

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

EXPERIMENTAL EVALUATION METHODOLOGY FOR SPACECRAFT PROXIMITY MANEUVERS IN A DYNAMIC ENVIRONMENT

by

Richard Salvatore Zappulla II

June 2017

Dissertation Supervisor: Marcello Romano

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 2017	Dissertation September 29, 2014 – June 16, 2017

4. TITLE AND SUBTITLE

EXPERIMENTAL EVALUATION METHODOLOGY FOR SPACECRAFT PROXIMITY MANEUVERS IN A DYNAMIC ENVIRONMENT

5. FUNDING NUMBERS

6. AUTHOR(S)

Richard Salvatore Zappulla II

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

In an effort to pursue advanced space missions, improved onboard trajectory optimization and path-planning capabilities are necessary. Regardless of the mission, the paramount requirement for any candidate guidance-and-control method is the ability to react in real time to a dynamic environment, followed by fuel efficiency. Ground-based kinodynamic test beds are critical in developing, testing, and verifying these requirements. The NPS POSEIDYN test bed is introduced here as a state-of-the-art (SOTA) dynamic, hardware-in-the-loop test bed. Key improvements to the software architecture, enabling the development of multi-rate guidance, navigation, and control (GNC) algorithms, are presented in addition to a detailed system characterization. To aid in the experimental evaluation of GNC algorithms, a Standard Test Framework is proposed in addition to a new guidance comparison metric previously missing in the literature, which simultaneously captures the computational complexity and algorithm performance. Guidance and control methods representing the SOTA are experimentally evaluated and compared against a proposed rapidly exploring random tree-based guidance method using the proposed test framework and the comparison metric.

14. SUBJECT TERMS

guidance, control, GNC, real-time guidance and control, hardware-in-the-loop, dynamic simulator, pulse-width modulation, sigma-delta modulation, standard test framework, guidance comparison methodology, guidance comparison metric, dynamic environment, guidance and control trade space

15. NUMBER OF PAGES 221

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

EXPERIMENTAL EVALUATION METHODOLOGY FOR SPACECRAFT PROXIMITY MANEUVERS IN A DYNAMIC ENVIRONMENT

Richard Salvatore Zappulla II
Civilian, Department of the Air Force
B.S., Georgia Institute of Technology, 2011
M.S., Georgia Institute of Technology, 2013

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN ASTRONAUTICAL ENGINEERING
from the
NAVAL POSTGRADUATE SCHOOL
June 2017

Approved by: Marcello Romano
Professor of Mechanical and
Aerospace Engineering
Dissertation Supervisor

Roberto Cristi
Professor of Electrical and
Computer Engineering

Oleg Yakimenko
Professor of Mechanical and
Aerospace Engineering

Jae Jun Kim
Research Associate Professor of
Mechanical and Aerospace
Engineering

Josue Muñoz
Research Aerospace Engineer
Air Force Research Laboratory,
Space Vehicle Directorate

Approved by: Dr. Garth Hobson
Chair, Department of Mechanical and Aerospace Engineering

Approved by: Douglas Moses
Vice Provost for Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In an effort to pursue advanced space missions, improved onboard trajectory optimization and path-planning capabilities are necessary. Regardless of the mission, the paramount requirement for any candidate guidance-and-control method is the ability to react in real time to a dynamic environment, followed by fuel efficiency. Ground-based kinodynamic test beds are critical in developing, testing, and verifying these requirements. The NPS POSEIDYN test bed is introduced here as a state-of-the-art (SOTA) dynamic, hardware-in-the-loop test bed. Key improvements to the software architecture, enabling the development of multi-rate guidance, navigation, and control (GNC) algorithms, are presented in addition to a detailed system characterization. To aid in the experimental evaluation of GNC algorithms, a Standard Test Framework is proposed in addition to a new guidance comparison metric previously missing in the literature, which simultaneously captures the computational complexity and algorithm performance. Guidance and control methods representing the SOTA are experimentally evaluated and compared against a proposed rapidly exploring random tree-based guidance method using the proposed test framework and the comparison metric.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1 Introduction **1**

1.1 Motivation 1

1.2 State of the Art Real-Time Guidance and Control Methods 4

1.3 Research Objectives 7

1.4 Dissertation Contributions 7

1.5 Dissertation Overview 9

1.6 Publication History 11

2 Constrained Dynamic System Development **13**

2.1 Reference Scenario 13

2.2 Dynamical Environment 13

2.3 Constraints 16

2.4 Optimal Control Problem Formulation. 18

3 The POSEIDYN Test Bed and TRIDENT-GNC Software Architecture **25**

3.1 Motivation for Ground-Based Test Bed 25

3.2 Survey of Relevant Test Beds 26

3.3 Experimental Facility Overview: The POSEIDYN test bed 27

3.4 Hardware Architecture Overview. 27

3.5 Software Architecture Overview 32

3.6 POSEIDYN Test Bed Characterization 39

3.7 Case Study 46

3.8 Summary 54

4 Spacecraft Thruster Modulation Techniques **57**

4.1 Overview of Thruster Modulation Techniques 57

4.2 Pulse Modulator Development 59

4.3 Closed-Loop Spacecraft ACS with a Modulator-in-the-Loop 62

4.4 Experimental Comparison 64

4.5 Summary 71

5 Experimental Evaluation Methodology 73
5.1 Introduction 73
5.2 Standard Test Framework 74
5.3 General Formulation of the Guidance Control Metric 78
5.4 Interpretation of the Guidance Control Metric 80
5.5 System-Specific Formulations 81
5.6 Summary 84

6 Evaluation of Real-Time Spacecraft Guidance and Control Methods 87
6.1 Introduction 87
6.2 Optimal Control Solutions to the Standard Test Framework 88
6.3 Formulation of Evaluated Real-Time Guidance and Control Methods . . . 92
6.4 Closed-Loop Rapidly-Expanding Random Trees 102
6.5 Attitude Control 116
6.6 Experimental Setup 117
6.7 Experimental Results and Discussion 121
6.8 Simulation-Based Case Study Evaluating the Effect of Dynamic Obstacles on G&C Methods 128
6.9 Comparison of Guidance and Control Methods 129
6.10 Summary 134

7 Development of the Guidance and Control Trade Space 137
7.1 Case Study Overview 137
7.2 Bounding the Control Effort 137
7.3 Bounding the Sampling Time 138
7.4 Analysis and Discussion 142

8 Summary and Future Work 145
8.1 Summary of Contributions 145
8.2 Contribution Traceability 149
8.3 Future Work 149

Appendix A Validation of the Minimum Sampling Rate Metric **153**
A.1 Numerical Case Study Overview 153
A.2 Minimum Sampling Rate for Tracking Transient Responses. 157

Appendix B Taguchi Analysis Method **159**
B.1 Overview 159
B.2 Methodology 159
B.3 Taguchi Orthogonal Arrays 160
B.4 Computing the Average Effects 161

Appendix C Publication History **163**
C.1 Patents 163
C.2 Supplemental Media 163
C.3 Journal Publications 163
C.4 Conference Proceedings 164

List of References **167**

Initial Distribution List **185**

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

Figure 1.1 Evolution of Real-Time Autonomous Guidance and Control. . . . 2

Figure 1.2 Guidance algorithm trade-off between increased optimality and computational burden. 3

Figure 2.1 Illustration of the RSW frame for a Target spacecraft in a near-circular orbit with the Chaser in close-proximity. 15

Figure 2.2 Notional illustration of a keep-out zone combined with the docking cone corridor constraint for the rendezvous and docking scenario. 17

Figure 3.1 Lineup of the first- to fourth-generation floating spacecraft simulators (FSSs) used on the Proximity Operation of Spacecraft: Experimental hardware-In-the-loop DYNamic simulator (POSEIDYN) test bed. 28

Figure 3.2 Overview of the main elements of the POSEIDYN test bed. . . . 28

Figure 3.3 2.5 N·m·s Ball Aerospace reaction wheel assembly mounted atop one of a FSS. 30

Figure 3.4 Overview of POSEIDYN communications architecture. 31

Figure 3.5 Overview of the onboard software architecture of the Floating Spacecraft Simulators. 33

Figure 3.6 Toolkit for Real-tIme DevelopmENT of Guidance, Navigation & Control (TRIDENT-GNC) logo. 34

Figure 3.7 Description of the DKF health telemetry item bit fields. 35

Figure 3.8 Reaction wheel speed mode controller. 40

Figure 3.9 Comparison of operating system latencies between an unpatched and patched Linux kernel. 40

Figure 3.10 FOG data histogram with mean output and 1σ indications and Allan variance plot. 43

Figure 3.11 FSS thruster characterization data with first-order fit and estimated 3σ bounds. 44

Figure 3.12 Average residual acceleration at nine locations on the granite monolith. 46

Figure 3.13 Overhead view of the FSS path with four equally-spaced attitude snapshots in time across the tangential intercept guidance segment and a single revolution about the fixed point. 49

Figure 3.14 Position and velocity of the FSS during the entire maneuver. . . 50

Figure 3.15 Position and velocity errors of the FSS during circumnavigation. 51

Figure 3.16 Commanded and actuated control inputs (in the inertial frame) throughout the circumnavigation maneuver. 52

Figure 3.17 Covariance time histories throughout the circumnavigation maneuver. 53

Figure 3.18 Detailed illustration of the Attitude Rate Covariance recovery. . . 54

Figure 4.1 First-order Sigma-Delta Modulator ($\Sigma\Delta$M) topology and linearized block diagram. 60

Figure 4.2 Pulse-Width Modulator (PWM) topology and linearized block diagram. 61

Figure 4.3 Notional spacecraft attitude control system block diagram. . . . 63

Figure 4.4 Test setup for the comparison between $\Sigma\Delta$M and PWM. 64

Figure 4.5 Direct comparison of the $\Sigma\Delta$M and PWM trajectories averaged over 10 runs. 67

Figure 4.6 Experimental results averaged over 10 runs comparing the response of the $\Sigma\Delta$M to PWM for a 90 deg slew. 68

