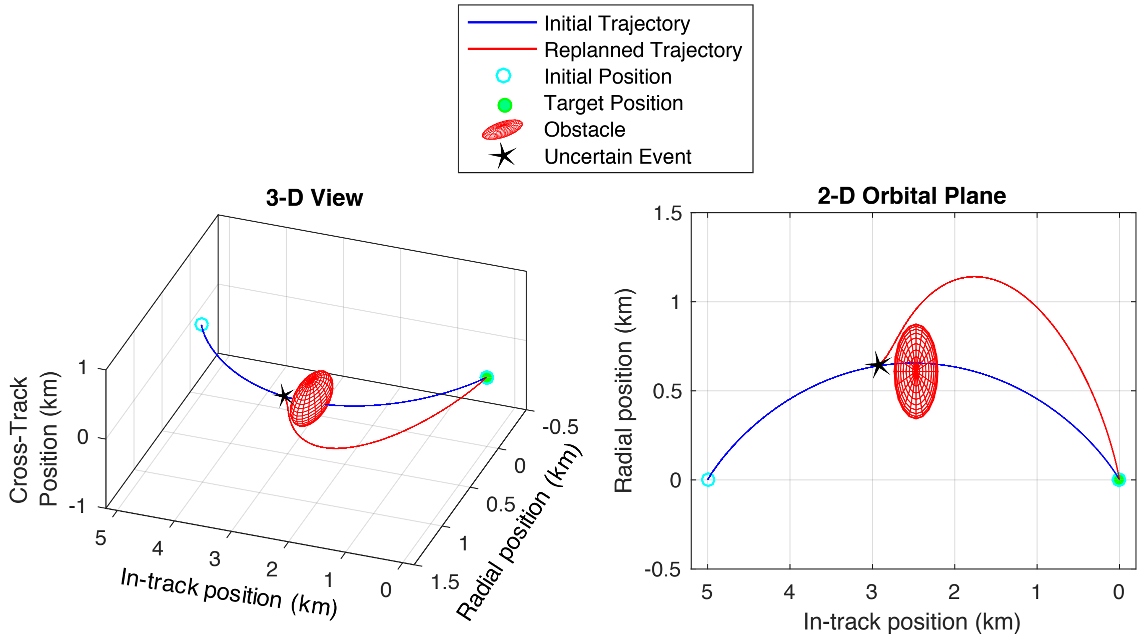


Figure 4-18: Example of replanning a trajectory in real-time when an unknown obstacle is sensed (an uncertain event). Blue line shows the optimal initial trajectory, red shows the optimal avoidance trajectory.

4.4 Probabilistic Trajectory Planning

In previous sections of this chapter, optimization has been performed without considering any uncertainty in the sources of the path constraints. In reality, however, there will never be exact knowledge of the true path constraints as there is always some uncertainty present in the system. This section describes a major contribution of this thesis, which is probabilistically computing optimal trajectories in the presence of these uncertainties. The objective is that these trajectories will perform the best on average over all possible instantiations of error from the known probability distribution of the uncertainty. Section 4.4.1 will motivate this in the context of the rendezvous, docking and joint maneuvering problem. Section 4.4.2 will describe the general process for completing this type of trajectory optimization, while Section 4.4.3 and Section 4.4.4 will detail the process and results from two case studies: uncertain obstacle avoidance and uncertain target attitude. Each of the case studies will implement a different objective function for the trajectory optimization to show that the process works for both the minimum energy and minimum fuel formulations.

4.4.1 Motivation and Hypothesis

Recent trends in spacecraft engineering and deployment have created an increasing level of uncertainty in orbit. Spacecraft are being built smaller (e.g., cubesats and other nanosats), which results in a smaller cross-section for radar and optical sensors to measure and corresponds to a decrease in the accuracy of these measurements at the same distance. This affects both on-orbit sensors and ground-based tracking. Satellites are malfunctioning on orbit, and missions are being designed to service these satellites. Unfortunately, sometimes these malfunctioned satellites are unable to communicate their state estimate to the servicer spacecraft as would normally occur during rendezvous missions. Furthermore, the amount of orbital debris is growing consistently and most of this debris does not have active sensors communicating state information. Thus, there is the potential for highly uncertain states to be involved in planning for missions in the future. This uncertainty can sometimes translate directly to the path constraints in a trajectory optimization problem if the mission involves maneuvering in the proximity of other objects.

Specifically concerning the path constraints explored in this thesis, obstacle constraints have obvious dependence on the uncertainty of the obstacles state. Additionally, the position and attitude of the target spacecraft can affect the LOS constraint as discussed in Section 3.3 and Section 4.2.2. As the attitude or position of the target change, its own sensors' Field Of View (FOV) would rotate and move with the target. Or, if fiducial markings on the target are being used by the chaser, the LOS region where the chaser could see these markings would also rotate and move with the target. Furthermore, if the object is passive, there may be no estimate of its attitude or, most optimistically, there will be a very uncertain estimate.

Attempting to quantify the size of possible uncertainties in position tracking of orbital debris or other passive obstacles, we can look to the tracking of cubesats in LEO. Cubesats are often tracked by the Joint Space Operations Center (JSpOC) similarly to how orbital debris is tracked using a Two-Line-Element (TLE) method. Unfortunately, with objects the size of cubesats, there can be significant errors in in-track, cross-track and radial directions on the order of a couple kilometers to hundreds of kilometers [203]. Velocity is usually known much better. Considering that the extremely large errors from the analysis in [203] may only be seen in certain situations, we will assume uncertainty on the level of a couple of kilometers for our obstacle avoidance constraints. Of course, this distance far exceeds the physical dimensions of any obstacle in Earth orbit (excluding asteroids and the Moon).

If acting in an extremely risk-averse manner, one could bound the obstacle estimate by a 4-σ or higher ellipsoid constraint as formulated and discussed in Section 4.3, which would avoid the obstacle roughly 99.9% of the time. This conservatism will, however, result in a decrease in performance as shown by the notional initial trajectory cost (red line) in Figure 4-19. So, there is a question on how much margin to leave around the estimated obstacle position.

In any case, given the size of these uncertainties from ground tracking and the critical and precise nature of rendezvous and docking missions, onboard sensors are most certainly required to reduce this uncertainty before proceeding to dock. Unfortunately, most sensors that can precisely measure attitude or range will only work within a certain range of the subject and, at the time of trajectory planning, may not be able to provide the relevant information. Assuming, as shown in the previous section, that trajectory replanning can be performed on the spacecraft in real-time, the most efficient method will be to plan an initial trajectory as we see fit, and then replan the trajectory once our onboard sensors come in range and we have a more precise lock on the state of our target or obstacle. The key will be to maintain the same, acceptable level of risk of violating the path constraints upon the replan. The question then becomes, how do we plan the initial trajectory or baseline trajectory such that we will probabilistically achieve the best performance given the uncertainty distribution associated with the path constraint? Or phrased slightly differently: how do we best use our current (potentially very uncertain) knowledge to plan proximity operation missions, while incorporating the fact that better information will be available in the future?

Given this framework, where the spacecraft will replan its trajectory after learning new information about the system (an uncertain event), there is a tradeoff between the size of the safety margin in the path constraints and the optimal fuel or tracking performance as seen in Figure 4-19. There are thus two competing costs: the initial trajectory cost and the replanning cost. A large initial margin given to the path constraint will result in fewer and smaller replanning maneuvers, thus reducing replanning cost, but will increase the initial trajectory cost. A small initial margin will reduce the initial trajectory cost, but will increase the frequency and size of replanning maneuvers. Therefore, looking at the summation of

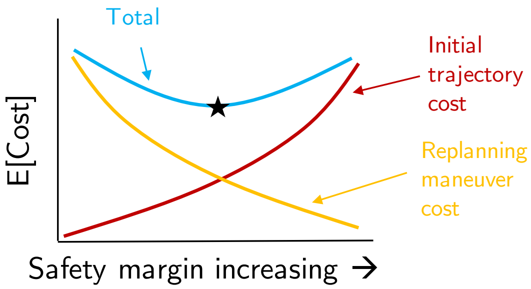


Figure 4-19: Representation of the tradeoff between size of the safety margin in path constraints and optimal fuel or tracking performance

the two costs, there will likely be a minimum total cost at which the best performance will be achieved on average. This hypothesis drives the subsequent methodology and case study sections.

4.4.2 Methodology

As motivated in the previous section, the goal of this probabilistic trajectory planning problem is to find the appropriate margin by which to avoid path constraints such that the expected performance will be minimized over the initial knowledge of the uncertainty in the source of the path constraint. Assuming a Gaussian uncertainty distribution, the problem will be posed in terms of what σ-level margin will be applied to the path constraint. This level will be referred to as the variable, n_σ, which is a multiplier on the standard deviation, σ, of the Gaussian probability distribution that represents the uncertainty in the source of the path constraint. The generalized optimization problem then becomes finding the

$$\underset{n_\sigma}{\operatorname{argmin}} \ \mathrm{E}\Big[J_{init}(n_\sigma) + J_{replan}(n_\sigma)\Big], \tag{4.43}$$

where $J_{init}(n_\sigma)$ is a generic trajectory optimization full-mission cost evaluation as a function of modifying the path constraint by adding a margin of n_σ to the nominal constraint and $J_{replan}(n_\sigma)$ is the additional replanning cost associated with reacting to the uncertain event of resolving the path constraint estimate. $J_{replan}(n_\sigma)$ is dependent on the initial trajectory designed for an n_σ avoidance and does not change its constraints directly as a function of n_σ (it always implements a fully risk-averse constraint matching that of the mission requirements). What adding this n_σ margin actually means will be discussed individually in the two case study sections that follow, however in brief would be the size of the covariance ellipsoid in the obstacle constraint in Section 4.4.3 and an angular margin added to the LOS zone constraint for the target attitude problem in Section 4.4.4.

The process for solving this generalized optimization problem comes down to a Monte Carlo approach. An expected value could also be computed other ways, such as sampling the constraint space with a fine n-dimensional grid and integrating it after weighting with an appropriate probability distribution. The Monte Carlo approach is used here simply due to the ease of implementation. Other methods could be used to increase the computational speed if it is desired to implement this process onboard the spacecraft, although at this point this is meant to be implemented as a baseline trajectory computed on the ground and sent to the spacecraft before the mission, and thus computation time is not a big concern.

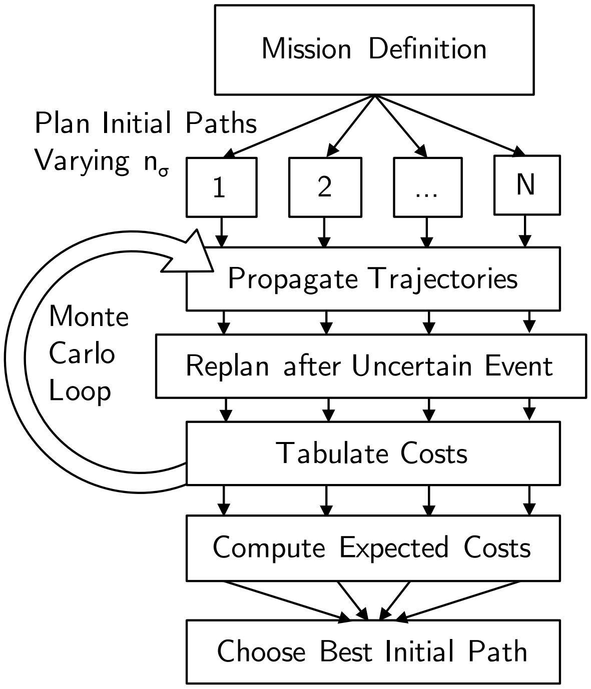


Figure 4-20: Monte Carlo process for the probabilistic trajectory planning

The process follows that described in Figure 4-20. Starting out with a definition of the upcoming mission and the current best estimate of the path constraints, several trajectories are planned that each avoid the path constraint by a different margin, n_σ. Each of these trajectories is propagated forward until a specific instantiation of an uncertain event occurs. For the purpose of the case studies in this thesis, the uncertain event corresponds to resolving a higher precision estimate of the path constraint with onboard sensors and subsequent replanning of the trajectory to adhere to this new path constraint if necessary. This event could be dependent on a specific time or location in the mission or even a distance with respect to the source of the path constraint. Therefore, each of the planned trajectories may trigger the uncertain event at a different point, or may not encounter this event at all. This is where the Monte Carlo simulation comes into play. A specific instantiation of the resolved path constraint is drawn from the uncertainty distribution and each of the trajectories are propagated until the uncertain event is triggered. The trajectory is then replanned using this new information and adhering to the requisite safety margin given the smaller uncertainty with this new knowledge. This process repeats itself for a large number of trials, each drawing a different result from the uncertainty distribution. The results for each of these replanning maneuvers are tabulated and kept until the full Monte Carlo process has completed. At this point, because the Monte Carlo process itself will approximate the uncertainty distribution, all that is left is to average the results to compute the expected costs for each of the initial trajectories. Then the process is simple to choose the n_σ corresponding to the trajectory with the lowest expected cost.

The Monte Carlo selection of instantiations of these uncertain events is performed using a Gaussian Latin Hypercube Sampling (GLHS) that has the potential to reduce the number of trials required to run when compared to a purely random sampling [204]. Using GLHS, and even LHS in general, can be shown to converge to the true values of the probability distribution faster than traditional random sampling techniques. The difference in GLHS is that probability bins are created such that there is an equal chance that a value is drawn from each of the bins. Thus, the bins are narrower close to the mean for a Gaussian and grow wider as they move away from the mean. Then a uniform random variable is drawn and placed appropriately into the corresponding bin. The effect is that the distribution will match that theoretically predicted with a fewer number of samples than is required for simply drawing from a Gaussian probability distribution alone due to its inherent structure. This process can reduce the number of trials run by a factor of 10 or more depending on the application.

The process for finding the minimum in these results can be improved slightly such that a full enumeration of all relevant n_σ trajectories is not necessary as shown in Figure 4-20. Given the results from the case studies in this thesis, a simple bisection search algorithm that converges to the minimum is feasible and could slightly improve performance depending how precise of solution is required. Since the number of trajectories chosen only amounts to about 10-20 values of n_σ between 0.5 and 4 to return a precise enough solution, this technique has the potential to reduce computation time by a factor of about 4-5, with marginally improved performance.

Several assumptions need to be made before this problem can be solved. First, we assume that a current steady-state covariance defining the path constraint with a coarse sensor is known as well as an anticipated steady-state covariance that will be obtained when in range of the higher-precision onboard sensor. Furthermore, a simpler assumption, is that we know the current mean estimates of the chaser, target and path constraint. We will also need to know the detection range of the onboard sensor and the requirement on the final replanning maneuver's avoidance risk. To scope the problem, we also assume that the uncertainty distributions are Gaussian, that we are not considering error in the position and velocity states of the target and chaser, that the onboard sensors will detect and have a steady-state estimate of the refined path constraint 100% of the time, and finally that linear CWH dynamics dominate other orbital disturbances. Specifically, resolving the assumption that no error is present in the chaser state is handled well with chance constraints as implemented in [127]. However, to better showcase the dominating behavior and drive discussion of the methods in this thesis, this simplified problem is explored.

The problem can be extended to multiple sensor updates and multiple uncertain path constraints in a recursive and combinatorial formulation, although this significantly increases computation costs as it suffers from the curse of dimensionality. The problem is that for every choice of n_σ for the first path constraint, another needs to be chosen for the subsequent path constraint. Thus, this has the property of exponentially increasing the number of trials required to find the optimal n_σ for each updated constraint. This extension is not discussed in the thesis due to the complexity and drastic increase in computation time, although would prove to be a very interesting topic to study.

Finally, this process is applied to two case studies in the next two sections. These sections will discuss the customized processes specific to their application, rather than the general form and process discussed in this section. The goal of these case studies is to allow the reader to get a more concrete understanding of how this work can be applied to real problems and show the benefits of planning in this manner through a collection of results.

4.4.3 Case Study 1: Uncertain Obstacle Avoidance

The first case study examined for the topic of probabilistic trajectory planning under uncertain path constraints is that of a large uncertainty in the state of an obstacle. This problem will be explored with a minimum energy, LQR cost function as described in Section 4.1.1 and the rotating-hyperplane, ellipsoid obstacle constraint formulation from Section 4.3.2. The idea is that there is an obstacle being tracked with ground radar with substantial uncertainty associated with its estimate on the order of about a 1 km, 3-σ Gaussian distribution. However, the spacecraft has an onboard sensor, such as LIDAR, that is capable of detecting any object within a specified range of the spacecraft and reducing the uncertainty in the estimate by a known sensor improvement factor. The major gain in this new measurement is that it is from a unique viewpoint which can both reduce the uncertainty in the estimate and resolve a better estimate of the state of the obstacle, possibly removing bias from the ground measurements.

While the sensor improvement factor will help reduce the size of the obstacle uncertainty ellipsoid and make it easier for the spacecraft to avoid, there is no guarantee that this new position will not be directly in the path of the optimal trajectory. Upon detection, we can expect the new measurement of the obstacle will likely not place it at the exact mean of the original uncertainty distribution, but essentially randomly drawn from that initial distribution. Since it is a Gaussian distribution, there is the possibility, however small, that the obstacle could pop up anywhere no matter what the initial avoidance plan was.

Thus, after detecting the obstacle, we would like to replan the trajectory to avoid the ellipsoid corresponding to the newly found obstacle. The final replanning trajectory needs to maintain the requisite risk margin, specified as the 99.9%, 4-σ avoidance trajectory under the uncertainty of the new sensor. As discussed in the last section, the goal is to find the probabilistically optimal initial trajectory (as a function of having an n_σ margin) such that a balance is found between the increasing initial cost of being more risk averse and the increasing replanning cost of taking a higher risk initial trajectory.

The n_σ margin directly corresponds to the size of the covariance ellipsoid defined in Section 4.3.1, thus the ellipsoid constraint can be modified as

$$\mathbf{P} = \left(n_\sigma^2 \Sigma\right)^{-1}, \tag{4.44}$$

where $\mathbf{P}$ is the shape matrix of the ellipsoid, Σ is the covariance matrix for the position estimate of the obstacle, and (4.29) through (4.32) still hold. Thus, we vary the value of n_σ to change the size of the obstacle ellipsoid and try to find the level of risk margin that enables the best performance in terms of LQR cost.

Following the process outlined in Section 4.4.2 for this application may help clarify exactly what is going on. Figure 4-21 shows a step-by-step procedure that outlines this process. The first step is to develop an initial estimate of the obstacle's location from ground measurements and setup avoidance ellipsoids at the varying levels of n_σ that we wish to explore. The second step is to optimize trajectories for each of these n_σ-levels of margin around the obstacle. The third step is to propagate the spacecraft through each of the n_σ-margin trajectories until the uncertain event is triggered when the spacecraft is within 1 km of the updated obstacle position drawn in the Monte Carlo process. At this point in step four, an ellipsoid constraint is created to force the spacecraft to avoid the updated obstacle by a level of $n_\sigma = 4$, corresponding to a 99.9% guarantee of avoidance based on the anticipated covariance present in the onboard sensor. Finally, the spacecraft replans the trajectory such that it avoids the obstacle and reaches the goal location. All of the results (initial trajectory cost and replanning cost) are tabulated for this particular Monte Carlo run. Then steps three through five are repeated until the required number of trials is run for the Monte Carlo to return statistically significant results through an average of all of the trials' results. A selection of 10,000 trials seems to achieve this statistical significance.

As depicted in Figure 4-21, this particular mission is designed for Phase 3 or the joint maneuvering phase of the generalized mission architecture presented in Section 3.2. Because this phase covers the longest distance and is supposed to represent a transfer of a space station module to an assembly location, the possibility of obstacles was thought to be higher than some of the other phases. The spacecraft spend more time in this phase and are navigating to a potentially congested area. There is the potential that another module of the space station has drifted off, another assembler satellite is maneuvering to the assembly location, or debris has been created from the assembly area. Thus, for our purposes, any of these situations have produced an obstacle that the spacecraft must avoid before reaching the assembly location. Of course, the techniques and results shown here will transfer to the other phases of the mission if necessary and could be combined with additional uncertain path constraints if necessary.

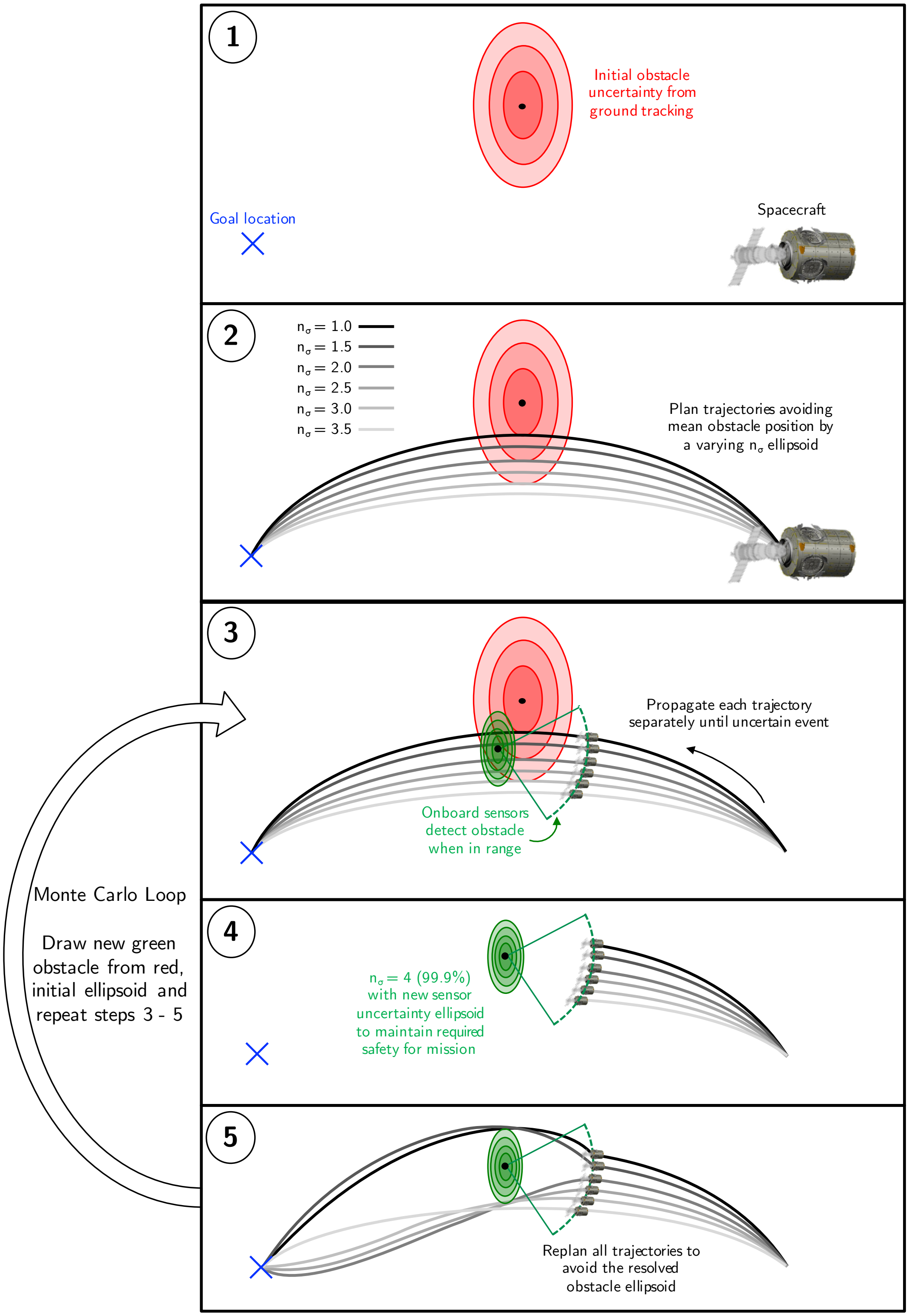

1
Initial obstacle uncertainty from ground tracking
Goal location
Spacecraft
2
$n_\sigma = 1.0$
$n_\sigma = 1.5$
$n_\sigma = 2.0$
$n_\sigma = 2.5$
$n_\sigma = 3.0$
$n_\sigma = 3.5$
Plan trajectories avoiding mean obstacle position by a varying n_σ ellipsoid
3
Propagate each trajectory separately until uncertain event
Onboard sensors detect obstacle when in range
Monte Carlo Loop
Draw new green obstacle from red, initial ellipsoid and repeat steps 3 - 5
4
$n_\sigma = 4$ (99.9%) with new sensor uncertainty ellipsoid to maintain required safety for mission
5
Replan all trajectories to avoid the resolved obstacle ellipsoid

Figure 4-21: Steps of the Monte Carlo process for probabilistic trajectory planning with uncertain obstacle constraints

This process has been implemented and executed in the MATLAB simulation described in Section 3.4. The full 3DOF problem is investigated although the goal location and initial position of the spacecraft are in the same orbital plane, so although there is a third DOF, the optimal (unobstructed) motion is to stay in the decoupled, 2DOF plane. Note that the spacecraft still has the ability to avoid the obstacles in the out-of-plane direction, although, given the advanced distance and the size of the simulated obstacles, it tends to remain in the orbital plane. Thus, for this simulation we only rotate the hyperplane in positive and negative directions about the $\hat{k}$-axis to account for the non-convexity rather than sampling the full 3DOF boundary. We use a quadratic solver to solve for the minimum LQR cost trajectory.

Figure 4-22 shows an example of how three choices of the n_σ avoidance ellipsoid can affect the results of the total trajectory cost. Note that these results are only for one instantiation of an obstacle randomly drawn in the Monte Carlo process, and therefore

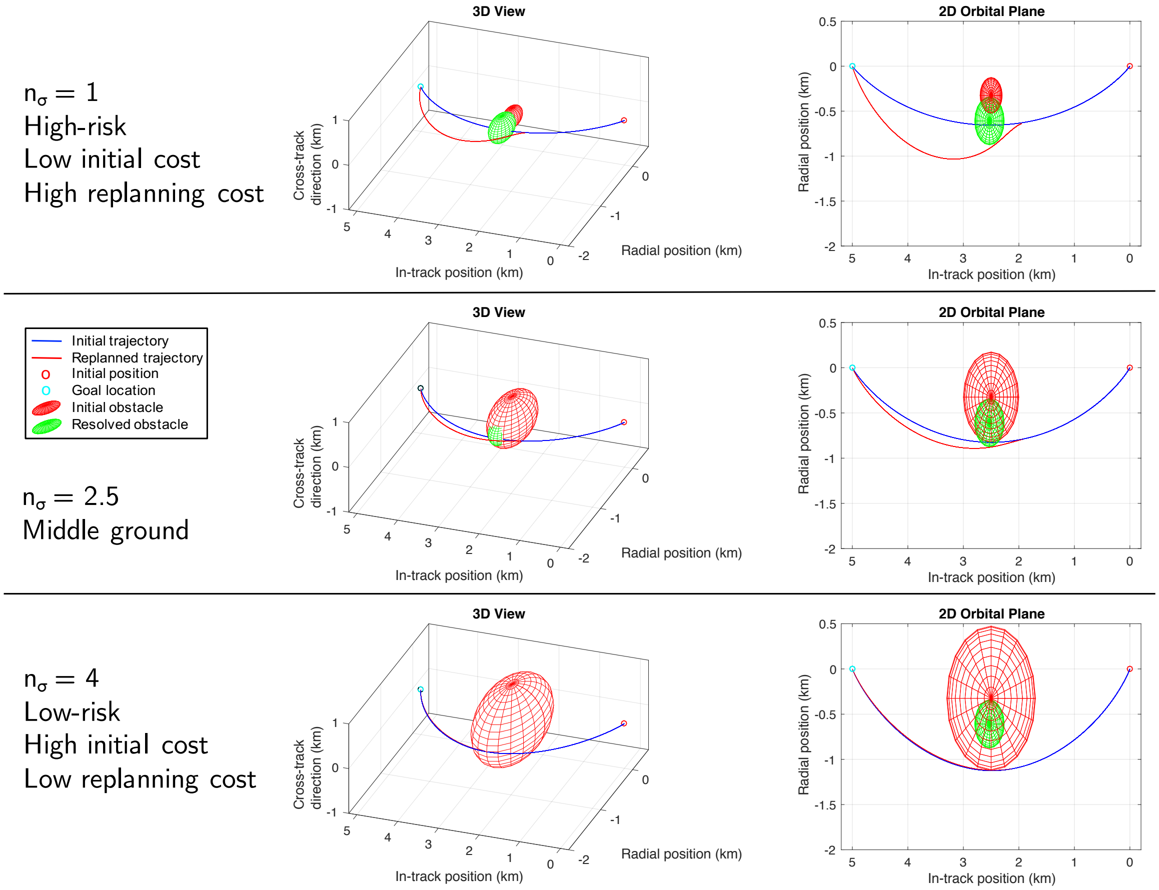


Figure 4-22: The effects of replanning on three initial trajectories, each avoiding the initial obstacle by a different margin

only show a general trend that could be experienced and vastly exaggerate the results at a wholistic level. Additionally, note that the obstacle does move, but it is difficult to represent in a static plot, and the hyperplane constraint, which also moves, is not depicted. The red, initial, n_σ-level obstacle uncertainty ellipsoid is shown at the point where the original trajectory is tangent (i.e., where the constraint is active), while the green, updated 4σ-level ellipsoid is shown where the specific replanned trajectory is tangent. This is not the location where each of these obstacles are detected and does not represent the full constraint that the spacecraft needs to avoid.

From Figure 4-22, there is a larger replanning maneuver for the higher risk scenario and barely any maneuvering present in the low-risk initial trajectory. The high-risk initial trajectory is identical to the true minimum-energy trajectory as the 1-σ ellipsoid does not intersect it. At 2.5-σ, the trajectory is perturbed by the initial ellipsoid resulting in a lower replanning cost but higher initial cost. The 4-σ ellipsoid is so large that the trajectory does not even need to be replanned when the onboard sensors detect the obstacle. In fact, the low-risk option actually replans to have a slightly better solution that the original trajectory, because the obstacle is not in its path at all. This behavior is important and will occur often throughout the Monte Carlo process. Thus, sometimes the replanning cost will be negative as performance will actually be improved after sensing the location of the obstacle with onboard sensors if the original trajectory was too conservative. Another peculiar behavior seen in Figure 4-22 is that the 1-σ initial obstacle size is smaller than the updated 4-σ size. Although strange at first glance, this is appropriate as the improvement in the sensor quality was not 4 times as precise, only 3 times for this example.

The process in Figure 4-21 is run by tabulating all of the results from the Monte Carlo simulation that produces individual trials similar to what is seen in Figure 4-22. New obstacle instantiations are drawn randomly in the Monte Carlo, and the average initial cost and replanning cost are computed for each level of margin. These averages are the expected value of the costs in a probabilistic sense. From this, we can determine the level of margin that produces the minimum total cost. Figure 4-23 shows the results from a simulation with three times improvement in sensing from onboard sensors at a detection distance of 1 km, with 10 n_σ values, and an obstacle with an initial uncertainty of 200 m, 1-σ. The results from this example show the exact behavior that was expected from the notional chart in Figure 4-19. The results prove that this tradeoff between safety and performance does

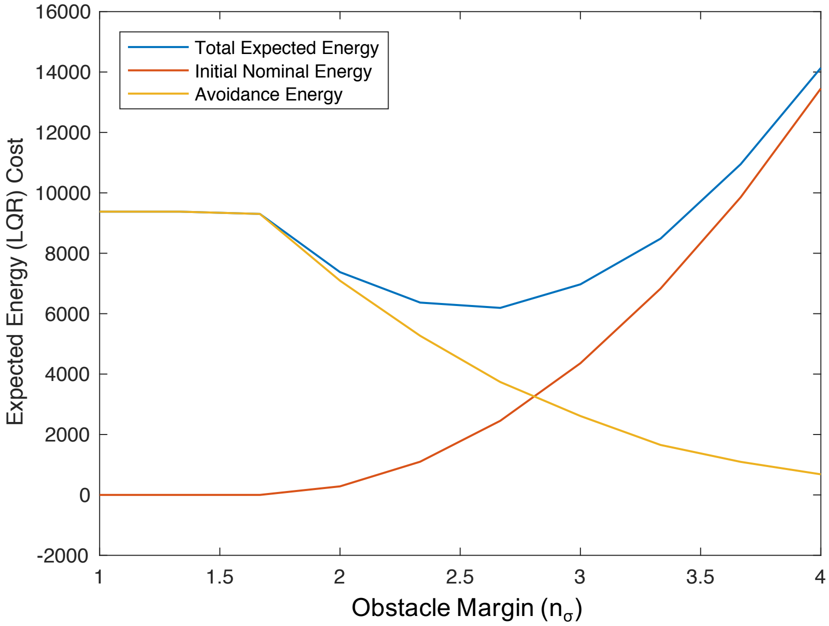


Figure 4-23: Results showing the tradeoff between initial and replanning costs and a definitive minimum in the total cost for the obstacle avoidance problem

exist and is measurable for a mission with reasonable parameters. Thus, our hypothesis is confirmed for the case of large uncertainties in initial knowledge of an obstacle's state. By selecting an initial trajectory with an avoidance ellipsoid constraint sized with $n_\sigma = 2.5$, the probabilistically optimal, minimum-energy cost trajectory can be implemented as a baseline for this mission. These results are calculated for the same simulation properties as for Figure 4-22, and thus the middle plot shows the probabilistically optimal initial trajectory.

Specifically, to note about Figure 4-23 is that in the range of n_σ from 0 through about 1.5, the minimum-energy trajectory does not intersect the initial avoidance ellipsoid in this problem, thus the same trajectory is implemented for all of these cases. Given the same initial trajectory, all results are the same in this range. Additionally, Figure 4-23 only shows the cost in excess of the unconstrained minimum-energy trajectory, so the initial nominal energy is zero with an n_σ value from 0 through 1.5, and the total cost is simply the avoidance energy or replanning cost.

As the detection distance and sensor improvement factor are modified, several interesting results can be gleaned from a comparison of their tradeoff curves. Figure 4-24 shows how the sensor improvement factor affects the results in terms of the general magnitude and behavior of initial trajectory cost, replanning or avoidance cost and the minimum of the total cost. Figure 4-25 shows the same for the detection distance.

The results presented in Figure 4-24 mainly show the trend of how the magnitude of the avoidance cost changes as the sensor improvement factor deceases (in the left column) and increases (in the right column), and how this affects the choice of the minimum total energy trajectory. The nominal case is shown in the top-center with a detection distance of 1 km and sensor improvement factor of 3, meaning that the onboard sensor is capable of detecting a 66 m, 1-σ estimate at a distance of 1 km. Note the value of the avoidance cost

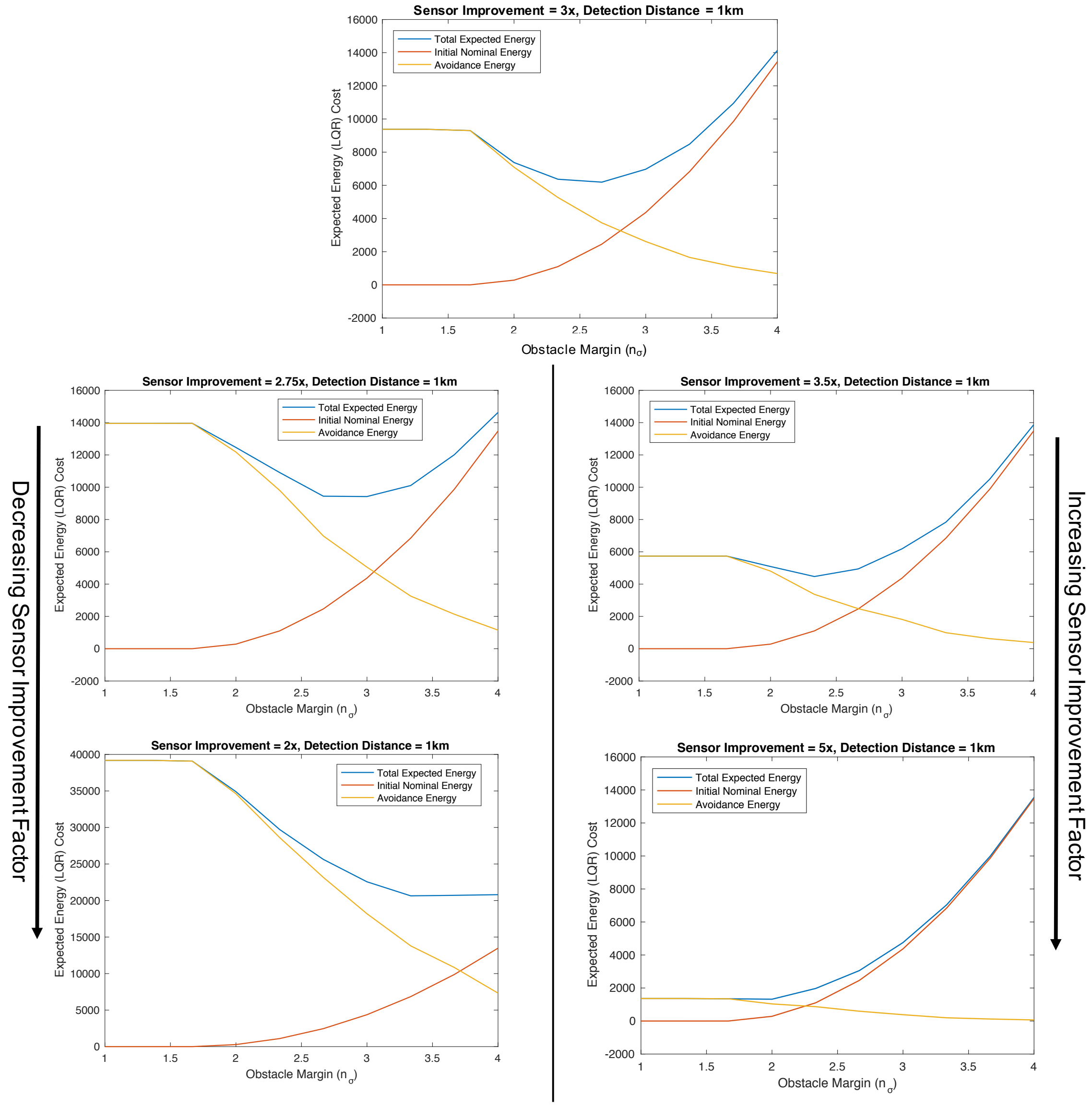


Figure 4-24: Results showing the tradeoff between initial and replanning costs while varying the sensor improvement factor for the obstacle avoidance problem

plateau on the left of each plot to see how a changing sensor improvement factor affects the results. The initial trajectory cost remains the same for all of these parameter choices, so the effects seen in the total energy cost of the trajectories is solely due to the changing avoidance cost. As the sensor improvement factor gets worse, there is a sharp increase in the avoidance energy, such that at an improvement factor of 2, the avoidance energy more than quadruples. The bottom left plot is the only one on a different scale, as the cost was too high to warrant changing all other plots to its scale. The increasing avoidance energy behavior causes the optimal n_σ to increase and suggests more risk-averse initial trajectories are appropriate for lower performing onboard sensors. The converse happens when increasing the sensor improvement factor, as can be seen by the rightmost two plots. The avoidance energy shrinks, causing the optimal n_σ to decrease until the unconstrained initial trajectory will perform the best on average with an improvement factor of 5. At a very high level, these results match what one might expect: investment in higher precision sensors will reduce the on-orbit cost of unnecessarily avoiding obstacles.

The results presented in Figure 4-25 show a similar trend of how the magnitude of the avoidance cost changes as the sensor detection distance deceases (in the left column) and increases (in the right column), and how this affects the choice of the minimum total energy trajectory. The nominal case is shown in the top-center again. A difference in this scenario is that the initial trajectory energy costs actually change with the detection distance. This mainly shows up for larger detection distances, as there are more cases in which there is very early detection of non-threatening obstacles and the path can be replanned to have an optimal, unconstrained trajectory at this point which reduces the cost compared to the initial trajectory. This reduction in cost is lumped in with the initial trajectory cost, because it does not relate to avoidance per se. Additionally, we wanted to keep the avoidance cost positive for easier interpretation of the results.