Figure 4.7 Comparison of the linearized model response to the simulated and experimental responses. 70

Figure 5.1 Schematic of the Standard Test Framework test cases. 76

Figure 5.2 Notional illustration on how to interpret the Guidance Comparison Metric. 81

Figure 5.3 Illustration of the conversion of a torque to a force couple. . . . 82

Figure 5.4 Distribution of thrust-to-mass ratios for surveyed spacecraft. . . 83

Figure 5.5 ν_1 scaling study results showing its scaling invariance. 83

Figure 5.6 Power-to-weight ratios of surveyed automobiles. 86

Figure 6.1 Trajectories of the optimal control solution for the Standard Test Framework test cases with six equally-spaced (in time) attitude snapshots. 91

Figure 6.2 Notional Target keep-out zone and docking cone path constraints (blue) with overlaid docking cone constraint (red) in the Target body frame. 93

Figure 6.3 Example path planning solutions using the RRT* algorithm for various numbers of nearest neighbors, k, and 150 nodes. 108

Figure 6.4 Elementary 2-D uniform flow centered at the goal position. . . . 110

Figure 6.5 Elementary 2-D Doublet Potential Flow. 112

Figure 6.6 Test setup for the real-time guidance and control algorithm experimental campaign. 118

Figure 6.7 Comparison of the translational control effort and maneuver time. 122

Figure 6.8 Comparison of the attitude error versus distance remaining. . . 125

Figure 6.9 Case 0 experimental results. 126

Figure 6.10 Case 1 experimental results. 126

Figure 6.11 Case 2 experimental results. 127

Figure 6.12 Case 3 experimental results. 127

Figure 6.13 Average effect of varying the rotation rate and the radius of rotation of the obstacle for an analytical (APF), hybrid (AAPF), and an optimization-based (IDVD) guidance and control methods. . . . 129

Figure 6.14 Comparison of the APF and AAPF to other guidance and control methods tested using the *Standard Test Framework*. 130

Figure 6.15 Comparison of guidance and control methods using the Guidance Comparison metric. 130

Figure 6.16 Comparison of guidance and control methods using the Guidance Comparison metric. 131

Figure 6.17 Comparison of guidance and control methods using the Guidance Comparison metric. 132

Figure 6.18 Comparison of the Initial Trees Generated for Case 1. 133

Figure 7.1 Annotated guidance and control trade space based on the Guidance Comparison metric. 143

Figure 8.1 Notional computational architecture consisting of a CPU, Coprocessor, and GPU. 152

Figure A.1 Average steady-state error versus controller sample rate. 154

Figure A.2 Average steady-state error versus controller sample rate for a 3-axis spacecraft reorientation maneuver. 156

Figure A.3 Average steady-state error versus controller sample rate for a simple harmonic oscillator. 157

Figure A.4 Average steady-state error versus controller sample rate for a double-integrator system under the minimum-time optimal control policy. 157

Figure A.5 Maximum change in error during the transient phase for a simple harmonic oscillator. 158

Figure B.1 Notional average effect plot for a $L_9(3^2)$ DOE. 162

Figure B.2 Factor A and B average effects overlaid for the notional $L_9(3^2)$ DOE. 162

List of Tables

Table 1.1 Summary of real-time system categories. 4

Table 1.2 Mapping of the research objectives to the supporting chapters. . 8

Table 1.3 Summary of contributions mapped to research objectives. 9

Table 1.4 Summary of relevant publications utilized throughout the dissertation. 12

Table 2.1 Relevant orbital parameters for the reference rendezvous and docking scenario. 13

Table 3.1 Summary of the relevant test beds characteristics. 26

Table 3.2 Comparison of space-qualified and typical terrestrial processors. 31

Table 3.3 Summary of the `CyclicTest` latency results. 40

Table 3.4 Vicon sensor noise characterization results. 42

Table 3.5 Summary of FOG noise analysis. 43

Table 3.6 Summary of relevant FSS physical properties. 45

Table 3.7 Case study test parameters. 48

Table 3.8 Summary of the circumnavigation time, position, speed, and attitude errors. 50

Table 3.9 Mean estimated error standard deviation over the circumnavigation maneuver. 50

Table 4.1 Summary of relevant experiment and FSS parameters. 65

Table 4.2 Comparison of the 10-run averaged experimental metrics (with 1σ error) between the $\Sigma\Delta$M & PWM. 66

Table 4.3 Comparison of transient and settling characteristics. 69

Table 5.1 *Standard Test Framework* parameters. 77

Table 5.2 Obstacle location and size for each test case. 77

Table 5.3 Summary of surveyed RPO spacecraft design specifications. . . 85

Table 6.1 Summary of the optimal control solutions. 90

Table 6.2 Summary of closest approach for each test case. 90

Table 6.3 Summary of attitude control methods. 117

Table 6.4 Saturation Controller Parameter Selection. 117

Table 6.5 Summary of APF and AAPF guidance parameters. 119

Table 6.6 Summary of IDVD guidance parameters. 119

Table 6.7 Summary of LQ-MPC guidance parameters for Case 1. 120

Table 6.8 Summary of LQ-MPC and NMPC guidance parameters for Case 2. 120

Table 6.9 Summary of RRT* and RRT*-HPF guidance parameters. 120

Table 6.10 Ratio of the non-dimensional control effort ν_1 for each evaluated method to the optimal non-dimensional control effort ν_1^*. 123

Table 6.11 Case 0 comparison of guidance algorithm performance metrics. . 123

Table 6.12 Case 1 comparison of guidance algorithm performance metrics. . 123

Table 6.13 Case 2 comparison of guidance algorithm performance metrics. . 124

Table 6.14 Case 3 comparison of guidance algorithm performance metrics. . 124

Table 6.15 Summary of the values for the factors considered in the case study. 128

Table 7.1 Summary of parameters for the example case study. 138

Table 7.2 Summary of time constants and bounds on ν_1 and ν_2. 142

Table 8.1 Mapping of the contribution to the supporting chapters. 149

Table B.1 $L_9(3^2)$ Taguchi orthogonal array. 161

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AIAA	American Institute of Aeronautics and Astronautics
AAS	American Astronautical Society
APF	Artificial Potential Function
AAPF	Adaptive Artificial Potential Function
ARW	angular random walk
ATV	Automated Transfer Vehicle
AR&D	autonomous rendezvous and docking
CG&C	Computational Guidance & Control
CL-RRT*	closed-loop RRT*
CL-RRT*–HPF	closed-loop RRT*–HPF
COM	center of mass
CMG	control moment gyroscope
DH	dual hyperplane
DOE	design of experiment
DCC	docking cone corridor
DKF	Discrete Kalman Filter
DOF	degree of freedom
FOG	fiber-optic gyroscope
FSS	floating spacecraft simulator

G&C	guidance and control
GNC	Guidance, Navigation, and Control
HCW	Hill-Clohessy-Wiltshire
HPF	Harmonic Potential Function
HTV	H-II Transfer Vehicle
IDVD	Inverse Dynamics in the Virtual Domain
IEEE	Institute of Electrical and Electronics Engineers
IPOPT	Interior Point OPTimizer
IMU	inertial measurement unit
ISS	International Space Station
LQ-MPC	Linear-Quadratic Model Predictive Control (MPC)
LQR	Linear Quadratic Regulator
MITL	modulator-in-the-loop
MOI	moment of inertia
MPC	Model Predictive Control
NASA	National Aeronautics and Space Administration
NMPC	Nonlinear MPC
NPS	Naval Postgraduate School
OCP	optimal control problem
OS	operating system
POSEIDYN	Proximity Operation of Spacecraft: Experimental hardware-In-the-loop DYNamic simulator

PCM	pulse-control modulator
PWM	Pulse-Width Modulator
PWPFM	Pulse-Width Pulse-Frequency Modulator
RC	reaction control
RH	rotating hyperplane
RMSE	root mean square error
RPO	rendezvous and proximity operations
RT-G&C	Real-Time Guidance & Control
RTOS	real-time operating system
RRT	Rapidly-Exploring Random Tree
RRT*	Rapidly-Exploring Random Tree Star
RSO	resident space object
RW	reaction wheel
$\Sigma\Delta$**M**	Sigma-Delta Modulator
TRIDENT-GNC	Toolkit for Real-tIme DevelopmENT of Guidance, Navigation & Control
UDP	User Datagram Protocol
VM	virtual machine
ZOH	zero-order hold

THIS PAGE INTENTIONALLY LEFT BLANK

Nomenclature

Operators

$\hat{(\cdot)}$	=	unit vector of $(\cdot)$
$(\cdot)^{\mathsf{T}}$	=	transpose of $(\cdot)$
$(\cdot)^{+}$	=	Moore-Penrose pseudoinverse of the matrix $(\cdot)$
$(\cdot)^{-1}$	=	inverse of the matrix $(\cdot)$
$(\cdot)^{*}$	=	optimal solution of the variable $(\cdot)$
$(\cdot)^{\times}$	=	matrical representation of the cross-product operator
$(\cdot)_k$	=	sample-time index
$\bar{(\cdot)}_n$	=	average effect of factor$(\cdot)$ at level n
$\lvert(\cdot)\rvert$	=	absolute value operator
$\left\lVert\cdot\right\rVert_1$	=	L_1-norm
$\left\lVert\cdot\right\rVert_2$	=	L_2-norm
$\left\lVert\cdot\right\rVert \left\lVert\right\rVert_{\mathsf{F}}$	=	Frobenius norm of the matrix $(\cdot)$
$\delta(\cdot)$	=	variation of $(\cdot)$
$(\cdot)^{(p)}$	=	variable $(\cdot)$ in phase p
$\nabla_x(\cdot)$	=	gradient of $(\cdot)$ with respect to x
$\nabla_r(\cdot)$	=	gradient of $(\cdot)$ with respect to r
$\exp\left[\cdot\right]$	=	exponential function
$\mathrm{diag}(\cdot)$	=	square matrix with diagonal elements $(\cdot)$
$\mathrm{sign}\left(\cdot\right)$	=	sign, or signum, function