As the detection distance gets worse, there is a to-be-expected increase in the avoidance energy. This trend is reasonable because reacting later to the same obstacle will require more severe correction maneuvers. Again, this behavior causes the optimal n_σ to increase and suggests more risk-averse initial trajectories are appropriate for lower performing onboard sensors. The opposite trends can be seen when increasing the detection distance. The avoidance energy is reduced, causing the optimal n_σ to move left until the unconstrained initial trajectory will perform the best on average. Similar to the sensor improvement factor

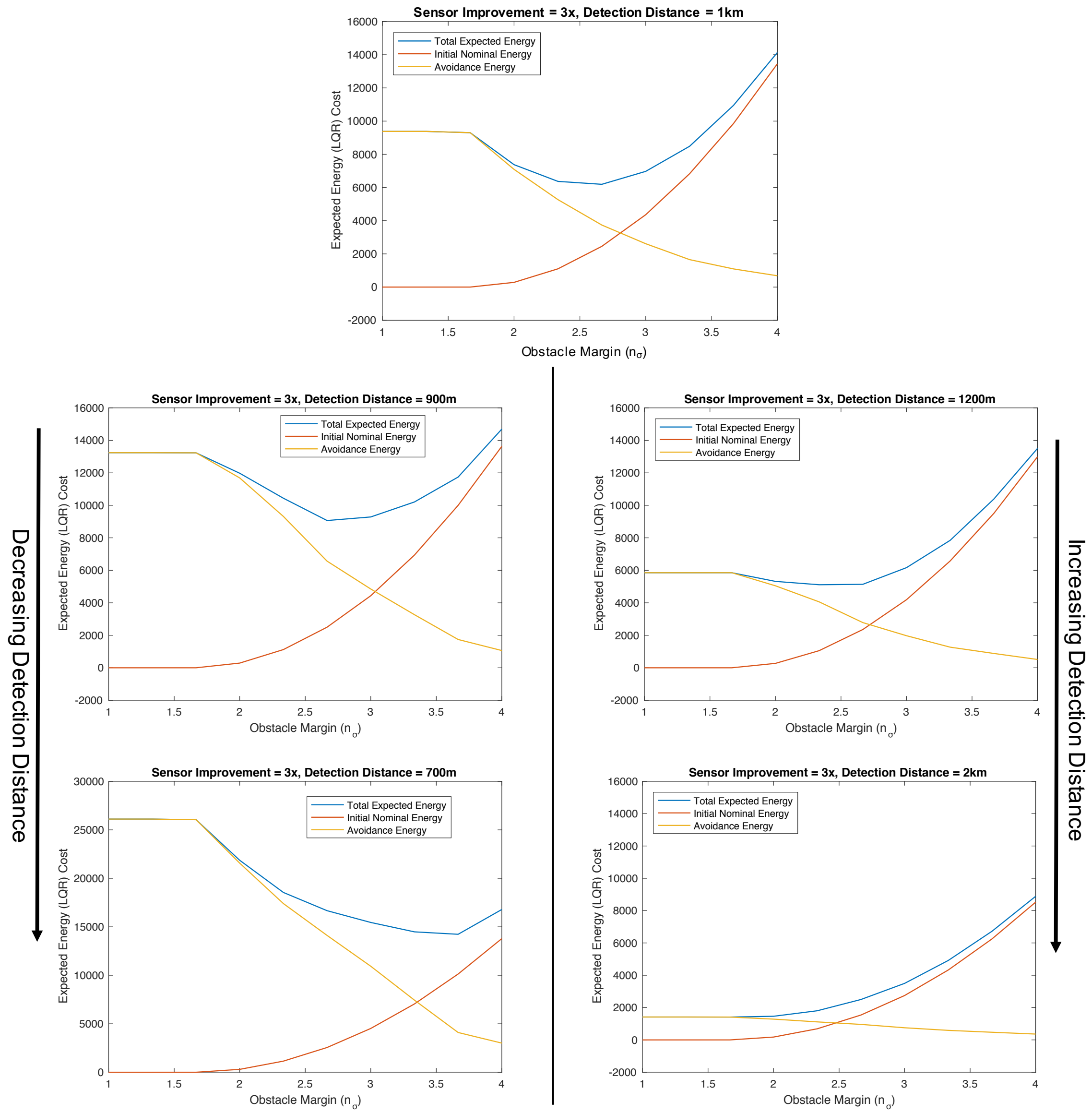


Figure 4-25: Results showing the tradeoff between initial and replanning costs while varying the detection distance for the obstacle avoidance problem

results and at a very high level, these results match what one might expect: investment in longer range sensors will reduce the on-orbit cost of avoiding obstacles by allowing earlier reaction.

These high-level trends that match conventional wisdom help to validate the process and give confidence in the results, however in a systems engineering context a more quantified approach may be required to determine which sensors are most appropriate for a particular spacecraft. By varying both the detection distance and sensor improvement factor across relevant ranges for existing sensors, a surface plot can be created that shows the probabilistic minimum energy cost of the proposed mission against variations in these sensor parameters. Figure 4-26 shows an example of this type of performance surface for the mission investigated in this section. In application, several mission examples should be examined when making system level decisions for spacecraft design, however, if we simply look at this mission scenario, we can draw several interesting conclusions.

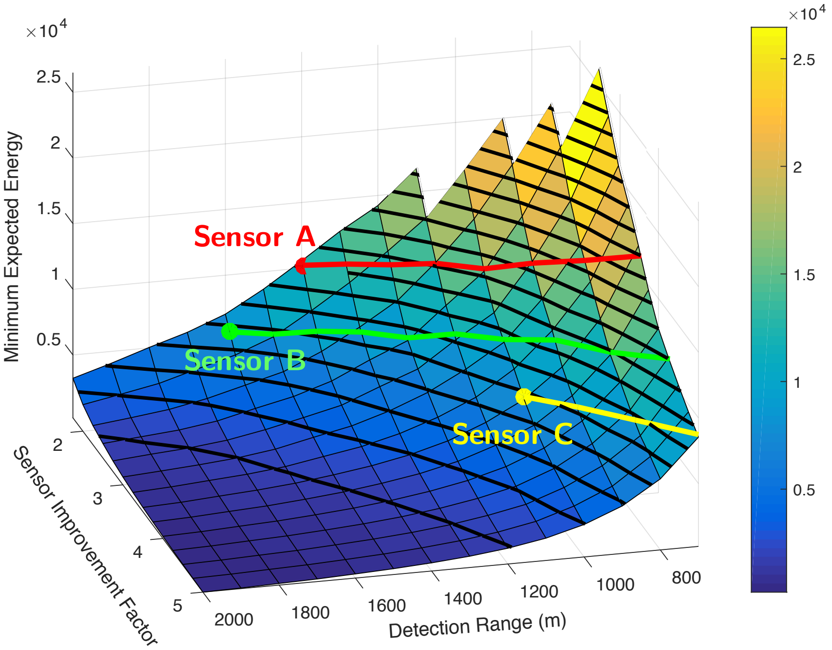


Figure 4-26: Results showing the tradeoff between initial and replanning costs while varying the detection distance and the sensor improvement factor for the obstacle avoidance problem. Three linear sensors are shown on the chart for comparison of performance.

A system engineer could use this surface plot to compare specific sensor performance in terms of total trajectory costs. Sensors A, B and C shown in Figure 4-26 are linear sensors. A linear sensor is considered to improve precision inversely proportional to the range from the subject. Thus, in a simple example a sensor would have twice the precision at half the distance from the subject. Sensors A and B both have sensor improvement factors of about 2 at their initial detection ranges of 1400 m and 1600 m respectively. Sensor C has a much better sensor improvement factor of 3.5 but can only detect from 1 km. The relationship between sensing performance and detection range can be plotted for each sensor and overlaid on the performance surface in Figure 4-26. By tracing the contours around, we can see that the initial probabilistic performance of Sensor B and Sensor C are nearly identical. This result shows an interesting comparison present between detection range and precision of the sensor. Additionally, following the line associated with each sensor dot, shows that using these linear sensors later than their initial detection will only have worse energy performance. This means that detection range is more important than precision in the context of these linear sensors. Other sensors may experience different behavior depending on the sensor model and additional sensors could be explored as necessary. There is the potential to be able to find a minimum performance detection range at which to start using the measurements from the sensor. However, for these linear sensors, the initial detection range represents the best choice.

Another viewpoint a system engineer could take is that the minimum expected cost (in this case the energy cost) is essentially the cost of the uncertainty in the system. Because the initial, unconstrained optimal cost is subtracted from all results in the charts here, the values shown are indeed the cost due to this uncertainty. Then the question arises as how to deal with this cost of uncertainty. The choice could potentially lie between investing in better sensors or in additional fuel (in the case that fuel cost was being optimized). Thus, analyses can be performed on whether investing in better sensors is worthwhile or if it would simply suffice to add additional fuel margin to account for the uncertainty. Plots like that in Figure 4-26 can go a long way in supporting these decisions. This decision point is of course subjective and depends entirely on the mission and vehicle, however does provide an interesting view point.

This section has motivated and discussed results from a case study of probabilistic trajectory optimization in the presence of large uncertainties in obstacle path constraints. It can be seen that there are performance improvements to be gained from analyzing the problem in this manner and finding the appropriate balance between risk and performance. The next section will discuss another case study in the context of the LOS zone constraint and uncertainty in the target's attitude.

4.4.4 Case Study 2: Uncertain Target Attitude

The second case study examined for the topic of probabilistic trajectory planning under uncertain path constraints is that of uncertainty in the attitude of the target and thus the direction the LOS zone constraint points. This problem will be explored with a minimum fuel cost function as described in Section 4.1.2 and the linear LOS zone constraint formulation from Section 4.1.3. The concept is that the target does not have active sensors onboard or at least has very poor attitude sensors onboard and the chaser spacecraft does not have a good estimate of its attitude. The target, which could be a defunct satellite, a space station module, or orbital debris in the form of a discarded rocket body, has a required docking location and therefore an associated LOS zone tied to its attitude. This poor initial estimate of the attitude will be improved when the spacecraft gets within range of its onboard sensors (LIDAR) that are able to measure the attitude more precisely. Again, the major gain in this new measurement is that it is from a unique viewpoint which can both reduce the uncertainty in the estimate and resolve a better estimate of the state of the obstacle, possibly removing bias from the initial measurements.

While the sensor improvement factor will help reduce the uncertainty in the attitude, there is no guarantee that this new attitude will not cause the original trajectory to miss the LOS zone. Upon detection, we can expect the new measurement of the attitude will likely not place it at the exact mean of the original uncertainty distribution, but essentially randomly drawn from that initial distribution. Since it is a Gaussian distribution, there is the possibility, however small, that the attitude could pop up anywhere no matter how robust the initial plan was.

Thus, after detecting the target's attitude, we would like to replan the trajectory to adhere to the new LOS zone corresponding to the newly resolved attitude. The final replanning trajectory needs to maintain the requisite risk margin, specified as the 99.9%, 4-σ avoidance trajectory under the uncertainty of the new sensor. As discussed previously, the goal is to find the probabilistically optimal initial trajectory (as a function of having an n_σ margin) such that a balance is found between the increasing initial cost of being more risk averse and the increasing replanning cost of taking a higher risk initial trajectory. The concern here is with cutting the corner of the LOS zone.

The exact definition of n_σ in this case study differs from the previous obstacle ellipsoid application, however should be just as intuitive. We are now concerned with the LOS zone rotating with the target attitude and thus making corner cutting of the original constraint very risky. Therefore, we are going to apply an angular margin to both edges of the LOS zone constraint as shown in Figure 4-27. To simplify, the attitude of the target will only vary in the $\hat{k}$-axis and will be oriented correctly in the out-of-plane attitude. Therefore, the uncertainty in the target attitude can be represented by a Gaussian univariate distribution with standard deviation, σ. The LOS zone margin is then defined by the multiplier, n_σ.

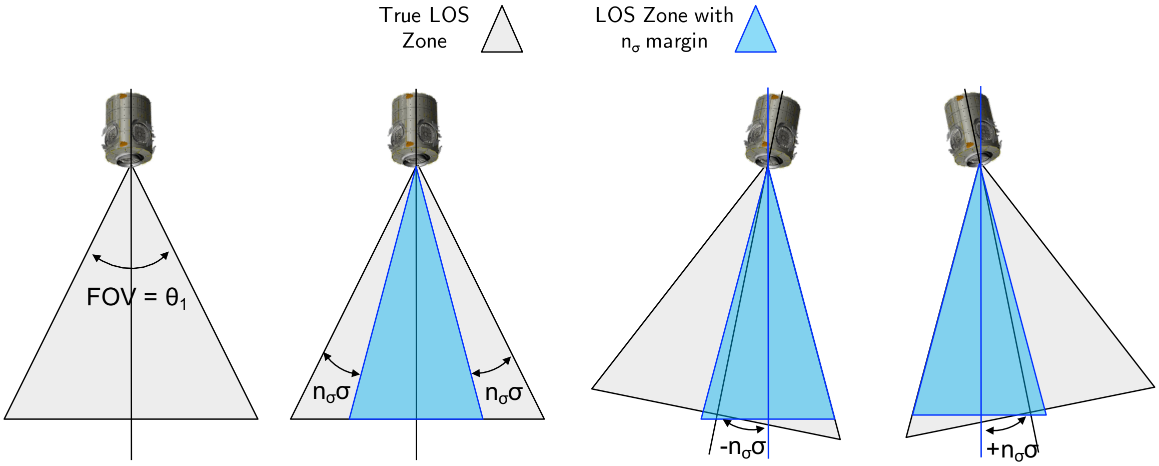


Figure 4-27: Description of what the n_σ margin means in the context of uncertain target attitude and the LOS zone constraint, where σ is the standard deviation in attitude error

The pyramid version of the LOS constraint is described in Section 3.2.4 and further detailed for implementation in (4.23) and (4.24). It is the angle, ϕ, which will be the uncertain parameter and will be measured by onboard sensors. The only change to this constraint will be to change the value of θ_1 to $\theta_1 - 2n_\sigma\sigma$. When replanning after the resolved attitude is known, the updated mean estimate for ϕ will be implemented in the constraint. Although the LOS zone with the margin applied is smaller than the true LOS zone, it is reducing risk of cutting the corner or exiting the LOS zone on either side rather. If the margin exceeds the half-width of the LOS zone, there will be no trajectory that can meet that margin as the constraint is infeasible. Thus, we are limited to scenarios in which the 4-σ value of the constraint is less than the half-width of the true LOS zone.

The process of determining the probabilistically optimal trajectory for this case study is described in detail in Figure 4-28. The first step is to determine the true LOS zone FOV angle, θ_1, and construct a set of constraints that vary the margin to this true LOS zone through different n_σ values. The second step is to compute a fuel optimal trajectory individually for each of these n_σ valued constraints. The third step is to propagate each of these trajectories individually until the onboard sensors are within range of the target spacecraft and a new increased precision measurement is obtained (the uncertain event). The fourth step, once the new target attitude information is obtained, is to formulate the new LOS zone constraint by adding a 4-σ margin to the true LOS zone based on the smaller uncertainty associated with the new measurements. Potentially at this point, the target attitude will have moved in a manner such that the original trajectory now does not adhere

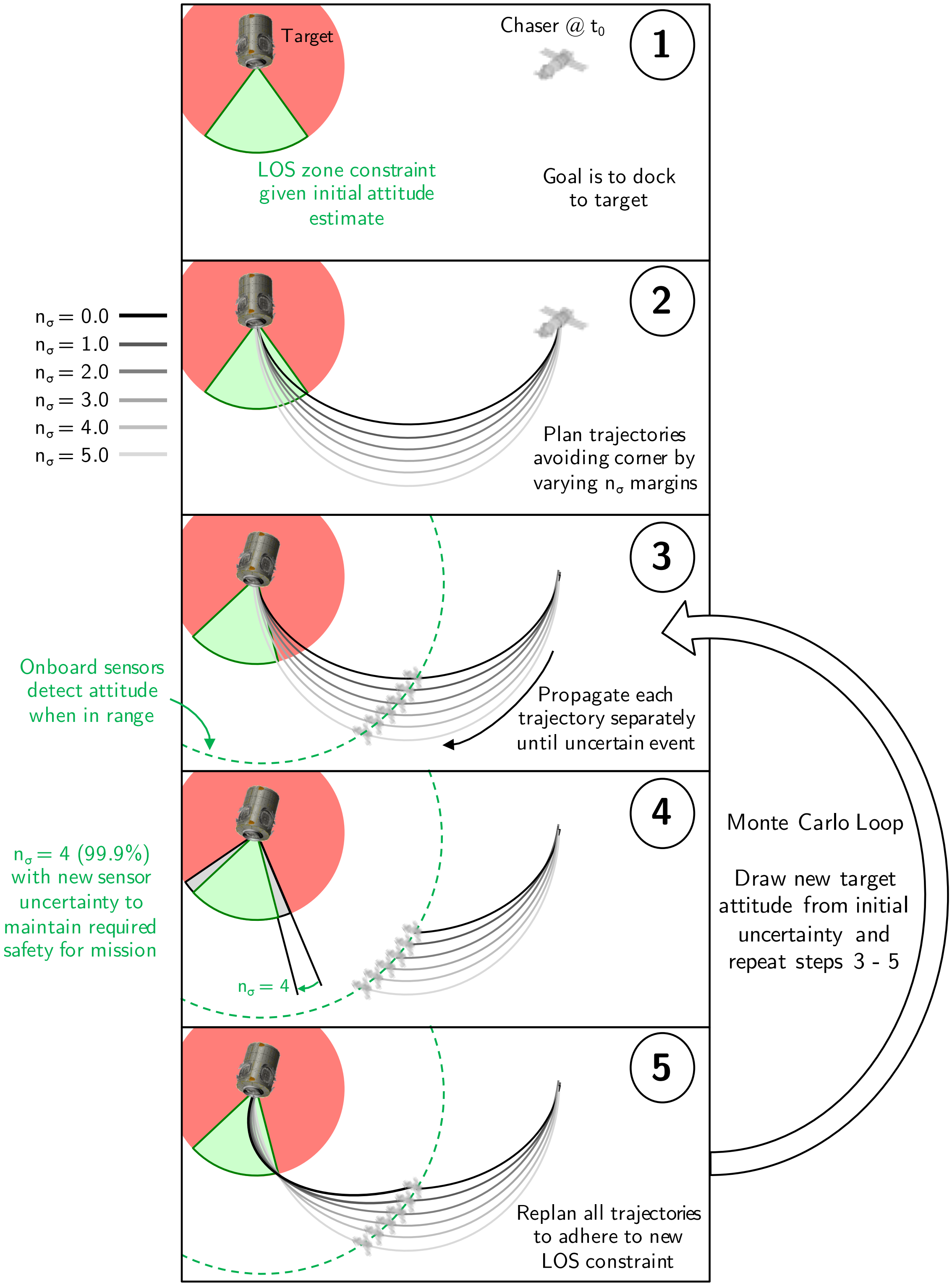

Target
Chaser @ t_0
1
LOS zone constraint given initial attitude estimate
Goal is to dock to target
$n_\sigma = 0.0$
$n_\sigma = 1.0$
$n_\sigma = 2.0$
$n_\sigma = 3.0$
$n_\sigma = 4.0$
$n_\sigma = 5.0$
2
Plan trajectories avoiding corner by varying n_σ margins
3
Onboard sensors detect attitude when in range
Propagate each trajectory separately until uncertain event
4
$n_\sigma = 4$ (99.9%) with new sensor uncertainty to maintain required safety for mission
$n_\sigma = 4$
Monte Carlo Loop
Draw new target attitude from initial uncertainty and repeat steps 3 - 5
5
Replan all trajectories to adhere to new LOS constraint

Figure 4-28: Steps of the Monte Carlo process for probabilistic trajectory planning with uncertain target attitude and LOS constraints

to the required constraints and replanning must be performed. Therefore, in step five, the goal is to optimize the trajectory from this point forward to adhere to the new 4-σ constraint, resulting in a 99.9% probability of mission success. Steps three through five are then repeated by drawing a new resolved attitude from original uncertainty distribution at the detection distance. This is the Monte Carlo portion of the process and must be repeated for about 10,000 trials to obtain statistically significant results. For each trial, the results are tabulated and then averaged at the end to determine the expected value of the initial trajectory cost and replanning cost. From these results, the best n_σ initial trajectory can be chosen as the optimal baseline trajectory.

The case study is implemented through Phases 1 and 2 of the thesis mission problem. Specifically, the main concern here is with the transition from Phase 1b to Phase 2. The process discussed in Section 4.2.1 is used to solve this phase transition problem in the minimum fuel, linear programming form. The case study is implemented and executed in the MATLAB simulation discussed in Section 3.4 for only the 2DOF problem. We use a linear solver to compute the minimum fuel trajectories and, consequentially, use linear versions of all constraints.

Figure 4-29 presents the results from three initial trajectories, each with different n_σ margins and the required replanning maneuver based on a single trial of the Monte Carlo simulation. For this analysis, we use an attitude standard deviation of $\sigma = 5°$ and a FOV angle of $\theta_1 = 60°$. The detection distance is 300m and the sensor improvement factor is 10. There are several interesting points to discuss about these results. The 0-σ initial trajectory skirts the edge of the true LOS zone, such that when the resolved attitude of the target is determined, it must perform a large maneuver to meet the 4-σ constraint with the new uncertainty. The 2-σ initial trajectory instead shows more moderate results with a smaller avoidance maneuver upon the updated LOS zone. Finally, the 4-σ initial trajectory barely makes a correction at all. Notice how the blue, initial LOS constraints decrease in size as the n_σ margin increases to force the initially planned trajectory to enter the docking phase in a more centralized location. Note that the green, updated LOS zone is determined as each of the spacecraft enter the range of their onboard sensor (marked as a dashed black line in the figure), and may be larger or smaller than the initial LOS zone constraint. This is purely a consequence of the comparative levels of uncertainty and does not explicitly affect results.

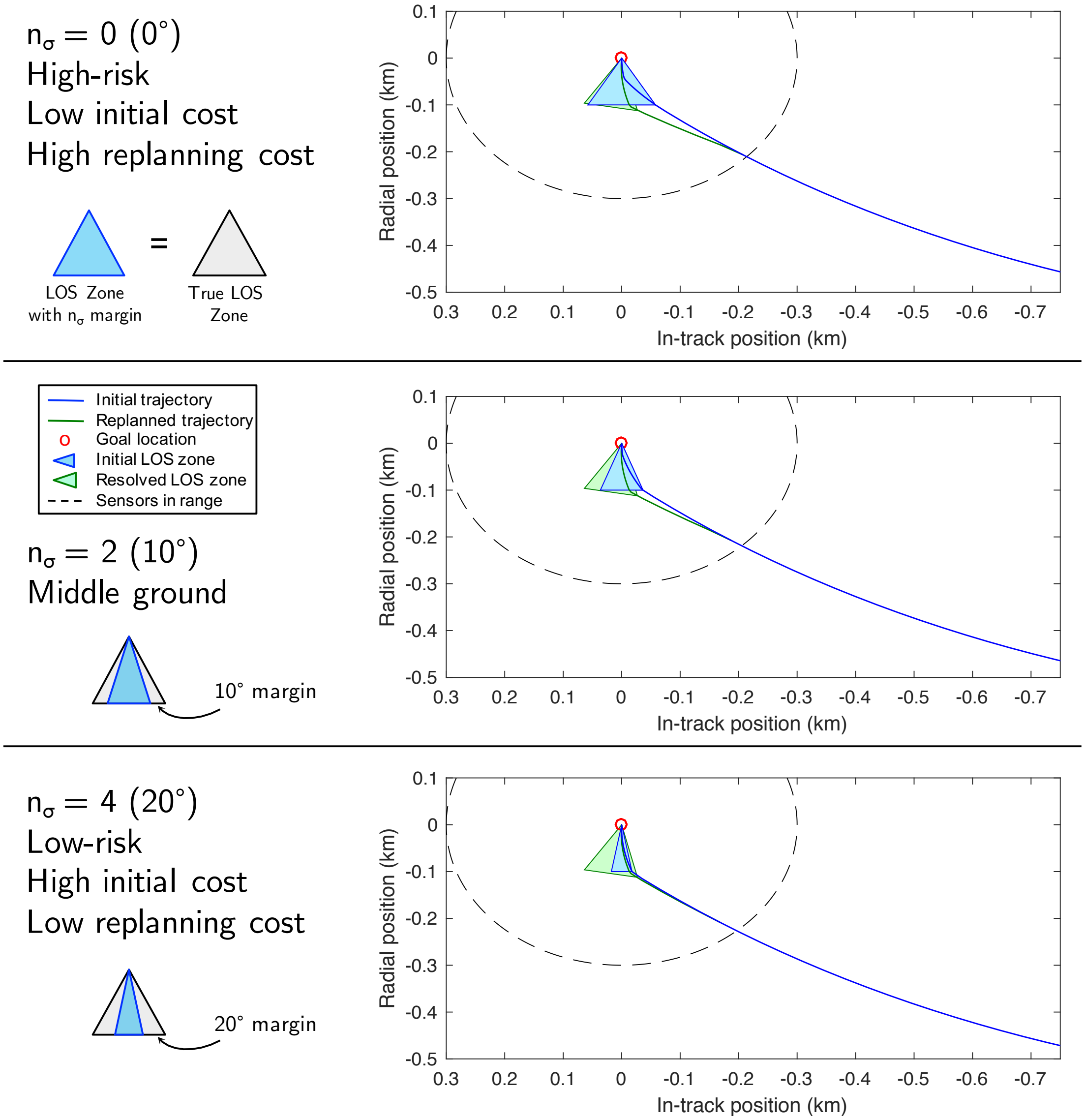


Figure 4-29: The effects of replanning on three initial trajectories, each adhering to the initial LOS region constraint by a different margin

The process in Figure 4-28 repeats, producing results similar to those shown in Figure 4-29 for each n_σ trajectory and for each target attitude sampled from the Gaussian distribution during the Monte Carlo process. All of the costs associated with these trajectories are tabulated and used to compute an expected value of the initial trajectory cost and replanning cost in terms of fuel consumption. The results in Figure 4-30 show these expected fuel costs as a function of the n_σ risk margin. These results are less convincing than the obstacle case study, however we can still find an optimal n_σ risk margin of 3.5 and begin to show increased cost as the risk margin increases further. This optimal trajectory

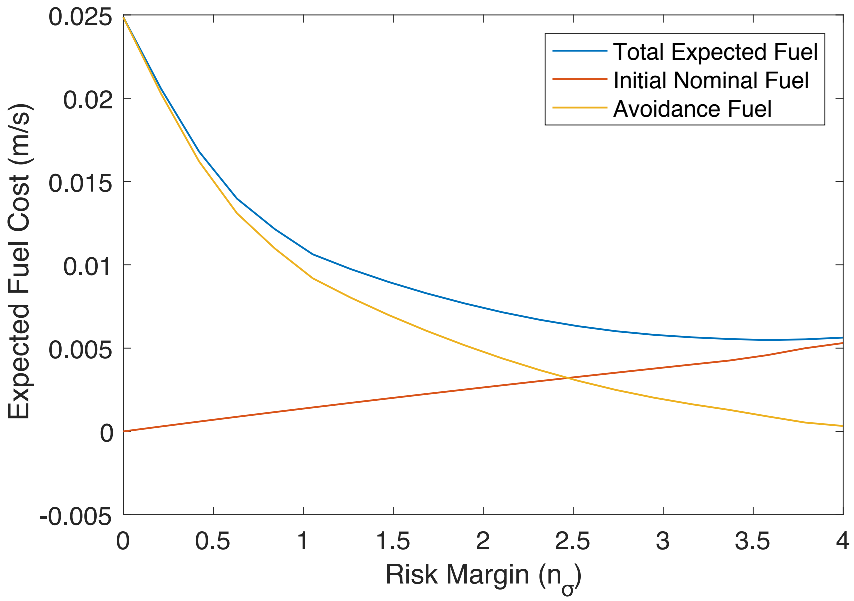


Figure 4-30: Results showing the tradeoff between initial and replanning costs and a minimum in the total cost for the uncertain target attitude problem

is similar to that seen in the bottom plot of Figure 4-29 although slightly riskier. As expected, the replanning fuel cost decreases and the initial trajectory fuel cost increases as the risk margin increases. Therefore, the tradeoff is experienced in this problem as well and is capable of being exploited to pick a probabilistically optimal baseline solution. With this, we can confirm that probabilistically planning trajectories under uncertain path constraints is applicable and could be potentially useful for missions with uncertain target attitude and an LOS zone constraint. The improvement over an extremely risk-averse trajectory (a very simple way to handle the uncertainty), however, would vary depending on the specific mission parameters.

Figure 4-31 shows how the expected replanning, or avoidance, fuel cost changes as a function of the detection range that specifies the distance that the chaser's onboard sensors are able to measure the target's attitude. As the detection range increases, similarly to the obstacle avoidance problem, the expected avoidance fuel decreases, causing the minimum total expected fuel trajectory to have a smaller and smaller n_σ risk margin. This trend makes sense as the earlier the spacecraft can react to the new information, the cheaper that replanned trajectory is going to be. The closer the chaser gets to the target without knowing the updated target attitude, the more severe a correction maneuver will need to be. In this way, detection range is a very valuable metric for a sensor. However, there is

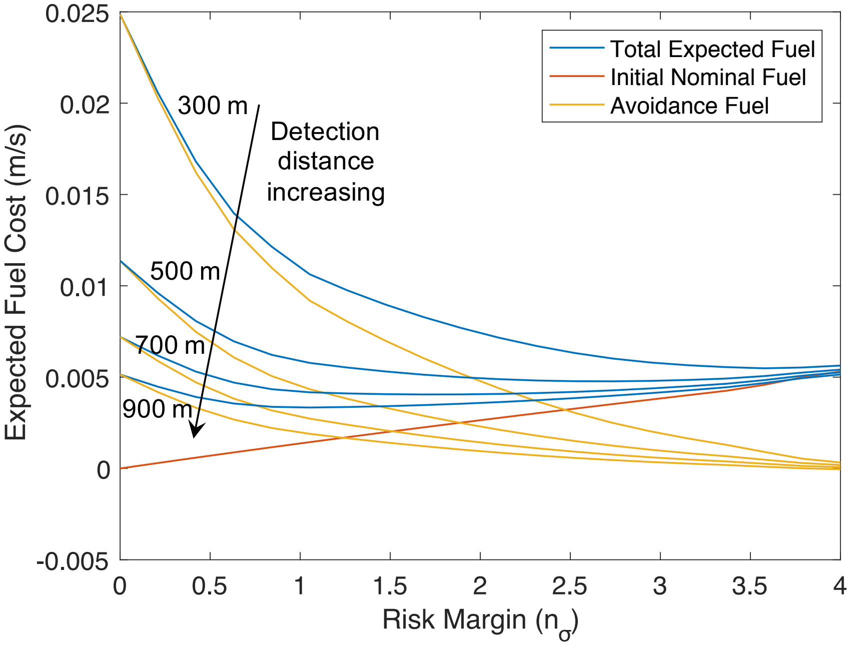


Figure 4-31: Results showing the tradeoff between initial and replanning costs while varying the detection distance for the uncertain target attitude problem

decreasing marginal utility as the detection distance continues to increase and there are not significant gains after about 1 km in this mission scenario. The initial trajectory fuel remains fairly constant in this case study as compared to the obstacle avoidance problem. This fact is a consequence of the minimum fuel problem setup. In the minimum fuel problem, there is not as much gain in replanning the maneuver when the initial trajectory will meet a specific instantiation of the updated LOS zone, because to do so would require a fuel burn. In the minimum energy case, fuel is not weighted as heavily and state error can be reduced significantly by performing different burns in a mission replan. For the minimum fuel, uncertain target attitude problem, negative replanning costs are therefore very rare compared to the minimum energy, obstacle avoidance problem.

Similarly, Figure 4-32 depicts how a variation in the sensor improvement factor, or how much more precise of a measurement is available with onboard sensors, affects the total expected fuel costs. These example results are shown for a detection distance range of 500 m as opposed to the 300 m distance previously used as nominal. The 500 m detection range better shows the behavior relevant to this discussion. Although an increase in the sensor improvement factor reduces the expected avoidance maneuver fuel costs in the range of small n_σ, the reduction in avoidance fuel for larger n_σ is not as significant. As a result,

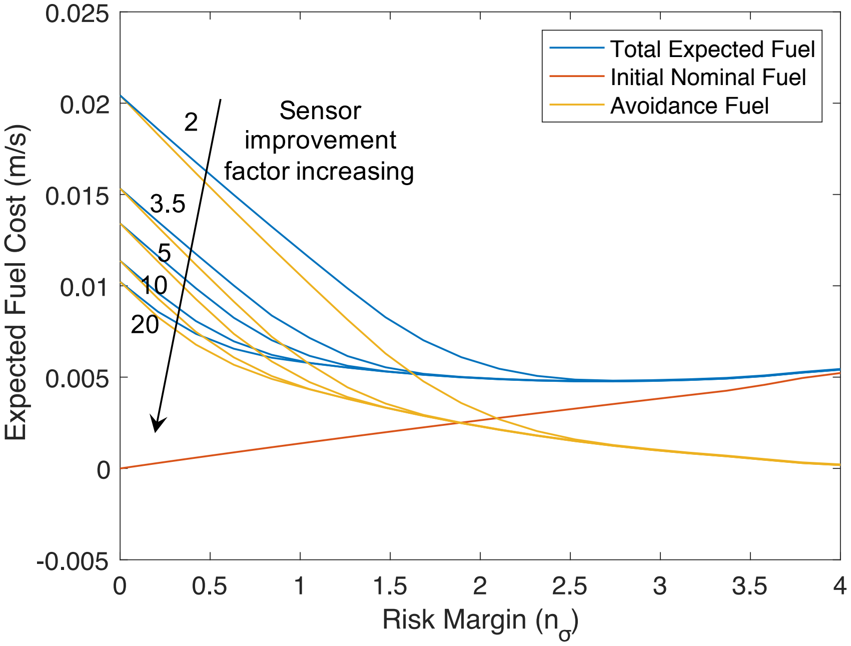


Figure 4-32: Results showing the tradeoff between initial and replanning costs while varying the sensor improvement factor for the uncertain target attitude problem

the total expected fuel does not vary significantly between these sensor precisions. In fact, the minimum n_σ trajectory is not very sensitive to this parameter overall and maintains an optimal value in the vicinity of 2.75 for all sensor improvement factors shown on the plot. Also there is a significant decrease in marginal utility after an improvement factor of 10. In addition to the obstacle avoidance problem, it appears that in this problem the detection range of a sensor has a larger effect on the subsequent replanning maneuvers than does the sensor precision. Thus, a general comment that can be drawn from these results is that detection range should be a premium metric when choosing sensors for spacecraft involved in rendezvous and docking missions.

Overall, in this section, the probabilistic trajectory planning process has been implemented for an uncertain LOS zone constraint problem and is proven to be beneficial at improving performance for minimum fuel formulations as well. In conjunction with the previous section on obstacle avoidance, the first objective of the thesis has been achieved. We have shown that baseline trajectories computed through probabilistic path planning can outperform current techniques given large uncertainties in mission critical parameters and uncertain events. Additionally, in the spirit of the main thesis objective, we have shown a tradeoff between safety and performance and have shown advanced planning can help optimize trajectories under this tradeoff.

Chapter 5

Reference-Tracking Controller Selection

Given the trajectories generated in the previous chapter, the aim of this chapter is to describe how to plan appropriate reference-tracking controllers to follow these trajectories throughout all phases of the mission. These controllers will be designed with the objective of achieving robustness to the uncertainties in the problem, both aleatoric and epistemic, while maintaining a high level of performance. This overall approach will be discussed in more detail in Section 5.1. Each of the investigated controllers will be described in Section 5.2, and the probabilistic method of evaluating the controllers against varying uncertainty distributions is covered in Section 5.3. Finally, the results from applying this methodology to the thesis problem are presented and explained in Section 5.4.

5.1 Overview of Approach

The subsections included in this major section discuss, at a high level, the process used for the reference-tracking controller selection portion of this thesis. First, the concept is motivated and the hypothesis is described in Section 5.1.1. Then, the process developed for pre-mission baseline controller selection is outlined in Section 5.1.2 and specific uncertainties covered in this chapter are described in Section 5.1.3. Last, in Section 5.1.4, the method used to perform this controller scheduling in real-time, onboard the spacecraft is presented. This section is mainly an overview and specific details about the process, the controllers and the metric evaluation will be explained in the following major sections.

5.1.1 Motivation and Hypothesis

As has been covered thoroughly, uncertainty plays an important and potentially detrimental role in spacecraft rendezvous, docking, and joint maneuvering missions. One cannot expect that the optimal control solution consisting of control inputs to be implemented at specific times during the mission will be successful in the presence of unaccounted for process noise or additional unmodeled dynamics. Any error in knowledge of the design model of the system will affect the performance of these open-loop optimal trajectories, thus it is important to account for this error by estimating the spacecraft's state and making corrections based on the error between the optimal trajectory and the current state in a closed-loop manner. The tool that lets us perform these corrections and track the optimal trajectory well in the presence of errors and noise in the system is called a reference-tracking controller.

Throughout the phases of the mission as described in Section 3.2, the spacecraft will experience different levels and types of uncertainty. Different controllers could potentially perform better or worse in specific phases depending on the sources and amount of uncertainty associated with that phase. The question then becomes, how do we choose the best controller for each phase of a mission given different uncertainties and objectives in each phase and a set of controllers with varying performance and robustness?

The hypothesis in this work is that there is a tradeoff between performance and robustness to errors in the knowledge of uncertain parameters used to design the specific reference-tracking controller. In a generalized form, this notional hypothesis is presented in

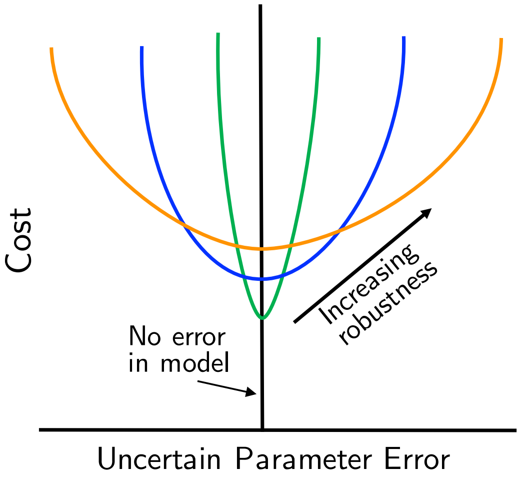


Figure 5-1: Notional tradeoff hypothesized between controller robustness and performance across levels of error in the design model

Figure 5-1. As one tries to increase the robustness of a controller to errors in model parameters, there will be some loss in nominal performance of that controller in cases where there is no error in the model. Hopefully though, the performance will improve over a less robust controller for cases in which there is large parameter error. In Figure 5-1, the green curve has the lowest cost when evaluated against the nominal model, but is more fragile when the error increases such that the cost skyrockets with a small deviation from the nominal plant. The blue curve represents a slightly more robust controller whose cost does not skyrocket as quickly, yet has some overhead associated with this additional range of performance and therefore does not perform as well when there is very little error in the model. The orange curve represents the same behavior, but takes it a step further. In general, controllers designed optimally for the nominal model without considering any error will be very fragile and act like the green curve, while robust and adaptive controllers will begin to look more like the blue and orange curves as they consider the possibility of these model errors.

Given this hypothesis, controllers with different levels of robustness may be appropriate for different levels of uncertainty in specific system parameters. Therefore, the choice of the best controller will likely differ between the phases of the mission, given the varying levels of uncertainty in the different phases. It is the goal of the chapter and the second thesis objective to solve this challenge for a multi-stage rendezvous, docking, and joint maneuvering mission. This problem may sometimes be referred to as controller *scheduling* or controller *selection*. To clearly define the terminology, *scheduling* refers to a selection of a controller for each individual phase, while *selection* is a narrower term only referring to choosing the best controller for one particular phase or for the whole mission. Loosely though, the terms are used interchangeably for the most part.

5.1.2 Selection Process Overview

The process followed here can be used to answer the question raised in the previous motivation section and find the best performing controller for each of the mission phases under their unique assortment of parametric uncertainties and noises. This process involves balancing performance and robustness of the controllers to the appropriate proportion.

To start out, it is very important to understand the particular uncertainties present in the mission and have a general idea of how they will affect the reference-tracking controllers. These uncertainties involved in this problem are discussed in Section 3.3, however only specific ones will be investigated for the chapter as explained in Section 5.1.3. Knowing these uncertainties, it is possible to design specific controllers that each have a varying degree of robustness to the uncertainties. These controllers are grouped into what is known as the controller library. This library and the controllers contained for this thesis are described extensively in Section 5.2.

With both of these components defined, Figure 5-2 proceeds to outline the tasks that make up the full reference-tracking controller selection process. For each of the uncertain parameters of interest to the mission, both an uncertainty probability distribution and range need to be defined. This probability density function (PDF) can be anything desired from uniform to Gaussian to incredibly nonlinear, but should be the best-known representation of the expected error in our knowledge of the uncertain parameter. The uncertainty range should be defined such that the overwhelming majority of the probability distribution is covered, and the integral over this range is essentially equal to one. Expressed in more detail later, these uncertainty distributions and ranges can either be defined in a univariate

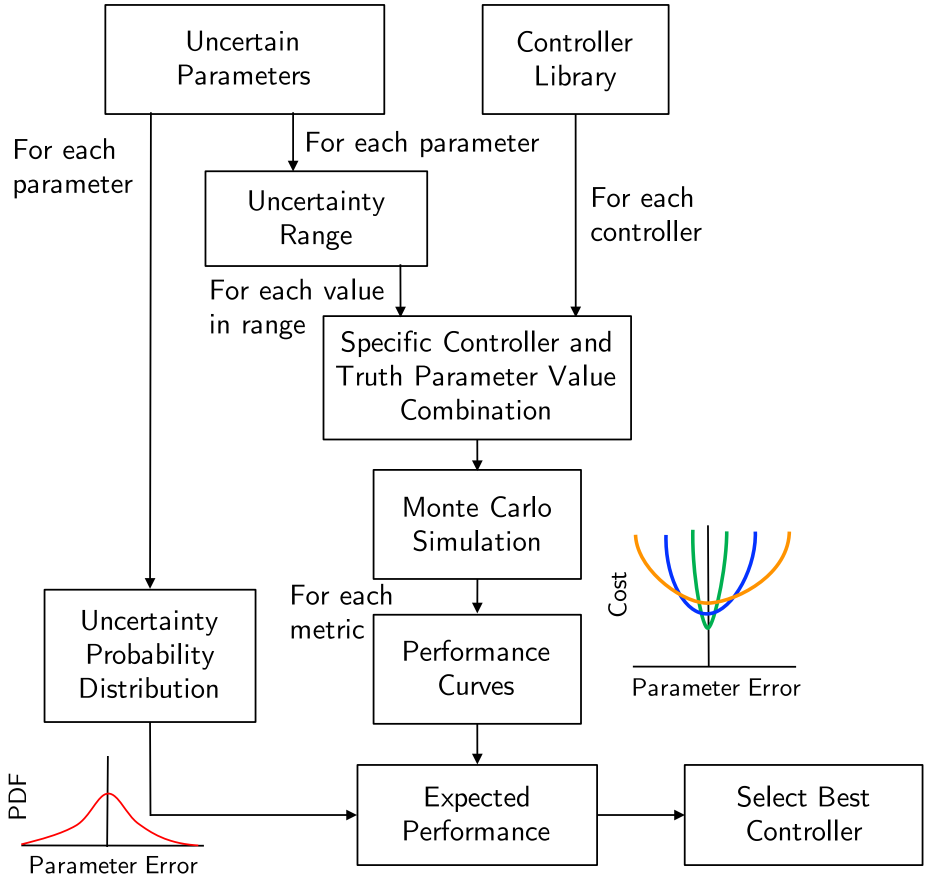


Figure 5-2: Overview of the reference-tracking controller selection process

manner individually or multivariate manner considering multiple uncertainties interacting at once.

Next, the uncertainty range needs to be sampled at a level sufficient for the performance trends and uncertainty distribution to be well captured. Usually a meshed grid will do. Each of these sampled levels of parameter error is paired with a controller from the controller library and then needs to be run through a Monte Carlo simulation to account for the aleatoric noises present. Once all of the combinations of controllers and parameter error instantiations are run through the Monte Carlo simulation, the average results are used to form the performance curves (or performance hypersurfaces in the generalized multivariate case) for each performance metric. This specific Monte Carlo process of computing the performance is discussed in Section 5.3.1. Once the performance hypersurfaces are obtained, they need to be integrated over the PDF associated with these uncertain parameters to compute the probabilistic expected performance for each desired performance metric as explained in Section 5.3.2. With these expected performance evaluations for each of the controllers, all that is left is to determine which controller performs the best in terms of the preferences of the user for particular performance metrics. This selection process is detailed in Section 5.3.3

This whole process should be completed for each phase of the mission, selecting the best controller for each phase. The selection of the four individual controllers that are best for each of the four mission phases is known as the controller schedule. Given the Monte Carlo process involved, it is only reasonable to run through all of these steps prior to the mission for a baseline controller schedule and choice of potentially useful controllers. The real-time, onboard version of this process is simplified and discussed in Section 5.1.4, although still relies on this analysis to be performed before the mission.

Additionally, it serves the mission planning well if the process is run for several levels of uncertainty, such that the designers understand how particular levels of uncertainty in mission critical parameters affect controller performance and how the ranges over which the choice of the best controller will change. The process is amenable to this, as long as the uncertainty range selected early in the process covers all of the uncertainty distributions of interest. If that is the case, the Monte Carlo simulation would not need to be run again to determine performance curves for the new range.

5.1.3 Uncertainties in Dynamics

The classes of uncertainty accounted for in the reference-control design and analyzed in this chapter involve both aleatoric and epistemic uncertainties. With the exception of the unexpected combined spacecraft mass after docking to a target with an uncertain mass, uncertain events are not discussed in the context of pre-mission planning, but only in the context that the onboard controller scheduler is capable of replanning the control selection based on new knowledge obtained from the occurrence of an uncertain event.

Specific aleatoric uncertainties investigated include measurement noise from the angle and range sensors, thruster noise, higher-order orbital dynamics and orbital perturbations. The measurement noise in the angle and range sensors varies throughout the phases as described in Section 3.2.8. In Phase 1a, there is large uncertainty in the angular measurements and no range measurements available. In Phase 1b, there is similarly performing angular measurements although a high-uncertainty range measurement has been added. In Phase 2, very precise angular and range measurements are available. When performing the joint maneuver in Phase 3, we return to the higher-uncertainty range and angular measurements, although now from a different source. The same thruster noise is present during all phases and is represented as a Gaussian 2.5% error added to each thruster firing, as scaled by the magnitude of the commanded force. Additionally, the full nonlinear orbital mechanics model is used in the truth simulation as described in Section 3.1.2, including inverse-square gravity, J2-6, SRP, and 3rd body from the Sun and Moon. Since the mission is in GEO, the atmospheric drag term has negligible impact. The initial target orbit is set to be circular, however, note that the inclination is set to three degrees such that the J2-6 perturbation will be more noticeable. As the mission proceeds, the orbit of the target gradually gains eccentricity. These nonlinearities are implemented to address the modeling errors associated with the linear CWH control form. Because of the aleatoric uncertainties, the Monte Carlo simulation is required such that enough missions can be simulated to statistically determine performance effects and mission success criterion under these random noises affecting each time step of the mission differently.

Epistemic sources of uncertainty, or uncertain parameters, explored in this analysis are limited to the thruster force magnitude and the initial orbital radius of the target spacecraft that defines the CWH frame and dynamics. These two were chosen on purpose, such as to account for two other uncertain parameters and effectively kill four birds with two stones. Taking a look at the CWH equations in (3.1) again, although in an expanded form, we have

$$\begin{aligned} \ddot{x} - 2\sqrt{\frac{\mu}{r_0^3}}\dot{y} - 3\frac{\mu}{r_0^3}x &= \frac{f_{thr}u_x}{m}, \\ \ddot{y} + 2\sqrt{\frac{\mu}{r_0^3}}\dot{x} &= \frac{f_{thr}u_y}{m}, \text{ and} \\ \ddot{z} + \frac{\mu}{r_0^3}z &= \frac{f_{thr}u_z}{m}. \end{aligned} \tag{5.1}$$

Notice that the thruster force magnitude, f_{thr}, has an exact inverse relationship of the mass on the CWH dynamics. Therefore, designing a controller to be robust to an uncertainty in either of these parameters could result in an identical result. Whether the uncertainty lies in the mass or the thruster force level is pretty much irrelevant, so we just choose the thruster force level as it is the fraction of nominal force the spacecraft performs and has nicer scale and no units to worry about. Nominally, $f_{thr} = 1$. If the spacecraft instead only had half of the thrusting force anticipated or had double the mass, for our purposes we understand that f_{thr} would equal 0.5 in both cases.

Also notice in (5.1), that we have replaced the n from (3.1) with its expanded form. By doing this, we can more closely address the actual parameter that the ground sensors are measuring (i.e., the altitude). By looking at r_0, the target's initial orbital radius, we also are investigating changes in the mean motion of the orbit. The math is also a little more fun, since we have the square root of a cubic rather than the simpler CWH representation with the mean motion, n.

We treat these two parameters, f_{thr} and r_0, as epistemic uncertain parameters, meaning that it is possible to determine their true value, but in practice we do not know them. They can be considered biases in a sense, where there could potentially be an error between what value we use to construct the model for the controller and what value exists in reality. These are the uncertain parameters referred to in Figure 5-2 and Section 5.1.2, that will be varied and over which controller performance will be integrated. This means that the controllers will be designed to nominal values of these parameters, but the simulation will add a constant error to the true parameter value and simulate the dynamics with this error incorporated, yet still run the controller which was designed for the nominal values. In this manner, we can see how different controllers behave when run on dynamics that do match their design model. This is key in assessing robustness performance.

Other potential uncertain parameters that could be investigated in future work, but are not investigated this thesis, include the magnitude of the sensor covariances in different phases, the direction thrusters are pointed, constant sensor bias in angle or range, and constant or periodic external forces.

5.1.4 On-line Controller Scheduling

In fitting all of these pieces together, Figure 5-3 shows where the onboard controller scheduling component fits into the larger architecture. Section 5.1.2 details what happens during the controller scheduling and downselection block in this diagram. After that process is completed prior to the mission, the performance curves (or hypersurfaces) are handed to the on-line controller scheduler, so as not to perform the time-consuming task of rerunning all of the Monte Carlo simulation trials with added noises and nonlinear dynamics onboard the spacecraft. Instead, as the uncertainty distributions of mission critical parameters change during a mission, the onboard planner is capable of performing the expected performance computations on-line. These calculations are far less computationally intensive and only require the knowledge of the current uncertainty distribution and the storing of the performance curves (or hypersurfaces) in memory.

Note that this assumes that a significant amount of memory is available to store these

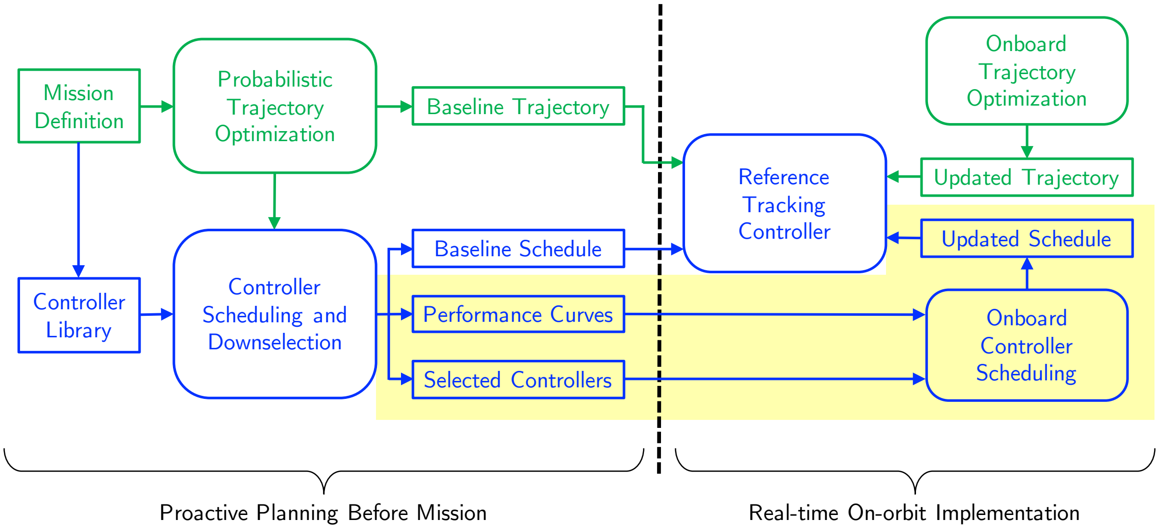

Mission Definition
Probabilistic Trajectory Optimization
Baseline Trajectory
Onboard Trajectory Optimization
Reference Tracking Controller
Updated Trajectory
Updated Schedule
Controller Library
Controller Scheduling and Downselection
Baseline Schedule
Performance Curves
Selected Controllers
Onboard Controller Scheduling
Proactive Planning Before Mission
Real-time On-orbit Implementation

Figure 5-3: Overall process of guidance and controller planning both before and during a mission, with the on-line controller scheduling portion highlighted

performance results. This memory requirement grows depending on how fine the sampling mesh is over the uncertainty range and grows exponentially with the number of uncertain parameters considered. To reduce this requirement, any controllers that were completely dominated in all relevant performance metrics and over all ranges of expected uncertainty can be discarded. Then, only the results from the smaller downselected set of controllers needs to be uploaded to the spacecraft. Additional methods to deal with a growing memory burden would be to only include a set of important performance metric information rather than the full set. Given the initial analysis, areas of the performance hypersurfaces that are well represented by interpolation schemes could also be left out of the upload package.

Although this controller schedule update process could be performed continuously throughout the mission as the uncertainty distributions change, the author does not suggest this implementation method as it could lead to chattering, or the rapid switching back and forth between different controllers. Instead, it is suggested only to plan controllers for the upcoming phases based on the current knowledge. Before an upcoming phase, the current levels of uncertainty could be determined, and the controller scheduling process could be performed, returning the suggested controller to use for the next phase. If significant analysis were performed on the continuous controller scheduling problem to show the architecture was robust to chattering or other unintended behavior, the author anticipates there could be significant performance gains over a mission, however no such analysis is performed or provided in this thesis.

Something to consider amongst all of this is that when an uncertainty distribution changes during a mission, the mean of that distribution will also change most of the time. Because the controllers and the analysis process were designed around the nominal mean values of the uncertain parameters, and now those mean values have changed, the question is raised as to whether these results are still valid. Under certain assumptions they can be. If we assume that the controller itself will not be redesigned to account for the new mean of the distribution, and we are only choosing between the same nominally designed controllers, then this process is perfectly valid. If we instead plan on updating the controllers onboard the spacecraft in real-time to be designed to a model with the new mean parameter values, we must make one of a couple assumptions for this onboard scheduling process to remain valid. One assumption we can make is that the same relative or normalized performance will remain constant about the new mean parameter values. Then, we just need to shift the uncertainty distribution's mean back to the original and only update the covariance in our onboard analysis. Alternately, we could perform analysis prior to the mission to create performance hypersurfaces for a range of possible nominal mean parameter values and then select the appropriate curves onboard the spacecraft. However, this is not a great alternative as the data volume would increase significantly. For the purpose and results from this thesis, we consider either of the first two assumptions to be true, with the third as an unfortunate backup.

5.2 Controller Library

The controller library is comprised of a set of reference-tracking controllers that are specifically designed to handle the dynamics and uncertain parameters of interest to a particular mission. These reference-tracking controllers are each hand designed and tuned appropriately by an engineer to adhere to reasonable stability margins. The library should consist of a varying range of controller performance levels and robustness to uncertainties, such that the controller scheduler has sufficient choices of controllers to cover the span of possible uncertainties throughout the full multi-stage mission. The subsequent subsections each describe in mathematical detail the controllers chosen for analysis in this thesis. At a high

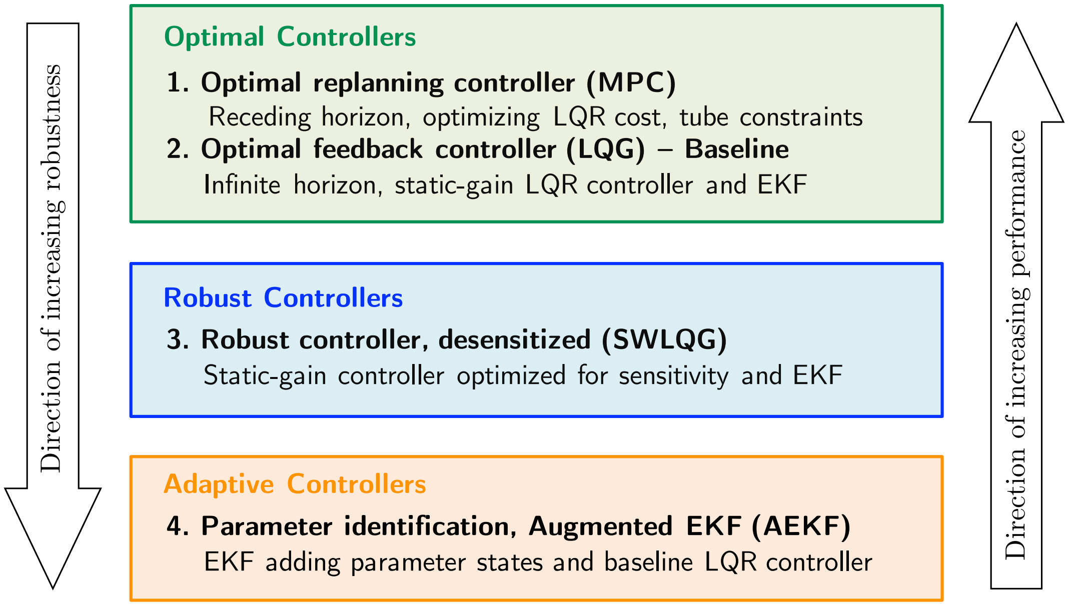

Direction of increasing robustness
Optimal Controllers
1. Optimal replanning controller (MPC)
Receding horizon, optimizing LQR cost, tube constraints
2. Optimal feedback controller (LQG) – Baseline
Infinite horizon, static-gain LQR controller and EKF
Robust Controllers
3. Robust controller, desensitized (SWLQG)
Static-gain controller optimized for sensitivity and EKF
Adaptive Controllers
4. Parameter identification, Augmented EKF (AEKF)
EKF adding parameter states and baseline LQR controller
Direction of increasing performance

Figure 5-4: An example of the controllers contained in the controller library for this thesis and ranking based on nominal performance and robustness to uncertainties

level however, Figure 5-4 briefly describes the controllers implemented for this problem and ranks them based on anticipated nominal performance and robustness. The baseline for all of the analysis will be the Linear-Quadratic-Gaussian (LQG) controller, meaning a Linear-Quadratic Regulator (LQR) paired with an Extended Kalman Filter (EKF). Each of the other controllers implemented make slight tweaks to this design such that the performance goal should remain relatively constant. In this manner, the true comparison is being made between control architectures and design choices rather than the tuning of specific knobs to increase or decrease performance within one controller type.

The controller library spans optimal, robust, and adaptive control types. Model Predictive Control (MPC) is used as the example of a controller using online optimization to improve performance at the expense of increased computation power. The LQG controller is used as the baseline, constant-gain feedback controller. The Sensitivity Weighted LQG (SWLQG) controller is a desensitizing robust controller which also is a constant-gain feedback controller, but is designed to be more robust to changes in the uncertain parameters. Finally, a parameter estimation techniques known as an Augmented EKF (AEKF) is used as a varying-gain feedback controller that optimizes gains every time step based on a current estimate of the uncertain parameters.

These controllers need to be designed in the same format such that they have common inputs and outputs. If designed in this manner, swapping one controller for another will be simple, as it only required to switch the controller function call in the code. Inputs should include the current estimated state, the reference trajectory to be followed, any other updated variables from the estimator or other software modules, and maybe the appropriate mission phase. The output should be the commanded control input for that time step. Internal variables should store things like integral error or other persistent variables. The controllers are each formatted in a manner similar to this in the simulation described in Section 3.4, such that any user can easily swap in and out a specifically designed controller as desired for a particular mission phase.

5.2.1 General Control Setup

This section will discuss the control set-up that several of the controller methods will refer to in the upcoming sections. In this thesis, the standard form of a linear feedback controller in state space will be defined as:

$$
\begin{aligned}
\dot{\mathbf{x}} &= A\mathbf{x} + B_w\mathbf{w} + B_u\mathbf{u} \\
\mathbf{z} &= C_z\mathbf{x} + D_{zw}\mathbf{w} + D_{zu}\mathbf{u} \\
\mathbf{y} &= C_y\mathbf{x} + D_{yw}\mathbf{w} + D_{yu}\mathbf{u}\ ,
\end{aligned}
\tag{5.2}
$$

where $\mathbf{x}$ represents the state vector of positions and velocities in the CWH frame; $\mathbf{z}$ is the vector of performance variables; $\mathbf{y}$ is the measurement vector; $\mathbf{u}$ is the control input to the system; $\mathbf{w}$ is unit Gaussian disturbance noise; and A, B_w, B_u, C_y, C_z, D_{zw}, D_{zu}, D_{yw}, and D_{yu} are matrices of appropriate size that specify the linear system dynamics. In general, all of the controllers in this thesis have $B_w = 0$, $D_{zw} = 0$, and $D_{yu} = 0$, unless directly specified otherwise. In reality, there are nonlinear perturbations to this model and process noise is present through thruster force noise. These terms are ignored in the general controller set-up here, yet do show up in the simulation. Additionally, note that all of the controllers in this chapter are presented for the full 3DOF system as it is simple to decouple the out-of-plane component to reduce to a 2DOF controller if desired.

Unfortunately, we do not have full state feedback, and the measurement models described for each phase of our mission in Section 3.2 are nonlinear, thus the above linear equation for $\mathbf{y}$ is not suitable and we must use the nonlinear form,

$$
\mathbf{y} = h_n\left(\mathbf{x}, \mathbf{v}\right), \tag{5.3}
$$

where $\mathbf{v}$ is the Gaussian measurement noise defined by $\mathrm{E}\left[\mathbf{v}\mathbf{v}^T\right] = R_n$, and where the appropriate models and noise characteristics are defined for the n-th mission phase in Section 3.2.

Additionally, since the controllers will eventually need to be implemented in discrete time, we obtain the general form of the controller to be referenced in the following sections as

$$
\begin{aligned}
\mathbf{x}(k+1) &= A_d\mathbf{x}(k) + B_d\mathbf{u}(k) \\
\mathbf{z}(k) &= C_z\mathbf{x}(k) + D_{zu}\mathbf{u}(k) \\
\mathbf{y}(k) &= h_n\left(\mathbf{x}(k), \mathbf{v}(k)\right),
\end{aligned}
\tag{5.4}
$$

where A_d and B_d are found in Section 3.1.1, k is the current time-step, and T is the discretization time, or time between successive steps k and $k+1$.

The goal of the controller is then to stabilize these dynamics with respect to a reference trajectory defined by $\mathbf{r}(k)$. When designing a linear feedback controller, we are thus looking for a compensator of the form

$$\begin{aligned} \mathbf{x}_c(k+1) &= A_c\mathbf{x}_c(k) + B_c\mathbf{y}(k) \\ \mathbf{u}(k) &= -C_c\mathbf{x}_c(k) \end{aligned}, \tag{5.5}$$

where $\mathbf{x}_c(k)$ is the compensator state at step k which is equal to the error between the state and reference, $\mathbf{x}_c(k) = \mathbf{x}(k) - \mathbf{r}(k)$.

5.2.2 Linear-Quadratic-Gaussian (LQG)

The discrete-time LQG compensator is the combination of a Linear-Quadratic Regulator (LQR) and a Linear-Quadratic Estimator (LQE) or Kalman Filter (KF). The goal when finding an infinite-horizon LQG compensator is to minimize the quadratic cost function,

$$J = \lim_{N\to\infty} \frac{1}{N}\mathrm{E}\left[\sum_{k=1}^{N}\Big(\mathbf{x}(k)^T R_{xx}\mathbf{x}(k) + 2\mathbf{x}(k)^T R_{xu}\mathbf{u}(k) + \mathbf{u}(k)^T R_{uu}\mathbf{u}(k)\Big)\right], \tag{5.6}$$

where N is the length of the horizon, and R_{xx}, R_{uu}, and R_{xu} are weighting matrices for the cost on the state error, control input, and cross-terms respectively. They can be derived in a useful form with the following relationship from [129]:

$$R = \begin{bmatrix} C_z^T \\ D_{zu}^T \end{bmatrix} R_{zz} \begin{bmatrix} C_z & D_{zu} \end{bmatrix} = \begin{bmatrix} R_{xx} & R_{xu} \\ R_{xu}^T & R_{uu} \end{bmatrix} \geq 0,\ R_{uu} > 0, \tag{5.7}$$

where C_z and D_{zu} are defined in the previous section and R_{zz} is a (usually diagonal) relative weighting matrix capable of emphasizing state or control penalty over one another. Additionally, although not directly present in the cost function, we need to know the relative magnitudes of the process and measurement noise to determine the expectation of the cost. These are defined in the following relationship from [129]:

$$V = \begin{bmatrix} B_w \\ D_{yw} \end{bmatrix} V_{ww} \begin{bmatrix} B_w^T & D_{yw}^T \end{bmatrix} = \begin{bmatrix} V_{xx} & V_{xy} \\ V_{xy}^T & V_{yy} \end{bmatrix} \geq 0,\ V_{yy} > 0, \tag{5.8}$$

where B_w and D_{yw} are defined in the previous section and V_{ww} is a (usually diagonal) relative weighting matrix capable of emphasizing process or measurement noise over one another.

Derived from [205], the solution to the discrete-time LQG compensator problem takes the form of (5.5), where

$$\begin{aligned}
A_c &= A_d - B_d K - L C_y \\
B_c &= L \\
C_c &= K \\
K &= \left(B_d^T P B_d + R_{uu}\right)^{-1} \left(B_d^T P A_d + R_{xu}^T\right) \\
L &= \left(A_d Q C_y^T + V_{xy}\right) \left(C_y Q C_y^T + V_{yy}\right)^{-1},
\end{aligned} \tag{5.9}$$

where P and Q solve the steady-state, Discrete-time Algebraic Riccati Equations (DARE):

$$\begin{aligned}
P &= A_d^T P A_d - \left(A_d^T P B_d + R_{xu}\right) \left(R_{uu} + B_d^T P B_d\right)^{-1} \left(B_d^T P A_d + R_{xu}^T\right) + R_{xx}, \\
Q &= A_d Q A_d^T - \left(A_d Q C_y^T + V_{xy}\right) \left(V_{yy} + C_y Q C_y^T\right)^{-1} \left(C_y Q A_d^T + V_{xy}^T\right) + V_{xx}.
\end{aligned} \tag{5.10}$$

The separation principle shows that we can design the estimator and controller separately using the above equations as the stability of the estimator and feedback controller are independent. We do just this.

The LQR feedback controller is designed using (5.9) and (5.10) and implemented as shown in (5.5). The C_z and D_{zu} matrices are chosen to abide to Bryson's Rules for the formation of LQR weighting matrices. Bryson's Rules tailor the R_{xx} and R_{uu} matrices through normalization of the signals by their maximum expected value and additional relative weightings if desired. The goal is to obtain

$$R_{xx} = \begin{bmatrix} \frac{\alpha_x^2}{x_{max}^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\alpha_y^2}{y_{max}^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\alpha_z^2}{z_{max}^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\alpha_{\dot{x}}^2}{\dot{x}_{max}^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\alpha_{\dot{y}}^2}{\dot{y}_{max}^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\alpha_{\dot{z}}^2}{\dot{z}_{max}^2} \end{bmatrix} \text{ and } R_{uu} = \begin{bmatrix} \frac{\beta_x^2}{(u_x)_{max}^2} & 0 & 0 \\ 0 & \frac{\beta_y^2}{(u_y)_{max}^2} & 0 \\ 0 & 0 & \frac{\beta_z^2}{(u_z)_{max}^2} \end{bmatrix}, \tag{5.11}$$

where the sum of the α's is one, the sum of the β's is one, and the maximum state error and control input on each axis are specified. This results in

$$C_z = \begin{bmatrix} \frac{\alpha_x}{x_{max}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\alpha_y}{y_{max}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\alpha_z}{z_{max}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\alpha_{\dot{x}}}{\dot{x}_{max}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\alpha_{\dot{y}}}{\dot{y}_{max}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\alpha_{\dot{z}}}{\dot{z}_{max}} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } D_{zu} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{\beta_x}{(u_x)_{max}} & 0 & 0 \\ 0 & \frac{\beta_y}{(u_y)_{max}} & 0 \\ 0 & 0 & \frac{\beta_z}{(u_z)_{max}} \end{bmatrix}. \tag{5.12}$$

For the implementation in this thesis, the α's are chosen as $\sqrt{1/6}$ and the β's as $\sqrt{1/3}$ to maintain an equal weighting on the importance of each axis of control. Also, the maximum position and velocity error expected varies depending on the mission phase and is set roughly equal to the standard deviation of the noise defined in Section 3.2.8. The maximum control input is constrained to 10 Newtons, so this value is used in the above matrices. On a final note, the R_{zz} matrix is set up to weigh the control input 100 times as costly as the state error.

Unfortunately, as mentioned previously, we have nonlinear measurements, and thus cannot implement the LQE portion of this controller as described above. Instead, we implement the discrete-time EKF from [206].

The EKF uses nonlinear dynamics and measurements to propagate the state and compute the predicted measurement, but then linearizes these dynamics and measurements at each time step to perform the normal KF measurement update steps and covariance propagation. The process starts by linearizing the nonlinear dynamics and measurements,

$$\begin{gathered} \mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k), T) \text{ and} \\ \mathbf{y}(k) = h_n(\mathbf{x}(k), \mathbf{v}(k)), \end{gathered} \tag{5.13}$$

where the measurements, $\mathbf{y}$, are different for each mission phase n as defined in Section 3.2.

To proceed we want to find the linearizations,

$$\tilde{A}(\hat{\mathbf{x}}(k|k-1)) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k-1)} \qquad \text{and} \qquad \tilde{C}(\hat{\mathbf{x}}(k|k-1)) = \left.\frac{\partial h_n}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k-1)}. \tag{5.14}$$

The continuous dynamics are already linear as shown in (5.2) so $\tilde{A}(\hat{\mathbf{x}}(k)) = A_d$ and is linear time-invariant (LTI) as described in Section 3.1.1. The linearization of Phase 2 and Phase 3 measurements is sufficient to show the form for the other phases. In Mission Phase 2, we have

$$\tilde{C}_2(\hat{\mathbf{x}}(k|k-1)) = \left.\begin{bmatrix} \frac{-y}{x^2+y^2} & \frac{x}{x^2+y^2} & 0 & 0 & 0 & 0 \\ \frac{-xz}{\rho^3\left(\sqrt{1-(z/\rho)^2}\right)} & \frac{-yz}{\rho^3\left(\sqrt{1-(z/\rho)^2}\right)} & \frac{1}{\rho}\sqrt{1-(z/\rho)^2} & 0 & 0 & 0 \\ \frac{x}{\rho} & \frac{y}{\rho} & \frac{z}{\rho} & 0 & 0 & 0 \end{bmatrix}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k-1)}, \tag{5.15}$$

and in Mission Phase 3, we have

$$\tilde{C}_3(\hat{\mathbf{x}}(k|k-1)) = \left.\begin{bmatrix} \frac{r_y}{r_x^2+r_y^2} & \frac{-r_x}{r_x^2+r_y^2} & 0 & 0 & 0 & 0 \\ \frac{-r_x r_z}{r^3\left(\sqrt{1-(r_z/r)^2}\right)} & \frac{-r_y r_z}{r^3\left(\sqrt{1-(r_z/r)^2}\right)} & \frac{1}{r}\sqrt{1-(r_z/r)^2} & 0 & 0 & 0 \\ \frac{r_x}{r} & \frac{r_y}{r} & \frac{r_z}{r} & 0 & 0 & 0 \end{bmatrix}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k-1)}, \tag{5.16}$$

where the only notable differences are the sign changes in the first row and the fact that, in Phase 3, the chaser is getting measurements as the relative distance to a partner spacecraft instead of to the target spacecraft. The definitions for $r_x,\ r_y,\ r_z,\ r$ and ρ are in Section 3.2.

With these linearizations, we can now show the individual steps of the EKF. To start, we propagate the previous estimate to the current time step using

$$\hat{\mathbf{x}}(k|k-1) = A_d\mathbf{x}(k-1|k-1) + B_d\mathbf{u}(k-1), \tag{5.17}$$

and propagate the previous covariance matrix to the current time step using

$$Q(k|k-1) = A_d Q(k-1|k-1)A_d^T + V_{xx}. \tag{5.18}$$

Next, we compute the predicted measurement by feeding the propagated state, $\hat{\mathbf{x}}(k|k-1)$, into the nonlinear measurement equations. We subtract this from the real measurement to get a term known as the innovation, then multiply the innovation by the Kalman gain, $L(k)$, and add the result to the propagated estimate to get our new estimate. This process can be represented in a set of equations as

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + L(k)\left[\mathbf{y}(k) - h_n(\hat{\mathbf{x}}(k|k-1)\right], \text{ with} \tag{5.19}$$

$$L(k) = Q(k|k-1)\tilde{C}_n^T(\hat{\mathbf{x}}(k|k-1))\left[\tilde{C}_n(\hat{\mathbf{x}}(k|k-1))Q(k|k-1)\tilde{C}_n^T(\hat{\mathbf{x}}(k|k-1)) + V_{yy}\right]^{-1}. \tag{5.20}$$

Finally, we update the covariance with the recent measurement by setting

$$Q(k|k) = \left[I - L(k)C_n(\hat{\mathbf{x}}(k|k-1))\right]Q(k|k-1), \tag{5.21}$$

where I is the 6×6 identity matrix. Note that although the dynamics do not involve process noise, we fictitiously create a small amount of noise such that we do have a nonzero V_{xx}. Correspondingly, the filter will always have a base level of uncertainty, as V_{xx} is a constant additive matrix during covariance propagation. This base level of uncertainty will help push the filter out of stagnation when otherwise it could potentially get stuck at an incorrect estimate as the covariance converges to zero.

The estimated state from the EKF is passed to the LQR controller. The LQR controller computes the error and uses the gains calculated above to control the system. The control inputs to the system at that time step are passed to the EKF and the process repeats.

Although the LQG framework guarantees stability for the plant as designed, it does not have any guarantees on robustness or stability margins. Additionally, it is a steady-state optimal feedback controller, and thus performance could be improved if optimizing for a specific horizon length. Different approaches, each using these LQG gains and framework as a baseline, are discussed in the following sections. The goal will be to compare these control techniques across the same performance metrics, where each controller should be designed with the same baseline performance as described in this section. The aim would be to objectively compare control techniques rather than tuning parameters to make one type better than another.

5.2.3 Sensitivity Weighted Linear-Quadratic-Gaussian (SWLQG)

The Sensitivity Weighted LQG (SWLQG) compensator falls directly out of the LQG formulation and is known as a desensitizing robust control technique, as there are no guarantees on stability margins and other robustness metrics. The goal of SWLQG is to minimize the sensitivity of a compensator to uncertain parameters in the system dynamics, thereby making the closed-loop system more robust to these uncertainties. The sensitivity of the dynamics to the uncertain parameters is added to the LQG cost function and the objective is to minimize this combined cost. The following implementation is similar to that shown in [129], except in discrete-time. A first-order sensitivity can be computed by the partial derivative of the state with respect to a particular uncertain parameter. The SWLQG cost function for a single uncertain parameter becomes

$$J_s = \lim_{N\to\infty} \frac{1}{N} \mathrm{E}\left[\sum_{k=1}^{N} \left(\mathbf{x}(k)^T R_{xx}\mathbf{x}(k) + \frac{\partial \mathbf{x}(k)}{\partial s}^T R_{ss}\frac{\partial \mathbf{x}(k)}{\partial s} + 2\mathbf{x}(k)^T R_{xu}\mathbf{u}(k) + \mathbf{u}(k)^T R_{uu}\mathbf{u}(k)\right)\right], \tag{5.22}$$

where s is the particular uncertain parameter present in the dynamics, and R_{ss} is a weighting matrix that can affect the importance of the sensitivity in comparison to the other metrics. The remaining is equivalent to the LQG cost function in (5.6).

As shown in [129], this expression can be reorganized in a very convenient form of

$$J_s = \lim_{N\to\infty} \frac{1}{N} \mathrm{E}\left[\sum_{k=1}^{N} \left(\mathbf{x}(k)^T R'_{xx}\mathbf{x}(k) + 2\mathbf{x}(k)^T R'_{xu}\mathbf{u}(k) + \mathbf{u}(k)^T R'_{uu}\mathbf{u}(k)\right)\right], \tag{5.23}$$

where

$$\begin{aligned} R'_{xx} &= R_{xx} + \frac{\partial A_d}{\partial s}^T A_d^{-T} R_{ss} A_d^{-1} \frac{\partial A_d}{\partial s}, \\ R'_{xu} &= R_{xu} + \frac{\partial A_d}{\partial s}^T A_d^{-T} R_{ss} A_d^{-1} \frac{\partial B_d}{\partial s}, \text{ and} \\ R'_{uu} &= R_{uu} + \frac{\partial B_d}{\partial s}^T A_d^{-T} R_{ss} A_d^{-1} \frac{\partial B_d}{\partial s}, \end{aligned} \tag{5.24}$$

and correspondingly for the LQE problem,

$$\begin{aligned} V'_{xx} &= V_{xx} + \frac{\partial A_d}{\partial s} A_d^{-1} V_{ss} A_d^{-T} \frac{\partial A_d}{\partial s}^T, \\ V'_{xy} &= V_{xy} + \frac{\partial A_d}{\partial s} A_d^{-1} V_{ss} A_d^{-T} \frac{\partial C_y}{\partial s}^T, \text{ and} \\ V'_{yy} &= V_{yy} + \frac{\partial C_y}{\partial s} A_d^{-1} V_{ss} A_d^{-T} \frac{\partial C_y}{\partial s}^T. \end{aligned} \tag{5.25}$$

The formulation in (5.23) is precisely the same as the LQG cost function. Only the weighting matrices have changed as described in (5.24) and (5.25). This allows us to use the same methods as described in Section 5.2.2 to solve for the appropriate LQR and EKF gains, only changing the weighting matrices.

So far, the SWLQG problem has only been presented for the presence of one uncertain parameter. However, due to the modular nature of adding the sensitivity to the cost function, it is simple to expand this approach to multiple uncertain parameters. For multiple parameters, the cost function becomes

$$J_s = \lim_{N\to\infty} \frac{1}{N} \mathrm{E}\left[\sum_{k=1}^{N} \left\{ \mathbf{x}(k)^T R_{xx} \mathbf{x}(k) + \sum_{i=1}^{n_s} \left(\frac{\partial \mathbf{x}(k)}{\partial s_i}^T R_{ss_i} \frac{\partial \mathbf{x}(k)}{\partial s_i} \right) + 2\mathbf{x}(k)^T R_{xu} \mathbf{u}(k) + \mathbf{u}(k)^T R_{uu} \mathbf{u}(k) \right\}\right], \tag{5.26}$$

where n_s is the number of uncertain parameters. This again can be simplified to (5.23), but with

$$\begin{aligned} R_{xx}^{'} &= R_{xx} + \sum_{i=1}^{n_s} \frac{\partial A_d}{\partial s_i}^T A_d^{-T} R_{ss_i} A_d^{-1} \frac{\partial A_d}{\partial s_i}, \\ R_{xu}^{'} &= R_{xu} + \sum_{i=1}^{n_s} \frac{\partial A_d}{\partial s_i}^T A_d^{-T} R_{ss_i} A_d^{-1} \frac{\partial B_d}{\partial s_i}, \\ R_{uu}^{'} &= R_{uu} + \sum_{i=1}^{n_s} \frac{\partial B_d}{\partial s_i}^T A_d^{-T} R_{ss_i} A_d^{-1} \frac{\partial B_d}{\partial s_i}, \\ V_{xx}^{'} &= V_{xx} + \sum_{i=1}^{n_s} \frac{\partial A_d}{\partial s_i} A_d^{-1} V_{ss_i} A_d^{-T} \frac{\partial A_d}{\partial s_i}^T, \\ V_{xy}^{'} &= V_{xy} + \sum_{i=1}^{n_s} \frac{\partial A_d}{\partial s_i} A_d^{-1} V_{ss_i} A_d^{-T} \frac{\partial C_y}{\partial s_i}^T, \text{ and} \\ V_{yy}^{'} &= V_{yy} + \sum_{i=1}^{n_s} \frac{\partial C_y}{\partial s_i} A_d^{-1} V_{ss_i} A_d^{-T} \frac{\partial C_y}{\partial s_i}^T. \end{aligned} \tag{5.27}$$

For this particular problem, we will explore three uncertain parameters: the orbital radius of the target spacecraft, the thrust force magnitude, and the mass of the target spacecraft. These parameters are directly present in the dynamics discussed in Section 5.2.1. First, the orbital radius of the target spacecraft, r_0, is present in the dynamics in the mean motion parameter, $n = \sqrt{\mu/r_0^3}$, which comes up several times in A_d and B_d. Thus, computing the partial derivatives of A_d and B_d with respect to the target's orbital radius yields:

$$\frac{\partial A_d}{\partial r_0} = \begin{bmatrix} -\frac{9nT}{2r_0}\mathrm{s} & 0 & 0 & \frac{3}{2nr_0}\left(\mathrm{s} - nT\mathrm{c}\right) & -\frac{3}{nr_0}\left(nT\mathrm{s} + \mathrm{c} - 1\right) & 0 \\ -\frac{9nT}{r_0}\left(\mathrm{c} - 1\right) & 0 & 0 & \frac{3}{nr_0}\left(nT\mathrm{s} + \mathrm{c} - 1\right) & \frac{6}{nr_0}\left(\mathrm{s} - nT\mathrm{c}\right) & 0 \\ 0 & 0 & \frac{3nT}{2r_0}\mathrm{s} & 0 & 0 & \frac{3}{2nr_0}\left(\mathrm{s} - nT\mathrm{c}\right) \\ -\frac{9n}{2r_0}\left(\mathrm{s} + nT\mathrm{c}\right) & 0 & 0 & \frac{3nT}{2r_0}\mathrm{s} & -\frac{3nT}{r_0}\mathrm{c} & 0 \\ \frac{9n}{r_0}\left(nT\mathrm{s} - \mathrm{c} + 1\right) & 0 & 0 & \frac{3nT}{r_0}\mathrm{c} & \frac{6nT}{r_0}\mathrm{s} & 0 \\ 0 & 0 & \frac{3n}{2r_0}\left(\mathrm{s} + nT\mathrm{c}\right) & 0 & 0 & \frac{3nT}{2r_0}\mathrm{s} \end{bmatrix}, \tag{5.28}$$

$$\frac{\partial B_d}{\partial r_0} = \frac{f_{thr}}{(m_c + m_t)} \begin{bmatrix} -\frac{3}{2n^2 r_0}\left(nT\mathrm{s} - 2\mathrm{c} + 2\right) & \frac{3}{n^2 r_0}\left(nT - 2\mathrm{s} + nT\mathrm{c}\right) & 0 \\ -\frac{3}{n^2 r_0}\left(nT - 2\mathrm{s} + nT\mathrm{c}\right) & -\frac{6}{n^2 r_0}\left(-2 + nT\mathrm{s} + 2\mathrm{c}\right) & 0 \\ 0 & 0 & -\frac{3}{2n^2 r_0}\left(nT\mathrm{s} - 2\mathrm{c} + 2\right) \\ \frac{3}{2nr_0}\left(\mathrm{s} - nT\mathrm{c}\right) & -\frac{3}{nr_0}\left(nT\mathrm{s} + \mathrm{c} - 1\right) & 0 \\ \frac{3}{nr_0}\left(nT\mathrm{s} + \mathrm{c} - 1\right) & \frac{6}{nr_0}\left(\mathrm{s} - nT\mathrm{c}\right) & 0 \\ 0 & 0 & \frac{3}{2nr_0}\left(\mathrm{s} - nT\mathrm{c}\right) \end{bmatrix}, \tag{5.29}$$

where c represents $\cos(nT)$ and s represents $\sin(nT)$.

Secondly, the thruster force magnitude is only present in the B_d matrix and is essentially a scalar multiple of the entire matrix, so taking the partial derivative yields

$$\frac{\partial B_d}{\partial f_{thr}} = \frac{1}{(m_c + m_t)} \begin{bmatrix} \frac{1}{n^2}\left(1 - \mathrm{c}\right) & \frac{2}{n^2}\left(nT - \mathrm{s}\right) & 0 \\ -\frac{2}{n^2}\left(nT - \mathrm{s}\right) & \frac{4}{n^2}\left(1 - \mathrm{c}\right) - \frac{3}{2}T^2 & 0 \\ 0 & 0 & \frac{1}{n^2}\left(1 - \mathrm{c}\right) \\ \frac{1}{n}\mathrm{s} & \frac{2}{n}\left(1 - \mathrm{c}\right) & 0 \\ -\frac{2}{n}\left(1 - \mathrm{c}\right) & \frac{4}{n}\mathrm{s} - 3T & 0 \\ 0 & 0 & \frac{1}{n}\mathrm{s} \end{bmatrix}. \tag{5.30}$$

Thirdly, the target spacecraft mass is also only present in the B_d matrix, and because it affects the dynamics in a very similar manner to the thruster force magnitude, is also a scalar multiple of the entire matrix. Taking the partial derivative yields

$$\frac{\partial B_d}{\partial m_t} = -\frac{f_{thr}}{(m_c+m_t)^2}\begin{bmatrix} \frac{1}{n^2}(1-\mathrm{c}) & \frac{2}{n^2}(nT-\mathrm{s}) & 0 \\ -\frac{2}{n^2}(nT-\mathrm{s}) & \frac{4}{n^2}(1-\mathrm{c})-\frac{3}{2}T^2 & 0 \\ 0 & 0 & \frac{1}{n^2}(1-\mathrm{c}) \\ \frac{1}{n}\mathrm{s} & \frac{2}{n}(1-\mathrm{c}) & 0 \\ -\frac{2}{n}(1-\mathrm{c}) & \frac{4}{n}\mathrm{s}-3T & 0 \\ 0 & 0 & \frac{1}{n}\mathrm{s} \end{bmatrix}. \tag{5.31}$$

These results can be used directly in (5.27) to compute the desired weighting matrices for the LQR solution process described in Section 5.2.2. Notably, the partials of A_d will also be needed for the EKF weighting matrices, however since the measurements do not depend on these parameters, the partials of C_y will be zero and only the V_{xx} matrix will be modified for SWLQG.