Notation

$\mathbf{a} : \mathbb{B} \mapsto \mathbb{C}$	=	$\mathbf{a}$ is function from $\mathbb{B}$ to $\mathbb{C}$
$a \propto b$	=	a is proportional to b
$\mathbb{A} \setminus \mathbb{B}$	=	the set containing all elements of the set $\mathbb{A}$ not in $\mathbb{B}$
$q \in \mathbb{X}$	=	q is an element of the set $\mathbb{X}$
$\mathbb{U} \subseteq \mathbb{R}$	=	the set $\mathbb{U}$ is a proper subset of the set $\mathbb{R}$

Variables

$\mathbf{0}_{n\times n}$	=	$n \times n$ matrix of zeros
a	=	semi-major axis of an orbit, m
a_1, a_2, a_3	=	semi-major axes of an ellipse, m
$\mathbf{b}(\cdot)$	=	column matrices of $\mathbf{B}$
$\mathbf{B}$	=	continuous-time control input matrix
b_0	=	reference moment arm (1 m), m
b_1, b_2	=	semi-minor axes of an ellipse, m
$\mathbf{c}_p$	=	constraint vector associated with the optimal control problem
c_e	=	control effort
D	=	distance from the FSS to the reference trajectory, m
d_{a_i}	=	i^{th} coefficient of the polynomial describing the x trajectory
d_{b_i}	=	i^{th} coefficient of the polynomial describing the y trajectory
$d_i(\cdot)$	=	distance from the center of mass of the i^{th} obstacle to the boundary its keep-out zone boundary, m
d_{off}	=	docking cone interface offset, m
$\mathbf{e}$	=	error function (vector)
$\dot{\mathbf{e}}$	=	time derivative of the error function (vector)
$\mathbf{e}_f$	=	terminal condition vector associated with the optimal control problem
$e(t)$	=	time-varying error function
F	=	force, N
$\mathbf{f}$	=	vector of forces applied by each thruster, N
f_{pwm}	=	PWM sampling rate, Hz
$f_{\Sigma\Delta\text{M}}$	=	$\Sigma\Delta$M sampling rate, Hz
f_{r}	=	reference waveform sampling rate, Hz
g_0	=	gravitational acceleration constant at sea-level, m/s^2
$g(\cdot)$	=	running cost
$g_a(\cdot)$	=	augmented running cost
$\mathbf{H}_k$	=	measurement mapping matrix at sample time k
$\mathscr{H}(\cdot)$	=	Hamiltonian
$h(\cdot)$	=	endpoint cost
$\mathbf{I}_{n\times n}$	=	$n \times n$ Identity matrix
$\hat{\mathbf{I}}$	=	orthonormal basis vector of the Earth-Centered Inertial frame

I_{RW}	=	reaction wheel inertia, kg·m^2
$\hat{\mathbf{J}}$	=	orthonormal basis vector of the Earth-Centered Inertial frame
$\mathbf{J}$	=	inertia matrix, kg·m^2
J, J_{zz}	=	moment of inertia about the z axis, kg·m^2
$J(\cdot)$	=	cost functional associated with the optimal control problem
$J_a(\cdot)$	=	augmented cost functional associated with the optimal control problem
j	=	$\sqrt{-1}$
$\hat{\mathbf{K}}$	=	orthonormal basis vector of the Earth-Centered Inertial frame
$\mathbf{K}_a$	=	gain matrix
$\mathbf{K}_k$	=	Kalman gain at sample time k
k	=	strictly-positive constant
k_a	=	strictly-positive attractive potential gain
k_r	=	strictly-positive repulsive potential gain
k_d	=	derivative gain
k_p	=	proportional gain
ℓ_c	=	length of the docking cone corridor, m
$\mathbf{M}$	=	thruster to control input mapping matrix
m	=	mass, kg
$\mathbf{N}_i$	=	shaping matrix for the i^{th} obstacle
$\hat{\mathbf{n}}_{\text{DCC}}$	=	docking axis unit vector
$\hat{\mathbf{n}}_{\text{c}}$	=	normal vector
N	=	horizon length
N_c	=	number of constraints
N_e	=	number of terminal conditions
N_f	=	number of thrusters
N_m	=	number of measurements
N_o	=	number of obstacles
N_u	=	number of control inputs
N_x	=	number of states
n	=	mean motion, s^{-1}
n_{t_x}	=	number of coefficients describing t_x
n_{t_y}	=	number of coefficients describing t_y
$\mathbf{P}$	=	solution to the discrete Riccati equation

$\mathbf{P}_k$	=	error covariance at sample time k
$\mathbf{p}_c$	=	Chaser location, m
$\dot{\mathbf{p}}_c$	=	Chaser velocity, m/s
$\mathbf{p}_{c,\text{obs}}$	=	obstacle center of rotation, m
$\mathbf{p}_{\text{dock}}$	=	point on the hyperplane
$\mathbf{p}_f$	=	desired configuration, m
$\dot{\mathbf{p}}_f$	=	desired velocity, m/s
$\mathbf{p}_{\text{ref}}$	=	reference configuration, m
$\mathbf{p}_t$	=	Target location, m
$\dot{\mathbf{p}}_t$	=	Target velocity, m/s
$P_{out}(t)$	=	power output, W
$\mathbf{Q}$	=	intermediate state weighting matrix
$\mathbf{Q}_a$	=	attractive potential shaping matrix
$\mathbf{Q}_k$	=	process noise covariance at sample time k
$\mathbf{q}$	=	configuration variable
q_0	=	root node of the tree $\mathcal{T}$
q_f	=	goal configuration
q_i	=	a node belonging to the tree $\mathcal{T}$
q_k	=	scalar quantity for Q_k at sample time k
q_{nearest}	=	nearest node
q_{new}	=	newly added node to the tree $\mathcal{T}$
q_{par}	=	parent node
q_{rand}	=	randomly sampled configuration
$\mathbf{R}(\cdot)$	=	rotation operator
$\mathbf{R}$	=	control effort weighting matrix
$\mathbf{R}(t)$	=	time-varying upper triangular matrix
$\mathbf{R}_k$	=	sensor noise covariance at sample time k
$\mathbb{R}$	=	real number
$\mathbb{R}_+$	=	strictly-positive real number
$\bar{\mathbb{R}}_+$	=	non-negative real number
$\mathbb{R}_{<0}$	=	strictly negative real number
$\mathbb{R}^{m\times n}$	=	$m \times n$ matrix containing real numbers
$\mathbf{r}_{cd}$	=	relative position of the Chaser with

		respect to the docking interface, m
$\mathbf{r}_{cf}$	=	relative position of the Chaser with
		respect to the terminal position, m
$\mathbf{r}_{co_i}$	=	relative position of the Chaser with respect
		to the center of mass of the i^{th} obstacle, m
$\mathbf{r}_{ct}$	=	relative position of the Chaser with respect
		to the center of mass of the Target spacecraft, m
$\mathbf{r}_{\text{target}}$	=	position vector of the target
		in the Earth-Centered Inertial frame, m
r_{ref}	=	reference radius, m
R, R_i	=	obstacle radius, m
$r(t)$	=	repeating reference waveform output
$s(\cdot)$	=	switching function
s_a, s_b	=	saturation controller parameters
$\mathrm{T}_{(\cdot)}(s)$	=	closed-model attitude control system with
		the $(\cdot)$ modulator-in-the-loop
$\mathcal{T}$	=	a tree
T_b	=	Executive sampling period, s
T_g	=	guidance algorithm sampling period, s
T_p	=	time to peak response, s
T_{pwm}	=	PWM sampling period, s
T_{rso}	=	time constant associated with the motion of a RSO, s
T_{RT}	=	maximum real-time sampling period, s
$T_{\Sigma\Delta\text{M}}$	=	$\Sigma\Delta$M sampling period, s
T_{r}	=	reference waveform sampling period, s
T_s	=	settling time, s
t	=	time, s
t_0	=	initial time, s
t_f	=	final time, s
$t_x(\kappa_t)$	=	polynomial describing time in the x direction
$t_y(\kappa_t)$	=	polynomial describing time in the y direction
$\mathbb{U}$	=	set of all admissible controls
$\mathbf{u}$	=	control input vector

$\mathbf{u}_k$	=	control input vector at sample time k
$\mathbf{u}^+$	=	maximally negative control input vector
$\mathbf{u}^-$	=	maximally positive control input vector
${}^{\mathcal{B}}\mathbf{u}(t)$	=	control input vector in the body frame
${}^{\mathcal{I}}\mathbf{u}(t)$	=	control input vector in the inertial frame
v_x, v_y	=	velocity, m/s
$\mathbf{v}_{\text{target}}$	=	velocity vector of the target
		in the Earth-Centered Inertial frame, m/s
V_{max}	=	maximum output of a modulator
$v(t)$	=	time-varying input signal representing the commanded force
$\bar{v}$	=	average speed, m/s
v_r	=	radial speed, m/s
v_θ	=	tangential speed, m/s
v_{ref}	=	reference speed, m/s
$w(t)$	=	time-varying $\Sigma\Delta$M intermediate signal
$\mathbb{X}$	=	configuration space
$\mathbb{X}_{\text{free}}$	=	obstacle-free configuration space
$\mathbb{X}_{\text{near}}$	=	set of all nearest nodes
$\mathbb{X}_{\text{obs}}$	=	set containing the obstacle(s)
$\mathbf{x}^0$	=	numerical initial condition
$\mathbf{x}_f$	=	terminal state
$\mathbf{x}_k$	=	state vector at sample time k
$\hat{\mathbf{X}}$	=	orthonormal basis vector of the Hill-Clohessy-Wiltshire frame
$\hat{\mathbf{x}}_k$	=	state estimate at time step k
X_{avg}	=	performance criterion grand average
x, y	=	position, m
$\tilde{x}, \tilde{y}$	=	position error, m
x_c, y_c	=	position of the center of the reference trajectory, m
$x_1(t)$	=	first-order model of the FSS translational time history, m
$x_1(t)$	=	second-order model of the FSS translational time history, m
$x(\kappa_t)$	=	polynomial describing the x trajectory
$\dot{x}, \dot{y}$	=	velocity, m/s
$\ddot{x}, \ddot{y}, \ddot{z}$	=	acceleration, m/s^2