As mentioned earlier, SWLQG is a form of desensitizing robust control and does not have guarantees on robustness and stability margins. That being said, it does limit the impact of particular uncertain parameters by desensitizing the controller's response to errors corresponding to these type of dynamics. For the most part however, it turns out the dynamics are not particularly sensitive to these parameters for our mission in geosynchronous orbit. Effectively, these additions to the LQG weighting matrices act to scale the control gains lower as a function of how large R_{ss} is set. There are some minor changes elsewhere in the control gains matrix, but predominantly this scaling is the effect. The result is a controller very similar in appearance and behavior to the baseline SWLQG.

5.2.4 Model Predictive Control (MPC)

MPC, sometimes known as receding horizon control, is an optimal control technique that solves full optimization problems in the real-time control loop. The concept starts by defining a discretized prediction horizon over which to optimize the state and control input with respect to a quadratic cost function. The optimal trajectory and the optimal control input required to enact the trajectory are computed, such that a vector of future states and a vector of future control inputs are planned. The controller then executes a set number of these planned control inputs. Usually, this is only the first step of the control input plan, however depending on the time required to solve the optimization problem and the speed of the dynamics in the problem, sometimes several steps are carried out. After this set of control inputs is implemented, the optimal control problem is solved again and the same number of control input steps are carried out. This process repeats itself until the state is controlled within a tolerance of the goal state. So, in typical use, the process loops between solving the optimal control problem and implementing the first step of this solution at each time step, marching forward in time with the same prediction horizon length. Typically, an EKF will be used in conjunction with this controller, and we will use the one described in Section 5.2.2.

In the problem posed in this thesis, we are attempting to track a reference trajectory that has been computed previously. Thus, the reference-tracking form of MPC is used here, where the goal is to optimize the state error with respect to the reference trajectory, rather than the state to the origin as in the typical MPC formulation. The objective of this MPC reference-tracking problem is to find the

$$\begin{aligned}\min_{\mathbf{U}} \Bigg\{ \Bigg(\sum_{i=k}^{k+N-1} \Big(\mathbf{x}(i) - \mathbf{r}(i)\Big)^T R_{xx} \Big(\mathbf{x}(i) - \mathbf{r}(i)\Big) + \mathbf{u}(i)^T R_{uu} \mathbf{u}(i) \Bigg) \\ + \Big(\mathbf{x}(k+N) - \mathbf{r}(k+N)\Big)^T P \Big(\mathbf{x}(k+N) - \mathbf{r}(k+N)\Big) \Bigg\}\end{aligned} \tag{5.32}$$

$$\begin{aligned}\text{subject to} \quad \mathbf{x}(i+1) &= A_d \mathbf{x}(i) + B_d \mathbf{u}(i), \ \forall i = k, \ \dots \ , k+N-1, \\ ||\mathbf{u}(i)||_\infty &\leq \bar{u}, \ \forall i = k, \ \dots \ , \ k+N-1, \\ \mathbf{x}(i) &\leq \mathbf{r}(i) + \Delta_{max}, \ \forall i = k+1, \ \dots \ , k+N, \text{ and} \\ \mathbf{x}(i) &\geq \mathbf{r}(i) - \Delta_{max}, \ \forall i = k+1, \ \dots \ , k+N,\end{aligned} \tag{5.33}$$

in which $\mathbf{U} = \Big[\mathbf{u}(k)^T, \quad \mathbf{u}(k+1)^T, \quad \dots, \quad \mathbf{u}(k+N-1)^T\Big]^T$ is the $3N \times 1$ stacked vector of control input, $\mathbf{x}(k)$ is the chaser's current position, $\mathbf{x}(i)$ is the chaser's planned position at the i-th step of the N-step horizon, $\mathbf{r}(i)$ is the reference position at the i-th step, R_{xx} is the tracking error cost weighting matrix from Section 5.2.2, R_{uu} is the control cost weighting matrix from Section 5.2.2, and P is the solution to the discrete algebraic Riccati equation for the unconstrained, steady-state LQR problem from Section 5.2.2. For all but the last step in the trajectory, the cost is a function of a combination of reference tracking error and applied control. For the last step, there is a penalty based upon the state only, as no control is computed for this step. The first constraint upholds the dynamics. The second sets a maximum commanded control force on a single axis, which is specified by $\bar{u}$ as defined in Section 3.2.6. The last two constraints impose what are known as tube-constraints on the position and velocity, where Δ_{max} is a vector describing the maximum desired deviance from the reference trajectory for each state.

This objective function is very similar to the one in the trajectory optimization problem in Section 4.1.1, and as such can be simplified to the same expression in (4.6) and solved via a quadratic solver. However, in Section 4.1.1, we did not consider the reference tracking form of the problem and implemented different constraints. Specifically, from [200], the tube constraints can be implemented in linear form as

$$\begin{bmatrix} \Omega \\ -\Omega \end{bmatrix} \mathbf{U} \geq \begin{bmatrix} \mathbf{X}_{max} - \Psi \mathbf{x}(k) \\ -\mathbf{X}_{min} + \Psi \mathbf{x}(k) \end{bmatrix}, \quad \text{where}$$

$$\mathbf{X}_{max} = \begin{bmatrix} \mathbf{r}(k+1) + \Delta_{max} \\ \vdots \\ \mathbf{r}(k+N) + \Delta_{max} \end{bmatrix} \quad \text{and} \quad \mathbf{X}_{min} = \begin{bmatrix} \mathbf{r}(k+1) - \Delta_{max} \\ \vdots \\ \mathbf{r}(k+N) - \Delta_{max} \end{bmatrix}, \tag{5.34}$$

while the $6N \times 3N$ Ω matrix and $6N \times 6$ Ψ matrix are defined in Section 4.1.1 and are a consequence of simplifying the optimization problem for a quadratic solver.

For the implementation here, a horizon length of $N = 6$ was chosen through trial and error to reduce the computational burden and still constrain the control sufficiently. The tube constraint magnitudes vary throughout different phases from 1 m in position and 5 cm/s in velocity in Phases 1 and 3 to 10 cm and 1 cm/s in Phase 2. The tube constraint for the last step in the horizon is shrunk by a factor of 10 in order to constrain the planned motion closer to the trajectory without over-constraining the previous steps in the horizon.

Unfortunately, due to parameter uncertainty and other disturbances in the dynamics it cannot be guaranteed that, with these tube constraints, the optimization problem posed in (5.32) will have feasible results. Therefore, in the event that the solver returns an infeasible solution, a scheme is implemented to increase the size of the tube constraints slowly until a feasible solution can be found. Generally, this is known in the literature as constraint tightening MPC, and sometimes works in the reverse direction as described in this section [207]. In its basic form, this scheme scales the tube constraints from the previous values each time the solver returns an infeasible result, always starting from the base values on a new optimization step in the MPC horizon. Thus, the smallest possible feasible tube constraints will be met for cases in which the original problem is infeasible. Since the upscaling is small, this technique can be computationally intensive and could be dramatically improved. Nonetheless, it proves quite effective.

The quadratic solver is able to solve this MPC problem in roughly 0.1 s on average, even considering the constraint tightening iteration described in the previous section. Because this solution time is much smaller than the step size in our horizon, we can implement the form of MPC where only the first planned control input is implemented and then the problem is re-optimized on the next time step.

MPC is expected to be able to outperform a nominal LQG feedback controller, because in essence it solves the finite-horizon form of the LQR problem which will be able to better optimize the cost when compared to constant-gain, infinite-horizon feedback. The same weighting matrices are used in this optimization as were used for the LQR problem and the same EKF is estimating the state. Naturally, with this expected increase in performance, there is a tradeoff; computation time increases dramatically by a few orders of magnitude, and the robustness to uncertainties will drop significantly. Therefore, MPC is the high performance, low robustness option in the controller library.

5.2.5 Parameter Estimation using an Augmented Extended Kalman Filter (AEKF)

Parameter estimation or parameter identification is a technique frequently used in adaptive control to help determine the values of specific uncertain constants in the dynamics of the system. The form of adaptive control used here relies on augmenting the EKF states with the uncertain parameters, estimating the values of these parameters, and then sending them to the control law to recompute controller gains based on the new plant information. The parameter identification part of the solution is known as an AEKF and is described fully in [208], while the controller being used is the LQR controller described in Section 5.2.2 but updated every time step with the current estimate of the parameters. So, in general this adaptive control technique will be referred to as AEKF in shorthand for the remainder of the thesis, fully knowing that AEKF technically only describes the method of parameter estimation.

For this implementation, we only consider two uncertain parameters: the target's orbital radius and the magnitude of the thruster force. Because both the mass and thruster force magnitude affect the problem in a similar manner (one exactly inversely of the other), this will not allow the AEKF to estimate both simultaneously. By only choosing one of these uncertainties to estimate, we are effectively estimating the net effects of both parameters. As thruster force magnitude is scaled in a nicer manner, being unitary as default, we choose it. Alternately one could call the ratio between the thruster force magnitude and mass the uncertain variable and achieve similar results.

In this application, and for the parameters chosen, we do not receive direct measurements of any of the uncertain parameters and will need to rely on the cross-terms in the dynamics to help us identify the values of these parameters. We choose the dynamics of these parameters to be a persistence model with a small amount of additive Gaussian noise such that the parameter estimate can perform a random walk and essentially trust the estimate more as it moves in the direction of true value. With larger controlled movements, the problem becomes richer in terms of the signal-to-noise ratio and the estimate will perform better, while under slow and small control input, the estimate will suffer.

The EKF state is augmented to become $\hat{\mathbf{x}}^a$ and propagates according to the equation,

$$\begin{aligned} \hat{\mathbf{x}}^a(k+1|k) &= \begin{bmatrix} \hat{\mathbf{x}}(k+1|k) \\ r_0(k+1|k) \\ f_{thr}(k+1|k) \end{bmatrix} = \begin{bmatrix} A_d\hat{\mathbf{x}}(k|k) + B_d\mathbf{u}(k|k) \\ r_0(k|k) + w_{r_0}(k|k) \\ f_{thr}(k|k) + w_{f_{thr}}(k|k) \end{bmatrix} \\ &= \mathbf{f}^a\left(\hat{\mathbf{x}}(k|k), \mathbf{u}(k|k), r_0(k|k), f_{thr}(k|k), w_{r_0}(k|k), w_{f_{thr}}(k|k)\right), \end{aligned} \tag{5.35}$$

where we have augmented the state of the EKF to include the two uncertain parameters, r_0 and f_{thr}, and their dynamics. Additionally, the covariance now propagates as

$$Q^a(k+1|k) = A_d^a|_{\hat{\mathbf{x}}^a(k+1|k)}\, Q^a(k|k)\; A_d^a|_{\hat{\mathbf{x}}^a(k+1|k)}^T + V_{xx}^a, \tag{5.36}$$

where the linearized dynamics are of the form

$$\begin{aligned} A_d^a|_{\hat{\mathbf{x}}^a(k+1|k)} &= \left.\frac{\partial \mathbf{f}^a}{\partial \mathbf{x}^a}\right|_{\mathbf{x}^a=\hat{\mathbf{x}}^a(k+1|k)} \\ &= \begin{bmatrix} A_d & \frac{\partial A_d}{\partial r_0}\hat{\mathbf{x}}(k+1|k) + \frac{\partial B_d}{\partial r_0}\mathbf{u}(k+1|k) & \frac{\partial B_d}{\partial f_{thr}}\mathbf{u}(k+1|k) \\ 0_{1\times n} & 1 & 0 \\ 0_{1\times n} & 0 & 1 \end{bmatrix}, \end{aligned} \tag{5.37}$$

and V_{xx} is appended to add the small Gaussian noise for the random walk in two extra diagonal entries. Fortunately, we have already computed the partial derivative matrices used in this expression in Section 5.2.3. The remaining measurement update equations do not change in the AEKF as we do not have any measurements of these parameters. Thus, the rest of the AEKF is exactly the same as the EKF implemented in Section 5.2.2.

Similar to some of the other controllers, the AEKF does not have any guarantees on performance, robustness or stability, however should in general perform better than most of the others when the true value of the uncertain parameter varies widely from the nominal expected value. Nominal performance will suffer because the random walk will inform the controller of a slightly incorrect plant. This overhead should subside as parameter uncertainty grows, until the AEKF actually performs better after a cross-over point.

5.3 Controller Metric Evaluation

Now that the set of controllers to be evaluated has been defined, it is time to discuss exactly how these controllers will be assessed and compared across different levels of uncertainty. This section covers the evaluation process from the creation of performance hypersurfaces through Monte Carlo simulation in Section 5.3.1 to the calculation of the expected performance metrics based on specific levels of uncertainty in Section 5.3.2 to conclude with the method proposed to select and schedule the best controllers for each mission phase in Section 5.3.3. In general, this section follows the process outlined in Figure 5-2, although further expanded and with more rigor.

5.3.1 Performance Hypersurfaces

The goal of creating performance hypersurfaces is to determine how well a specific controller will behave when subject to errors in the model of the system. The hypersurfaces are metric-specific functions of the error in uncertain parameters, and as such are n-dimensional functions of the n uncertain parameters. Hence, the term hypersurface arises as a generic concept of a surface that exists in $n + 1$ dimensions, the last of which being the evaluation metric. Thus, with one uncertain parameter, this hypersurface is a curve that can be plotted in 2D. With two uncertain parameters, the hypersurface is a surface that can be plotted in 3D. Higher-order hypersurfaces exist, following the same trend, but are much more difficult to plot.

Individual hypersurfaces need to be created for each important performance metric and for each controller in the library. Depending on the number of metrics and controllers, this can result in a lot of data and can take a significant amount of time to compute. Therefore, the creation of these hypersurfaces is limited to planning prior to the mission, and the onboard controller scheduler will only use the previously computed data in its selection process.

The process for computing a performance hypersurface starts with defining an appropriate uncertainty range. This uncertainty range should cover nearly all of the largest uncertainty distribution expected. For a uniform distribution, developing an appropriate uncertainty range is easy, as it should span the full width of the distribution. For Gaussians, one could choose a range that covers 99.9% or higher of the distribution. Technically, the Gaussian distribution extends to infinity, so it would be impossible to cover the entire uncertainty range. The goal is to limit the amount of the distribution left out, such that when integrating over it, the result is very close to unity. As seen later, this will allow for precise expected value computation. Remember, in multivariate analysis, this uncertainty range will be specified across n-dimensions and is permitted to be nonlinear (i.e., not necessarily a hypercube). For a 2D Gaussian with different variance in each dimension, an ellipse could be defined as the range for example, as opposed to a rectangle.

The next step is to create a mesh over this uncertainty range, such that each point in the mesh corresponds to a particular instantiation of error in each uncertain parameter. In the univariate case, the mesh is a discretization of points through the range of uncertain parameter values. In 2D, the mesh would be a grid of points over the uncertainty range area. Iteration with analysis of the resulting performance hypersurfaces can be used to determine if an appropriately fine mesh discretization level has been chosen. If there seem to be nonlinearities in the results that are not captured by the chosen mesh level, a finer mesh should be implemented. This process can be performed in an iterative manner such that results will not need to be recomputed, and only intermediate values of the grid are explored.

For each point in the uncertainty range mesh, specific performance metrics need to be evaluated for each controller. These metrics can include things such as LQR tracking cost, fuel consumption, phase completion time, meeting mission success criterion, and computation time required, among others. These five are the only metrics evaluated in this thesis, however a slew of other possible performance based evaluations could be implemented as desired. LQR tracking cost is computed as defined for the LQR controller in Section 5.2.2, with the same weighting matrices to truly measure the performance metric that each controller was attempting to achieve. Fuel cost is measured in terms of the amount of propellant the spacecraft has expended in truth rather than as commanded, such that the thruster noise and potential error in thruster force level are incorporated. Phase completion time is measured as the time from first entering a particular phase to the time of first exiting it. Phase 2 and 3 in particular are set to end when the spacecraft reaches a tolerance around the desired goal locations, while Phase 1a and 1b are tied specifically to a radial distance with no tolerance included. Mission success is determined by the successful maneuvering through all phases of the mission in the required time, without violating any constraints. Computation time is the average time per control period required to complete the estimation and control algorithms over all control periods in a phase. In this thesis, these times are calculated from the clock times in MATLAB on a late-2013 MacBook Pro with a 2 GHz Intel Core i7 processor and 16 GB of 1600 MHz DDR3 memory.

To evaluate these metrics, a Monte Carlo simulation of the full, multi-stage mission is run for each point in the uncertainty range mesh, and the metrics are tracked for each phase of the mission. The Monte Carlo simulation is meant to account for the full interaction of aleatoric noises present in the problem. The idea is that, when complex random processes are run for a large number of trials, the average results will tend to the true average results as the number of trials increases. The number of trials required to run can be evaluated such that statistically significance is achieved. The results presented in this thesis have undergone error analysis, and in specific cases, this analysis is directly called out. Thus, the Monte Carlo simulation is run to account for the random processes of thruster noise and measurement noise. However, a different process is used for the uncertain parameters. At each instantiation of a particular set of uncertain parameters (e.g., a thruster magnitude -15% off nominal and an orbital radius +10% off nominal) over the full uncertainty range mesh defined earlier, this Monte Carlo process is run. The performance metrics returned as an average of this Monte Carlo process make up the associated evaluation of the particular controller for that specific combination of uncertain parameter values. The uncertain parameters are treated differently because they are not random processes. They are similar

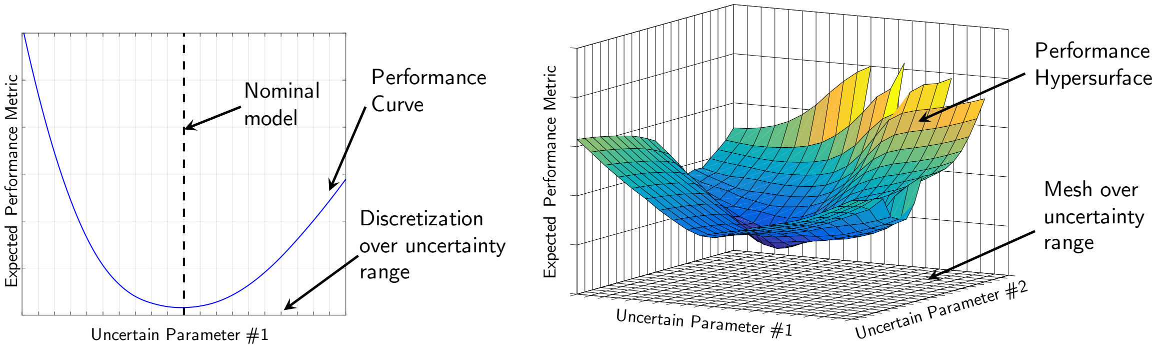


Figure 5-5: Notional examples of a univariate (left) and multivariate (right) performance hypersurface and associated uncertainty range meshes

to biases in the knowledge of system parameters as they remain constant throughout a mission. Therefore, the simulation only needs to be run once for each of the combinations in the uncertainty range.

After running through the full evaluation process for all controllers and all mesh points, the performance hypersurfaces are known as a function of the values of the uncertain parameters. Figure 5-5 shows an example of a performance hypersurface and associated uncertainty sampling mesh for both a univariate and multivariate case. It is these hypersurfaces that would be uploaded to the spacecraft for online controller scheduling. The process in the following sections is common to both ground and onboard implementation and discusses the calculation of probabilistic expected performance and eventual weighting of metrics and selection of the best controller.

5.3.2 Probabilistic Metric Calculation

The computation of the expected values of performance metrics over a specific level of uncertainty relies heavily upon a performance hypersurface being created. Therefore, before anything can be done here, the previous section must be followed to generate these performance hypersurfaces. Onboard implementation consists of the same process from here on out, assuming that the performance hypersurfaces were created on the ground prior to the mission.

Additionally, the probabilistic analysis herein relies on the uncertainty range and mesh to be sized large enough as discussed in the previous section. If they are sized too small, there could be important missing portions of the probability distribution left out of the proceeding calculation.

With both the performance hypersurface and uncertainty distribution of interest, the probabilistic performance, or expectation of the performance, can be computed using the law of the unconscious statistician. For the univariate case, this means that

$$\mathrm{E}\left[m_{kj}\right] = \int\limits_{-\infty}^{+\infty} f_k(p_k) m_{kj}(p_k)\ dp_k, \tag{5.38}$$

where m_{kj} is the value of the j-th performance metric subject to the k-th uncertain parameter, p_k represents a specific instantiation of the k-th uncertain parameter, and $f_k(p_k)$ is the value of the PDF associated with the k-th uncertain parameter evaluated at the specific instantiation of that uncertain parameter. The result is the integral of the multiplication of the PDF with the performance curve as shown pictorially in Figure 5-6.

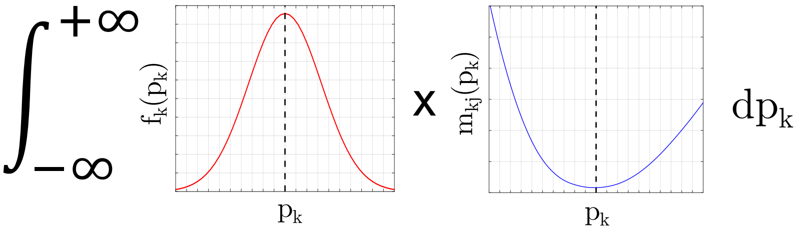


Figure 5-6: Graphical representation of the computation of the expected performance for a univariate uncertain parameter analysis

Note that the univariate form can be used if desired to reduce the increased computation required for the multivariate case. Even when there are more uncertain parameters present in the problem, separate expected performance metrics can be computed, each subject to a univariate uncertainty analysis. The metrics would then rate how well certain controllers handle individual uncertain parameters. The assumption one must make here is that the controllers that perform well under individual uncertainties will also perform well under a combination of uncertainties. If this is deemed acceptable for the mission planning process, and analysis is performed to show stability for at least some chosen cases with combined uncertainties, then computational time complexity reduces to a linear dependence on number of uncertainties instead of exponential.

However, since the Monte Carlo process in determining performance hypersurfaces can be run in parallel, this work can run on multiple cores of a personal computer or on a large-scale distributed computing server. This parallelization can reduce the running time to reasonable durations. And since we have not planned to run the hypersurface creation onboard the spacecraft, this should not be too much trouble. Therefore, if access to a large-scale distributed computing server exists, the multivariate analysis would be far preferred to the univariate option presented above. Nonetheless, the thesis presents both as options going forward.

For the multivariate case, we also have from the law of the unconscious statistician that

$$\mathrm{E}\left[m_j\right] = \int\limits_{-\infty}^{+\infty} \cdots \int\limits_{-\infty}^{+\infty} f(p_1, \ldots, p_{n_p}) m_j(p_1, \ldots, p_{n_p}) \; dp_1 \cdots dp_{n_p}, \tag{5.39}$$

where m_j is the j-th performance metric, $f(p_1, \ldots, p_{n_p})$ is the value of the joint, multivariate PDF at the specific instantiation of each uncertain parameter, and n_p is the number of uncertain parameters involved in the multivariate PDF. In 2D, the result is the volume integral of the multiplication of the PDF with the performance surface as shown pictorially in Figure 5-7.

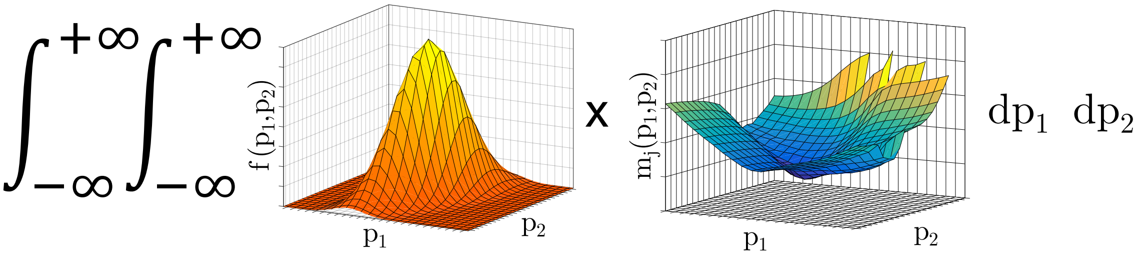


Figure 5-7: Graphical representation of the computation of the expected performance for a 2D multivariate uncertain parameter analysis

This integration (for either the univariate or multivariate case) can be completed numerically through trapezoidal integration provided the mesh is fine enough. Rather than the linear interpolation used in the trapezoidal integration, other interpolation schemes could be used as desired. For this thesis, an interpolation scheme known as a shape-preserving piecewise cubic interpolation is used for univariate analysis, and a spline interpolation scheme is used for multivariate analysis. The choice of interpolation scheme may depend on the number of uncertain parameters required to be integrated over, precision required and computation time allowed.

With these computed expected values, we now have probabilistically predicted the performance of the controller over the expected range of uncertainty in the mission. All that is left is to examine this data and decide which controllers have the best characteristics based on a set of preferences that describe the relative value of the different metrics. The next section details this process for both the univariate and multivariate analysis.

5.3.3 Controller Selection Method

The last step of the reference-tracking controller selection process is the most subjective. It involves picking the best controllers based on the expected performance computed for all of the metrics. Because this process involves a preference as to which metric is more important than other metrics, it is inherently subjective. Nonetheless, given a set of preferences, the same process can be followed to determine the best performing controller. This process can be performed individually for each mission phase if expected performance metrics have been computed for each phase, or it can be used to select the best controller over the full mission.

We will first discuss the process assuming that a set of relative preferences or importance weightings between performance metrics is known. Later, we will discuss how these weightings can be created. For the time being however, it suffices to know that these weightings are scalar multipliers on each performance metric, such that when all metrics are summed together, they will stress the importance of particular metrics over others.

Given this framework, the discrete optimization problem of choosing the best controller from the controller library is represented in the univariate uncertain parameter form as

$$\begin{gathered}\underset{c}{\operatorname{argmin}}\left\{\left(\sum_{k=1}^{n_p}\sum_{j=1}^{n_m}\pi_k\lambda_j\int\limits_{-\infty}^{+\infty} f_k(p_k)m_{kj}(p_k)\ dp_k\right)_c\right\}\\ \text{subject to} \\ \mathrm{P}_k[failure] = 1-\int\limits_{-\infty}^{+\infty} f_k(p_k)\mathrm{P}_k[success|p_k]\ dp_k \leq 0.01\quad \forall\ k=1\ldots n_p,\end{gathered} \tag{5.40}$$

where c is the specific controller being evaluated, π_k is the relative importance weighting on the k-th uncertain parameter, λ_j is the relative importance weighting on the j-th performance metric, n_p is the number of uncertain parameters, n_m is the number of metrics, and the integral in the argmin function is the j-th expected performance metric subject to the k-th uncertain parameter as computed in Section 5.3.2. Additionally, the constraint implies that the probability of mission failure is equal to one minus the expected value of the probability of mission success as computed in the integral, and that it should be less than 1% as an example. This percentage is tunable of course. Mission success involves completing the full mission in the requisite time period and meeting all constraints placed during the mission and on resulting metrics.

For the multivariate case, the optimization problem becomes

$$\underset{c}{\operatorname{argmin}} \left\{ \left(\sum_{j=1}^{n_j} \lambda_j \int\limits_{-\infty}^{+\infty} \cdots \int\limits_{-\infty}^{+\infty} f(p_1, \ldots, p_{n_p}) m_j(p_1, \ldots, p_{n_p}) \; dp_1 \cdots dp_{n_p} \right)_c \right\}$$
$$\text{subject to}$$
$$\mathrm{P}[failure] = 1 - \int\limits_{-\infty}^{+\infty} \cdots \int\limits_{-\infty}^{+\infty} f(p_1, \ldots, p_{n_p}) \mathrm{P}[success|p_1, \ldots, p_{n_p}] \; dp_1 \cdots dp_{n_p} \leq 0.01, \tag{5.41}$$

where all of the same definitions apply, except that $m_j(\cdot)$ and $f(\cdot)$ no longer depend on a specific uncertain parameter, p_k, but all n_p uncertain parameters being investigated. Additionally, the probability of failure constraint is now computed from integration of a joint probability function of all uncertain parameters.

Essentially, what these optimization problems express is that we want to pick the controller that has the lowest possible cost over the weighted sum of all expected metrics, while adhering to constraints on the probability of mission success. Really the optimization problem is just choosing one controller out of a reasonably small set from the controller library, so advanced techniques do not need to be performed to solve it. Simply running the weighted-sum cost function on all of the controllers and picking the one with the lowest cost will suffice. Given this formulation, the question then turns to how we set these relative importance weightings. As usual, the answer is that it depends. That is, it depends on the types of preferences the user has.

Sometimes, a user cares about a particular metric, but only that it is below a certain point. As an example, this could mean that the user would prefer that the maneuver completion time be under a specific time and would want to ignore all solutions that result in longer times. Or the user wants to limit the fuel in a particular maneuver to be under a specific value, but does not care how far under this value the controller is able to achieve. In these cases, these metrics need only be added to the mission failure criterion constraint, such that if the metric exceeds these set values, the controller is deemed infeasible and excluded from the selection. The relative weightings on these metrics would then be set to zero such that only this constraint influences the choice, not performance improvements beyond meeting the constraint.

Other times, a user will care both about a constraint and performance beyond that constraint value. As far as weightings go, this is equivalent to the case where there are no constraints and performance is only desired to be minimized, except that the constraint should still be implemented as discussed in the last paragraph.

After determining the set of metrics that fall in these two categories, we now wish to create non-zero relative weightings between these metrics. Because all of the metrics will likely have different units, it is easiest to first normalize the metrics such that each starts out on a level playing field. This normalization is worked into the weighting for each performance metric, λ_j. It is suggested to normalize the results based on the value of the performance metric when evaluated on the nominal system model, such that nominal performance will be one for all metrics. In this manner, the rate of increase in a metric above one will show how sensitive the metric is in that regime of uncertainty.

After λ_j has normalized all the metrics onto the same playing field, additional multipliers can be placed based on the user's preferences. The process is trivial if the user only cares about one metric over all improvement in others, as it is extremely unlikely there will be any ties in this process. In this case that metric is weighted to one and all others to zero. Otherwise, the user needs to come up with weightings that express the degree to which he cares about each metric and multiply this factor by the normalization factor to form λ_j. Preferences are very difficult to quantify, however, and there is danger they will just be tweaked to produce the results the user was looking for in the first place.

Ideally, a Pareto front of controllers could be explored such that non-dominated controllers are compared against the different metrics. With the Pareto front, an engineer is able to see if there are any large jumps in improvement of one metric with a small worsening in another. With the small number of controllers in the controller library of this thesis, however, this would not be very fruitful. Although not implemented in this thesis and left for future work, the concept exists to vary tunable parameters in each of these controllers and fill the Pareto front more completely such that an analysis like this would be possible. The time required to compute performance hypersurfaces for all of these controllers was much too great for the author to run given hardware and financial constraints. In any case, pre-mission analysis could still result in a satisfactory set of weightings for the onboard controller scheduler to implement during the mission.

The previous discussion has only discussed λ_j, the weighting on metrics, while there is also a weighting on uncertain parameters, π_k, in the univariate case. This is easier, however, as we do not wish to normalize the results and therefore can jump straight to ranking the uncertain parameters in order of their importance to the mission and assigning subjective relative weights. Usually though, the covariance or level of uncertainty in the particular uncertain parameter should provide enough weighting information itself that will show objectively how that parameter affects the mission. Therefore, in most cases, each of the weightings can be set to one. In fact, because of this, the author would not suggest altering the weightings from one. Nonetheless, the weightings remain in the formulation to maintain a general form.

As a final note, with respect to treating metrics as constraints or attempting to minimize them, there is an important distinction that needs to be made. Depending on the point at which this analysis is being performed in the full mission lifecycle, the treatment of these metrics may change. Typically, most of the control and software work for space missions is completed well after the design of the spacecraft is set in stone. Thus, controllers must conform to the limitations of the hardware. Instead, it is suggested to perform this reference-tracking controller analysis when performing trade studies for the spacecraft design phase. In this situation, a variety of controllers can be explored that each try to minimize certain performance metrics and show the gains that can be achieved if parameters are tweaked and the guidance and control software can instead dictate some design choices. If this methodology is implemented, constraints on fuel consumption and computation time could turn into minimization problems instead. Once the hardware is set, however, it will make more sense to use the constraint formulation and not include them in the overall cost evaluation function.

With this commentary, the reference-tracking controller selection process has been fully explained. The following section will cover its implementation and hopefully illuminate answers to any questions that are still on the mind of the reader with respect to how certain parts of the process are carried out.

5.4 Results

Given the previous methodology sections making up this chapter, the full reference-tracking controller scheduling process is carried out on the multi-stage rendezvous, docking, and joint maneuvering mission defined in detail in Chapter 3. Key results throughout the process will be presented in this section, which explore how different controllers handle uncertainties when following the previously computed optimal trajectories from Chapter 4 and how to select the best performing controllers over expected uncertainties for the mission. First though, nominal controller performance will be investigated.

As a quick reminder, this mission is that of a chaser (tug) spacecraft rendezvousing and docking to a target spacecraft (a space station module), and proceeding to relocate it to an assembly location. This full mission takes place in GEO and is permitted to last up to 8 hours. Nominally, the trajectory is designed for a 4 hour combined rendezvous and docking phase from a position 3 km behind the target and 2-hour relocation phase to a position 5 km away. The extra 2-hour margin is a maximum time constraint placed on the system when under severe uncertainties. Finally, the mass of the target is four times that of the chaser, such that when they dock, the chaser is in command of a mass five times as large. All nonlinear constraints described in Section 3.2 are implemented and enforced throughout the mission. The trajectory is first optimized using the methods in Chapter 4, excluding an obstacle avoidance scenario, yet leaving a 1-σ risk margin to the LOS constraint.

The controllers from the controller library described in Section 5.2 are coded into the MATLAB simulation framework described in Section 3.4. Each controller is designed and tested with the nominal system parameters defined in Section 3.2.8 and to the same nominal LQR weighting matrices. The goal of each controller is to track the optimized full-mission trajectory throughout all mission phases, meeting all constraints along the way, and reaching the final location before the mission time limit is exceeded.

As mentioned, first we will examine the nominal reference-tracking controller performance without any uncertainties in the system. No measurement noise, no thruster noise, no nonlinear orbital dynamics. However, we still run the estimator and take the (noiseless) measurements. Each of the controllers achieves very similar performance under this easiest of tests. Figure 5-8 shows the tracking performance of the baseline controller (the LQG controller) plotted in the orbital plane, which is nearly identical to the other controllers. In

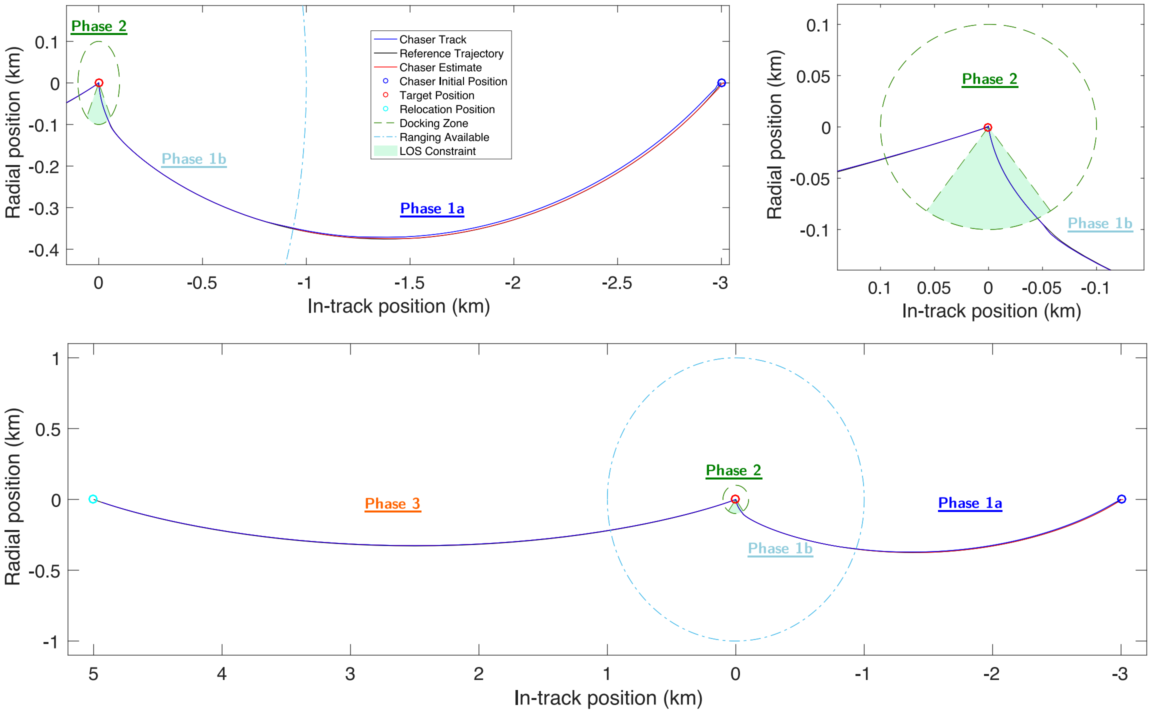


Figure 5-8: Full mission tracking performance of the baseline LQR reference-tracking controller and EKF estimator in the full nonlinear model simulation with no noise

Figure 5-8 and several others like it, the true simulation position of the spacecraft achieved throughout the mission is shown in blue, the estimated position from the EKF in red, and the reference trajectory in black. Note that in this case, the curves overlap each other substantially, and the only major noticeable difference is in Phase 1a, where there is a slight drift in the true position of the satellite from the reference and estimate due to the range unobservability.

When measurement noise, thruster noise, orbital nonlinearities and perturbations are added into the simulation, the spacecraft are still able to track the reference trajectory very well. Again, the baseline LQG tracking performance is shown in Figure 5-9 and fairly closely matches a high-level comparison of a single trial's performance of the other controllers. With the aleatoric noises added to the problem, the estimator does a bit worse in estimating the true position as seen by the small deviances in Phases 1b and 3. The drift in Phase 1a has grown now that there is some noise in the measurements and thruster execution. Because of the range unobservability in this phase, small errors in position are not likely to be picked up, and the estimator will think it is tracking the reference well (as it does here), because the model of the system predicts that it should be. Once a range measurement is available

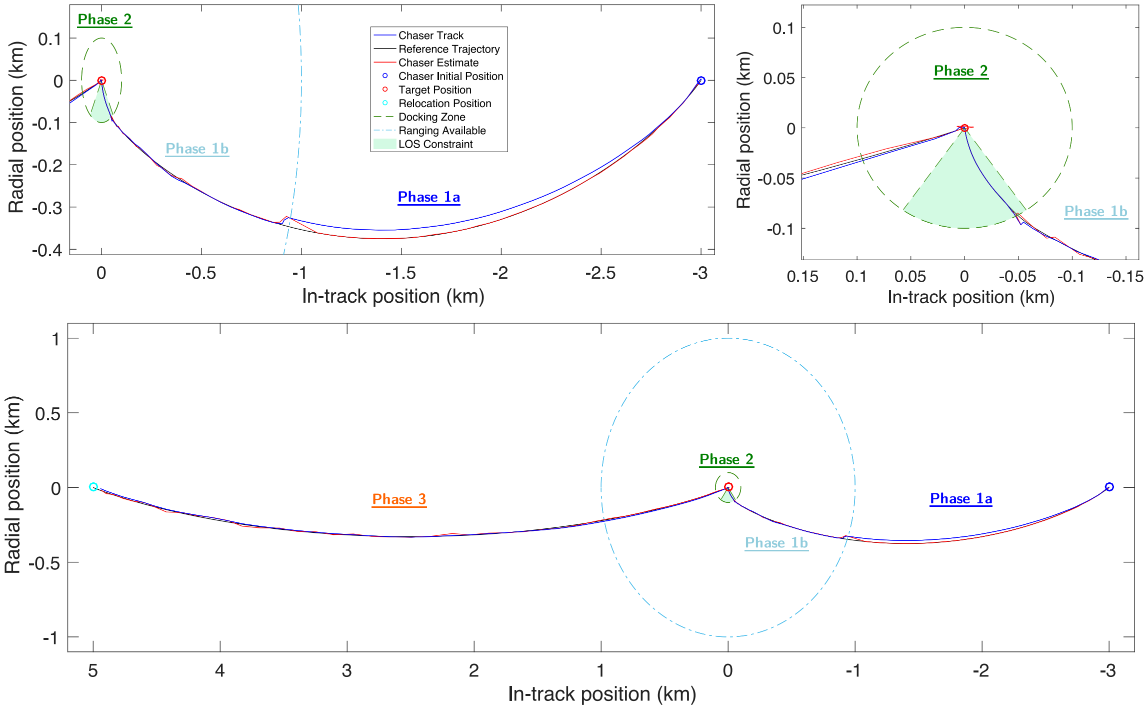


Figure 5-9: Full mission tracking performance of the baseline LQR reference-tracking controller and EKF estimator in the full nonlinear model simulation with the addition of full measurement and thruster noise

when the true position of the spacecraft enters Phase 1b, the estimate is quickly able to recognize this error and snap to the correct position as can be seen in the upper left plot of Figure 5-9. Additionally, because of the highly precise measurements available in the docking phase, the estimate, true position, and reference trajectory overlap.

Figure 5-10 shows the growth and tightening of the covariance in the estimated position and velocity states from the standard EKF. Notice the drift in the position error during Phase 1a until about 115 minutes into the mission, where the range measurement is available. Then there is a period of higher measurement noise in Phase 1b until about 180 minutes, where the highly precise docking sensors are in use. There is a steady decline in the covariance in the position states during this time. After docking at 240 minutes, the sensors have larger error again, and the commanded thrusts are much larger and thus have larger noise characteristics in the x and z directions, which are mostly orthogonal to the direction of thrust. This causes the covariance in the estimate to grow, but the true error is not actually that bad. These noise characteristics match what are seen in the position plots in Figure 5-9.

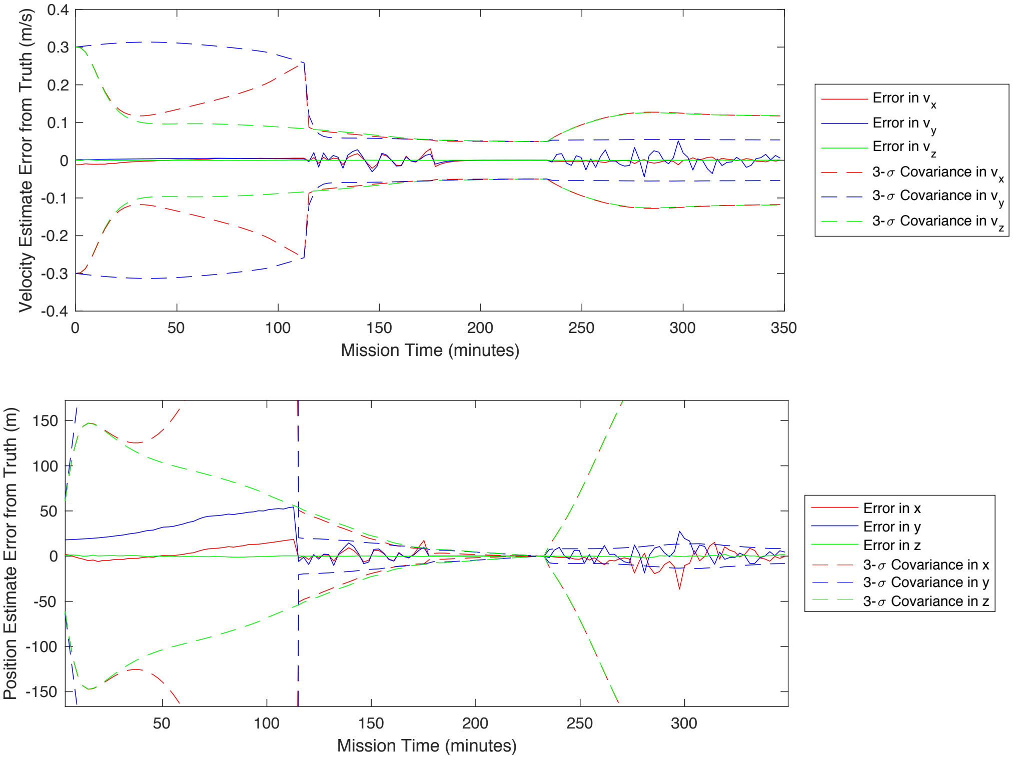


Figure 5-10: Estimation error and covariance in velocity and position states as result from the EKF paired with the LQR controller

Under full noise, the other controllers in the library perform similarly to the LQG controller when looking at tracking performance plots like Figure 5-9. However, there are some slight differences that will affect performance in the long run. For example, occasionally the MPC method will encounter some difficulties when entering Phase 1b with a large deviance from the reference trajectory. Instead of correcting for the error immediately like the other controllers, the MPC will predict that this will take a lot of propellant and will instead skirt the allowable tube constraint boundary before slowly edging its way over

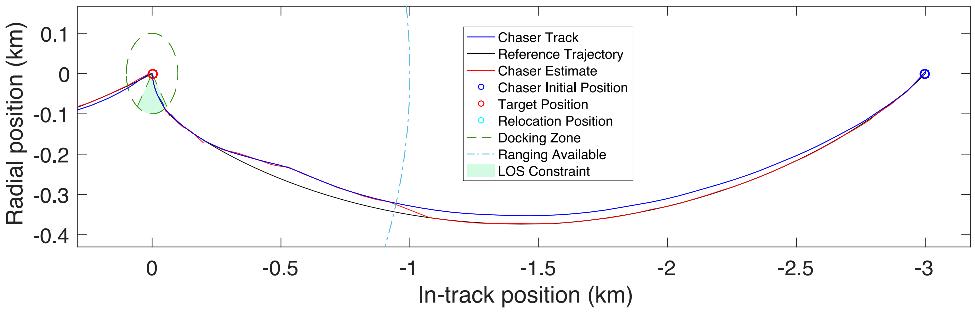


Figure 5-11: Infrequent transition issue with the MPC controller resulting in lower tracking performance in Phase 1b although improved fuel consumption

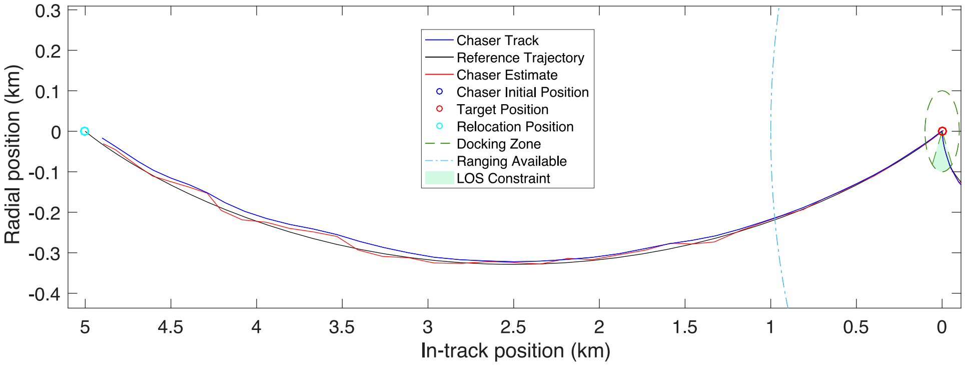


Figure 5-12: Phase 3 drifting of the AEKF on a nominal system model showing the decrease in tracking performance from the baseline LQG case

to the reference. Figure 5-11 shows this behavior. However, this happens fairly infrequently, and the tracking performance is much better once it nears the reference trajectory.

The AEKF controller experiences some inefficiencies due to the constant switching of LQR gains based on the current best estimate of the thruster force level and target orbital radius. These estimates will be slightly noisy on the nominal system, and since they are used directly to compute the gains every control period, there will be some less-than-optimal gains chosen at times due to the error in the estimate. This effect can be seen best in Phase 3, when the gains are larger due to the extra mass. Figure 5-12 shows the additional noise and tracking error present for the AEKF on a fairly typical trial run. Figure 5-13 shows the source of this estimate in the steady state estimate of the thruster force level in Phase 3. However, Figure 5-13 shows the estimate of the thruster force, when there was a significant error (-50%) in the controller's nominal model. One can see that the deviation in the estimate is much smaller than the error that would exist on a standard LQG controller if the thrust magnitude were significantly off nominal. This is where the AEKF technique gains performance as the error in uncertain parameters, like the thruster force magnitude,

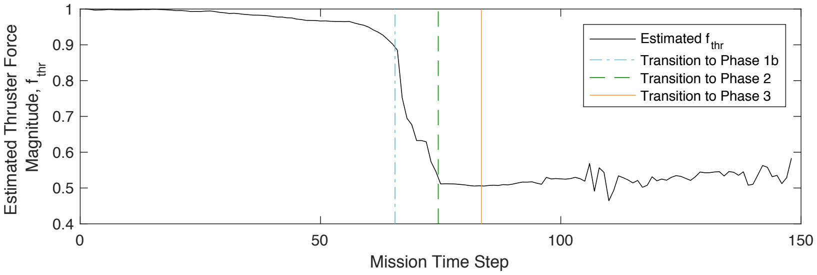


Figure 5-13: The progression of the thruster force level estimation performed by the AEKF for a single, typical simulation run over a full mission

grow. The estimation improves as more thrust is commanded in the mission and as the measurements increase precision, such that by the end of Phase 2, the AEKF has almost nailed the true force magnitude. When more noise is present in Phase 3, the estimate starts to suffer a little bit. This same behavior still happens in Phase 3 when implemented on the nominal thrust magnitude case.

One of the main benefits of the SWLQG controller is due to the fact that it attempts to desensitize the optimal gains to errors in both thruster force magnitude and orbital radius. The main effect of this desensitization is that the gains are decreased. And decreased gains lead to better fuel performance. Additionally, due to the desensitization, the SWLQG controller is also less sensitive to noise in the thrusters, not only errors in the force bias. These decreased gains cause the SWLQG to track the trajectory somewhat slower than the others, however this only amounts to a lag and the desired shape of the reference is still attained. Results thus look very similar to Figure 5-9 for the baseline LQG controller.

Before completely transitioning to discussing performance of the controllers when there are large errors in uncertain parameters during a mission, a comparison of the controllers' average performance to the nominal system will first be analyzed. Fuel consumption differs somewhat significantly between the controllers as the optimization is geared toward LQR cost instead. Figure 5-14 shows a typical mission timeline of fuel consumption for each of the controllers. Immediately, the MPC and SWLQG controllers are seen to be much more fuel efficient, while the baseline LQG and AEKF have similar fuel use. The AEKF has a slightly worse variance in results from run to run due to the inefficiencies discussed. The SWLQG has the best fuel performance due to the lower gains. Another interesting note

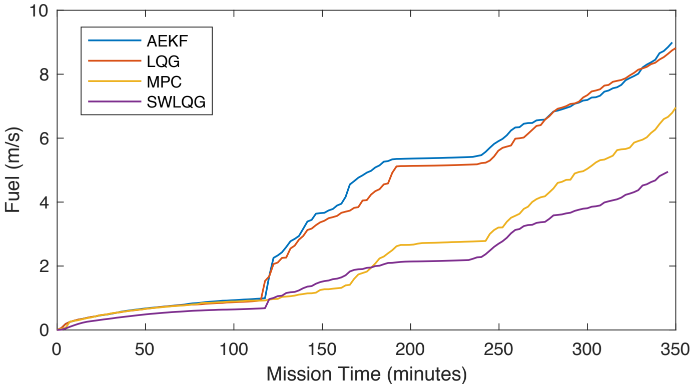


Figure 5-14: Comparison of fuel consumption histories over a single, typical run of a full mission for each controller in the controller library

is that MPC does not have a large jump in fuel consumption around the 115-minute mark like the other controllers do. The absence of this jump is because it does not transition as abruptly when entering Phase 1b as discussed earlier and shown in Figure 5-11.

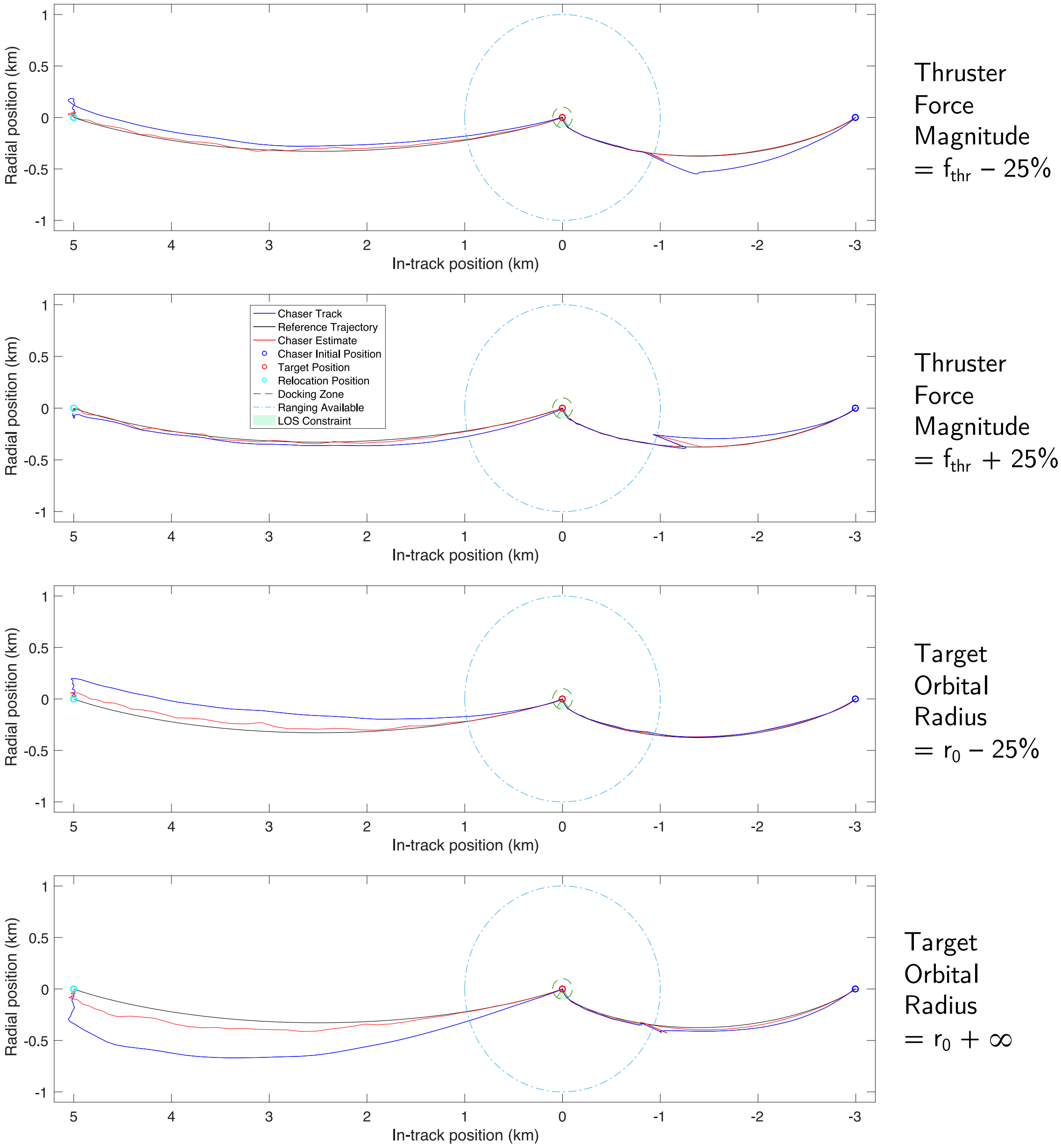
Table 5.1: Nominal performance metric evaluation and comparison for each controller in the controller library on a full mission with aleatoric but no epistemic uncertainty

Controller	LQR Cost	Fuel (m/s)	Completion Time (min)	Computation Time (s)
AEKF	483	9.1	356	6.2×10^{-3}
LQG	324	9.0	354	2.0×10^{-4}
MPC	280	7.6	355	5.8×10^{-2}
SWLQG	379	5.6	353	2.5×10^{-4}

Table 5.1 shows a wider, full-mission comparison of averages of each of the metrics of interest over a 10,000 trial Monte Carlo simulation. The metrics consist of LQR cost, fuel consumption, mission completion time and computation time required per control period. There are several differences between these controllers that are illuminated by these nominal results. First though, let it be known that the LQR cost shown is weighted differently than that used to design the controllers. It only takes into account the state error between the reference trajectory and the true spacecraft position throughout the mission. The change was made to decouple this metric from the fuel consumption metric so as to not count fuel twice and to have a specific metric for how well the tracking is performed. With that said, when comparing the LQR cost, the MPC controller performs the best followed in order by LQG, SWLQG and AEKF. This ordering was more or less expected from a high-level ranking of performance tradeoff with robustness. To gain extra robustness to errors in the nominal model, the controller needs to give up some performance when executed on the nominal model. In terms of fuel consumption, the average results match the trends that were seen in Figure 5-14 for the individual trials. The AEKF and LQG have statistically insignificant differences in fuel usage, while the MPC and SWLQG each perform better. Differences in mission completion times are statistically insignificant. Computation times are orders of magnitude different, with MPC being overwhelmingly slower as expected, and the LQG and SWLQG having comparatively fast computation times. Computation times are calculated based on the clock times in MATLAB on a late-2013 MacBook Pro with a 2

GHz Intel Core i7 processor and 16 GB of 1600 MHz DDR3 memory.

Given the nominal performance of each of the controllers, the discussion now shifts to off-nominal performance. The aim here is not only to see which controller performs the best under ideal circumstances, but also to see how the controllers react to errors in uncertain parameters in the dynamics. Therefore, we are looking for a good balance of performance and robustness depending on the uncertainty level. Specifically, modeling errors in the



Thruster Force Magnitude $= f_{thr} - 25\%$

Thruster Force Magnitude $= f_{thr} + 25\%$

Target Orbital Radius $= r_0 - 25\%$

Target Orbital Radius $= r_0 + \infty$

Figure 5-15: Comparison of the baseline LQG controller tracking performance to specific biases in uncertain parameters over the full mission

target's orbital radius and the thruster force magnitude are considered in this problem as discussed in Section 5.1.3. These errors can be considered constant unknown biases throughout the mission and are not stochastic processes like the measurement and actuator noise.

Figure 5-15 shows the LQG controller tracking performance when there are errors in the knowledge of the target's initial orbital radius that defines the CWH frame and the force magnitude of the thrusters. Each plot introduces a specific error as noted by the text on the right of the plot. The LQG controller is designed to nominal values without knowing this error is present.

In the top plot of Figure 5-15, when the thruster force is overestimated, there is a strong drift in Phase 1a, where the unobservability prevents the EKF from knowing of this bias. Eventually, the angle to the target reaches a point where the measurements do not match the model, and the EKF begins to trust the measurements more and model less. In Phase 1b and 2, the performance degradation is more subtle. However, in Phase 3, due to the large thrusts required, drift is seen again to the point where the spacecraft crosses the relocation point and begins to loop backward. As the error in this direction grows, the loop here grows as well until the spacecraft does not have enough time to reach the goal location.

When thruster force is underestimated, as in the second plot from the top, there is drifting in the opposite directions. In this case, during Phase 1a, the chaser gets much ahead of the reference trajectory such that it enters Phase 1b early, gains a better estimate of its location, and immediately performs a large correction to get back to the reference trajectory (which was still lagging in Phase 1a). This action requires a lot of fuel and therefore causes a major decrease in performance.

Similar but more pronounced drifting can be seen for errors in the target orbital radius during Phase 3, however the drifting in Phase 1a is reduced in comparison. Note that the first and third plots in Figure 5-15 exhibit drifting in the same direction. In these cases, the orbital dynamics are stronger than expected in comparison to the thruster force level. The last plot is for an essentially near-infinite orbital radius, which approaches a case with zero orbital mechanics present. Interestingly, the behavior is relatively constant after a five-fold increase in the orbital radius since the magnitude of the orbital dynamic forces is much smaller than the thruster forces.

These same general trends in performance degradation can be seen in the other controllers to varying degrees. This raises the question of which controller can handle the most error in these uncertain parameters before failing the mission by violating path constraints or exceeding the maximum mission duration. To determine this, the Monte Carlo process described in Section 5.1.2 is run for the full range of uncertainty expected for each uncertain parameter separately. This analysis will represent the univariate case, while the multivariate case will be explored later to see how a combination of these two uncertainties affects robustness.

Figure 5-16 plots the mission success rate as a function of error in the model of the thruster force magnitude for each of the controllers as a comparison of their robustness ranges. Each point on the plot is computed by determining the number of successful missions from a reduced Monte Carlo simulation of 1,000 trials for each specific value of thruster force error bias. Due to increased computation time only 500 trials are run for AEKF and 100 for MPC. Again, the controllers are designed to nominal values without knowing this constant error is present. The width of the 100%-mission-success plateau represents the range of thruster force magnitude errors that the controller can withstand before the mission fails. The nominal thruster force magnitude is represented as 1, a value of 2 represents a 100% increase in the magnitude, and so forth. From this analysis, the AEKF controller seems to be substantially more robust than the other controllers, covering a range from 0.26 (-74%) through 4.5 (+350%). LQG and SWLQG have similar ranges, however SWLQG is shifted to the right due to the reduction in controller gain from desensitization. The MPC has the worst robustness range, only covering from about 0.8 (-20%) to 1.15 (+15%).

It is worth it to note at this point, that most of the techniques are not failing because they go unstable, but because they violate the mission completion time constraints in the problem. Thus, although SWLQG may have worse range in the reduction of thrust direction, it may have larger stability margins, and only have a longer response time. Because

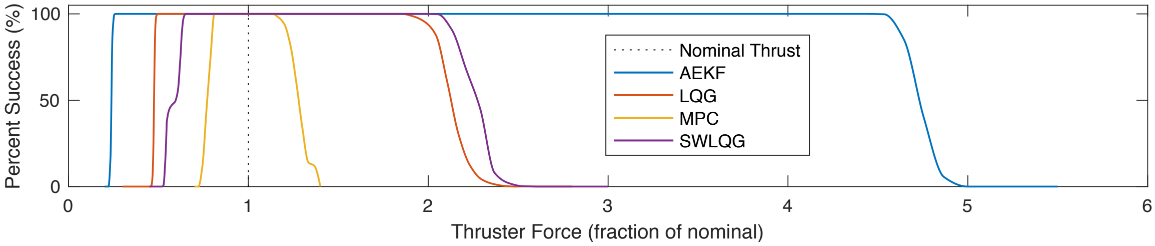


Figure 5-16: Percent mission success throughout the Monte Carlo trials plotted against uncertainty in the thruster force magnitude for each controller

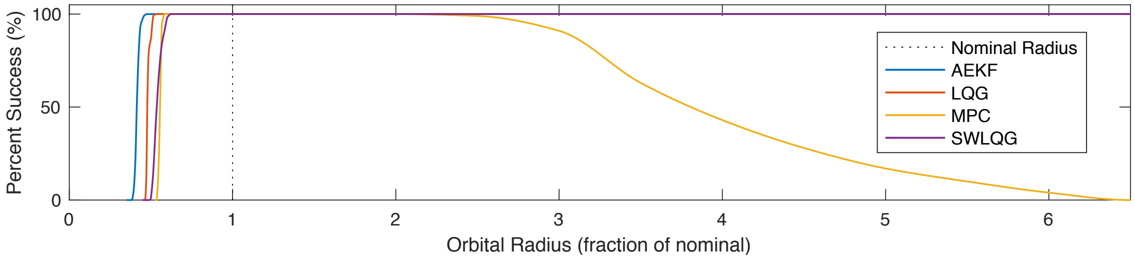


Figure 5-17: Percent mission success throughout the Monte Carlo trials plotted against uncertainty in the orbital radius for each controller

of the constraints in this problem, this is not an analysis of phase and gain margin robustness of each of controllers, but more of a practical evaluation of mission success criterion. Nonetheless, the takeaway here is that there is a tremendous gain in robustness from using an AEKF controller, and that MPC is the most fragile of the techniques explored.

A similar plot is created and shown in Figure 5-17 for the mission success rate as a function of error in the model of the target's initial orbital radius for each of the controllers. Again, we see the best robustness from the AEKF and the worst from the MPC controller. Interestingly, each of the LQG, SWLQG and AEKF are robust to any increase in the true target's orbital radius. Essentially, this just means that they are robust to an "F=ma" truth model without any orbital dynamics present. AEKF completes missions successfully until about a 53% reduction in orbital radius, LQG a 47% reduction, and both MPC and SWLQG about a 40% reduction. MPC also starts to fail at about double the radius. These robustness ranges are actually quite impressive and one would never expect an error in the estimate of the target's orbital radius by these margins. This is likely a consequence of the analysis being performed in GEO, where the orbital dynamics are much weaker than in LEO. If the mission were instead in LEO, one could expect these ranges to decrease significantly.

Now the discussion has moved into the realm of creating performance curves and surfaces for this problem. The univariate case will continue to be presented to show the controllers' performance metrics against errors in one uncertain parameter at a time. In a sense, the robustness plots in Figure 5-16 and Figure 5-17 are performance curves for mission success probability, except they will be used primarily for probability of success constraint adherence when considering which controllers to pick for specific uncertainty distributions. The performance curves for the LQR cost will first be explained, and then other metrics will be discussed briefly as they follow the same process of creation.

Figure 5-18 gives a detailed view of the LQR cost performance curves for unknown biases in the thruster force magnitude. These curves were created for each controller by running a Monte Carlo simulation for specific values of the uncertain parameter, in this case the thruster force magnitude. This Monte Carlo simulation varies aleatoric noises to evaluate the average LQR cost at a specific instantiation of an unknown bias in the thruster force magnitude. The performance curve itself is then the average LQR cost as a function of the value of the uncertain parameter. LQR performance curves are computed for each controller over the same uncertainty range using the same random seeds for the Monte Carlo simulation. The LQG and SWLQG controller each could be run for 10,000 Monte Carlo trials, while due to computation time constraints the AEKF could only run for 1,000 trials and the MPC for 500 trials. To account for this, 95% confidence error bars are added

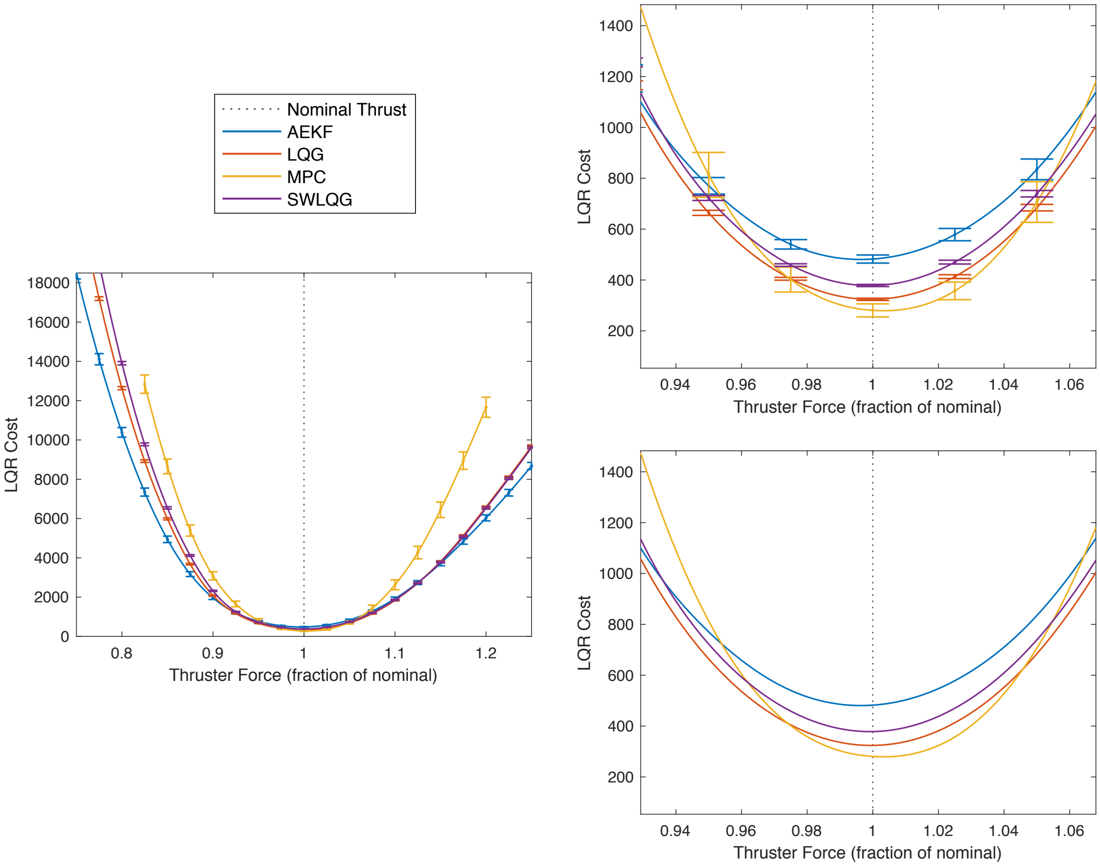


Figure 5-18: Performance curves for the LQR cost metric of each controller across uncertainty in the thruster force magnitude. Different zoom levels shown, with no error bars on one of the furthest zoomed.

to the plots to show statistical significance. This is especially important for the LQR cost performance curves as they fall very close to each other in critical ranges.

In the left of Figure 5-18, over the whole uncertainty range of interest the AEKF controller seems to have the lowest LQR cost outside the range of about 0.9 to 1.1 times the nominal thruster force magnitude, while MPC seems to have the highest LQR cost outside this range. LQG an SWLQG appear to have very similar performance, with LQG having lower cost for negative thruster force errors and SWLQG having ever so slightly lower LQR cost for positive thruster force errors. The two plots on the right show a zoom in on the range of about 7% from the nominal value (one with error bars to show statistical significance and one without for clarity). With small errors in thruster force from the nominal model, MPC has the lowest LQR cost (from about 0.975 to 1.05), and otherwise the baseline LQG controller has the lowest LQR cost. These plots show the overhead present in the more robust controllers as nominal performance decreases roughly in the order of how robust each controller is as shown in Figure 5-16.

Figure 5-19 shows the same LQR performance plots for the uncertainty in the target's

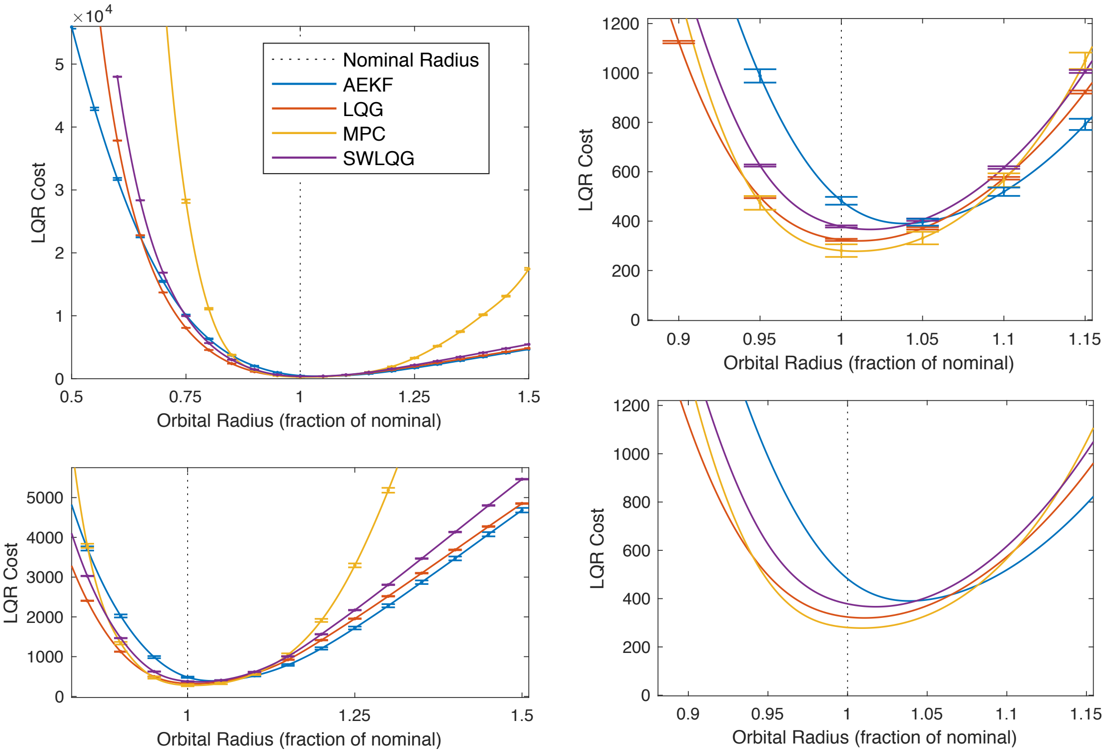


Figure 5-19: Performance curves for the LQR cost metric of each controller across uncertainty in the orbital radius. Different zoom levels shown, with no error bars on one of the furthest zoomed.

orbital radius. These are computed with nominal thruster force values, but varying an unknown bias in the orbital radius uncertain parameter. Qualitatively the results are very similar to the thruster force magnitude analysis. The AEKF has the lowest LQR cost when the error is large as shown in the plots on the left. The LQG has the lowest LQR cost in the middle range and MPC has the lowest cost when small errors exist. Again, 95% confidence error bars are shown to support statistical significance of the results. Interesting here is that with positive error, the MPC controller transitions directly to the AEKF controller on the rightmost plots rather than transitioning through the LQG controller as happens in the negative direction and in the thruster force case. The dependence of the MPC optimization on knowing the true model is shown by how poorly it performs with modest error in the model and how well it performs with little error. Note again that ranges of error in the orbital radius will likely be a small percentage of the nominal and this full range is not quite realistic. Nevertheless, it is worthwhile academically to see where and how these controllers tradeoff their performance in comparison to each other.

The same analysis can be completed to find the performance curves for the other metrics as well. In fact, these metrics should be tracked in the same simulation runs as the LQR cost, such that these simulations only need to be run once. The resulting performance curves for both the thruster force magnitude and the orbital radius univariate analyses are found in Figure 5-20. These curves do not have error bars shown as they are much smaller than any difference between the curves and only serve to clutter the results.

As depicted in the topmost two plots in Figure 5-20, the computation time per control period for the LQG, SWLQG and AEKF does not depend on the value of the uncertain parameter, yet for MPC there is an increase in computation time the larger the errors from the nominal model. This increase is due to the constraint tightening in the MPC algorithm. With larger errors in the uncertain parameters, the constraint tightening algorithm must iterate through more constraints before a feasible solution can be found. Eventually there is a cutoff on the number of iterations and missions that exceed this cutoff point are deemed failures as they cannot find a feasibly optimal plan. Data is not shown on these performance curve plots when the success probability computed for that parameter error is less than 90%.

Mission completion times for the different controllers vary with the same general trend for the most part, however there is one major difference with respect to thruster force magnitude uncertainty. In this case, the AEKF is able to estimate the correct thruster force

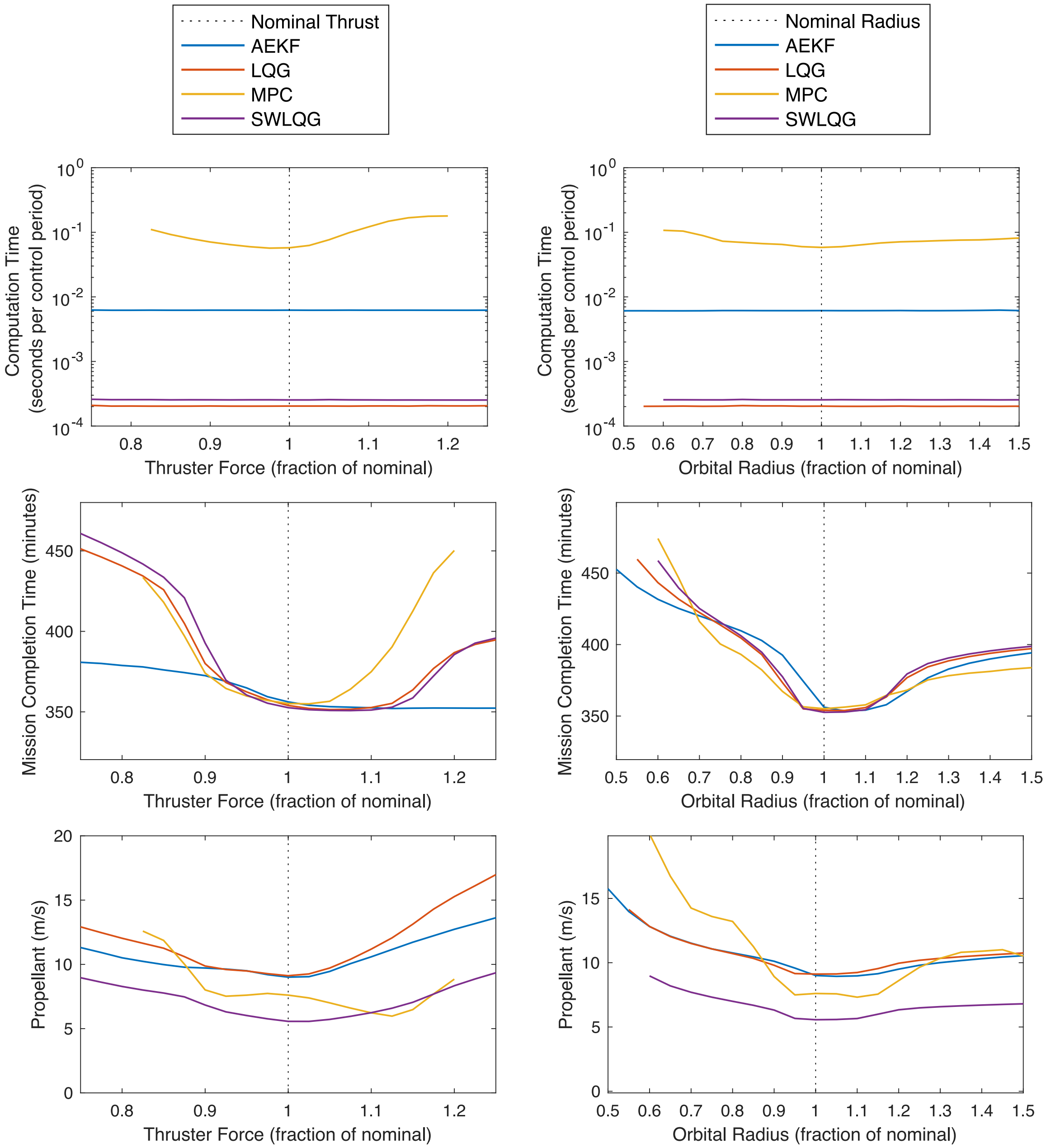


Figure 5-20: Performance curves for the computation time, fuel consumption and mission completion time metrics for each controller as compared for uncertainty in the orbital radius and the thruster force magnitude separately

fairly well such that the completion time is not affected significantly when large uncertainties are present. For the orbital radius case, the AEKF is not quite as good at estimating, so this behavior is not seen to the same effect. A final note with regard to completion times is that when the thruster force is larger than expected, the controllers tend to lead the reference by a slight margin such that mission times drop on average as the thruster force error increases until controllers start overcorrecting for it.

Finally, when looking at fuel costs in the bottom plots in Figure 5-20, it is quite apparent that the SWLQG controller performs the best in this respect over the full uncertainty ranges. MPC also performs well with small errors from the nominal model, but is even worse than the baseline LQG and AEKF fuel consumption under larger errors. The SWLQG has this better fuel performance as discussed before because of the reduction in gains from the desensitization process.

To finish the univariate analysis, the LQR cost is treated as the only performance metric of interest and the best performing controller over the full mission in terms of the LQR cost is therefore desired. The analysis could easily be done individually for each phase by tracking phase-by-phase metrics and going through the same process four times. This phase-by-phase analysis is left for the multivariate case however. Figure 5-21 shows the ranges of three uncertainty distribution types in which each controller type has the best expected LQR cost. The expected LQR cost is computed as the integral of the performance curve multiplied point by point by the probability density function of interest. This expected LQR cost scalar value is computed for each controller and the controller with the lowest value is chosen as the optimal reference-tracking controller for that specific uncertainty distribution.

In Figure 5-21, the colored bars above the example PDFs show the ranges in which each controller is deemed optimal. For the uniform distribution, the range is defined by the half-width of the distribution. For the Gaussian distribution, the range is defined by the standard deviation. For the Weibull distribution, it is defined by the location of the maximum of the PDF. Since, the Weibull distribution is asymmetric, it highlights how the ranges change in positive and negative directions better than the others. Additionally, under each of the colored bars is an example distribution from that class as colored by which controller would be optimal under that distribution. Table 5.2 and Table 5.3 show the ranges in numerical form for the thruster force magnitude uncertainty and orbital radius uncertainty respectively.

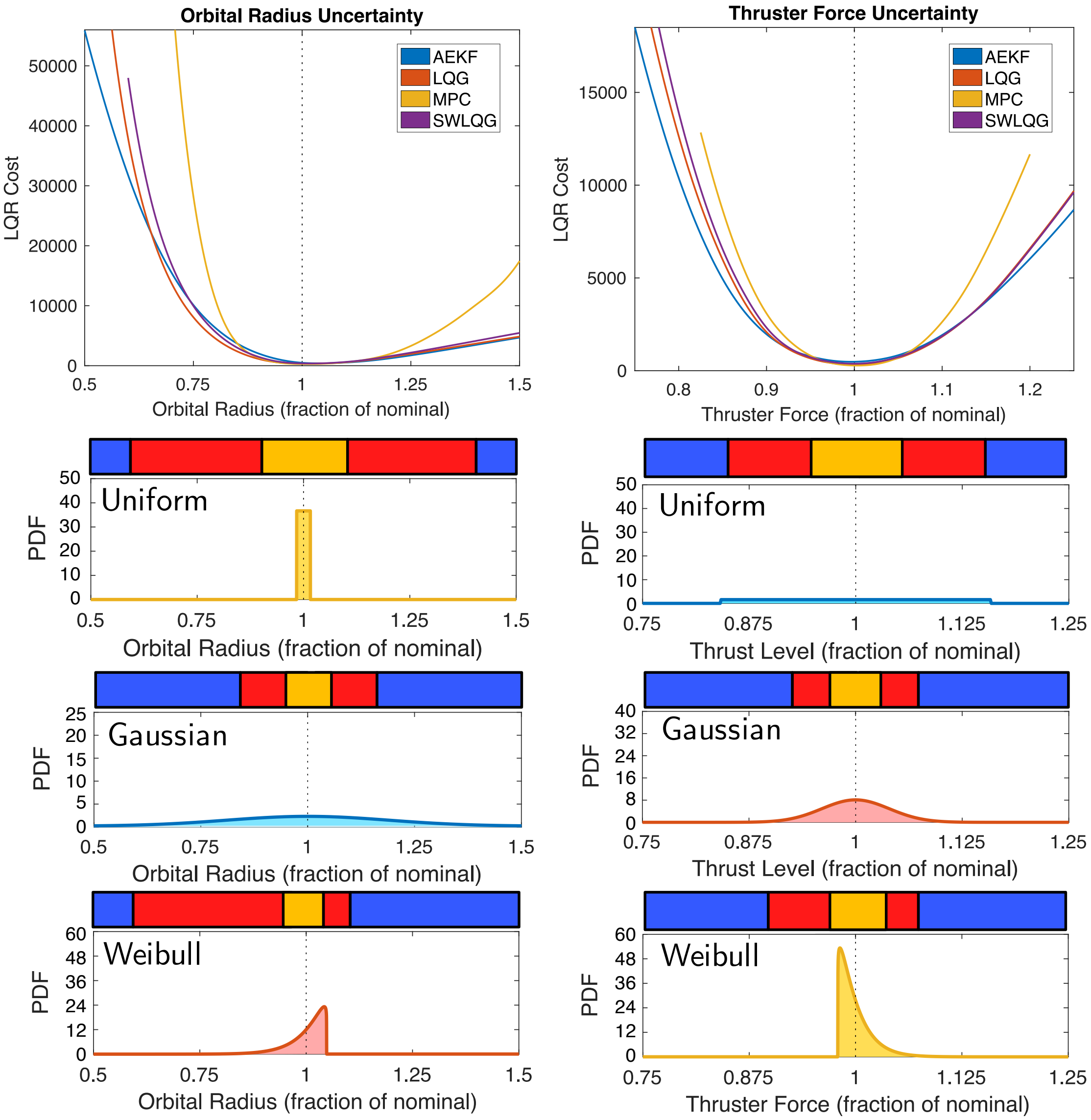


Figure 5-21: Best performing controllers with respect to LQR cost over different uncertainty distributions and levels. Ranges and several example distributions shown colored by which controller would have the optimal LQR cost over that distribution.