$\hat{\mathbf{Y}}$	=	orthonormal basis vector of the Hill-Clohessy-Wiltshire frame
$y(t)$	=	modulator output
$y(\kappa_t)$	=	polynomial describing the y trajectory
$\mathbf{z}$	=	Guidance Comparison metric
$\mathbf{z}_k$	=	measurement vector at sample time k
$\hat{\mathbf{Z}}$	=	orthonormal basis vector of the Hill-Clohessy-Wiltshire frame
α	=	angle, rad
α_f	=	angle with respect to the Target, rad
α_g	=	angle with respect to the goal position, rad
α_{RW}	=	reaction wheel acceleration, rad/s^2
β	=	docking cone corridor half-angle, rad
β_t	=	angle between FSS and point tangent to reference trajectory, rad
$\mathbf{\Gamma}_k$	=	discrete-time input gain matrix at sample time k
γ_{d_i}	=	angle with respect to the i^{th} doublet, rad
γ_t	=	angle of the Target with respect to i^{th} doublet, rad
$\Delta\omega$	=	difference between ω_{desired} and ω_{Measured}, rad/s
Δt_c	=	computational delay, s
Δt_{DKF}	=	discrete Kalman filter time step, s
Δt_m	=	maneuver time, s
ϵ_{DKF}	=	discrete Kalman filter convergence threshold
$\varepsilon_{\text{error}}$	=	vector component of the error quaternion
ε	=	$\Sigma\Delta$M quantizer threshold level
ζ	=	χ^2 test statistic for measurement association
ζ_{BW}	=	dynamic bandwidth, m/s^2
η_{error}	=	scalar component of the error quaternion
θ	=	attitude angle, rad
θ_{error}	=	error angle, rad
θ_f	=	desired attitude, rad
$\theta(t)$	=	second-order model of the FSS attitude time history, rad
θ_{ref}	=	reference attitude angle, rad
$\dot{\theta}$	=	angular rate, rad/s
κ	=	strictly positive non-dimensionalizing factor
κ_{d_i}, κ_i	=	strength of the i^{th} doublet

κ_t	=	virtual time
κ_{tar}	=	strength of the target doublet
Λ_i	=	convex set for the i^{th} element of $\boldsymbol{\rho}$
$\boldsymbol{\lambda}$	=	Lagrange multiplier vector
$\mu_\oplus$	=	gravitational parameter of Earth, m^3/s^2
ν_1	=	non-dimensional control effort, element of $\mathbf{z}$
ν_2	=	non-dimensional sampling time, element of $\mathbf{z}$
$\boldsymbol{\rho}$	=	vector of adaptive update parameters
$\dot{\boldsymbol{\rho}}$	=	time derivative of the vector of adaptive update parameters
$\tilde{\boldsymbol{\rho}}$	=	augmented vector of adaptive update parameters
$\rho_i j$	=	adaptive update parameter; ij component of $\mathbf{R}(t)$
σ_i	=	width parameter of the i^{th} obstacle repulsive potential
σ_x,σ_y	=	position sensor noise, m
σ_θ	=	attitude sensor noise, rad
σ_ω	=	fiber-optic gyroscope sensor noise, rad/s
τ	=	torque (scalar), N·m
$\boldsymbol{\tau}$	=	torque (vector), N·m
$\boldsymbol{\Upsilon}_k$	=	process noise covariance matrix at sample time k
Υ	=	vernal equinox
$\boldsymbol{\Phi}_k$	=	State transition matrix at sample time k
ϕ	=	angle of FSS relative to center of the reference trajectory, rad
ϕ_a	=	attractive potential
ϕ_b	=	Target boundary constraint repulsive potential
ϕ_d	=	potential due to a doublet
ϕ_r	=	obstacle repulsive potential
ϕ_{tot}	=	total potential
ϕ_{uf}	=	potential due to a uniform flow
$\xi(t)$	=	time-varying desired pulse-width
ψ_i	=	height parameter of the obstacle repulsive potential
$\dot{\psi}_{\text{ref}}$	=	user-specified desired angular rate, rad/s
$\boldsymbol{\omega}$	=	angular velocity vector of the spacecraft body frame with respect to the inertial frame, rad/s
$\boldsymbol{\omega}_k$	=	Process noise at sample time k

$\omega, \omega_{\text{BW}}$	=	system bandwidth, rad/s
ω_{desired}	=	desired flywheel velocity, rad/s
ω_{obs}	=	rate of rotation of the obstacle about $\mathbf{p}_{c,\text{obs}}$, rad/s
$\omega_{\text{Earth,local}}$	=	local rotation rate of the Earth, rad/s
ω_{Measured}	=	measured flywheel velocity, rad/s
ω_z	=	z component of the angular velocity, rad/s

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

I would like to first acknowledge the team of individuals at the United States Air Force Research Laboratory who allowed me attend the Naval Postgraduate School (NPS) to pursue this degree and complete this work. Specifically, I would like to thank Mr. Kevin Schum as well as the late Mr. Christopher Sillence for their enthusiastic support and guidance throughout the application process and Major Klayton Bobsein, Ms. Debra Fogle, Ms. Kristin Gallette, and Ms. Paula "D'Anne" Smith for their daily support that allowed me to be in-residence and focus on the work. I would also like to thank Dr. Frank Chavez for his guidance and feedback throughout this process. Furthermore, I would like to acknowledge the financial support of The Department of Defense Science, Mathematics, and Research for Transformation (SMART) Scholarship in pursuing this endeavor.

I would next like to thank Mr. Alessio Grompone, Dr. Marco Ciarcià, and the previous Spacecraft Robotics Laboratory (SRL) members who worked on improving the facilities during their time at NPS, my work would not have been possible without the previous efforts and contributions you have made to the facilities. I would like to thank the current members of the SRL with whom I have shared my time at NPS and have acted as my sounding board for my ideas, including LT Andrew Bradstreet and LT Neehar Pandya. I would also like to thank Dr. Josep Virgili-Llop for his assistance in "tinkering" and implementing various aspects of the test bed (albeit, well after normal business hours) and for implementing the IDVD method used in this work. I appreciate his patience in answering my many questions along the way. To Dr. Hyeongjun Park, thank you for your assistance in running some of the experiments for the various conference and journal papers we have collaborated on as well as for implementing the NMPC formulation used in this work. To Major Costantinos "Costas" Zagaris, thank you for sharing your knowledge of model predictive control as well as for implementing the LQ-MPC methods used in the this work. Your continuous support, both academically and professionally, has made my time here much more enjoyable.

Next, I would like to thank my dissertation committee members for their assistance and invaluable insights. I would like to thank Dr. Jae Jun Kim for challenging me academi-

cally throughout the process. To Dr. Oleg Yakimenko, thank you for teaching me never to gloss over the “simple” things, as they are often the most important – it took three attempts, but I think the title of this dissertation is finally the right one! To Dr. Roberto Cristi, thank you for the inspiration and encouragement to further pursue the modulation technique presented in this dissertation. To Dr. Josue Muñoz, thank you challenging me throughout the process as well as for always having an open door and taking the time to brainstorm through ideas and helping to tie the work together. Lastly, I am indebted to my advisor, Dr. Marcello Romano, for not only taking me on as a student, but for the numerous suggestions and pushes that helped me to grow academically and made me more confident in my work.

Lastly, I would like to thank my support system, my friends and family, throughout this endeavor. Thank you for the words of encouragement to lift my spirits when I needed it. Specifically, thank you to Major Steve Wojdakowski and Captain Patrick Acox for befriending me. Thank you, Giovanni Minelli for the support and encouragement you provided along the way, including the impromptu mock qualifying exams the night before my actual exam. A special shout out to my long-distance friends who are going through a similar undertaking, especially Kevin Reilley and Grant Ross. And lastly, I would like thank my family for their support throughout this journey, especially my girlfriend, Marjorie Christensen, and my mother and father, Barbara and Richard Zappulla, who have instilled in me the work ethic, passion, and discipline necessary to take on this great task in such a short amount of time.

CHAPTER 1:
Introduction

> The formulation of the problem is often more essential than its solution, which may be merely a matter of mathematical or experimental skill. To raise new questions, new possibilities, to regard old problems from a new angle requires creative imagination and marks real advance in science.
>
> —Albert Einstein and Leopold Infield
> *The Evolution of Physics* [1]

1.1 Motivation

Suppose for a moment, you are tasked with designing the guidance and control (G&C) architecture for an interplanetary sample-return mission. For this notional mission, the spacecraft will rendezvous with an object, travelling through the expanse of the Solar System, that is believed to be (partially) composed of some rare and highly-desired material. After it has rendezvoused, it will perform proximity maneuvers to collect various measurements as well as a physical sample from the object. After completing the mission, the spacecraft will rendezvous and dock with a pre-positioned fuel depot. The question becomes, *what guidance and control methods will you choose? Why?*

Do the intrinsic characteristics and properties of your selected method make it an attractive choice? Were there any environmental factors – such as communication delays and other naturally occurring bodies (e.g., micrometeoroids) – that require an autonomous response from the spacecraft? If these are important factors, is the original G&C still the best choice? Moreover, after you select a method, how are you going to test and validate the implementation?

While this is just a notional thought experiment, it illustrates some of the needs presented in the National Research Council's decadal study titled *Voyages for Planetary Science in the Decade 2013-2022* [2] and technologies identified in the National Aeronautics and Space Administration (NASA) *Technology Roadmaps* [3]. Both studies emphasize the need for improvements to onboard autonomous trajectory optimization, path planning, and replanning capabilities to support envisioned space missions [2]–[4]. These necessary advancements in onboard autonomous Guidance, Navigation, and Control (GNC) capabilities will enable more complex missions, such as autonomous

spacecraft rendezvous and proximity operations (RPO) as well as autonomous rendezvous and docking.