Table 5.2: Ranges of uncertainty for which each controller has the best LQR cost under different distributions types for errors in thruster force magnitude

	Uniform	Gaussian	Weibull
Controller	"σ" = halfwidth	"σ" = σ	"σ" = location of PDF max
AEKF	$\lvert\sigma\rvert > 15\%$	$\lvert\sigma\rvert > 8\%$	$\sigma < -10\%$ or $\sigma > 7\%$
LQG	$5\% < \lvert\sigma\rvert < 15\%$	$3\% < \lvert\sigma\rvert < 8\%$	$-10\% < \sigma < -3\%$ or $4\% < \sigma < 7\%$
MPC	$\lvert\sigma\rvert < 5\%$	$\lvert\sigma\rvert < 3\%$	$-3\% < \sigma < 4\%$
SWLQG	-	-	-

Table 5.3: Ranges of uncertainty for which each controller has the best LQR cost under different distributions types for errors in the target's initial orbital radius

	Uniform	Gaussian	Weibull
Controller	"σ" = halfwidth	"σ" = σ	"σ" = location of PDF max
AEKF	$\lvert\sigma\rvert > 42\%$	$\lvert\sigma\rvert > 17\%$	$\sigma < -43\%$ or $\sigma > 10\%$
LQG	$11\% < \lvert\sigma\rvert < 42\%$	$5\% < \lvert\sigma\rvert < 17\%$	$-43\% < \sigma < -6\%$ or $3\% < \sigma < 10\%$
MPC	$\lvert\sigma\rvert < 11\%$	$\lvert\sigma\rvert < 5\%$	$-6\% < \sigma < 3\%$
SWLQG	-	-	-

As the error bars were shown in Figure 5-18 and Figure 5-19 to demonstrate statistical significance of the performance curves, there is a similar approach that can be taken to show areas that have statistically significant best performing controllers under different uncertainty levels. Figure 5-22 shows this analysis, where the same format is used as in Figure 5-21. Although now, the ranges of uncertainty that do not show a statistically significant choice between the adjacent best performing controllers are colored grey. To determine statistical significance here, the concern is essentially the error bars around the boundaries between controllers in Figure 5-21. The process to compute these grey areas is to add or subtract two standard deviations to the performance curves such that the performance curves for adjacent controllers are given "best" and "worst" case bounds. At this point, the integration over these modified curves is performed again, and the area between these two bounds is colored grey.

In Figure 5-22, it can be seen that the grey areas are not insignificant. Although the

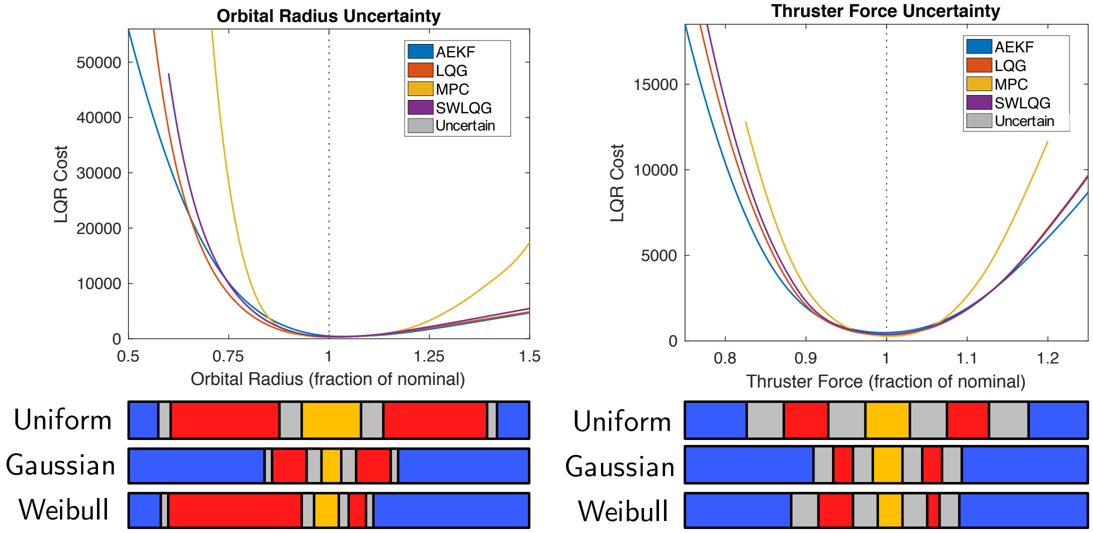


Figure 5-22: Best performing controllers with respect to LQR cost over different uncertainty distributions and levels. Shown in grey are the uncertainty ranges of the boundaries (i.e., when the choice between adjacent best performing controllers is not statistically significant to two standard deviations).

main effects of the analysis remain, there are significant ranges of uncertainty that do not show a statistically clear preference for one controller over another. This uncertainty can be resolved through simply running more Monte Carlo simulation trials. Of course, using more trials takes more time to run, and there must be a balance between acceptable ranges of ambiguity and processing time required. In this thesis, however, the levels of ambiguity between adjacent controllers were deemed acceptable, and the analysis will continue.

These results conclude the univariate analysis presented in this thesis. We have found optimal full-mission controllers in terms of LQR cost and constrained by mission completion time. For this analysis, the fuel and computation time neither had constraints nor any weighting in the final performance criterion. The multivariate case will study individual controller selection for each phase and will also show how the results change when different weightings are placed on fuel and LQR cost to create the final performance metric.

In reality, the uncertainties will not act individually on the system as is assumed in the univariate case. If there are multiple uncertain parameters, they will act on the system together. Therefore, an analysis needs to be performed on the multivariate, joint uncertainty distribution between all of the uncertain parameters involved in the system. For this thesis, both the orbital radius and thruster force magnitude uncertainty will be looked at together. Remember, however, that the multivariate process is both data intensive and time consuming to complete. If the user is heavily constrained in terms of data space or processing time, the univariate analysis may be all that is able to be performed. For this thesis, the multivariate analysis was performed in parallel on 12 cores from three different computers and took a couple days in total to run a performance and robustness evaluation for all four controllers (MPC taking a large majority of the time).

The same process is carried out as detailed when discussing the univariate results and in Section 5.1.2 and Section 5.3. This time the uncertainty range is in two dimensions, and thus performance surfaces are generated instead of performance curves. A Monte Carlo simulation is therefore run on each point in the 2D mesh grid of uncertain parameter values such that the average performance metrics are calculated for every point. Due to the substantial increase in evaluation points, the number of trials for the Monte Carlo simulation is reduced to 1,000 for the LQG and SWLQG controllers, 100 for the AEKF and 50 for the MPC.

The robustness comparison between the different controllers can now be represented in the 2D uncertainty space spanned by the orbital radius and thruster force uncertain parameter error. Figure 5-23 shows the 100% mission success contour lines for each controller. A larger area enclosed by the contour implies a more robust controller. Figure 5-16 and Figure 5-17 show the cross sections of this plot on the dotted, nominal model lines. From here we can see that the AEKF is far more robust than the others, and MPC is by far the

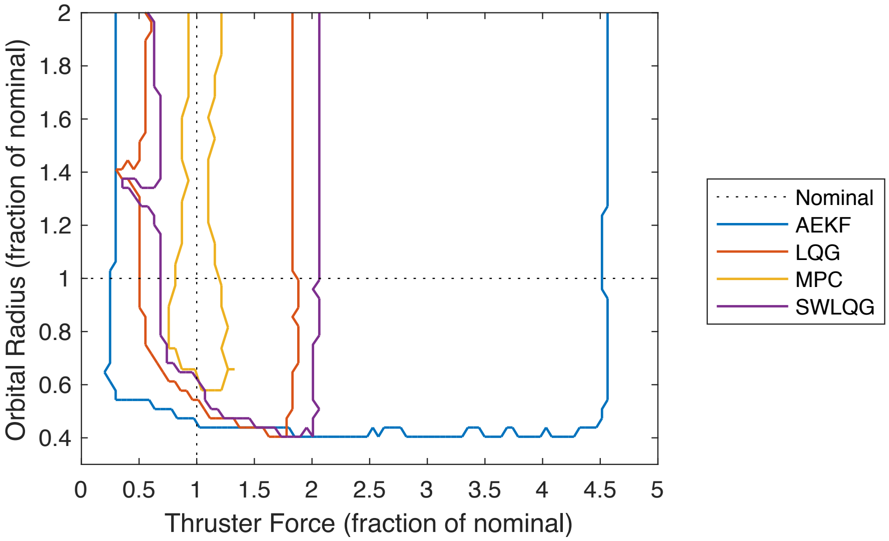


Figure 5-23: Robustness comparison showing 100% mission success contours for each controller over the multivariate uncertainty space of orbital radius and thruster force

most fragile. LQG and SWLQG have comparable enclosed areas but are shifted from each other slightly. There are some interesting outcrops on both of these controllers' contours when there is low thruster force and higher orbital radius. This behavior seems to be due to a cancelling out of effects. The low thrust combines with a low orbital dynamic force to perform similarly to the nominal case. The same behavior happens when the orbital radius shrinks and the thruster force grows. The fast-paced dynamics from lowering the radius seem to be countered by the increase in thruster force and better robustness is seen in these directions. When the thruster force is low and a low orbital radius makes the dynamics faster, the controllers have a more difficult time as shown by the reduced robustness in the bottom left quadrant of the plot. Note that the contours are fairly noisy due to the lower number of Monte Carlo trials being run in this analysis, but also because it is a 100% contour that will skip areas that have say a 99.9% success rate. The drop off in success rate seems to be fairly steep in this analysis however, so the approximation suits us well.

The performance surfaces are computed in the same uncertainty ranges for thruster force magnitude and target orbital radius as were used for the univariate analysis, except this time of course the mesh grid is in two dimensions instead, and the combinations of these uncertain parameter errors are explored. It becomes more difficult to show comparisons of the surfaces with respect to each other in 3D, so instead each of the surfaces will be shown on separate axes with the same axes limits and coloring, such that rough comparisons can be made. Additionally, contour plots can be examined in a qualitative manner. Figure 5-24 shows a comparison of the LQR performance surfaces for all four controllers and Figure 5-25 shows the comparison for the fuel consumption performance surfaces. Only results from the full mission analysis are shown here. Individual phase performance surfaces differ between phases due to the different levels of measurement noise, sensors available, and maneuver goals, and thus the results will vary depending on the particular phase. For the purposes of limiting the already large number of plots in this results section, the individual phase results will be presented without showing the individual performance surfaces. In any case, the full mission surfaces encompass contributions from all phases to give a more hearty test of each controller.

At first glance in Figure 5-24, all of the performance surfaces look very similar. They all share a common shape, almost like a paraboloid or a bowl with a distinct minimum near the nominal model. Looking closer however, there are several things noticeable. First,

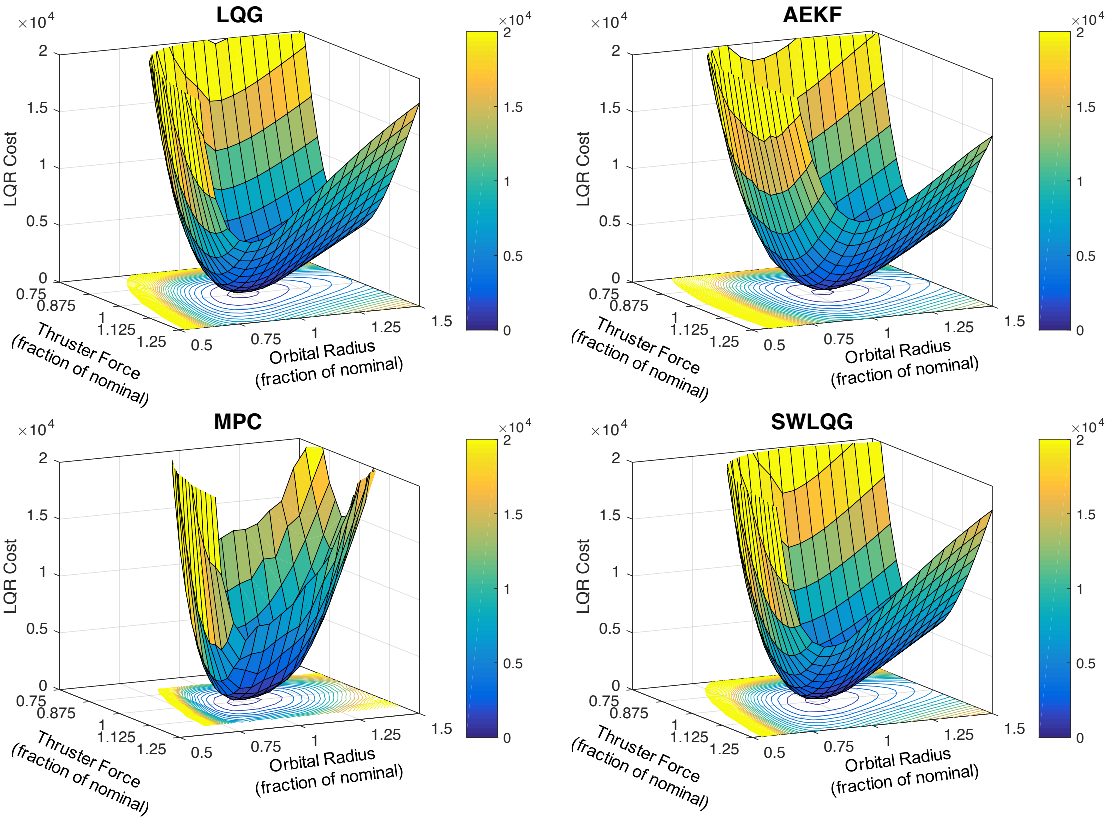


Figure 5-24: Multivariate LQR cost performance surfaces for each controller over orbital radius and thruster force magnitude uncertainty for the full mission

and most obvious is that the MPC performance surface is much narrower than the others and a bit noisier. The narrowness, shows how the performance degrades quickly as large errors are present in the model and the fact that the MPC surface was only computed over a smaller uncertainty range due to this fragility. The noisiness is due to the low number of Monte Carlo trials run. If we look at the first contour of each of the plots, however, we see that the MPC one is wider than the others and the AEKF one is the smallest. This implies that the MPC controller is better performing in terms of LQR cost when the error is small. We can also see looking at the outer edge of the contours that the AEKF covers a larger area in the top left corner, suggesting that the performance of the AEKF is better than the others when thruster force is low and orbital radius is low (i.e., the case where the uncertainties work against each other).

Fuel performance surfaces are shown in Figure 5-25, and contrasting the LQR cost performance surfaces, actually look quite different. A commonality however is that fuel consumption seems to be less, forming a valley in each of the surfaces in the direction where

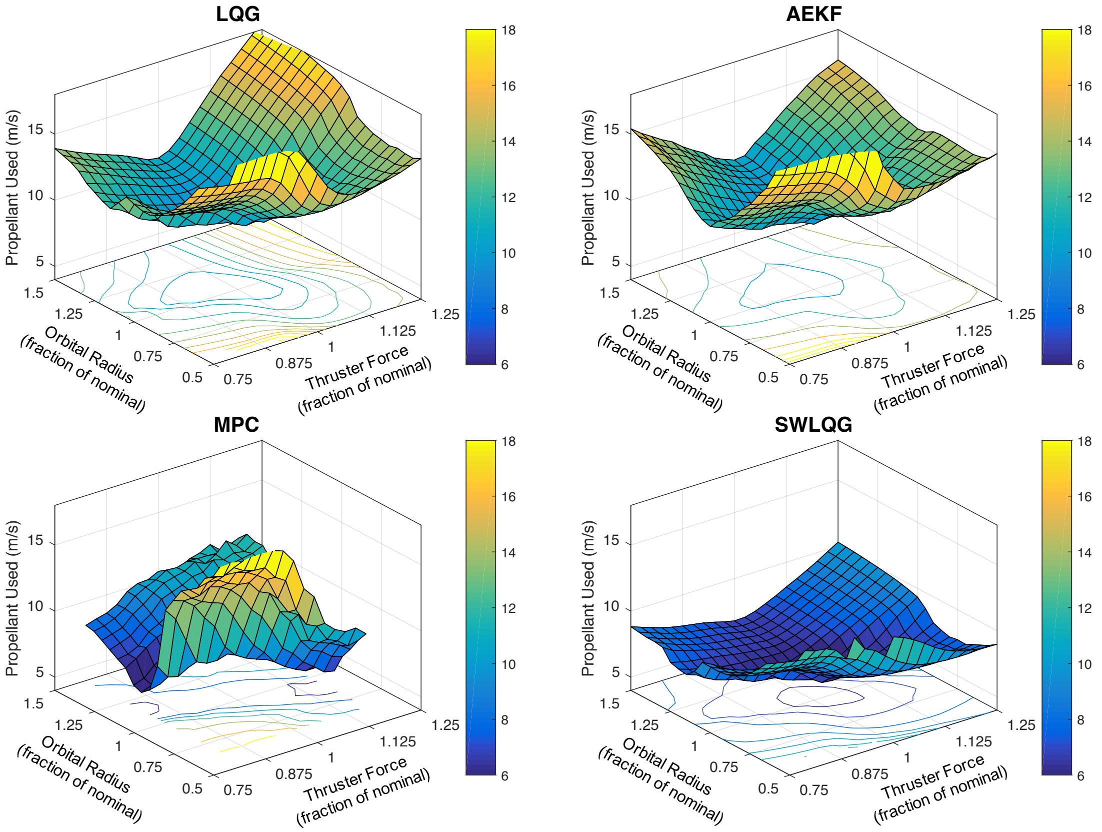


Figure 5-25: Multivariate fuel consumption performance surfaces for each controller over orbital radius and thruster force magnitude uncertainty for the full mission

the uncertainties work together (i.e., when thruster force is low and orbital radius is high or when thruster force if high and orbital radius is low). The contours in this direction are much wider than the orthogonal direction, where the uncertainties work against each other. As expected from the univariate results, the SWLQG far outperforms the others in terms of fuel consumption, hovering below 10 m/s for most of the area, while LQG and AEKF have the highest fuel consumption hanging above 10 m/s for the most part. MPC has some interesting behavior in that it has better fuel performance than nominal when the uncertainties work together. This effect is strange although could be explained by the dynamics doing more work than is expected from the optimization and thus advancing through the trajectory more quickly than is actually desired from an LQR cost perspective (which the MPC controller is attempting to optimize each time step).

Now, when computing the optimal controller for this analysis, the LQR cost and fuel consumption will be weighted individually and the mission completion time and computation time metrics will only be used for probabilistic constraints that guarantee within a certain probability that the mission will succeed and meet these additional metric constraints. These constraints will be applied differently based on a particular phase of the mission. In this situation, the sum of the expected performance in the weighted LQR cost and fuel consumption will dictate which controller is the best. For the full mission, we can plot the best controller over the 2D area of error in the uncertain parameters. Figures 5-26, 5-27, 5-28, 5-29, 5-30, and 5-31 show this for six different weightings between the two metrics. Figure 5-26 shows what happens when only LQR cost is considered (i.e., 0 weighting on fuel). Figure 5-27 depicts the situation when there is a 50% relative weighting on the fuel, Figure 5-28 depicts that when there is an equal weighting, Figure 5-29 when there is a 200% weighting on fuel, Figure 5-30 when there is a 400% weighting on fuel, and Figure 5-31 when there is a 1000% weighting on fuel. If the full fuel plot were shown it would be composed entirely of purple except a small yellow area, so it is left out.

As the importance of fuel increases, we can see the SWLQG controller gradually take over optimality from the LQG and MPC controllers. AEKF tends to have a stranglehold on the areas where thruster force and orbital radius act in opposing directions, and LQG on the areas in which they work together. In the first two plots, the MPC controller is still

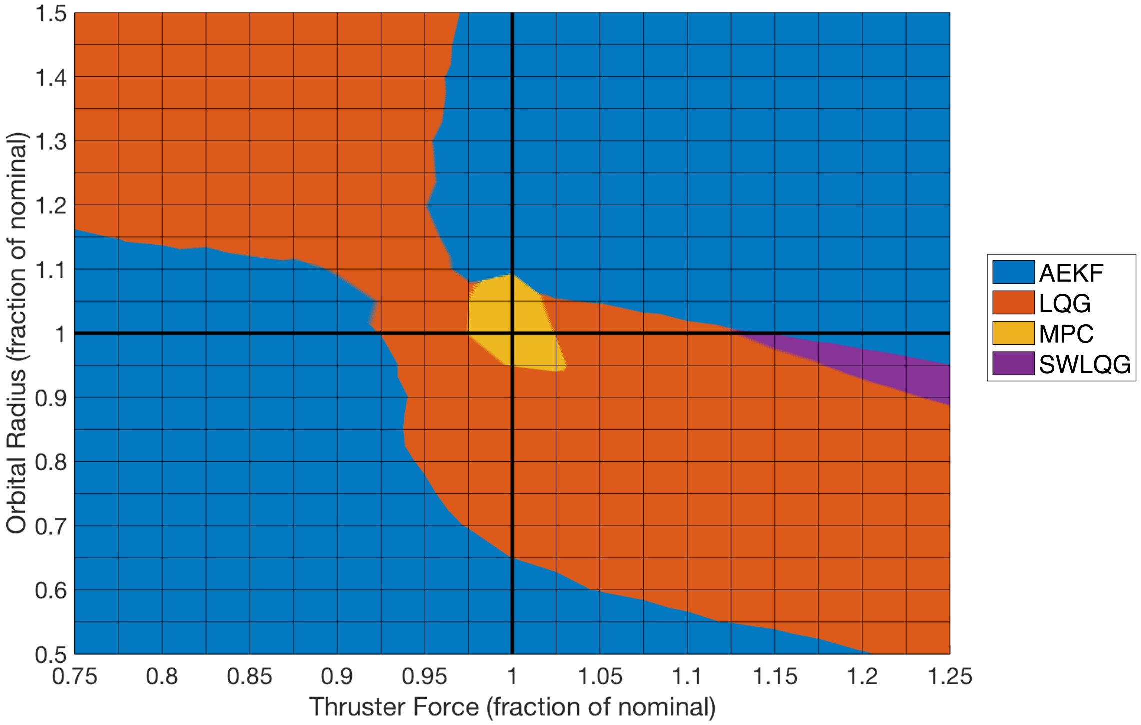


Figure 5-26: Areas of the multivariate uncertainty space over orbital radius and thruster force magnitude that each controller has the best performance (LQR cost)

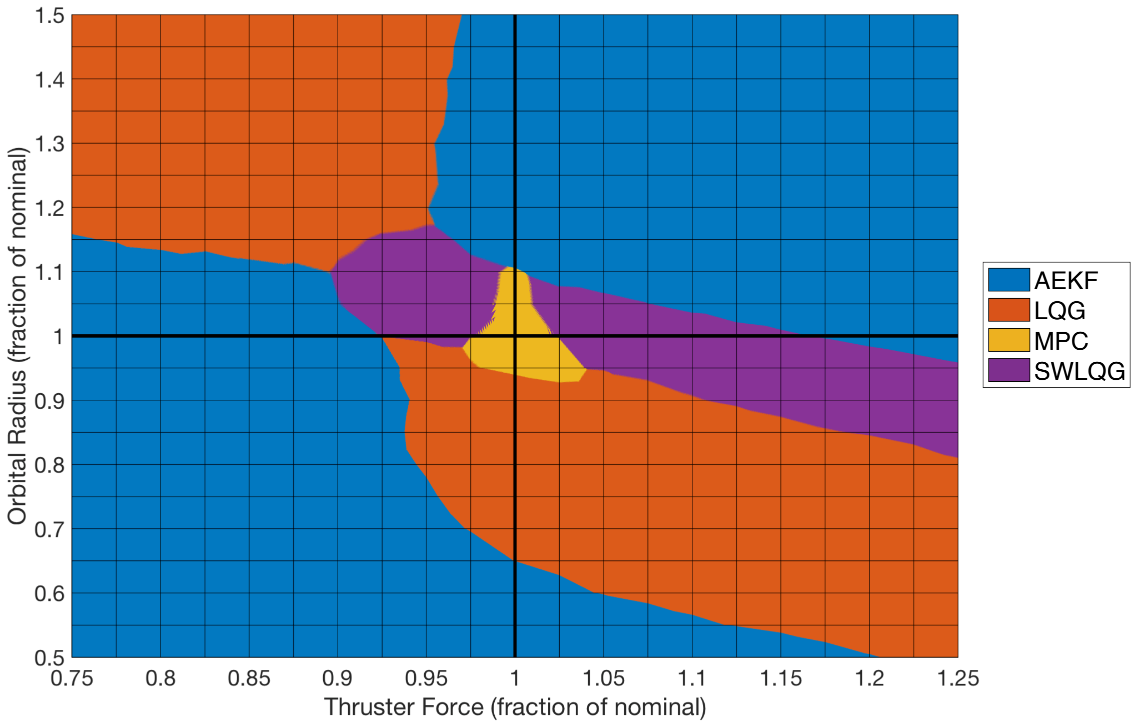


Figure 5-27: Areas of the multivariate uncertainty space over orbital radius and thruster force magnitude that each controller has the best performance (LQR cost + $\frac{1}{2}$×Fuel)

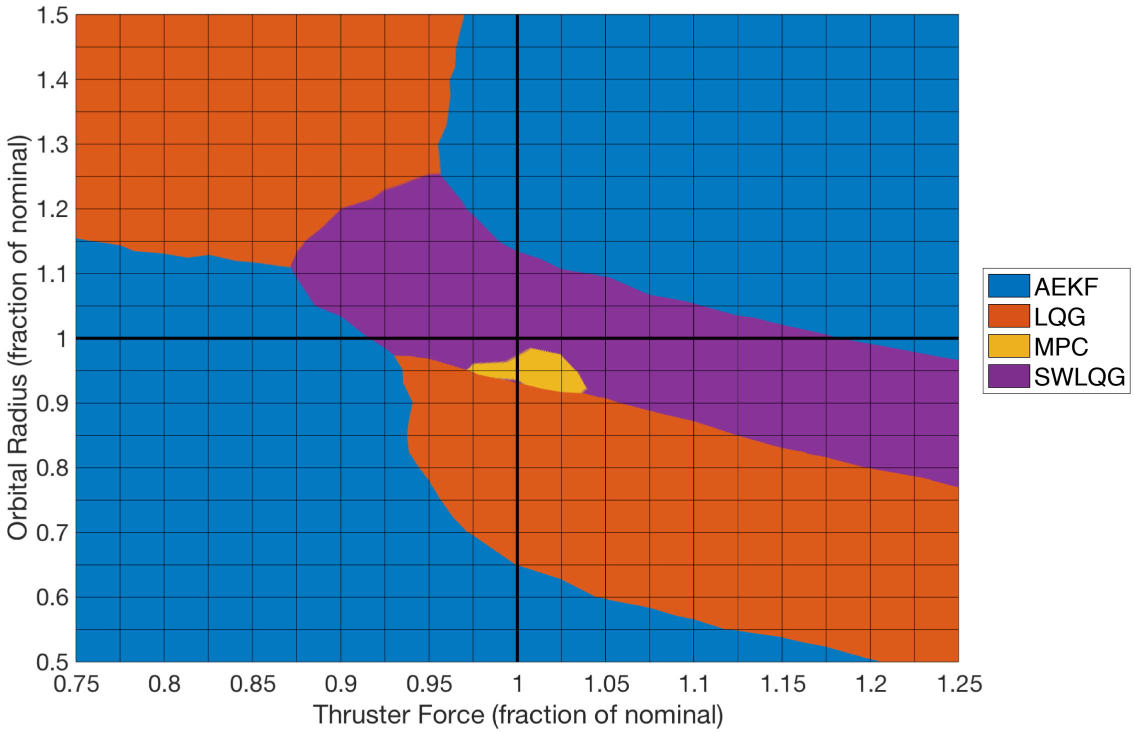


Figure 5-28: Areas of the multivariate uncertainty space over orbital radius and thruster force magnitude that each controller has the best performance (LQR cost + 1×Fuel)

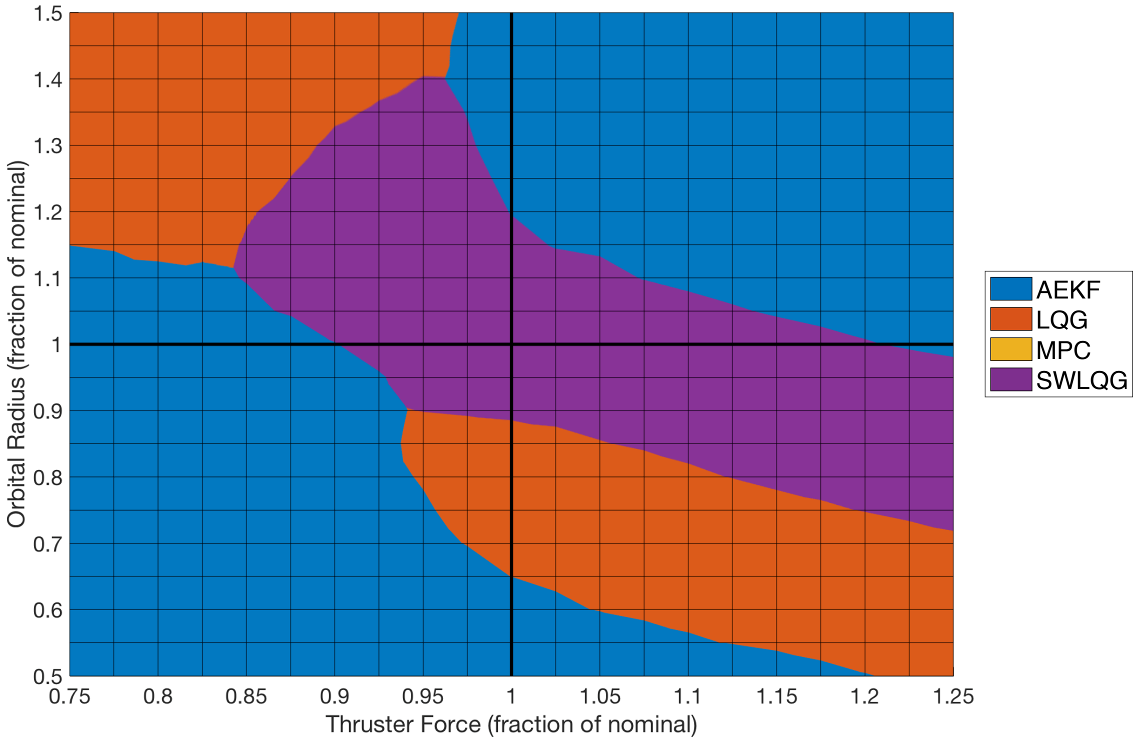


Figure 5-29: Areas of the multivariate uncertainty space over orbital radius and thruster force magnitude that each controller has the best performance (LQR cost + 2×Fuel)

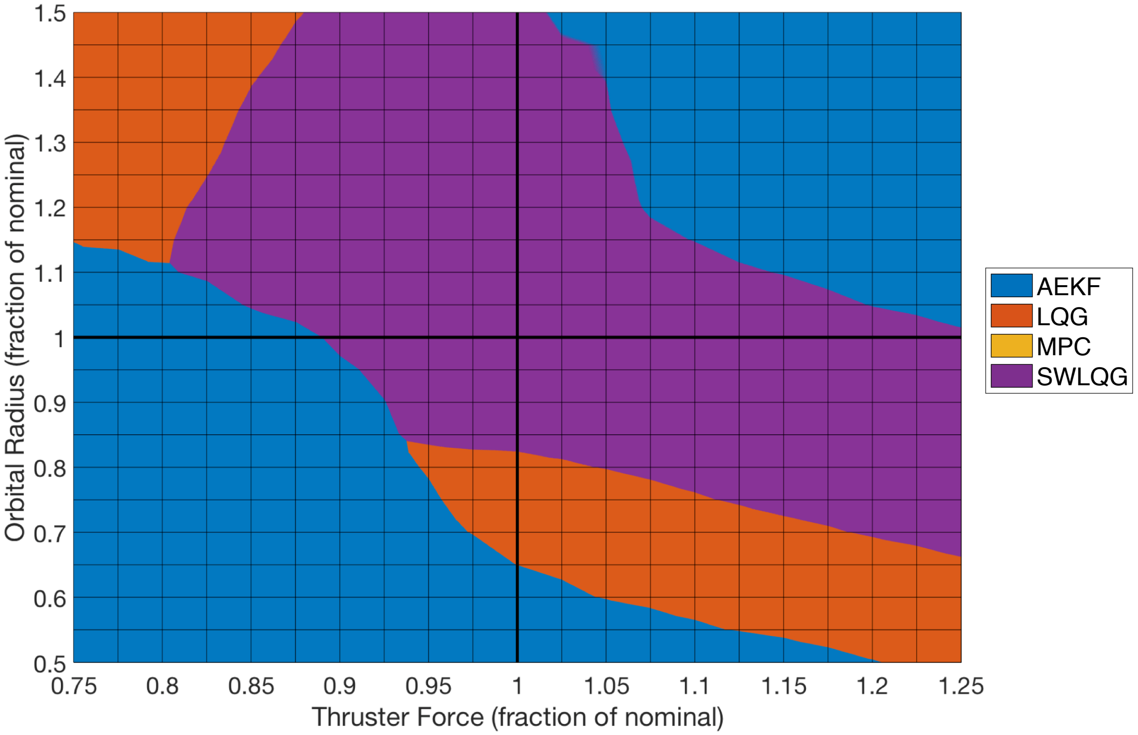


Figure 5-30: Areas of the multivariate uncertainty space over orbital radius and thruster force magnitude that each controller has the best performance (LQR cost + 4×Fuel)

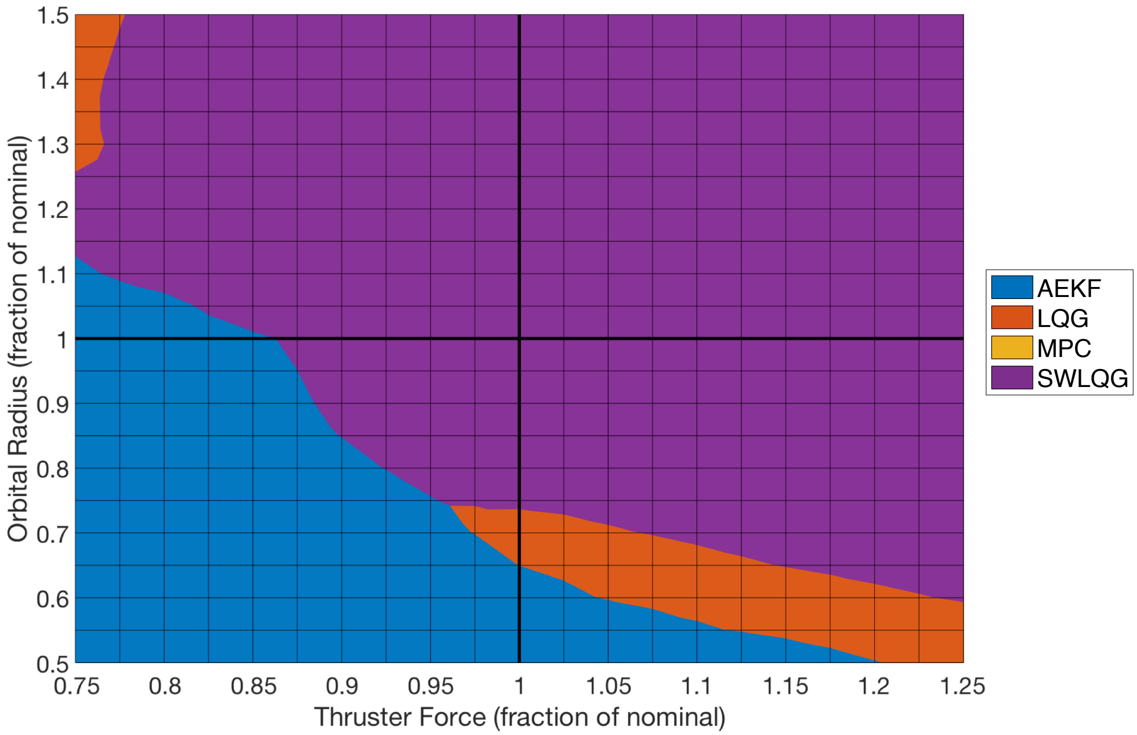


Figure 5-31: Areas of the multivariate uncertainty space over orbital radius and thruster force magnitude that each controller has the best performance (LQR cost + 10×Fuel)

optimal when there are small errors as expected. However, from the third plot on, the SWLQG controller takes over the nominally best spot. Note that Figure 5-18 and Figure 5-19 essentially show cross sections of Figure 5-26 along the nominal model lines.

Figure 5-32 presents a similar error analysis to that shown in the univariate case in Figure 5-22. In the multivariate case, the number of Monte Carlo trials run was an order of magnitude smaller than the univariate case, and as such the bounds are this time shown using only one standard deviation. The same process is followed, where this time one standard deviation is added or subtracted from each performance surface to determine “best” and “worst” case performances surfaces for each controller. The area between these two bounds is colored grey. This analysis must be done pairwise between each of the controllers as the intersecting areas are all combined into the contiguous grey region shown in Figure 5-32. Again, in the LQR-cost-only case, the underlying results for each controller are upheld, although the grey areas are shown to be non-negligible. To decrease the size of these grey areas of ambiguity, more Monte Carlo simulation trials would need to be run. The results at the end of this section fall into the regions where there is a statistically significant controller choice, and this error analysis will not be further explored.

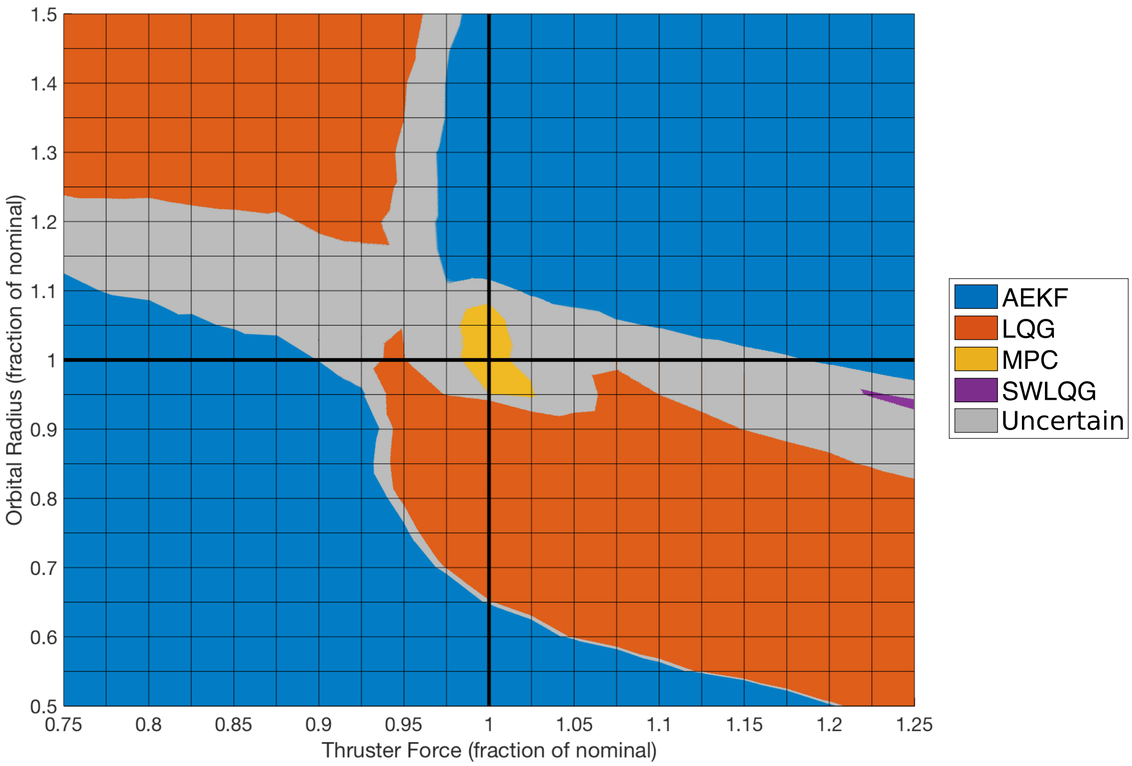


Figure 5-32: Areas of the multivariate uncertainty space over orbital radius and thruster force magnitude that each controller has the best performance (LQR cost + 0×Fuel). Shown in grey are the areas in which the choice between best performing controllers is not statistically significant to one standard deviation.

While interesting and useful to illustrate overall performance, the previous area charts are shown for the case of using the same controller over the full mission, while we are interested in the individual controllers that perform the best in each phase. The following results show examples of finding the best controllers over each individual phase.

First though, since the objectives in each phase are different, the overall performance metric weightings must be chosen to match these objectives. Table 5.4 shows a proposed weighting scheme for this mission. Alternate schemes can be used per the user's discretion. Specifically, in Phase 1a, there is a concern for having low fuel consumption but also not to drift significantly and cause excess error heading into Phase 1b. So, in Phase 1a, the LQR tracking cost and fuel cost are weighted equally. In Phase 1b, the tracking performance is more important since we need to ensure the spacecraft is staged well enough to begin the docking phase. So, we lower the fuel consumption weighting to a quarter. Both of these phases are fairly slow and allow for high computational cost, so we do not place any constraints on the other metrics. In Phase 2, docking is of the utmost importance, and therefore, it is critical to meet the stringent docking constraints and minimize error upon docking. Thus, in Phase 2, we weight the fuel consumption as zero and only assess the LQR cost. Additionally, we place a constraint on the computation time to account for extra computation costs from docking sensor processing and the more time critical reaction required in some cases. Following the mission CONOPS, there is also a constraint that docking happens before an eclipse blocks out the fiducial markings, and therefore, the phase end time is also constrained. In Phase 3, tracking performance is important in case an obstacle constraint were encountered as in Section 4.4.3, but it is also the most fuel intensive phase, thus we also care about fuel efficiency. Finally, there is a full mission duration timeout that is enforced in Phase 3. Table 5.4 summarizes this discussion.