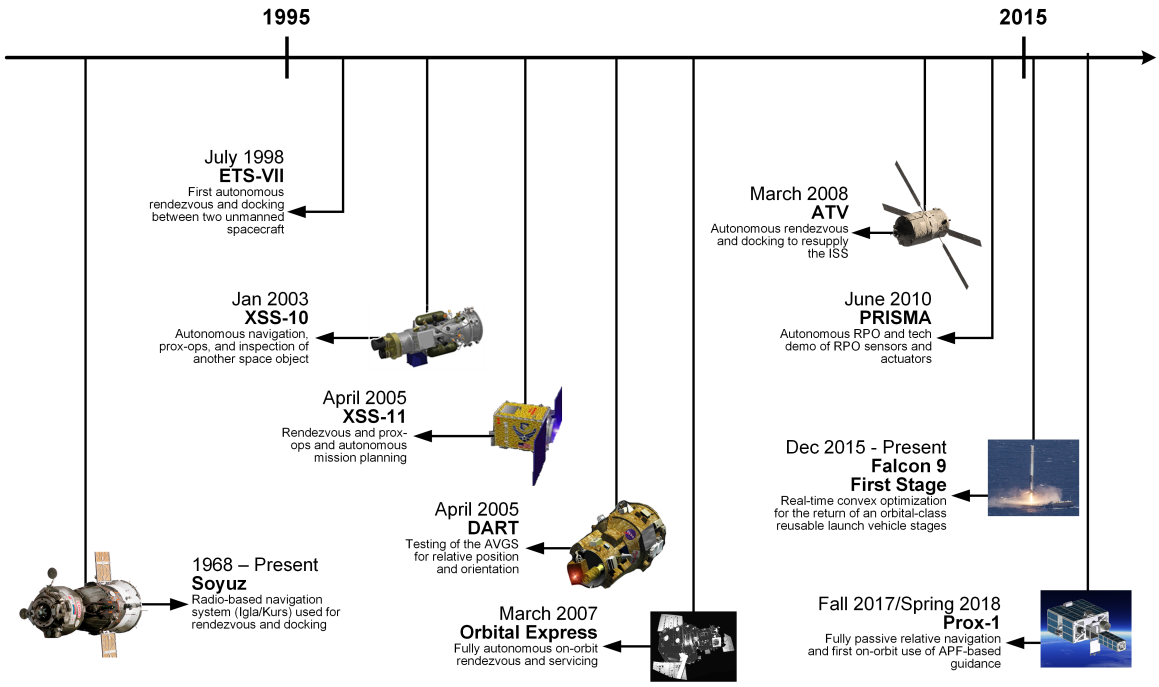


Figure 1.1. Evolution of Real-Time Autonomous Guidance and Control.[1]

As illustrated in Figure 1.1, numerous missions, which were either proposed or flower over the past few decades, have explored new techniques for safely conducting RPO [5], [7]. Arguably, RPO is considered to have begun in the early 1960s with the Vostok[2] and Gemini programs while autonomous RPO begin in 1967 with the Soyuz program [5]. The more recent push towards greater autonomy to enable more complex missions began with the Japan Aerospace Exploration Agency's Engineering Test Satellite VII (ETS-VII) [5]. Since then, several experimental missions have been completed (or proposed in the case of the Georgia Institute of Technology's Prox-1 [7], [8]), which have tested various algorithms and equipment necessary to conduct autonomous RPO [5], [9]–[11]. Yet despite active research in this area, few spacecraft can be found in literature which routinely perform autonomous RPO and docking. Currently, the only three spacecraft which autonomously dock with the International Space Station (ISS) are the European Space Agency's Automated Transfer Vehicle (ATV) [18] and the Russian Soyuz and Progress spacecraft [5]. Additionally, while not technically

[1]The data used to create this figure was adapted from [5]–[11]; the images were adapted from the following sources: [9], [10], [12]–[17].

[2]The rendezvous was performed by launching a manned capsule in such an orbit that it approached another on-orbit capsule (launched the previous day) to within 6.5 km [5].

a spacecraft, the Space Exploration Technologies Corporation (also known as SpaceX) has demonstrated the successful use of real-time convex optimization in recovering the first stage of its Falcon 9 launch vehicle [6] in an arguably more complex and dynamic environment than on orbit — effectively showing the potential for use on-orbit. Regardless of the mission, attention must be given to ensure safe operations in the vicinity of other spacecraft.

To achieve autonomous RPO, the guidance algorithm must not only ensure real-time, collision-free operations, but also reduce propellant consumption. This results in a conflicting set of requirements [19], [20] and is illustrated in Figure 1.2. For instance, obstacle avoidance requires the GNC subsystem to have the ability to respond and re-plan to a dynamic environment. This is critical, as collision avoidance is a hard requirement for any mission [7], [8], [21]. However, generating an efficient and safe trajectory involves solving a constrained optimal control problem (OCP). Once the optimal solution to the OCP is found, a reference trajectory can then be generated and tracked using a series of waypoints [18], [22] or feed-forward control [23]. Given the dimensionality of the OCP and possible numerical sensitivities, methods for finding optimal trajectories can be too computationally burdensome to be performed onboard a computationally constrained spacecraft. Thus, the generation of a sub-optimal but *feasible* solution is more important, as it will ensure the safety-of-fight requirement (i.e., collision avoidance) is satisfied at all times.

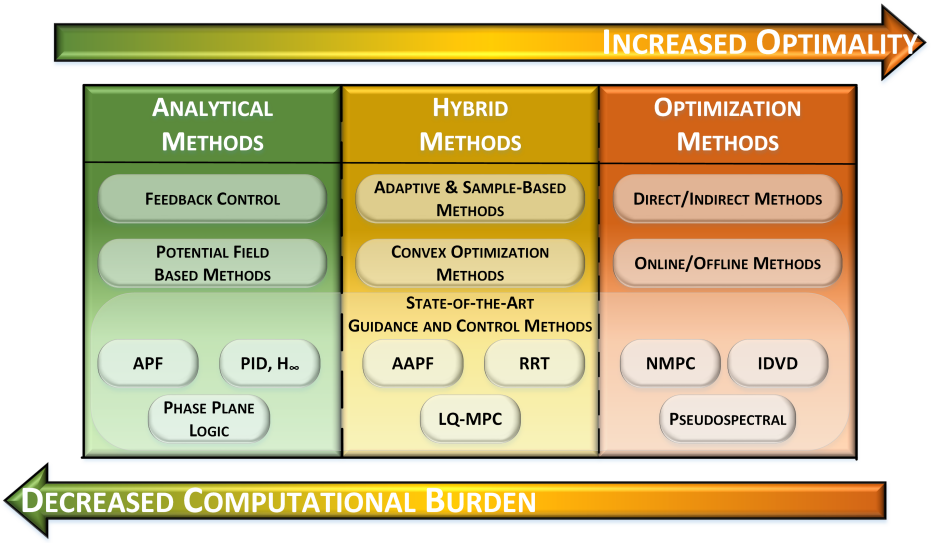


Figure 1.2. Guidance algorithm trade-off between increased optimality and computational burden.

1.2 State of the Art Real-Time Guidance and Control Methods

Real-Time Guidance & Control (RT-G&C) algorithms are those that can be implemented in the computationally constrained environment present onboard spacecraft and can be run in real-time such that they can provide feedback control. By real-time, it is meant that the algorithm readily provides a feasible solution within a prescribed amount of time regardless of the system load, and failure to meet these deadlines may result in a hazardous situation (e.g., a collision with another object) [24]. Notably, this time period can range from milliseconds to several seconds or minutes and is dependent upon several factors (and is further discussed in Section 7.3.1). Real-time systems are categorized as soft, hard, and firm [24]. The consequences of missing deadlines as well as corresponding examples for each of these categories are presented in Table 1.1.

Table 1.1. Summary of real-time system categories. Adapted from [24].

Category	Consequence of Missed Deadline(s)	Example Application
Soft	Degraded Performance	Computer Programs (e.g., Word Processor)
Hard	Entire System Failure	Safety-Oriented Systems (e.g., Fire Suppression System)
Firm	Tolerant of the Occasional Missed Deadline; Too Many Missed Deadlines May Result in System Failure	Embedded Controllers (e.g., Engine Management Controller)

The limiting factor in RT-G&C algorithms is the computational-constrained environment present onboard spacecraft. Despite modern computational technology, space-rated compute elements are subject to typical design constraints, such as high reliability in the presence of radiation, low power consumption, low thermal design power, and high computational power [25]. In general, power, volume, mass, and cost constraints make adding a dedicated processor for advanced G&C algorithms infeasible. Combined with the need to be respect time deadlines, RT-G&C algorithms are inherently limited in their computational complexity.

Different theoretical approaches have been explored to solve the guidance and control problem for achieving autonomous real-time, collision-free RPO in literature. Note, Figure 1.2 lists several state-of-the-art RT-G&C methods that will be discussed. Early RT-G&C methods, utilized by re-entry autopilot and reaction-control system in the Apollo spacecraft, include phase-plane logic and sliding mode control [26]. More recently,

proportional-integral-derivative as well as H-infinity controllers have been utilized by the ATV to track a reference trajectory and autonomously dock with the ISS [18]. Other approaches in literature are limited to experimental testing only, mostly on terrestrial test beds.

Artificial Potential Function (APF)-based guidance methods have the advantage of guaranteeing real-time on-board execution and collision-free RPO through there closed-form control law [19], [20], [27], [28]. Additionally, onboard passive or active sensors can be used for relative navigation, thus allowing the APF guidance method to achieve real-time obstacle avoidance. This method has been successfully implemented for autonomous RPO in either simulated or experimental environments [29]–[33]. The performance of the APF method is not guaranteed since there is no explicit minimization of a cost function, such as fuel consumption or maneuvering time.