Table 5.4: Phase-by-phase description of performance metric weightings and constraints over which to optimize a baseline controller schedule

	Phase 1a	Phase 1b	Phase 2	Phase 3
LQR Cost Weighting	1	1	1	1
Fuel Cost Weighting	1	0.25	0	0.5
Phase Time Constraint	No	No	Yes	Yes
Computation Constraint	No	No	Yes	No

Using the weightings and constraints in Table 5.4, Figure 5-33 shows the areas in which each controller is optimal for individual phases. With no error in the model of the system, the best performing controllers for each phase are SWLQG in Phases 1a and 1b, LQG in Phase 2, and MPC in Phase 3. Thus, only considering the different aleatoric noise levels and the different objectives in each phase, there are particular controllers that favor particular phases. This confirms some of the motivational material presented with regards to this topic in Chapter 1.

The individual phase area charts in Figure 5-33 show different behavior than those seen in the full-mission versions from Figure 5-26 through Figure 5-31 for the most part. Phase 3 shows the most similarity, with the AEKF, SWLQG and LQG regions remaining very close to the full-mission case, but the MPC region has grown significantly. The MPC technique does its best in the joint maneuver phase, while does oddly perform well with moderately high thruster force error in comparison to the others in Phase 1a. This odd behavior is

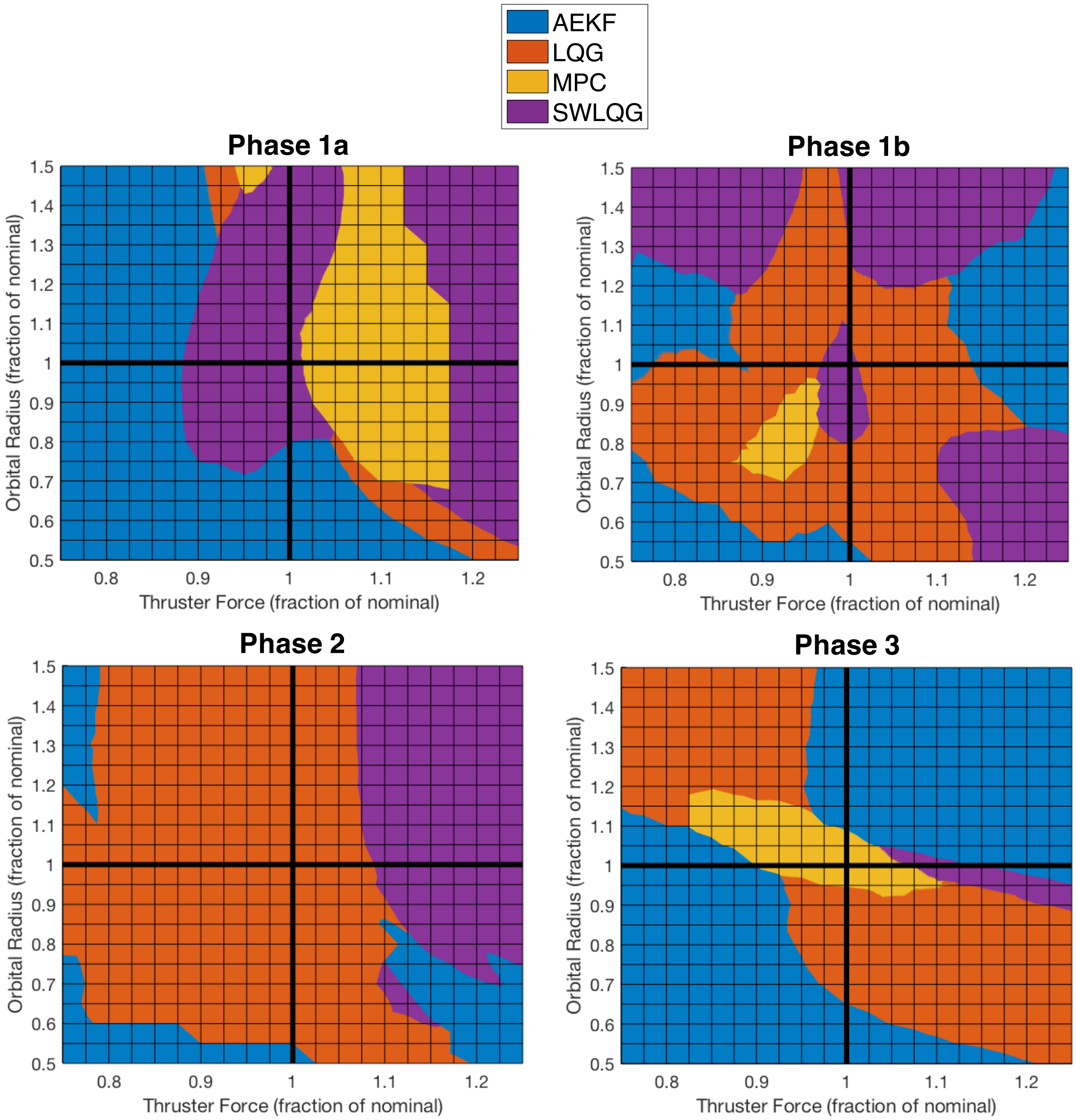


Figure 5-33: Areas of the multivariate uncertainty space over orbital radius and thruster force magnitude that each controller has the best performance (per weightings in Table 5.4) for each individual phase

mainly due to the better fuel performance as seen and discussed in Figure 5-20 for the mid-range thruster force error. The AEKF technique maintains dominance of the outer edges of the uncertainty range for the most part, and thus, in general, seems to still be the best choice when the error is large in the uncertain parameter. The LQG controller makes up a large portion of the Phase 1b, 2 and 3 charts, but not as much in Phase 1a, where there seems to be a benefit to fuel efficient techniques and those better able to react to deviances from the reference trajectory.

Another benefit of splitting this analysis into individual phases is that we can individually vary uncertainty levels in the different phases to see if lower uncertainty in these parameters is expected in certain phases, to only analyze that particular phase by those levels. For example, if we look at an uncertain target spacecraft mass. This will add additional uncertainty to our thruster force magnitude (which represents both uncertainties in mass and thruster force as discussed in Section 5.1.3), but only for Phase 3, the joint maneuver phase. By analyzing the results here in Phase 3 under the assumption of a higher level of uncertainty in Phase 3, we can account for cases like this.

These plots are interesting when looking at best performance in terms of a specific instantiation of the error in the uncertain parameters, but what we really want is to know the controller that has the best probabilistically expected performance over the full uncertainty distribution. These area plots do not tell the full story though, as due to the asymmetric nature of the chart, a controller could only be slightly worse performing in all areas and not show up in this plot, but still be best performing overall on a particular range. Thus, it is important to perform the expected performance calculation based on integrating over the full probability density function representing the uncertain parameter distribution. In the multivariate case for two uncertain parameters, this means performing the volume integral over the joint PDF. If this is performed for each controller in each phase, the result will be a single controller that has the probabilistically best performance over that specific uncertainty distribution. This controller will be the answer to the question of which reference-tracking controller would best balance the tradeoff between performance and robustness.

For the online controller scheduling process, all of the individual phase performance surfaces are uploaded to the spacecraft, and all that needs to be done onboard is to perform this integration given the current uncertainty distribution in the relevant parameters and select the best performing controller for the upcoming phases. As the process is the same from here on out, there will not be any distinction between the results that would be obtained for either the analysis before or during the mission. The only difference during the mission (not covered here) is that the uncertainty distribution may not have its mean at the nominal model location, and therefore the integration will be performed with an off-center distribution. This assumes the onboard controller designs are not updated with the information from the new mean of the distribution.

Table 5.5: Probabilistically optimal reference-tracking controllers chosen for each mission phase under three Gaussian uncertainty distributions

	Low Uncertainty	Medium Uncertainty	Large Uncertainty
Covariance (Σ)	$\begin{bmatrix} (10km)^2 & 0 \\ 0 & (0.2N)^2 \end{bmatrix}$	$\begin{bmatrix} (10km)^2 & 0 \\ 0 & (2N)^2 \end{bmatrix}$	$\begin{bmatrix} (100km)^2 & 0 \\ 0 & (4N)^2 \end{bmatrix}$
Phase 1a	SWLQG	AEKF	AEKF
Phase 1b	SWLQG	SWLQG	AEKF
Phase 2	LQG	LQG	LQG
Phase 3	MPC	AEKF	AEKF

In either case, if we perform this integration with the metric weightings and constraints given in Table 5.4, we can obtain optimal controller choices for each phase under specific uncertainty ranges. A summary of these results is shown in Table 5.5 for a few uncertainty distribution options ranging from low uncertainty in the range of 1% to high uncertainty of 40%. Note that the uncertainty of the target's initial orbital radius is kept to reasonable values instead of exploring the full range of ±50%, since this range was only explored for academic interest, and in reality, the uncertainties in this parameter will not grow that large. There is, however, one situation in which the orbital radius uncertainty may reach these large ranges. If an orbital debris removal spacecraft or a servicing spacecraft were only going to implement one controller over its lifetime and were to visit multiple targets at possibly very different orbital altitudes, the uncertainty distribution of target radii may be analyzed here to select the best controller needed in this case.

As the uncertainty level increases, we see that the AEKF controller gradually becomes the optimal controller for all of the phases except for the docking phase. In the docking phase, Phase 2, the measurements are very precise, and thus the EKF does not trust the model nearly as much, and the estimation does not suffer. The LQG controller performs the best in Phase 2 throughout any reasonable uncertainty range. When the estimation does not suffer, the LQG controller (although with suboptimal gains) is able to track the reference much better. Remember that there is a computation constraint in Phase 2, so the MPC technique was not in the running for the selection. MPC does have the best performance with small error in Phase 3, but cedes to the AEKF by about 9% variance in thrust and about 7% variance in radius.

In Phase 1b, the area chart makes it seem like LQG may be better under moderate amounts of uncertainty. In reality, the performance surfaces are very close to each other. And because the Gaussian PDF weighs the area closer to the nominal model heavier than elsewhere, the SWLQG controller's expected performance actually holds for a longer period than it looks at first. It then directly transitions to AEKF as the uncertainty increases rather than going through an LQG period. This is one example of the drawbacks of relying heavily on the area charts in Figure 5-33 for analysis rather than performing the true expected value computations.

With the probabilistically optimal reference-tracking controller schedule now found, it can be implemented on the spacecraft in the simulation discussed in Section 3.4. For resolution of the chapter and checking that the chosen schedule does indeed seem realistic, the simulation results of using this controller scheduler under a mission with the specified uncertainty are presented. The smallest uncertainty level analyzed above is used for this analysis as it is most typical of current missions (where the higher uncertainty levels are applicable mostly to future missions). The thruster force is increased and the orbital radius decreased by one standard deviation of this distribution. Additionally, a −100-kg error in the target spacecraft mass is added for Phase 3. Thus, following Table 5.5, the SWLQG controller will be used in Phases 1a and 1b, the LQG controller in Phase 2, and the MPC controller in Phase 3. For all of these controllers, the EKF is used to estimate the spacecraft state.

Figure 5-34 shows the resulting position and velocity states throughout the mission. From the position plot, the chaser starts at -3 km in the in-track direction, docks to the target at the end of Phase 2 at about the 4-hour mark of the mission (240 minutes), and finally tugs the target to the assembly location at 5 km in the in-track direction. The velocity plot shows the maximum velocity in the initial CWH frame is about 1 m/s and that the velocity is kept below the 0.05 m/s constraint in the docking phase. Noisier velocity states correspond to when there is higher noise in the estimate of the state due to the high covariance in the range measurement in Phases 1b and 3. Since the maneuver is in the orbital plane, the z-axis components of the position and velocity are nearly always close to zero as expected.

Figure 5-35 shows the estimator error and covariance over the full mission as the measurement uncertainty changes between each phase. For the most part, each phase boundary

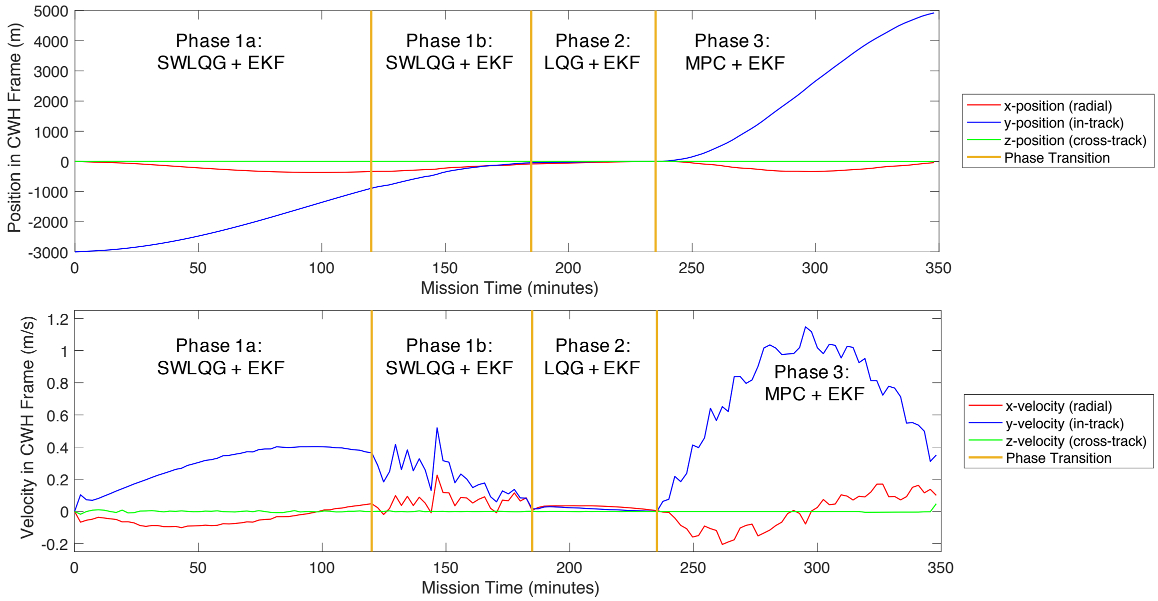


Figure 5-34: Position and velocity states throughout the full mission as result of implementing the previously computed best controller schedule while under the smallest uncertainty level

shows a distinct jump in the performance of the estimation. In Phase 1a, the unobservability due to the absence of a range measurement causes the error to grow without the spacecraft realizing it. In Phase 2a, the noisy range measurement and continual control corrections

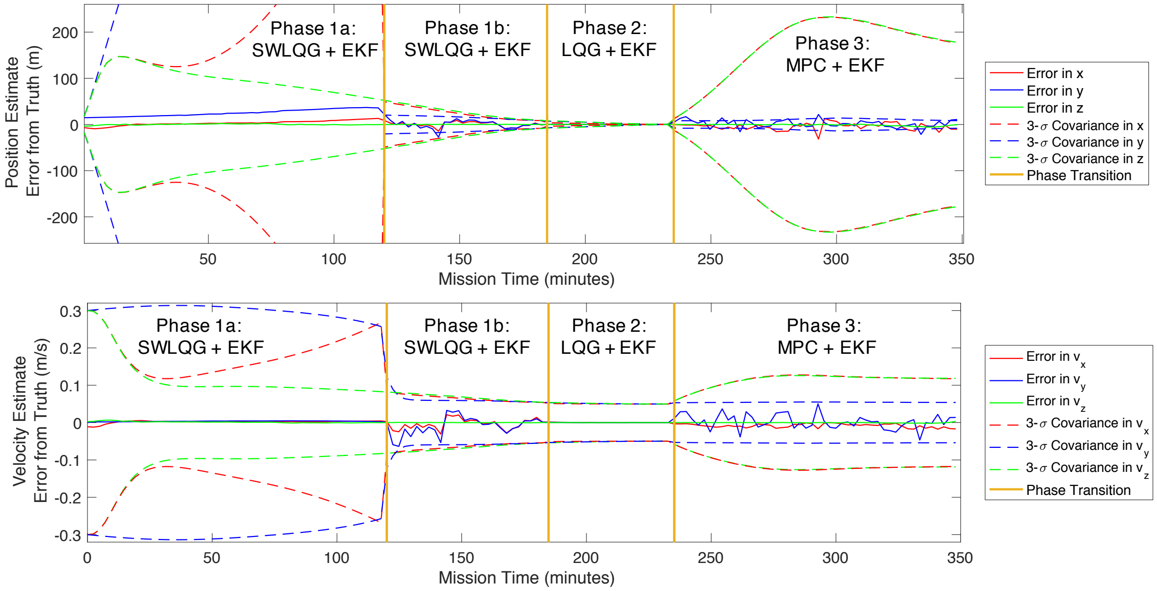


Figure 5-35: Estimation error and covariance in position and velocity throughout the full mission as result of implementing the previously computed best controller schedule while under the smallest uncertainty level

cause the error to be a bit more sporadic, although the drifting error in the position from Phase 1a is resolved within a few minutes. In Phase 2, the error and covariance continue to converge through the entire phase such that appropriate position and velocity error exists at docking (under 1 cm and 0.5 mm/s). In Phase 3, the estimation error and covariance grow again due to the larger range uncertainty present in this phase.

Figure 5-36 shows the tracking error of the true spacecraft position and velocity with respect to the reference trajectory. Important to note is the drift in the position tracking error in Phase 1a due to the unobservability, followed by a convergence to a modest 10-20m error in Phase 1b. The velocity tracking is actually quite good in Phase 1a as the model of the system is quite accurate for velocity. The error grows in velocity in Phase 1b due to the larger thruster commands and growing impact of range measurement error. In the plots on the right side, the convergence of the tracking error during the docking phase is illustrated to confirm that the tracking error is less than about 20cm in position and 1 mm/s in velocity. These values are well within the ranges required for the ISS docking system [1]. In Phase 3, the tracking error grows as the spacecraft begins to lead the trajectory by up to 100 m due to the increased thruster force, decreased orbital radius and decreased target spacecraft mass. Because tracking is not as important in this phase, these results are acceptable.

Finally, the fuel consumption and commanded control inputs from the mission are plot-

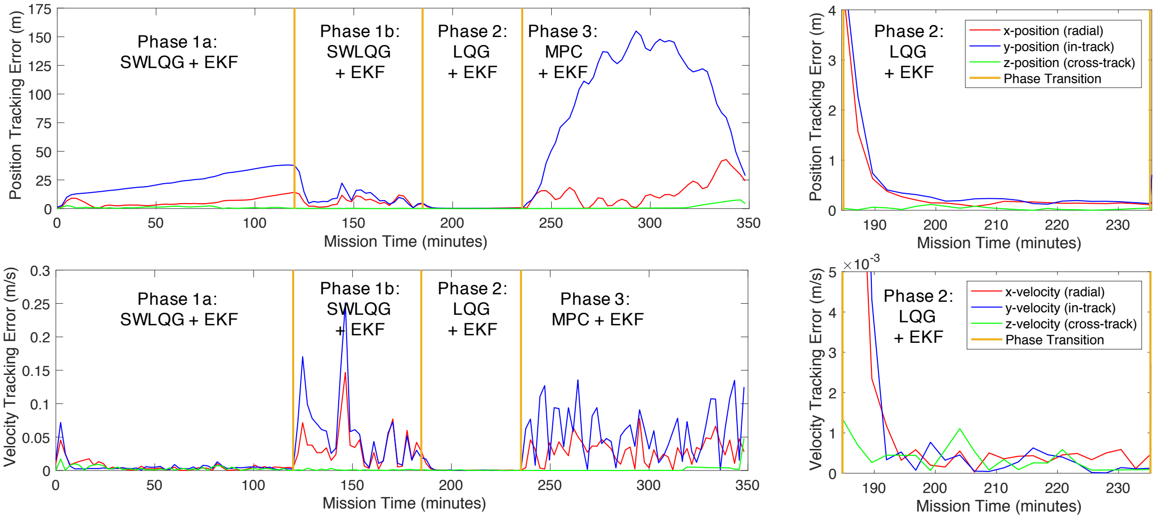


Figure 5-36: Reference tracking error in position and velocity throughout the full mission as result of implementing the previously computed best controller schedule while under the smallest uncertainty level

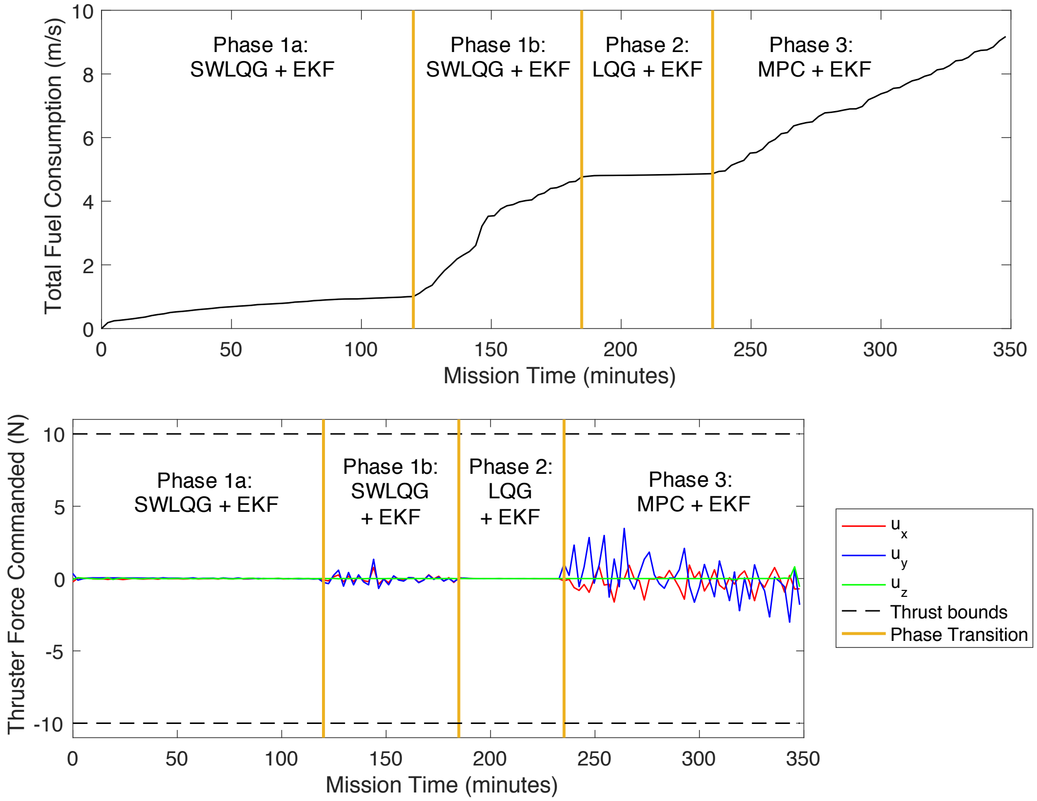


Figure 5-37: Total fuel consumption and control commanded throughout the full mission as result of implementing the previously computed best controller schedule while under the smallest uncertainty level

ted in Figure 5-37. Phases 1b and 3 are the largest contributors to the total fuel consumption mainly due to the noisy estimation present in these phases. The commanded control in these phases is also quite noisy due to this uncertainty in state. Phase 2 contributes almost nothing to the fuel cost of the mission, as is expected, since it is the phase with the highest measurement precision and lowest velocity. Therefore, all seems well within reason for a typical rendezvous and docking mission, and it would seem the combination of these controllers into the baseline optimal schedule proves effective.

With the results just presented, the second objective of the thesis has been satisfied. The process defined earlier in the chapter has been followed to schedule reference-tracking controllers throughout the multi-stage rendezvous and docking mission such that the selected controllers have the probabilistically best performance given aleatoric and epistemic uncertainties in the dynamics. These controllers can each be run in real-time on a spacecraft, as can the simplified process of onboard controller scheduling (given that enough data storage space is available to hold the set of performance hypersurfaces). Therefore, paired with the real-time trajectory optimization techniques discussed in Chapter 4, the third thesis objective has also now been satisfied.

Chapter 6

Conclusion

6.1 Summary

The popularity of rendezvous and docking missions is growing on several fronts. Future missions including satellite servicing and repair, orbital debris removal, and on-orbit assembly all require rendezvous and docking. Several of these missions also require maneuvering with the target of interest after docking. This thesis has divided missions like these into a common set of phases, each with their own objectives, dynamics, and constraints. A generic problem formulation has been created to define the different phases and allow analysis on techniques that optimize trajectories and select the best performing reference-tracking controllers throughout the full, multi-stage mission.

Today's rendezvous and docking missions are planned to very well-known, designed-to-be-docked-to targets. With these new mission types on the horizon, there also comes a higher degree of uncertainty in the system. When performing orbital debris removal or servicing of broken satellites, the state of the target object may be unknown. Tracking from ground sensors may not be able to resolve this uncertainty, and thus, missions must be able to perform well while considering the lack of concrete knowledge of system parameters. In the context of rendezvous, docking and joint maneuvering missions, this thesis studies both aleatoric uncertainty such as measurement and process noise and epistemic uncertainty such as errors in thruster force magnitude or other constant parameters in the dynamics.

To address these uncertainties in the planning process of multi-stage missions, the problem was split into two separate areas: trajectory optimization and reference-tracking controllers. The overall goal in the thesis was to find an appropriate balance of risk aversion and performance, such that sufficient safety margins can be achieved, while still performing well in terms of fuel consumption and tracking error. Safer strategies will inherently involve decreased performance in these areas, yet high performing strategies will be very risky. So, in the planning process, the probabilistically optimal mission plan was found, such that the performance was maximized on average across the full uncertainty range possible in the system.

To this end, a process was created for probabilistically optimizing baseline trajectories for both a minimum fuel and minimum energy objective function. These baseline trajectories were designed and tested for two case studies. An obstacle avoidance problem, where initial large uncertainty of an obstacle will be reduced when onboard sensors are in range. The goal was to find the best performing safety margin around the initially estimated obstacle state. The same concept was applied to a target satellite with an unknown attitude and thus with uncertainty in the direction of the docking axis. Risk margins of varying degrees were added to the path constraints in these problems, and the trajectory corresponding to the risk margin with the lowest expected cost was chosen as the optimal baseline trajectory.

Additionally, the problem of selecting the best reference-tracking controllers to follow these optimal trajectories was explored under significant uncertainties in the dynamics. Several controllers were designed with varying levels of nominal performance and robustness to uncertainty. Each of the mission phases involves different levels of noise, and there could also be uncertain parameters in the dynamics formulation. Both of these types of uncertainties were added to the problem, and a Monte Carlo based simulation was used to assess controller performance over a range of these possible uncertainties. The controllers were assessed and compared across different levels of uncertainty, and the probabilistically best performing controllers were chosen for each phase given the expected uncertainty distributions for the upcoming mission.

Finally, we can consider the multitude of anomalies and mishaps seen historically on orbit as detailed in Section 1.4. When applying the techniques discussed in this thesis, the impact of several of these events could have been reduced significantly. The main groupings of these events consist of actuator failures and estimation failures. It is not that these failures could have been prevented themselves, but that the trajectory design and controller selection techniques could have improved the performance in the event that they did happen. In the actuator failure scenario, prior work had addressed the fully failed thruster problem. This thesis addresses the case where there are anomalous firings or constant bias added to the level of the thruster force. We have analyzed and selected reference-tracking controllers given large uncertainties in the forces of thrusters. If these selected controllers were implemented onboard these previous missions with thruster force problems, the fuel and tracking performance of the mission would have improved given the added robustness. For estimation drift, sensing errors or similar anomalies, an online trajectory replanning technique was shown in this thesis that could plan a new trajectory from the current location and greatly reduce the stoppage time and fuel wasted during several of these historical-seen errors. Overall, the techniques in this thesis could not have prevented these mishaps, but would have led to better performance under them.

6.2 Contributions

The following is a list of contributions that this thesis will provide to the field. The major, numbered contributions align with the thesis objectives described in Section 1.6, while the subpoints go into more detail about the contribution, even expressing minor additional contributions in that area.

1. A process is developed for generating probabilistically optimal baseline trajectories while accounting for the possibility of actionable uncertain events throughout the multi-stage rendezvous, docking and joint maneuvering mission.

    - A general framework and methodology is presented in a form that predicts the best risk margin by which to avoid path constraints such that on average the total cost of the mission is minimized by trading off increases in initial trajectory cost and replanning cost.
    - Two case studies are investigated as examples of how effective this process can be. In one case study, obstacle avoidance is studied when there is a large initial uncertainty in the location of the obstacle. Onboard sensors will reduce this uncertainty later in the mission, and the trajectory will be replanned at this point to avoid the obstacle. It is shown that there is an intermediate safety margin to avoid the original estimate of the obstacle which produces better expected performance. Additionally, the same concept is explored for a target satellite with

  an uncertain attitude that is tied to a docking port alignment constraint, and similar results are presented showing performance improvement over traditional risk-averse maneuvers while maintaining the same level of safety.

  - Two cost functions are also covered in this optimization framework: a minimum energy cost posed as an LQR cost and a minimum fuel cost posed linearly. Results show that this technique is viable for use on both cost functions, yielding very similar results.

2. Another process is developed for selecting and scheduling the best reference-tracking controllers for use in multi-stage rendezvous, docking, and joint spacecraft systems under different levels and distributions of uncertainty.

   - Several reference-tracking controller formulations are applied to this problem: LQG, SWLQG, MPC and AEKF. All controllers are developed with the same modular inputs and outputs such that they are swappable and contained in a controller library. The controllers cover a varying range of performance and robustness capabilities such that interesting trades can be performed with respect to which perform the best under specific uncertainties.
   - A selection methodology is presented that ranks and schedules controllers based on expected performance over a particular uncertainty distribution. The methodology covers probabilistic metric evaluation of each of the controllers under aleatoric uncertainties such as sensor and actuator noise and epistemic uncertainties for parameter error in thruster force magnitude and initial radius of the target spacecraft (as both of these parameters help define the CWH dynamics).
   - Results analyze the optimal selection of controllers for the full mission and individual mission phases given this controller library and these uncertainties. Examples are shown for typical uncertainty distributions, as well as which controller performs the best under specific errors in uncertain parameters.
   - A simplification for on-line controller selection logic is suggested that will update the optimal controller schedule based upon current knowledge of uncertainty in the system. The simplification involves onboard integration of performance results (which were computed during the pre-mission analysis) over the current probability density function of the uncertain parameters in the problem.

3. Onboard guidance and control techniques are developed and analyzed for multi-stage close proximity spacecraft operations under uncertainties and uncertain events.

    - Trajectories are optimized continuously between multiple stages without stopping points resulting in increased performance over traditional methods that use waypoints between phases.
    - Several reactionary techniques are developed and analyzed to replan missions after uncertain events have occurred. These techniques include online trajectory optimization to react to obstacles, large tracking error when transitioning phases, and other knowledge illuminating events.
    - The trajectory optimization presented here is capable of being performed in real-time for the multi-stage missions including obstacle constraints in the form of an ellipsoidal, nonlinear constraint avoidance formulation and an improved upon form of a rotating-hyperplane constraint.
    - The thesis defines a generic multi-stage rendezvous and docking mission framework that has and can continue to be used by others to benchmark results.
    - A generalized multi-stage rendezvous, docking, joint maneuvering, and undocking simulation environment is coded to include uncertain events and uncertainties.

6.3 Future Work

As this thesis was completed in finite time, there is undoubtedly areas that could have been explored deeper and would serve as interesting areas of future study. In direct line with this work, additional case studies could be performed for new uncertain events or under different uncertain parameter errors. For example, the probabilistic trajectory optimization process could be thoroughly analyzed on initial obstacle uncertainty of varying size, location, and speed. Baseline trajectories could also be computed for the uncertain event of significant error experienced when transitioning from Phase 1, where there are only angular measurements, into Phase 1b, where range can be detected.

With regard to the reference-tracking controllers, additional uncertain parameters could be investigated such as sensor bias, thruster misalignment, and magnitudes of sensor and actuator noise. Extra controllers could be investigated as well to assess performance over a larger controller library, such as some of the control methods mentioned in the literature review chapter. With this larger library, a Pareto front analysis could be performed to better analyze the trades in a multi-objective framework.

Additionally, the studies in the thesis could be explored at different orbital radii such as in LEO instead of GEO, or even in more eccentric orbits. Sensitivity studies could be performed on particular parameters chosen for the mission example in this thesis, such as on spacecraft mass, maximum thruster force, range of sensors, precision of sensors, docking port location and tumbling of the target satellite.

Areas of future work that go beyond what is presented in this thesis could attempt to reduce the complexity of the multiple obstacle (or multiple uncertain event) planning framework such that it scales better with the number of obstacles. A POMDP framework could be explored for this development as has gained popularity in the field of planning under uncertainty. Also, the implementation of controller switching at intermediary points in the mission instead of only directly at the physical phase transitions could be studied. Additionally, a hybrid control robustness analysis could be performed using formal methods on the supervisory controller that switches between reference-tracking controllers as necessary. One other area would be to combine the probabilistic trajectory planning with the controller-scheduling problem such that controller tracking error is worked into the path constraint formulations in a chance constraint manner.

Finally, the work could be implemented on hardware and applied in a relevant environment to see if the techniques can be utilized effectively in real-world circumstances. This experimentation would serve to validate the results and simulation discussed in this thesis. A plan to validate this work on hardware should involve testing of the baseline computed trajectories and controller plans in an iterative manner to confirm the performance gains anticipated. Additionally, the online trajectory generation and controller scheduling methods should be implemented on the hardware and run in real time during an experiment such that onboard use can be verified. As a final suggestion, it would be beneficial to simulate uncertain events of all varieties to see how well these techniques can replan onboard the spacecraft in the event that something off-nominal were to occur during the mission. This validation is absolutely necessary before the techniques from this thesis are adopted in future rendezvous and docking missions.

References

[1] Wigbert Fehse. *Automated Rendezvous and Docking of Spacecraft.* Cambridge University Press, Cambridge, 2003.

[2] Emilio De Pasquale. ATV Jules Verne: a Step by Step Approach for In- Orbit Demonstration of New Rendezvous Technologies. *SpaceOps 2012 Conference*, pages 1–18, 2012.

[3] Isao Kawano, Masaaki Mokuno, Toru Kasai, and Takashi Suzuki. Result and evaluation of autonomous rendezvous docking experiment of ETS-VII. In *Guidance, Navigation, and Control Conference and Exhibit*, Reston, Virigina, Aug 1999. American Institute of Aeronautics and Astronautics.

[4] NASA. Overview of the DART Mishap Investigation Results. Technical report, NASA, 2005.

[5] David A Whelan, E Allen Adler, Samuel B Wilson III, and Gordon M Roesler Jr. Darpa orbital express program: effecting a revolution in space-based systems. *International Symposium on Optical Science and Technology*, 4136:48–56, 2000.

[6] Manny R. Leinz, Chih-Tsai Chen, Michael W. Beaven, Thomas P. Weismuller, David L. Caballero, William B. Gaumer, Peter W. Sabasteanski, Peter A. Scott, and Mark A. Lundgren. Orbital Express Autonomous Rendezvous and Capture Sensor System (ARCSS) flight test results. *Proceedings of SPIE*, 6958:69580A–69580A–13, 2008.

[7] Simone D'Amico, Per Bodin, Michel Delpech, and Ron Noteborn. PRISMA. In Marco D'Errico, editor, *Distributed Space Missions for Earth System Monitoring*, chapter 21. Springer New York, New York, NY, 2013.

[8] Satomi Kawamoto, Takeshi Makida, Fumiki Sasaki, Yasushi Okawa, and Shin ichiro Nishida. Precise numerical simulations of electrodynamic tethers for an active debris removal system. *Acta Astronautica*, 59(1-5):139–148, 2006.

[9] Shin-Ichiro Nishida, Satomi Kawamoto, Yasushi Okawa, Fuyuto Terui, and Shoji Kitamura. Space debris removal system using a small satellite. *Acta Astronautica*, 65(1-2):95–102, 2009.

[10] J. C. Liou, N. L. Johnson, and N. M. Hill. Controlling the growth of future LEO debris populations with active debris removal. *Acta Astronautica*, 66(5-6):648–653, 2010.

[11] Albert B. Bosse, W. J. Barnds, Michael A. Brown, N. G. Creamer, Andy Feerst, Carl G. Henshaw, Alan S. Hope, Bernard E. Kelm, Patricia A. Klein, Frank Pipitone, Bertrand E. Plourde, and Brian P. Whalen. SUMO: spacecraft for the universal modification of orbits. In Peter Tchoryk, Jr. and Melissa Wright, editors, *Proceedings of Spacecraft Platforms and Infrastructure*, volume 5419, pages 36–46. SPIE, Aug 2004.

[12] Sabrina Eberle, Andreas Ohndorf, and Ralf Faller. On-Orbit Servicing Mission Operations at GSOC. *SpaceOps 2010 Conference Delivering on the Dream Hosted by NASA Marshall Space Flight Center and Organized by AIAA*, pages 1–10, 2010.

[13] Christopher Jewison, David Sternberg, Bryan McCarthy, David W Miller, and Alvar Saenz-otero. Definition and testing of an architectural tradespace for on-orbit assemblers and servicers. In *International Astronautical Congress (IAC)*. International Astronautical Federation, 2014.

[14] Brandon Karlow, Christopher Jewison, David Sternberg, Sherrie Hall, and Alessandro Golkar. Tradespace investigation of strategic design factors for large space telescopes. *Journal of Astronomical Telescopes, Instruments, and Systems*, 1(2):027003, Apr 2015.

[15] Brook Sullivan and David Akin. A survey of serviceable spacecraft failures. In *AIAA Space 2001 Conference and Exposition*, pages 1–8, Reston, Virigina, Aug 2001. American Institute of Aeronautics and Astronautics.

[16] Geoffrey Landis, Sheila Bailey, and Renee Tischler. Causes of Power-Related Satellite Failures. In *2006 IEEE 4th World Conference on Photovoltaic Energy Conference*, volume 2, pages 1943–1945. IEEE, 2006.

[17] J L Goodman. Lessons Learned From Seven Space Shuttle Missions. Technical report, NASA, Jan 2007.

[18] Cornelius J Dennehy and J Russell Carpenter. A Summary of the Rendezvous , Proximity Operations , Docking , and Undocking (RPODU) Lessons Learned from the Defense Advanced Research Project Agency (DARPA) Orbital Express (OE) Demonstration System Mission. Technical report, NASA, 2011.

[19] R.O. Ambrose, B. D. Allen, R. Mueller, I. Nesnas, T. Fong, F. Chandler, L. Matthies, H. Hinkel, A. Sylvester, M. Redlinger, B. Wilcox, and R. Stillwater. 2015 NASA Technology Roadmaps TA 4: Robotics and Autonomous Systems. Technical Report July, NASA, 2015.

[20] Douglas Zimpfer, Peter Kachmar, and Seamus Tuohy. Autonomous Rendezvous, Capture and In-Space Assembly: Past, Present and Future. In *1st Space Exploration Conference: Continuing the Voyage of Discovery*, pages 1–12, Orlando, Florida, Jan 2005. American Institute of Aeronautics and Astronautics.

[21] David C. Woffinden and David K. Geller. The road to autonomous orbital rendezvous. *Advances in the Astronautical Sciences*, 129 PART 3(4):2395–2415, 2008.

[22] Yazhong Luo, Jin Zhang, and Guojin Tang. Survey of orbital dynamics and control of space rendezvous. *Chinese Journal of Aeronautics*, 27(1):1–11, 2014.

[23] James R. Wertz and Robert Bell. Autonomous Rendezvous and Docking Technologies: Status and Prospects. In *SPIE AeroSense Symposium*, volume 5088, pages 20–30, Aug 2003.

[24] Angel Flores-Abad, Ou Ma, Khanh Pham, and Steve Ulrich. A review of space robotics technologies for on-orbit servicing. *Progress in Aerospace Sciences*, 68:1–26, Jul 2014.

[25] John T. Betts. Survey of Numerical Methods for Trajectory Optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998.

[26] Daniel P Scharf, Fred Y Hadaegh, and Bryan H Kang. A Survey of Spacecraft Formation Flying Guidance. *Proceedings of the International Symposium Formation Flying*, 2002.

[27] Daniel P Scharf, Fred Y Hadaegh, and Scott R Ploen. A survey of spacecraft formation flying guidance and control. Part II: control. In *Proceeding of the 2004 American Control Conference*, volume 4, pages 1–13, 2004.

[28] Hari Hablani, Myron Tapper, and David Dana-Bashian. Guidance algorithms for autonomous rendezvous of spacecraft with a target vehicle in circular orbit. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1–5, Reston, Virigina, Aug 2001. American Institute of Aeronautics and Astronautics.

[29] Simon Nolet, Edmund Kong, and David W Miller. Autonomous docking algorithm development and experimentation using the SPHERES testbed. *Proceedings of SPIE, vol 5419, Spacecraft Platforms and Infrastructure*, 5419:1–15, 2004.

[30] Simon Nolet. *Development of a guidance, navigation and control architecture and validation process enabling autonomous docking to a tumbling satellite.* PhD thesis, Massachusetts Institute of Technology, 2007.

[31] Hari B. Hablani. Autonomous Inertial Relative Navigation with Sight-Line-Stabilized Sensors for Spacecraft Rendezvous. *Journal of Guidance, Control, and Dynamics*, 32(1):172–183, 2009.

[32] Jose Ruiz and Jeremy Hart. A Comparison Between Orion Automated and Space Shuttle Rendezvous Techniques. In *AIAA Guidance, Navigation, and Control Conference*, pages 1–10, Reston, Virigina, Aug 2010. American Institute of Aeronautics and Astronautics.

[33] Renato Zanetti. Optimal Glideslope Guidance for Spacecraft Rendezvous. *Journal of Guidance, Control, and Dynamics*, 34(5):1593–1597, 2011.

[34] Jin Zhang, Guo-jin Tang, Ya-Zhong Luo, and Hai-yang Li. Orbital rendezvous mission planning using mixed integer nonlinear programming. *Acta Astronautica*, 68(7-8):1070–1078, 2011.

[35] Gilberto Arantes and Luiz S. Martins-Filho. Guidance and Control of Position and Attitude for Rendezvous and Dock/Berthing with a Noncooperative/Target Spacecraft. *Mathematical Problems in Engineering*, 2014:1–8, 2014.

[36] Yassine Ariba, Denis Arzelier, Laura Sofia Urbina, and Christophe Louembet. V-bar and R-bar Glideslope Guidance Algorithms for Fixed-Time Rendezvous: A Linear Programming Approach. *IFAC-PapersOnLine*, 49(17):385–390, 2016.

[37] Christian Tournes and Yuri Shtessel. Automatic Docking Using Optimal Control and Second Order Sliding Mode Control. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, pages 3855–3860, Reston, Virigina, Aug 2007. American Institute of Aeronautics and Astronautics.

[38] Rajarshi Ghosh Dastidar. On the Advantages and Limitations of Sliding Mode Control for Spacecraft. In *AIAA SPACE 2010 Conference and Exposition*, Reston, Virigina, Aug 2010. American Institute of Aeronautics and Astronautics.

[39] Wei Lu, Yunhai Geng, Xueqin Chen, and Fan Zhang. Relative position and attitude coupled control for autonomous docking with a tumbling target. *International Journal of Control and Automation*, 4(4):1–22, 2011.

[40] Chutiphon Pukdeboon. Finite-time second-order sliding mode controllers for spacecraft attitude tracking. *Mathematical Problems in Engineering*, 2013(100):4965–4979, 2013.

[41] Binglong Chen and Yunhai Geng. Simulation of rendezvous and docking between service spacecraft and non-cooperative target with MATLAB-Simulink and STK. In *Proceeding of the 11th World Congress on Intelligent Control and Automation*, pages 5727–5732. IEEE, Jun 2014.

[42] Brian S Keller, Sivakumara S K Tadikonda, and Jalal Mapar. Autonomous Acquisition, Rendezvous, and Docking Using a Video Guidance Sensor: Experimental Testbed Results. *AIAA Guidance Navigation and Control Conference and Exhibit*, pages 1–10, Aug 2002.

[43] Qian Wang and Guang-Ren Duan. Robust Global Stabilization of Spacecraft Rendezvous System via Gain Scheduling. *International Journal of Automation and Computing*, 11(4):426–433, 2014.

[44] Ismael Lopez and Colin R. McInnes. Autonomous rendezvous using artificial potential function guidance. *Journal of Guidance, Control, and Dynamics*, 18(2):237–241, 1995.

[45] Alexander B Roger and Colin R McInnes. Safety constrained free-flyer path planning at the international space station. *Journal of Guidance, Control, and Dynamics*, 23(6):971–979, 2000.

[46] E St.John-Olcayto, C R McInnes, and F Ankersen. Safety-critical autonomous spacecraft proximity operations via potential function guidance. *2007 AIAA InfoTech at Aerospace Conference*, 2(May):1893–1902, 2007.

[47] Shawn B McCamish, Marcello Romano, Simon Nolet, Christine M Edwards, and David W Miller. Flight Testing of Multiple-Spacecraft Control on SPHERES During Close-Proximity Operations. *Journal of Spacecraft and Rockets*, 46(6):1202–1213, 2009.

[48] Dawei Zhang, Shenmin Song, and Run Pei. Safe Guidance for Autonomous Rendezvous and Docking with a Non-Cooperative Target. In *AIAA Guidance, Navigation, and Control Conference*, pages 1–19, Reston, Virigina, Aug 2010. American Institute of Aeronautics and Astronautics.

[49] Josue David Munoz. *Rapid Path-planning Algorithms for Autonomous Proximity Operations of Satellites*. PhD thesis, University of Florida, 2011.

[50] Richard Zappulla, Hyeongjun Park, Josep Virgili Llop, and Marcello Romano. Experiments on autonomous spacecraft rendezvous and docking using an adaptive artificial potential field approach. In *26th AAS/AIAA Space Flight Mechanics Meeting*, Napa, California, Feb 2016.

[51] S Li, R Mehra, Robert Smith, and Randal Beard. Multi-spacecraft trajectory optimization and control using genetic algorithm techniques. In *Aerospace Conference Proceedings*, volume 7. IEEE, 2000.

[52] Young Ha Kim and David B. Spencer. Optimal Spacecraft Rendezvous Using Genetic Algorithms. *Journal of Spacecraft and Rockets*, 39(6):859–865, 2002.

[53] Ya-Zhong Luo, Yong-Jun Lei, and Guo-Jin Tang. Optimal Multi-Objective Nonlinear Impulsive Rendezvous. *Journal of Guidance, Control, and Dynamics*, 30(4):994–1002, 2007.

[54] Ya-Zhong Luo, Hai-Yang Li, and Guo-Jin Tang. Hybrid Approach to Optimize a Rendezvous Phasing Strategy. *Journal of Guidance, Control, and Dynamics*, 30(1):185–191, 2007.

[55] Yechiel Crispin and Dongeun Seo. Rendezvous Between Two Active Spacecraft with Continuous Low Thrust. In *Advances in Spacecraft Technologies*. InTech, 2011.

[56] Grégory Saive and Massimiliano Vasile. Probabilistic Optimisation applied to Spacecraft Rendezvous on Keplerian Orbits. Technical report, Cranfield University, 2004.

[57] Michael A. Paluszek and Stephanie J. Thomas. Generalized 3D spacecraft proximity path planning using A*. *Collection of Technical Papers - InfoTech at Aerospace: Advancing Contemporary Aerospace Technologies and Their Integration*, 2(September):1230–1239, 2005.

[58] Per Bodin, Robin Larsson, Fredrik Nilsson, Camille Chasset, Ron Noteborn, and Matti Nylund. PRISMA: An In-Orbit Test Bed for Guidance, Navigation, and Control Experiments. *Journal of Spacecraft and Rockets*, 46(3):615–623, 2009.

[59] Griffin Francis, Emmanuel Collins, Oscar Chuy, and Aneesh Sharma. Sampling-Based Trajectory Generation for Autonomous Spacecraft Rendezvous and Docking. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, Reston, Virginia, Aug 2013. American Institute of Aeronautics and Astronautics.

[60] Jeff Phillips, Lydia Kavraki, and Nazareth Bedrossian. Spacecraft Rendezvous and Docking with Real-Time, Randomized Optimization. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Reston, Virigina, Aug 2003. American Institute of Aeronautics and Astronautics.

[61] Joseph A Starek, Edward Schmerling, Gabriel D Maher, Mathematical Engineering, Marco Pavone, and Brent W Barbee. Real-Time, Propellant-Optimized Spacecraft Motion Planning under Clohessy-Wiltshire-Hill Dynamics. In *Proceedings of the 2016 IEEE Aerospace Conference*. IEEE, 2016.

[62] Joseph A. Starek, Edward Schmerling, Gabriel D. Maher, Brent W. Barbee, and Marco Pavone. Fast, Safe, Propellant-Efficient Spacecraft Motion Planning Under ClohessyWiltshireHill Dynamics. *Journal of Guidance, Control, and Dynamics*, 40(2):418–438, Feb 2017.

[63] Morgan Baldwin, Avishai Weiss, Ilya Kolmanovsky, and R. Scott Erwin. Spacecraft debris avoidance using positively invariant constraint admissible sets. *Advances in the Astronautical Sciences*, 143:2209–2222, 2012.

[64] Avishai Weiss, Christopher Petersen, Morgan Baldwin, R. Scott Erwin, and Ilya Kolmanovsky. Safe Positively Invariant Sets for Spacecraft Obstacle Avoidance. *Journal of Guidance, Control, and Dynamics*, 38(4):720–732, 2015.

[65] Claus Danielson, Avishai Weiss, Karl Berntorp, and Stefano Di Cairano. Path Planning using Positive Invariant Sets. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 5986–5991, 2016.

[66] R. Bevilacqua and M. Romano. Quasi-optimal control for path constrained relative spacecraft maneuvers based on dynamic programming. *Nonlinear Dynamics and Systems Theory*, 8(2):137–150, 2008.

[67] Howie Choset and David Kortenkamp. Path Planning and Control for Free-Flying Inspection Robot in Space. *Journal of Aerospace Engineering*, 12(2):74–81, Apr 1999.

[68] L. Breger, J. How, and A. Richards. Model predictive control of spacecraft formations with sensing noise. *Proceedings of the American Control Conference*, 4:2385, 2005.

[69] Arthur Richards and Jonathan P. How. Model Predictive Control of Vehicle Maneuvers with Guaranteed Completion Time and Robost Feasibility. *Information and Software Technology*, 51(4):769–784, 2009.

[70] Hyeongjun Park, Stefano Di Cairano, and Ilya Kolmanovsky. Model Predictive Control of spacecraft docking with a non-rotating platform. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 18(PART 1):8485–8490, 2011.

[71] Hyeongjun Park, Stefano Di Cairano, and Ilya Kolmanovsky. Model Predictive Control for Spacecraft Rendezvous and Docking with a Rotating / Tumbling Platform and for Debris Avoidance. *American Control Conference*, pages 1922–1927, 2011.

[72] Rafael Vazquez, Francisco Gavilan, and Eduardo F. Camacho. Trajectory Planning for Spacecraft Rendezvous with On / Off Thrusters. *18th IFAC World Congress*, pages 8473–8478, 2011.

[73] S. Di Cairano, H. Park, and I. Kolmanovsky. Model Predictive Control approach for guidance of spacecraft rendezvous and proximity maneuvering. *International Journal of Robust and Nonlinear Control*, 22(12):1398–1427, 2012.

[74] 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.

[75] Edward N. Hartley, Paul A. Trodden, Arthur G. Richards, and Jan M. Maciejowski. Model predictive control system design and implementation for spacecraft rendezvous. *Control Engineering Practice*, 20(7):695–713, 2012.

[76] Avishai Weiss, I. Kolmanovsky, Morgan Baldwin, and R Scott Erwin. Model Predictive Control of three dimensional spacecraft relative motion. In *2012 American Control Conference (ACC)*, pages 173–178. IEEE, Jun 2012.

[77] Morgan Baldwin, Richard S Erwin, and Ilya Kolmanovsky. Robust Controller for Constrained Relative Motion Maneuvering with Disturbance Rejection. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, pages 1–14, Reston, Virginia, Aug 2013. American Institute of Aeronautics and Astronautics.

[78] P Li, X Yue, H Dai, and X Chi. 6Dof Terminal Guidance for Autonomous Spacecraft Capture Free Floating Objects Using State Dependent Model Predictive. *5th International Conference on Spacecraft Formation Flying Missions and Technologies*, 2013.

[79] M. Saponara, V. Barrena, a. Bemporad, E. N. Hartley, J. Maciejowski, a. Richards, a. Tramutola, and P. Trodden. Model predictive control application to spacecraft rendezvous in mars sample return scenario. *Progress in Flight Dynamics, Guidance, Navigation, Control, Fault Detection, and Avionics*, 6:137–158, 2013.

[80] Georgia Deaconu, Christophe Louembet, and Alain Théron. Minimizing the Effects of Navigation Uncertainties on the Spacecraft Rendezvous Precision. *Journal of Guidance, Control, and Dynamics*, 37(2):695–700, 2014.

[81] Piotr A Felisiak, Krzysztof Sibilski, Wieslaw Wroblewski, and Jurek Z Sasiadek. Spacecraft Rendezvous in Elliptical Orbit using Nonlinear Model Predictive Control. In *AIAA Guidance, Navigation, and Control Conference*, pages 1–12, Reston, Virginia, Jan 2014. American Institute of Aeronautics and Astronautics.

[82] Mirko Leomanni, Eric Rogers, and Stephen B. Gabriel. Explicit Model Predictive Control Approach for Low-Thrust Spacecraft Proximity Operations. *Journal of Guidance, Control, and Dynamics*, 37(6):1780–1790, 2014.

[83] Christopher Petersen, Andris Jaunzemis, Morgan Baldwin, Marcus Holzinger, and Ilya Kolmanovsky. Model predictive control and Extended Command Governor for improving robustness of relative motion guidance and control. *Advances in the Astronautical Sciences*, 152:701–718, 2014.

[84] Avishai Weiss, Morgan Baldwin, Richard Scott Erwin, and Ilya Kolmanovsky. Model Predictive Control for Spacecraft Rendezvous and Docking: Strategies for Handling Constraints and Case Studies. *IEEE Transactions on Control Systems Technology*, pages 1–23, 2015.

[85] Costantinos Zagaris, Morgan Baldwin, Christopher Jewison, and Christopher Petersen. Survey of Spacecraft Rendezvous and Proximity Guidance Algorithms for

on-Board Implementation. In *25th AAS/AIAA Spaceflight Mechanics Meeting*, pages 1–20, Williamsbug, VA, 2015.

[86] Robin Larsson, Sten Berge, Per Bodin, and Ulf Jönsson. Fuel efficient relative orbit control strategies for formation flying and rendezvous within PRISMA. *Advances in the Astronautical Sciences*, 125:25–40, 2006.

[87] Huijun Gao, Xuebo Yang, Peng Shi, Huijun Gao, Xuebo Yang, and Peng Shi. Multi-objective Robust H-Infinity Control of Spacecraft. *IEEE Transactions on Control Systems Technology*, 17(4):794–802, Jul 2009.

[88] Jia Jie, Yao Yu, and Ma Kemao. Continuous optimal terminal proximity guidance algorithm for autonomous rendezvous and docking. *Information Technology Journal*, 12(5):1011–1017, 2013.

[89] Gregory Lantoine and Richard Epenoy. Quadratically Constrained Linear-Quadratic Regulator Approach for Finite-Thrust Orbital Rendezvous. *Journal of Guidance, Control, and Dynamics*, 35(6):1787–1797, 2012.

[90] Richard Epenoy. Fuel-optimal trajectories for continuous-thrust orbital rendezvous with collision avoidance constraint. *Advances in the Astronautical Sciences*, 140(2):341–360, 2011.

[91] M. Ciarcià and M. Romano. Suboptimal Guidance for Orbital Proximity Maneuver with Path Constraints Capability. In *AIAA Guidance, Navigation, and Control Conference*, pages 1–24, Reston, Virigina, Aug 2012. American Institute of Aeronautics and Astronautics.

[92] Jacopo Ventura, Marco Ciarcià, Marcello Romano, and Ulrich Walter. Fast and Near-Optimal Guidance for Docking to Uncontrolled Spacecraft. *Journal of Guidance, Control, and Dynamics*, pages 1–17, 2016.

[93] Josep Virgili-Llop, Costantinos Zagaris, Hyeongjun Park, Richard I I Zappulla, and Marcello Romano. Experimental Evaluation of Model Predictive Control and Inverse Dynamics Control for Spacecraft Proximity and Docking Maneuvers. In *6th Internation Conference on Astrodynamics Tools and Techniques (ICATT)*, Darmstadt, Germany, 2016.

[94] Ping Lu and Xinfu Liu. Autonomous Trajectory Planning for Rendezvous and Proximity Operations by Conic Optimization. *Journal of Guidance, Control, and Dynamics*, 36(2):375–389, 2013.

[95] Xinfu Liu and Ping Lu. Solving Nonconvex Optimal Control Problems by Convex Optimization. *Journal of Guidance, Control, and Dynamics*, 37:750–765, 2014.

[96] Nick Martinson. Obstacle Avoidance Guidance and Control Algorithm for Spacecraft Maneuvers. In *AIAA Guidance, Navigation, and Control Conference*, Reston, Virigina, Aug 2009. American Institute of Aeronautics and Astronautics.

[97] Christopher Jewison, R. Scott Erwin, and Alvar Saenz-Otero. Model Predictive Control with ellipsoid obstacle constraints for spacecraft rendezvous. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, volume 48, pages 257–262. Elsevier Ltd., 2015.

[98] Mauro Pontani, Pradipto Ghosh, and Bruce a. Conway. Particle Swarm Optimization of Multiple-Burn Rendezvous Trajectories. *Journal of Guidance, Control, and Dynamics*, 35(4):1192–1207, 2012.

[99] Arthur Richards, Tom Schouwenaars, Jonathan P. How, and Eric Feron. Spacecraft Trajectory Planning with Avoidance Constraints Using Mixed-Integer Linear Programming. *Journal of Guidance, Control, and Dynamics*, 25(4):755–764, 2002.

[100] Georgia Deaconu, Christophe Louembet, and Alain Théron. Designing Continuously Constrained Spacecraft Relative Trajectories for Proximity Operations. *Journal of Guidance, Control, and Dynamics*, 38(7):1–10, 2014.

[101] Gregory E Chamitoff, Alvar Saenz-otero, Jacob G Katz, and Steve Ulrich. Admissible Subspace TRajectory Optimizer (ASTRO) for Autonomous Robot Operations on the Space Station. *AIAA Guidance, Navigation, and Control Conference*, pages 1–17, Jan 2014.

[102] Jian-feng Shi, Steve Ulrich, Andrew Allen, and Macdonald Dettwiler. Optimal Trajectory Guidance for Spacecraft Robotic Servicing. In *13th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, Noordwijk, the Netherlands, 2015.

[103] Benjamin J. Morrell, Gregory E. Chamitoff, Derek J. Kuether, Mauricio Coen, and Peter Gibbens. Integration of 3D SLAM, Rigid Body Landmarks and 3D Path Planning. In *AIAA SPACE 2016*, pages 1–20, Reston, Virginia, Sep 2016.

[104] Jian-Feng Shi, Steve Ulrich, Gregory E. Chamitoff, Benjamin J. Morrell, and Andrew Allen. Trajectory Optimization for Proximity Operations Around Tumbling Geometrical Constraints via Legendre Polynomials. In *AIAA/AAS Astrodynamics Specialist Conference*, pages 1–21, Reston, Virginia, Sep 2016.

[105] Roberto Lampariello. Motion Planning for the On-orbit Grasping of a Non-cooperative Target Satellite with Collision Avoidance. *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 1:636–643, 2010.

[106] Farhad Aghili. A prediction and motion-planning scheme for visually guided robotic capturing of free-floating tumbling objects with uncertain dynamics. *IEEE Transactions on Robotics*, 28(3):634–649, 2012.

[107] George Boyarko, Oleg Yakimenko, and Marcello Romano. Optimal Rendezvous Trajectories of a Controlled Spacecraft and a Tumbling Object. *Journal of Guidance, Control, and Dynamics*, 34(4):1239–1252, 2011.

[108] Zhanhua Ma, Ou Ma, and Banavara N. Shashikanth. Optimal Approach to and Alignment with a Rotating Rigid Body for Capture. *The Journal of the Astronautical Sciences*, 55(4):407–419, 2012.

[109] J. Michael, K. Chudej, M. Gerdts, and J. Pannek. Optimal rendezvous path planning to an uncontrolled tumbling target. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 19(PART 1):347–352, 2013.

[110] Amer Fejzic. *Development of Control and Autonomy Algorithms for Docking to Complex Tumbling Satellites*. PhD thesis, Massachusetts Institute of Technology, 2008.

[111] Norman Fitz-Coy and Ming-Cheng Liu. A modified proportional navigation scheme for rendezvous and docking with tumbling targets: The planar case. In *NASA Goddard Space Flight Center, Flight Mechanics/Estimation Theory Symposium*, pages 243–252, 1995.

[112] Hyeongjun Park, Richard Zappulla, Costantinos Zagaris, Josep Virgili-Llop, and Marcello Romano. Nonlinear model predictive control for spacecraft rendezvous and docking with a rotating target. In *27th AAS/AIAA Spaceflight Mechanics Meeting*, San Antonio, TX, 2017.

[113] Sivakumar Tadikonda, Brian Keller, and Jalal Mapar. Docked Two-Vehicle Dynamics, and Control by One-Vehicle: Experimental Results. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1–11, 2002.

[114] Alexander Parlos and John Sunkel. Adaptive attitude stability and control for Space Station/Orbiter berthing operations. In *Guidance, Navigation and Control Conference*, volume 1, Reston, Virigina, Aug 1992. American Institute of Aeronautics and Astronautics.

[115] Edward LeMaster, David Schaechter, and Connie Carrington. Experimental Demonstration of Technologies for Autonomous On-Orbit Robotic Assembly. In *Space 2006*, pages 1–14, Reston, Virigina, Sep 2006. American Institute of Aeronautics and Astronautics.

[116] Swati Mohan and David W Miller. *Quantitative Selection and Design of Model Generation Architectures for On-Orbit Autonomous Assembly*. PhD thesis, Massachusetts Institute of Technology, 2010.

[117] Matthew C. Sorgenfrei, Laura L. Jones, Sanjay S. Joshi, and Mason a. Peck. Testbed Validation of Location-Scheduled Control of a Reconfigurable Flux-Pinned Spacecraft Formation. *Journal of Spacecraft and Rockets*, 50(6):1235–1247, 2013.

[118] Panfeng Huang, Ming Wang, Zhongjie Meng, Fan Zhang, and Zhengxiong Liu. Attitude takeover control for post-capture of target spacecraft using space robot. *Aerospace Science and Technology*, 51:171–180, 2016.

[119] Christopher M. Jewison, Bryan McCarthy, David C. Sternberg, Daniel Strawser, and Cheng Fang. Resource Aggregated Reconfigurable Control and Risk-Allocative Path Planning for On-orbit Servicing and Assembly of Satellites. In *AIAA Guidance, Navigation, and Control Conference*, pages 1–20, Reston, Virginia, Jan 2014. American Institute of Aeronautics and Astronautics.

[120] Christopher Jewison. *Reconfigurable Thruster Selection Algorithms for Aggregative Spacecraft Systems*. Masters thesis, Massachusetts Institute of Technology, 2014.

[121] Christopher Jewison and R Scott Erwin. A spacecraft benchmark problem for hybrid control and estimation. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 3300–3305. IEEE, Dec 2016.

[122] Hasan A Poonawala and Ufuk Topcu. Filter-Based Stochastic Abstractions for Constrained Planning with Limited Sensing. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 3319–3324, 2016.

[123] Samira Farahani, Ivan Papusha, Catharine Mcghan, and Richard M Murray. Constrained Autonomous Satellite Docking via Differential Flatness and Model Predictive Control. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 3306–3311, Las Vegas, USA, 2016. IEEE.

[124] Meeko Oishi, Baisravan Homchaudhuri, Matt Shubert, Morgan Baldwin, and R Scott Erwin. Computing Reach-Avoid Sets for Space Vehicle Docking under Impulsive Thrust. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 3312–3318, 2016.

[125] Kewen Zhang and Michael A Demetriou. Synthesis of Adaptive Controllers for Spacecraft Rendezvous Maneuvers using Nonlinear Models of Relative Motion. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 395–400, 2016.

[126] Bharani P Malladi, Ricardo G Sanfelice, Eric Butcher, and Jingwei Wang. Robust Hybrid Supervisory Control for Rendezvous and Docking of a Spacecraft. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 3325–3330, Las Vegas, USA, 2016. IEEE.

[127] Masahiro Ono, Brian C. Williams, and Lars Blackmore. Probabilistic planning for continuous dynamic systems under bounded risk. *Journal of Artificial Intelligence Research*, 46:511–577, 2013.

[128] Angel Flores-Abad, Zheng Wei, Ou Ma, and Khanh Pham. Optimal Control of Space Robots for Capturing a Tumbling Object with Uncertainties. *Journal of Guidance, Control, and Dynamics*, 37(6):2014–2017, 2014.

[129] Simon Grocott. *Comparison of Control Techniques for Robust Performance on Uncertain Structural Systems*. PhD thesis, Massachusetts Institute of Technology, 1994.

[130] Puneet Singla, Kamesh Subbarao, and John L. Junkins. Adaptive Output Feedback Control for Spacecraft Rendezvous and Docking Under Measurement Uncertainty. *Journal of Guidance, Control, and Dynamics*, 29(4):892–902, Jul 2006.

[131] Antonio Teran and William Sanchez. On-board Parameter Learning Using a Model Reference Adaptive Position and Attitude Controller. In *Guidance Navigation and Control, 2017 Proceedings of the IEEE Aerospace Conference*, Big Sky, MT, 2017. IEEE.

[132] Steve Ulrich, Dustin L. Hayhurst, Alvar Saenz Otero, David Miller, and Itzhak Barkana. Simple Adaptive Control for Spacecraft Proximity Operations. In *AIAA Guidance, Navigation, and Control Conference*, pages 2219–2224, Reston, Virginia, Jan 2014. American Institute of Aeronautics and Astronautics.

[133] Steve Ulrich, Alvar Saenz-Otero, and Itzhak Barkana. Passivity-Based Adaptive Control of Robotic Spacecraft for Proximity Operations Under Uncertainties. *Journal of Guidance, Control, and Dynamics*, 39(6):1444–1453, 2016.

[134] Steve Ulrich and Kirk Hovell. Iterative Learning Control of Spacecraft Proximity Operations Based on Confidence Level. In *AIAA Guidance, Navigation, and Control Conference*, Reston, Virginia, Jan 2017.

[135] Martine Ganet-Schoeller, Jerome Bourdon, and Gregory Gelly. Non-Linear and Robust Stability Analysis for ATV Rendezvous Control. In *AIAA Guidance, Navigation, and Control Conference*, pages 1–18, Reston, Virigina, Aug 2009. American Institute of Aeronautics and Astronautics.

[136] Edward Wilson, Chris Lages, and Robert Mah. On-line gyro-based, mass-property identification for thruster-controlled spacecraft using recursive least squares. In *The 2002 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS-2002.*, volume 2, pages 334–337. IEEE, 2002.

[137] Jillian James. Adaptive control for post-dock maneuvers with an unknown semi-cooperative object. In *2016 IEEE Aerospace Conference*, pages 1–10. IEEE, Mar 2016.

[138] Divya Thakur, Sukumar Srikant, and Maruthi R. Akella. Adaptive Attitude-Tracking Control of Spacecraft with Uncertain Time-Varying Inertia Parameters. *Journal of Guidance, Control, and Dynamics*, 38(1):41–52, Jan 2015.

[139] Joshuah A. Hess, Eric D Swenson, Fred Leve, Jonathan Black, and Gary M. Goff. Adaptive Estimation of Nonlinear Spacecraft Attitude Dynamics with Time-Varying Moments of Inertia Using On-Board Sensors. In *AIAA Guidance, Navigation, and Control Conference*, pages 1–18, Reston, Virginia, Jan 2016. American Institute of Aeronautics and Astronautics.

[140] David C. Woffinden and David K. Geller. Optimal orbital rendezvous maneuvering for angles-only navigation. *Advances in the Astronautical Sciences*, 133(4):385–407, 2009.

[141] Simone D. Amico, J.-S. Ardaens, G. Gaias, H. Benninghoff, B. Schlepp, and J. L. Jørgensen. Noncooperative Rendezvous Using Angles-Only Optical Navigation: System Design and Flight Results. *Journal of Guidance, Control, and Dynamics*, 36(6):1576–1595, 2013.

[142] Jonathan Grzymisch and Walter Fichter. Optimal Rendezvous Guidance with Enhanced Bearings-Only Observability. *Journal of Guidance, Control, and Dynamics*, 38(6):1131–1140, 2015.

[143] Christopher Pong, Alvar Saenz-otero, and David W Miller. Autonomous thruster failure recovery on underactuated spacecraft using model predictive control. In *Proceedings of the 34th Annual AAS Rocky Mountain Section Guidance and Control Conference*. AAS, 2011.

[144] Johannes M. Hacker, James L. Goddard, and Peter C. Lai. Globalstar second generation hybrid attitude control on-orbit experience. *Advances in the Astronautical Sciences*, 152:3515–3531, 2014.

[145] Danyal Bustan, S K Hosseini Sani, and Naser Pariz. Adaptive Fault-Tolerant Spacecraft Attitude Control Design With Transient Response Control. *IEEE/ASME Transactions on Mechatronics*, 19(4):1404–1411, Aug 2014.

[146] Louis S. Breger and Jonathan P. How. Safe Trajectories for Autonomous Rendezvous of Spacecraft. *Journal of Guidance, Control, and Dynamics*, 31(5):1478–1489, 2008.

[147] Piero Miotto. Designing and Validating Proximity Operations Rendezvous and Approach Trajectories for the Cygnus Mission. In *AIAA Guidance, Navigation, and Control Conference*, pages 1–18, Reston, Virigina, Aug 2010. American Institute of Aeronautics and Astronautics.

[148] C Louembet, D Arzelier, and G Deaconu. Robust Rendezvous Planning Under Maneuver Execution Errors. *Journal of Guidance Control and Dynamics*, 38(1):76–93, 2015.

[149] S. He, W. Wang, D. Lin, and H. Lei. Robust output feedback control design for autonomous spacecraft rendezvous with actuator faults. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 230(6):578–587, 2016.

[150] P.S. Maybeck and R.D. Stevens. Reconfigurable flight control via multiple model adaptive control methods. *IEEE Transactions on Aerospace and Electronic Systems*, 27(3):470–480, May 1991.

[151] J.D. Boskovic, Sai-Ming Li, and R.K. Mehra. Reconfigurable flight control design using multiple switching controllers and online estimation of damage-related parameters. In *Proceedings of the 2000. IEEE International Conference on Control Applications. Conference Proceedings (Cat. No.00CH37162)*, pages 479–484. IEEE, 2000.

[152] John Davidson, Frederick Lallman, and W. Bundick. Integrated reconfigurable control allocation. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1–11, Reston, Virigina, Aug 2001. American Institute of Aeronautics and Astronautics.

[153] A Wander and R Förstner. Innovative fault detection, isolation and recovery strategies on-board spacecraft: State of the art and research challenges. In *Deutscher Luft- und Raumfahrtkongress 2012*, pages 1–9, 2012.

[154] Dimitry Gorinevsky, Gabriel M. Hoffmann, Marina Shmakova, Robert W. Mah, Scott Cryan, and Jennifer D. Mitchell. Fault tolerance of relative navigation sensing in Docking approach of spacecraft. *IEEE Aerospace Conference Proceedings*, 2008.

[155] Steve Chien, B Engelhardt, R Knight, G Rabideau, R Sherwood, Eric a Hansen, A Ortiviz, C Wilklow, and S Wichman. Onboard Autonomy on the Three Corner Sat Mission. In *International Symposium on Artificial Intelligence, Robotics and Automation for Space*, 2001.

[156] S. Knight, G. Rabideau, S. Chien, B. Engelhardt, and R. Sherwood. Casper: space exploration through continuous planning. *IEEE Intelligent Systems*, 16(5):70–75, Sep 2001.

[157] Steve Chien, Paul Zetocha, Ross Wainwright, Pete Klupar, Jim Van Gaasbeck, Pat Cappelaere, Dean Oswald, Rob Sherwood, Gregg Rabideau, Rebecca Castano, Ashley Davies, Michael Burl, Russell Knight, Tim Stough, and Joe Roden. The Techsat-21 autonomous space science agent. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems part 2 - AAMAS '02*, page 570, New York, New York, USA, 2002. ACM Press.

[158] Sylvain Damiani, Gérard Verfaillie, and Marie-Claire Charmeau. An earth watching satellite constellation. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems - AAMAS '05*, page 455, New York, New York, USA, 2005. ACM Press.

[159] Pei Wang, Gerhard Reinelt, Peng Gao, and Yuejin Tan. A model, a heuristic and a decision support system to solve the scheduling problem of an earth observing satellite constellation. *Computers & Industrial Engineering*, 61(2):322–335, Sep 2011.

[160] W A Fisher and Ella Herz. A Flexible Architecture for Creating Scheduling Algorithms as used in STK Scheduler. Technical report, Orbit Logic Incorporated, 2013.

[161] Philippe Monmousseau. *Scheduling of a Constellation of Satellites: Improving a Simulated Annealing Model by Creating a Mixed-Integer Linear Model.* PhD thesis, KTH, School of Engineering Sciences, 2015.

[162] B Smith, W Millar, J Dunphy, Yu-Wen Tung, P Nayak, E Gamble, and M Clark. Validation and verification of the remote agent for spacecraft autonomy. *Aerospace Conference, 1999. Proceedings. 1999 IEEE*, 1:449–468 vol.1, 1999.

[163] Alexei Iliasov, Elena Troubitsyna, Linas Laibinis, Alexander Romanovsky, Kimmo Varpaaniemi, Pauli Vaisanen, Dubravka Ilic, and Timo Latvala. Verifying Mode Consistency for On-Board Satellite Software. *Computer Safety, Reliability, and Security*, 6351(971):126–141, 2010.

[164] David Sternberg, Mark Chodas, Chris Jewison, Michael Jones, and Olivier De Weck. Multidisciplinary system design optimization of on orbit satellite assembly architectures. In *2015 IEEE Aerospace Conference*, pages 1–14. IEEE, Mar 2015.

[165] I.Y. Kim and O.L. de Weck. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and Multidisciplinary Optimization*, 29(2):149–158, Feb 2005.

[166] N Srinivas and Kalyanmoy Deb. Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, Sep 1994.

[167] Scott Uebelhart. *Non-Deterministic Design and Analysis of Parameterized Optical Structures during Conceptual Design.* PhD thesis, Massachusetts Institute of Technology, 2006.

[168] M.J. Rentmeesters, W.K. Tsai, and Kwei-Jay Lin. A theory of lexicographic multi-criteria optimization. *Proceedings of ICECCS '96: 2nd IEEE International Conference on Engineering of Complex Computer Systems (held jointly with 6th CSESAW and 4th IEEE RTAW)*, pages 76–79, 1996.

[169] A. V. Zykina. A lexicographic optimization algorithm. *Automation and Remote Control*, 65(3):363–368, 2004.

[170] Brent W. Barbee, Salvatore Alfano, Elfego Pinon, Kenn Gold, and David Gaylor. Design of spacecraft missions to remove multiple orbital debris objects. *Advances in the Astronautical Sciences*, 144:93–110, 2012.

[171] Brent William Barbee, J Russell Carpenter, Scott Heatwole, F Landis Markley, Michael Moreau, Bo J Naasz, and John Van Eepoel. A Guidance and Navigation Strategy for Rendezvous and Proximity Operations with a Noncooperative Spacecraft in Geosynchronous Orbit. *The Journal of the Astronautical Sciences*, 58(3):389–408, Jul 2011.

[172] C Guettier and J Poncet. Constraint Model-based Planning and Scheduling with Multiple Resources and Complex Collaboration Schema. *AIPS*, pages 284–293, 2002.

[173] I. Michael Ross and Christopher N. D'Souza. Hybrid Optimal Control Framework for Mission Planning. *Journal of Guidance, Control, and Dynamics*, 28(4):686–697, Jul 2005.

[174] Minh Hoang-Tuan Nguyen and Kok Kiong Tan. From Parametric Model-based Optimization to robust PID Gain Scheduling. *Elsevier*, May 2013.

[175] M. J. Mahmoodabadi, A. Bagheri, N. Nariman-Zadeh, A. Jamali, and R. Abedzadeh Maafi. Pareto design of decoupled sliding-mode controllers for nonlinear systems based on a multiobjective genetic algorithm. *Journal of Applied Mathematics*, 2012, 2012.

[176] E Frazzoli, M. Dahleh, E Feron, and R. Kornfeld. A randomized attitude slew planning algorithm for autonomous spacecraft. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1–8, Reston, Virigina, Aug 2001. American Institute of Aeronautics and Astronautics.

[177] Steve Chien, Russell Knight, Andre Stechert, Rob Sherwood, and Gregg Rabideau. Using iterative repair to improve the responsiveness of planning and scheduling. *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling*, pages 300–307, 2000.

[178] Kanna RaJan, Douglas Bernard, Gregory Dorais, Edward Gamble, Bob Kanefsky, James Kurien, William Millar, Nicola Muscettola, Pandurang Nayak, Nicolas Rouquette, Benjamin Smith, William Taylor, and Yu-wen Tung. Remote Agent: An Autonomous Control System for the New Millennium. *ECAI*, 14, 2000.

[179] Kamesh Subbarao and Atilla Dogan. Intelligent Integrated Sensor , Control and Maneuver Mapping Architecture For Autonomous Vehicle Rendezvous and Docking. *AIAA 3rd "Unmanned unlimited" Technical Conference, Workshop and Exhibit*, pages 1–11, 2004.

[180] Catharine L R McGhan, Richard M. Murray, Romain Serra, Michel D. Ingham, Masahiro Ono, Tara Estlin, and Brian C. Williams. A risk-aware architecture for resilient spacecraft operations. In *2015 IEEE Aerospace Conference*, volume 2015-June, pages 1–15. IEEE, Mar 2015.

[181] Pedro Santana, Tiago Vaquero, Andrew Wang, and Cheng Fang. PARIS: a Polynomial-Time, Risk-Sensitive Scheduling Algorithm for Probabilistic Simple Temporal Networks with Uncertainty. In *26th International Conference on Automated Planning and Scheduling*. ICAPS, 2016.

[182] Pedro Santana, Sylvie Thiebaux, and Brian Williams. RAO*: an Algorithm for Chance-Constrained POMDP's. In *Proceedings of the Thirtieth AAAI Conference*

on Artificial Intelligence (AAAI-16). Association for the Advancement of Artificial Intelligence, 2016.

[183] Keith Denoyer, David Hyland, Lawrence Davis, and David Miller. MACE II - A Space Shuttle experiment for investigating adaptive control of flexible spacecraft. *Guidance, Navigation, and Control Conference and Exhibit*, pages 1191–1196, 1998.

[184] Larry Davis, David Hyland, Gary Yen, and Alok Das. Adaptive neural control for space structure vibration suppression. *Smart Materials and Structures*, 8(6):753–766, 1999.

[185] David Hyland and Daniel Scharf. Adaptive neural control for MACE II. In *Space Technology Conference and Exposition*, Reston, Virigina, Sep 1999. American Institute of Aeronautics and Astronautics.

[186] John Louis Goodman. History of Space Shuttle Rendezvous and Proximity Operations. *Journal of Spacecraft and Rockets*, 43(5):944–959, 2006.

[187] Pierre Blanc-Paques, Emmanuel Gogibus, Christophe Louembet, and Mounir Kara-Zaitri. Evaluation of autonomous guidance techniques for space rendezvous and withdrawal strategy. In *AIAA Guidance, Navigation, and Control Conference*, pages 1–6, Reston, Virigina, Aug 2010. American Institute of Aeronautics and Astronautics.

[188] David K. Geller. Linear Covariance Techniques for Orbital Rendezvous Analysis and Autonomous Onboard Mission Planning. *Journal of Guidance, Control, and Dynamics*, 29(6):1404–1414, 2006.

[189] G Gaias, S D'Amico, and J.-S. Ardaens. Generalized Multi-Impulsive Maneuvers for Optimum Spacecraft Rendezvous. In *5th International Conference on Spacecraft Formation Flying Missions and Technologies*, pages 1–15, 2013.

[190] G. Gaias and S. D'Amico. Impulsive Maneuvers for Formation Reconfiguration Using Relative Orbital Elements. *Journal of Guidance, Control, and Dynamics*, pages 1–14, 2014.

[191] Christopher J Guerra. *Planning and Scheduling Proximity Operations for Autonomous Orbital Rendezvous*. PhD thesis, Massachusetts Institute of Technology, 2003.

[192] Christopher Guerra and Lance Page. Autonomous Onboard Planning for Satellite Rendezvous and Proximity Operations [invited]. In *Infotech@Aerospace*, pages 1–13, Reston, Virigina, Sep 2005. American Institute of Aeronautics and Astronautics.

[193] Giorgio Guglieri, Franco Maroglio, Pasquale Pellegrino, and Liliana Torre. Design and development of guidance navigation and control algorithms for spacecraft rendezvous and docking experimentation. *Acta Astronautica*, 94(1):395–408, 2014.

[194] Mark Jackson, Christopher D'Souza, and Hobson Lane. Autonomous mission management for spacecraft rendezvous using an agent hierarchy. *Collection of Technical Papers - InfoTech at Aerospace: Advancing Contemporary Aerospace Technologies and Their Integration*, 3(September):1385–1397, 2005.

[195] W.H. Clohessy and R.S. Wiltshire. Terminal Guidance Systems for Satellite Rendezvous. *Journal of the Aerospace Sciences*, 27(9):653–658, Jan 1960.

[196] Oliver Montenbruck and Eberhard Gill. *Satellite Orbits.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[197] David A. Vallado. *Fundamentals of Astrodynamics and Applications.* Springer, New York, 3rd edition, 2007.

[198] Charles H. Acton. Ancillary data services of NASA's Navigation and Ancillary Information Facility. *Planetary and Space Science*, 44(1):65–70, Jan 1996.

[199] G.E. Cook. Satellite Drag Coefficients. Technical report, Royal Aircraft Establishment, 1965.

[200] M Brand, V Shilpiekendula, C Yao, and S A Bortoff. A Parallel Quadratic Programming Algorithm for Model Predictive Control. *18th World Congress of the International Federation of Automatic Control (IFAC)*, pages 1031–1039, 2011.

[201] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization.* Athena Scientific series in optimization and neural computation: 6. Athena Scientific, Belmont, Massachusetts, 1997.

[202] Christopher Masaru Pong. *Autonomous thruster failure recovery for underactuated spacecraft.* PhD thesis, Massachusetts Institute of Technology, 2010.

[203] Kathleen Riesing. Orbit Determination from Two Line Element Sets of ISS-Deployed CubeSats. In *29th Annual AIAA/USU Conference on Small Satellites*, pages 1–9, Logan, UT, 2015. AIAA.

[204] Michael Stein. Large Sample Properties of Simulations Using Latin Hypercube Sampling. *Technometrics*, 29(May):143–151, 1987.

[205] Arthur E Bryson and Yu-Chi Ho. *Applied optimal control; optimization, estimation, and control.* Ginn and co., Waltham, Massachusetts, 1969.

[206] Arthur Gelb. *Applied optimal estimation.* M.I.T. Press, Cambridge, Massachusetts, 1974.

[207] A. Richards and J. How. Robust stable model predictive control with constraint tightening. In *2006 American Control Conference.* IEEE, 2006.

[208] Alberto Carrassi and Stéphane Vannitsem. State and parameter estimation with the extended Kalman filter: An alternative formulation of the model error dynamics. *Quarterly Journal of the Royal Meteorological Society*, 137(655):435–451, 2011.