To improve the performance of APF, a few approaches have been proposed [19], [20], [30], [34]. One such approach, the Adaptive Artificial Potential Function (AAPF) method, improves the optimality (with respect to a given cost functional) of the APF method by adapting the attractive potential shaping matrix to maintain a reference trajectory through the use of an adaptive update law [19], [20]. Note, the adaptive component of the AAPF method makes this a hybrid approach, as illustrated by Figure 1.2. The AAPF method was shown by Muñoz and Fitz-Coy to achieve improved consistency in maneuvering time and reduced propellant usage compared to the APF method via Monte Carlo analysis [19], [20]. However, all APF-based methods suffer from the existence of (stable) local minima which may trap a spacecraft and prevent it from completing the maneuver [20], [26]–[30], [34], [35].

To overcome the local minima issue, the use of potential fields based on analytic harmonic functions was explored by Akishita et al. Harmonic Potential Functions (HPFs) use analytical functions that have strong ties to irrotational and inviscid fluid flow; however, technically speaking, they do not completely eliminate the presence of local minima [36]. Any local minima present in the resulting potential field are unstable (i.e., saddle points) [36]–[38]. However, one drawback of HPFs is the difficultly associated with properly scaling the potential field elements to reduce numerical noise in the gradient computation [39], [40]. Despite this difficultly, HPFs can be utilized to generate a “Navigation Function,” as defined by Rimon and Koditschek [41], due to their construction and analytical formulation. HPF-based methods have been primarily utilized by autonomous kinematically-controlled and driftless[1] vehicles [43].

[1] A kinematically-controlled vehicle is one whose control input is its velocity. A driftless vehicle is a vehicle that "can be stopped instantaneously by setting the control input to zero" [42] – that is, “momentumless systems”. Examples of kinematically-controlled and driftless vehicles include autonomous vehicles [38], [43], [44] as well as robotic manipulators under kinematic control [45]–[48].

Unlike the previous RT-G&C algorithms, Inverse Dynamics in the Virtual Domain (IDVD) and Model Predictive Control (MPC) can be considered to be Computational Guidance & Control (CG&C) methods [49] as they attempt to solve the G&C task through an iterative, model-based approach. The dividing difference between RT-G&C and CG&C is that the latter has the implication of the method being iteration-based, or has otherwise been termed as *algorithmic control* [50] With this being the defining difference, CG&C algorithms can be considered a subset of RT-G&C algorithms. The IDVD has been experimentally evaluated for several tasks, including autonomous rendezvous and docking (AR&D) and docking with a rotating target [51]–[53]. The IDVD method is a quasi-optimal guidance method that is predicated on parameterizing both the trajectory and time as basis polynomials with a set number of coefficients [54]. The resulting polynomial coefficients are then obtained via a nonlinear programming optimization routine which minimizes the control effort while satisfying system and path constraints. Due to these approximations, the IDVD optimization method can be utilized as a feedback controller [52]. In contrast, the MPC framework solves a similar optimization problem, but over a limited horizon [55]. This receding horizon control approach allows for the efficient computation of the optimal solution to the constrained OCP. Additionally, the MPC framework allows for various constraint handling approaches, which can result in either a quadratic programming or nonlinear programming problem [52], [56]. The MPC approach has been widely applied to real-time spacecraft RPO and rendezvous and docking with both a rotating and non-rotating target [56]–[63].

Sampling-based methods such as Rapidly-Exploring Random Tree (RRT)-based methods (described in more detail in Section 6.4.1) offer a hybrid solution which leverages the computational simplicity of analytical methods, such APF and AAPF, while achieving near-optimal performance similar to optimization-based methods, such as IDVD and MPC [64], [65]. These methods have been widely used in autonomous kinematically-controlled and driftless vehicles, mainly operating in a static environment [44], [64], [66]–[68]. Due to their construction, sampling-based methods have the inherent advantage of efficiently exploring the configuration and arriving at a *feasible* solution relatively quickly [64], [67], [68].

Lastly, given the vast number of RT-G&C and emerging number of CG&C algorithms available in literature, the challenge arises of determining which method is “best” for a given application. Specifically, which algorithm enables the system to complete its objectives in an efficient manner while simultaneously meeting system-specific requirements (e.g., collision avoidance, computational deadlines). Additionally, when reporting on a new RT-G&C method, the majority of measurements presented in literature (e.g., control effort, maneuvering time, computational time) are implementation dependent. These system-specific measurements make it difficult to compare algorithms found in literature as the underlying system is fundamentally different (i.e., different physical and

actuator properties). Thus, it is apparent that the results be combined to form a metric that is agnostic to any specific implementation, denoted here as a “global" metric. The resulting global metric can then be used to aid in the selection of candidate algorithms to use for a particular application. To the best knowledge of this author, no such global metric that characterizes RT-G&C methods has been proposed in literature

1.3 Research Objectives

Motivated by the prevalence of RT-G&C algorithms available in literature and the future needs of space missions, the main focus of this dissertation is to further the experimental evaluation methodology for spacecraft guidance and control architectures. This is achieved through the following research objectives:

1. Advancement of a world-class, ground-based, system-level test bed with guaranteed real-time capabilities, rigorous characterization, and intuitive software architecture for the experimental evaluation and validation of guidance and control architectures

2. Development of a general purpose evaluation framework to compare guidance and control methods

3. Experimental evaluation of the proposed guidance and control method with a comparison to those available in literature.

4. Development of a hybrid guidance and control method based on the Rapidly-Exploring Random Tree Star (RRT*) framework that leverages the computational simplicity of analytical methods with the optimality of optimization-based methods.

The research objectives are cross-referenced with the their supporting chapters in Table 1.2.

1.4 Dissertation Contributions

The work presented in this dissertation focuses on the advancement of the state of the art in the experimental evaluation of Real-Time Guidance & Control (RT-G&C) and Computational Guidance & Control (CG&C) methods. This work is fundamental to the emerging field of CG&C algorithms as it provides the necessary methodology and tools to evaluate their performance. The contributions are summarized as follows:

Table 1.2. Mapping of the research objectives to the supporting chapters.

Research Objective	Dissertation Chapter				
	3	4	5	6	7
Advancement of a world-class, ground-based, system-level test bed with guaranteed real-time capabilities, rigorous characterization, and intuitive software architecture for the experimental evaluation and validation of guidance and control architectures	✓	✓	– –	✓	– –
Development of an general purpose evaluation framework to compare guidance and control methods	– –	– –	✓	✓	✓
Experimental evaluation of the proposed guidance and control method with a comparison to those available in literature.	– –	– –	– –	✓	✓
Development of a hybrid guidance and control method based on the RRT* framework that leverages the computational simplicity of analytical methods with the optimality of optimization-based methods.	– –	✓	– –	✓	– –

- Extensive characterization of the NPS POSEIDYN testbed, a world-class, ground-based, system-level GNC test bed.

- Development and implementation of the TRIDENT-GNC software architecture, enabling research into the emerging Computational Guidance & Control field.

- Development of a standardized evaluation framework to evaluate autonomous guidance and control (G&C) algorithms in an equitable manner.

- Development and validation of a global Guidance Comparison metric that simultaneously captures the algorithm performance and computational complexity of G&C algorithms.

- Experimental evaluation of the Sigma-Delta modulator as a thruster modulation technique.

- Application of a proposed RRT-based algorithm as a feedback controller for the translational G&C of a spacecraft in a dynamic environment.

- Development and validation of a metric to estimate the minimum sampling rate of a control system.

- Development and formulation of a G&C trade space for spacecraft design and GNC selection.

The contributions of this dissertation are mapped to the research objectives in Table 1.3.

Table 1.3. Summary of contributions mapped to research objectives.

Contribution	Research Objective			
	1	2	3	4
Extensive characterization of the NPS POSEIDYN test bed	✓	– –	✓	– –
Development and implementation of the TRIDENT-GNC software architecture, enabling research into the emerging Computational Guidance & Control field	✓	– –	✓	– –
Development of a standardized evaluation framework to evaluate autonomous G&C algorithms in an equitable manner	– –	✓	✓	– –
Development and validation of a global Guidance Comparison metric which captures the performance and computational complexity of algorithms	– –	✓	✓	– –
Experimental evaluation of the Sigma-Delta modulator as a thruster modulation technique	– –	– –	– –	✓
Application of a RRT-based algorithm as a feedback controller for the G&C of a spacecraft in a dynamic environment	– –	– –	✓	✓
Development and validation of a metric to estimate the minimum sampling rate	– –	✓	– –	✓
Development and formulation of a G&C trade space for spacecraft design and GNC selection	– –	✓	✓	– –

1.5 Dissertation Overview

This dissertation is constructed in such an order as to produce a logical flow. After the dynamics, constraints, and performance measures associated with a relevant scenario are presented, the dissertation details the specifics of the NPS POSEIDYN test bed and work performed to increase its robustness, reliability, and repeatability. After characterizing the test bed and the simulated spacecraft (i.e., test vehicle), the challenge associated of efficiently modulating the commanded control signal into a sequence of binary pulses to control the simulated spacecraft's thruster is explored. Next, a framework is presented which provides an equitable methodology to compare various RT-G&C

methods. Several guidance and control methods, including the proposed RRT-based method, are then detailed, evaluated, and compared using the proposed evaluation framework. The dissertation is then concludes in the traditional manner with a summary of contributions and recommendations for future work.

A detailed overview of the remaining chapters is as follows:

- **Chapter 2** introduces the reference scenario, the associated dynamics, relevant system and path constraints, as well as the key (optimal) performance measurements. These dynamics, path constraints, and performance measurements will be utilized in the development and testing of RT-G&C algorithms.

- **Chapter 3** presents an overview and characterization of the test bed which will be utilized to conduct the various experimental campaigns throughout this dissertation. The motivation for ground-based test beds is presented, followed by a survey of comparable test beds. A detailed description of the hardware and software architectures of the test bed is then presented. Next, the operating system latencies, sensor noise, thruster performance, physical properties, and end-to-end residual accelerations are characterized. Lastly, a case study is performed in order to demonstrate the capabilities of the test bed.

- **Chapter 4** investigates and experimentally evaluates and compares a new thruster modulation technique, the Sigma-Delta Modulator (which was briefly considered by Ciarcià et al. [69]), to the Pulse-Width Modulator. Linearized models of each modulator are presented. Next, these models, in combination with a simple PD control law, are used to develop candidate close-loop models. An experimental campaign is performed using the POSEIDYN test bed, comparing the two modulators in a relevant environment with hardware-derived noise and uncertainties. Finally, the results from the experiments, along with various numerical simulations, are analyzed to draw conclusions regarding the applicability of the candidate models as well as the Sigma-Delta Modulator's merit as a thruster modulation technique.

- **Chapter 5** establishes a proposed experimental framework in which to test and (partially) validate candidate guidance and control methods. This framework consists of several relevant test scenarios which attempt to extract various performance characteristics and qualities of the candidate algorithm. Next, a unique global comparison metric is proposed that captures key performance characteristics and allows them to be compared simultaneously.

- **Chapter 6** compares the proposed RRT-based feedback guidance and control technique against several other RT-G&C found in literature ranging from analytical to optimization-based methods. First, each algorithm being compared is introduced and summarized. Next, a detailed overview of sampling-based methods is presented, followed by a detailed development of the proposed guidance method. The individual results from the experimental campaign are presented. Finally, the proposed CG&C method is compared against the other RT-G&C methods evaluated.

- **Chapter 7** provides a case study, using the results from Chapter 6, illustrating the use of the Guidance Comparison metric in selecting a G&C method for a notional mission. To setup the guidance and control trade space, the Guidance Comparison metric is first bounded (in terms of control effort) using parameters typically available early in the design process. Next, time-related bounds are determined. In particular, a metric predicated on first-principles is introduced to estimate the minimum sampling rate required for a given guidance or control task.

- **Chapter 8** concludes the dissertation with a summary of the work performed, academic contributions, as well as areas for future work.

1.6 Publication History

A complete listing of *all* conference and journal publications, patents, as well as relevant supplemental material generated are listed in Appendix C. Table 1.4 cross-references the associated chapter or section to the relevant publication from which content was used.

Table 1.4. Summary of relevant publications utilized throughout the dissertation.

Reference Number	Chapter/Section	Journal/Conference	Status
[31]	Chapter 3	AIAA JSR[1]	Published
[33]	Chapter 3	2016 AIAA/AAS ASC[2]	Published
[70]	Chapter 4	AIAA JGCD[3]	Accepted
[71]	Section 5.2 Section 6.3.1 Section 6.7	IEEE TCST[4]	Under Review
[32]	Section 6.3.1	AAS/AIAA[5]	Published
[52]	Section 6.7	(2016) 6th ICATT[6]	Published
[72]	Section 6.7	CEAS Space Journal	Published
[61]	Section 6.7	AIAA/AAS ASC[2]	Published
[73]	Section 6.4 Section 6.4.2	2017 AAS/AIAA SFM[5]	To Be Published
[74]	Section 7.3.1 Appendix A	2017 AAS/AIAA SFM[5]	To Be Published
[75]	Chapter 5 Chapter 7	AIAA JGCD[3]	To Be Submitted

[1] *Journal of Spacecraft and Rockets*
[2] Astrodynamics Specialist Conference
[3] *Journal of Guidance, Control, and Dynamics*
[4] *Transactions on Control Systems Technology*
[5] Spaceflight Mechanics Conference
[6] International Conference on Astrodynamics Tools & Techniques

CHAPTER 2:
Constrained Dynamic System Development

A problem well put is a problem half solved.

—Anonymous
as quoted by Donald E. Kirk
Optimal Control Theory [76]

2.1 Reference Scenario

The necessity to perform autonomous rendezvous and docking (AR&D) to both a cooperative and uncooperative resident space object (RSO) is a fundamental requirement for many future missions [3], [4], [46]. Currently, AR&D is routinely performed between the ISS using vehicles such as the European Space Agency's ATV and the Russian Soyuz and Progress vehicles. However, vehicles such as the Japan Aerospace Exploration Agency's H-II Transfer Vehicle (HTV) and SpaceX's Dragon capsule are berthed to the ISS after autonomously rendezvousing. The reference scenario considered is a terminal rendezvous and docking with the ISS with the parameters listed in Table 2.1. The remainder of the chapter will detail the dynamical environment and the associated equations of motion, relevant system and path constraints associated with AR&D, as well as various (optimal) performance measures.

Table 2.1. Relevant orbital parameters for the reference rendezvous and docking scenario Adapted from [77].

Parameter	Value
Orbital Altitude	400 km
Eccentricity	0.00169
Inclination	51.950 deg
Mean Motion	0.00113 rad /s

2.2 Dynamical Environment

2.2.1 Translational Dynamics

The relative translational dynamics between two spacecraft, referred to as the Target and Chaser (or Chief and Deputy in other literature) spacecraft (denoted in Figure 2.1),

is described by the Hill-Clohessy-Wiltshire (HCW) framework. In this framework, it is assumed the Target spacecraft has a near-circular orbit. The dextral orthonormal basis for the HCW framework is referred to as the RSW frame or Local-Vertical Local-Horizontal (LVLH) frame in literature whose axes are anchored at the Target's center of mass (COM) and are defined as follows [78]:

$$\hat{\mathbf{X}} := \frac{\mathbf{r}_{\text{target}}}{\|\mathbf{r}_{\text{target}}\|_2} \tag{2.1a}$$

$$\hat{\mathbf{Z}} := \frac{\mathbf{r}_{\text{target}} \times \mathbf{v}_{\text{target}}}{\|\mathbf{r}_{\text{target}} \times \mathbf{v}_{\text{target}}\|_2} \tag{2.1b}$$

$$\hat{\mathbf{Y}} := \frac{\hat{\mathbf{Z}} \times \hat{\mathbf{X}}}{\left\|\hat{\mathbf{Z}} \times \hat{\mathbf{X}}\right\|_2} \tag{2.1c}$$

where $\mathbf{r}_{\text{target}}$ and $\mathbf{v}_{\text{target}}$ is the respective position and velocity of the Target spacecraft in the Earth-Centered Inertial (ECI) frame, defined by the unit vectors $(\hat{\mathbf{I}}, \hat{\mathbf{J}}, \hat{\mathbf{K}})$in Figure 2.1. Note, the unit vectors $(\hat{\mathbf{I}}, \hat{\mathbf{J}}, \hat{\mathbf{K}})$ of the Earth-Centered Inertial (ECI) frame are anchored at the center of the Earth and are defined as follows: the $\hat{\mathbf{I}}$ axis is in the direction of the vernal equinox (Υ); the $\hat{\mathbf{K}}$ axis extends from the origin toward the north pole; the $\hat{\mathbf{J}}$ axis completes the dextral orthonormal basis [79].

The relative translational dynamics between the Chaser and Target using the HCW framework becomes a function of the Target spacecraft's mean motion, n, and are given as [78],

$$\ddot{x} = f_x + 2n\dot{y} + 2n^2 x \tag{2.2a}$$

$$\ddot{y} = f_y - 2n\dot{x} \tag{2.2b}$$

$$\ddot{z} = f_z - n^2 z \tag{2.2c}$$

$$n = \sqrt{\frac{\mu_\oplus}{a^3}} \tag{2.2d}$$

where f_x, f_y, f_z are forces acting on the spacecraft in the x, y, and z directions respectively; $\mu_\oplus$ is the gravitational parameter of Earth and a is the semi-major axis of the Target spacecraft's orbit.

During the terminal rendezvous and docking phase, the Chaser is in close proximity of the Target with small relative velocities and the resulting motion is assumed to be planar. Additionally, over short durations of time relative to the orbital period of the Target, the resulting dynamics can be sufficiently approximated as double integrator

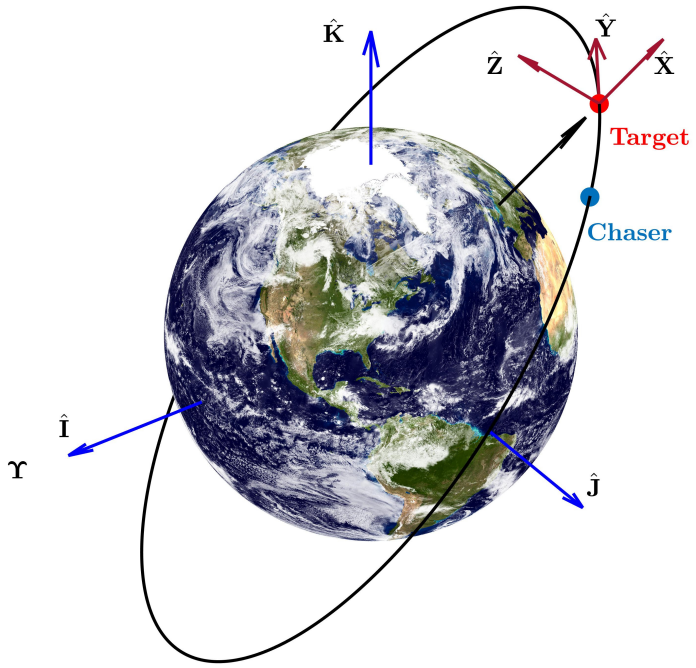


Figure 2.1. Illustration of the RSW frame for a Target spacecraft in a near-circular orbit with the Chaser in close-proximity. Adapted from [80].

dynamics [81]:

$$\ddot{x} = f_x \tag{2.3a}$$
$$\ddot{y} = f_y \tag{2.3b}$$

2.2.2 Rotational Dynamics

The attitude dynamics of the Chaser follow Euler's Equation [82],

$$\boldsymbol{\tau} = \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega}^{\times}\mathbf{J}\boldsymbol{\omega} \tag{2.4}$$

where $\boldsymbol{\tau} \in \mathbb{R}^3$ is the net torque acting on the spacecraft; $\mathbf{J} \in \mathbb{R}^{3\times3}$ is the inertia matrix with respect to the COM in the spacecraft body Cartesian coordinate system; $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity of the spacecraft body frame with respect to the inertial frame expressed in the spacecraft body Cartesian coordinate system; and $\mathbf{a}^{\times}$ is the matrical

representation of the cross-product operator, defined as,

$$\mathbf{a}^{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{2.5}$$

When the inertia is measured in the Principal Axis Cartesian coordinate system, the inertia matrix becomes diagonal. For small angular velocities, the gyroscopic term becomes negligible and Equation (2.4) becomes

$$\boldsymbol{\tau} = \mathbf{J}\dot{\boldsymbol{\omega}} \tag{2.6}$$

This implies the motion about each axis is decoupled and can be controlled independently. Furthermore, assuming no external disturbances, $\boldsymbol{\tau}$ becomes the control torque.

For the planar translational motion assumed in Equation (2.3), it is only necessary to consider the rotational dynamics about the z axis. Therefore, the resulting rotational dynamics can be simplified to,

$$\tau_z = J_{zz}\omega_z \tag{2.7}$$

where J_{zz} is the inertial about the z-axis in the spacecraft body frame and ω_z is the angular velocity about the z-axis of the spacecraft body frame with respect to the inertial frame.

2.3 Constraints

The constraints on the Chaser spacecraft can be broken up into two main categories: dynamical system constraints and path constraints. Constraints on a dynamical system (i.e., system constraints) are limitations imposed on the evolution of the system. These limitations are typically manifested through actuator saturation limits, such as the minimum impulse and maximum force a thruster can generate or the maximum torque generated by a momentum exchange device [83]. Path constraints, on the other hand, are typically used to describe regions of the state space that are considered admissible.

2.3.1 System Constraints

The system constraints considered herein are on the admissible controls. The set of all admissible controls, $\mathbb{U}$, considered herein is defined as the set of all controls in the inertial frame, ${}^{\mathcal{I}}\mathbf{u}(t) \in \mathbb{R}^{N_u}$ (where N_u is the number of control inputs) such that the control input in the spacecraft body-fixed frame, ${}^{B}\mathbf{u}(t)$, is bounded by some maximally negative control input $\mathbf{u}^{-}$ and some maximally positive control input $\mathbf{u}^{+}$.

This admissible set can be written as

$$\mathbb{U} := \left\{ {}^{\mathcal{I}}\mathbf{u}(t) \in \mathbb{R}^{N_u} \,|\, \mathbf{u}^- \leq {}^{\mathcal{B}}\mathbf{u}(t) \leq \mathbf{u}^+ \right\} \tag{2.8}$$

2.3.2 Path Constraints

The path constraints considered can be further broken up into two sub-categories: a keep-out zone which encompasses an obstacle and a docking cone corridor. These two types of path constraints are illustrated in one possible configuration in Figure 2.2.

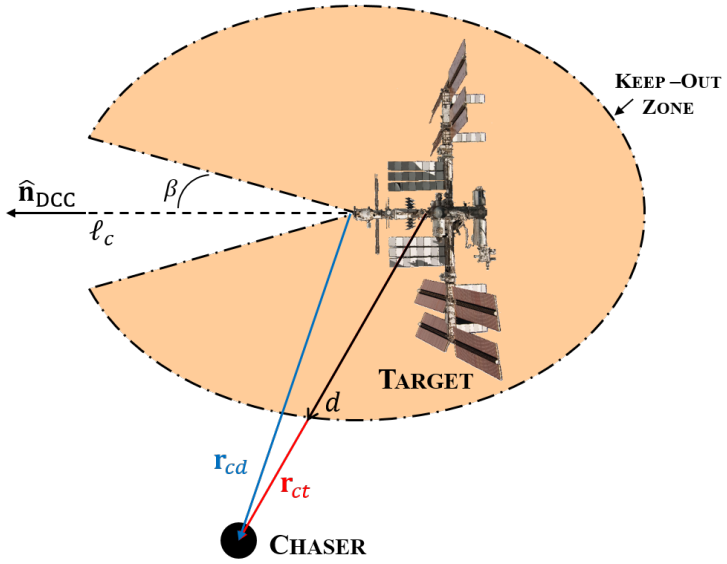


Figure 2.2. Notional illustration of a keep-out zone combined with the docking cone corridor constraint for the rendezvous and docking scenario. Adapted from [84].

The first constraint, the obstacle keep-out zone, encompasses the geometry of the object and defines an inadmissible portion of the configuration space. In order to satisfy the constraint, the Chaser COM must remain outside of the obstacle keep-out zone. The keep-out zone constraint can be formally stated as

$$\|\mathbf{r}_{ci}\|_2 \geq d_i\left(\mathbf{r}_{ci}\right) \tag{2.9}$$

where $\mathbf{r}_{ci}$ is the relative position of the Chaser COM with respect to the COM if the i^{th} obstacle and d_i is the distance from the COM of the i^{th} obstacle to the boundary of its keep-out zone along the line $\mathbf{r}_{ci}$. Note, as indicated by Equation (2.9), d_i is *not* constant and allowed to vary as a function of the relative position of the Chaser with respect to the obstacle. The only requirement placed on d_i is that the surface it describes is

continuous. One such instantiation of a keep-out zone is illustrated Figure 2.2. Other examples of keep-out zones are illustrated in [19], [32], [85], [86].

The second type of path constraint considered is a docking cone corridor (DCC). This path constraint ensures a safe, collision-free final trajectory to the Target in addition to ensuring the docking action occurs parallel to the docking axis [28], [87]. The DCC is anchored at the docking interface of the Target spacecraft and is defined by three parameters: the docking axis unit vector, $\hat{\mathbf{n}}_{\text{DCC}}$, the docking cone half-angle, β, and the length of the corridor, ℓ_c. The DCC path constraint is given as

$$\|\mathbf{r}_{cd}\|_2 \cos(\beta) \leq \hat{\mathbf{n}}_{\text{DCC}}^{\mathsf{T}} \mathbf{r}_{cd} \tag{2.10}$$

where $\mathbf{r}_{cd}$ is the relative position of the Chaser with respect to the Target docking interface. Unlike other constraints, the DCC constraint is only active when the distance between the Chaser COM and the Target docking interface is less than or equal to ℓ_c.

2.4 Optimal Control Problem Formulation

Optimal control is predicated on determining the control input $\mathbf{u} \in \mathbb{U} \subseteq \mathbb{R}^{N_u}$ necessary to transfer a system $\mathbf{x} \in \mathbb{R}^{N_x}$ from an initial state to a desired final state that minimizes a specified cost functional [88]. Through the application of the Calculus of Variations, the necessary conditions can be derived through the introduction of the costates [76], [88]. However, the dimensionality of the resulting optimization problem is increased by a factor of two from N_x to $2N_x$. Notably, this increase in dimensionality is commonly referred to as the *Curse of Dimensionality* [76], [88]. While the increased dimensionality is not necessarily an issue for modern (terrestrial) computational technology, implementation onboard a spacecraft provides unique challenges due to the computational constraints of space-based compute elements. In the remainder of this section, the general optimal control problem will be developed, along with two common optimal performance measures: the minimum-propellant and minimum-time optimal control problems.

The general optimal control problem can be succinctly stated as the following [76], [88], [89]:

$$\text{Minimize: } J(\cdot) = h(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t)\, dt \tag{2.11a}$$

$$\begin{aligned}
\text{Subject To: } \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) && \text{(2.11b)}\\
\mathbf{x}(t_0) &= \mathbf{x}^0 && \text{(2.11c)}\\
\mathbf{c}_p(\mathbf{x}(t), \mathbf{u}(t), t) &\leq 0 && \text{(2.11d)}\\
\mathbf{e}_f(\mathbf{x}_f, t_f) &= \mathbf{0} && \text{(2.11e)}
\end{aligned}$$

where the cost functional $J(\cdot) : \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \mapsto \mathbb{R}$ is composed of the endpoint cost $h(\cdot) : \mathbb{R}^{N_x} \times \mathbb{R} \mapsto \mathbb{R}$ and the running cost $g(\cdot) : \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \times \mathbb{R} \mapsto \mathbb{R}$. The system dynamics $\dot{\mathbf{x}}$ of the system is given by $\mathbf{f} : \mathbb{R}^{N_x} \times \mathbb{R}^{N_u}\mathbb{R} \mapsto \mathbb{R}^{N_x}$ with an initial condition $\mathbf{x}^0$ at time t_0 and terminal condition $\mathbf{e}_f : \mathbb{R}^{N_x} \times \mathbb{R} \mapsto \mathbb{R}^{N_e}$. Lastly, any constraints, such as path and/or actuator constraints, are given as $\mathbf{c}_p : \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \times \mathbb{R} \mapsto \mathbb{R}^{N_c}$.

Using the Calculus of Variations, the necessary conditions for optimality for a system with bounded control inputs, $\mathbf{u} \in \mathbb{U}$, can be derived. To begin, the cost functional given in Equation (2.11a) can be rewritten as an augmented cost funcitonal $J_a(\cdot)$ to include differential constraints [76],

$$\begin{aligned} J_a(\cdot) &= \int_{t_0}^{t_f} \left(g(\cdot) + \left[\frac{\partial h}{\partial t}(\cdot)\right]^{\mathsf{T}} \dot{\mathbf{x}}(t) + \frac{\partial h}{\partial t}(\cdot) + \boldsymbol{\lambda}^{\mathsf{T}}(\mathbf{f}(\cdot) - \dot{\mathbf{x}}(t)) \right) dt \\ &= \int_{t_0}^{t_f} g_a(\cdot)\, dt \end{aligned} \tag{2.12}$$

where $\boldsymbol{\lambda}(t) \in \mathbb{R}^{N_x}$ are the costates (which are also referred to as Lagrange Multipliers in some literature). Defining the Hamiltonian as [76], [88],

$$\mathscr{H}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) = g(\cdot) + \boldsymbol{\lambda}(\cdot)^{\mathsf{T}} f(\cdot) \tag{2.13}$$

the augmented integrand $g_a(\cdot)$ becomes

$$g_a(\cdot) = \mathscr{H}(\cdot) - \boldsymbol{\lambda}^T \dot{\mathbf{x}}(t) + \left[\frac{\partial h}{\partial t}(\cdot)\right]^{\mathsf{T}} \dot{\mathbf{x}}(t) + \frac{\partial h}{\partial t}(\cdot) \tag{2.14}$$

Invoking the Fundamental Theorem of the Calculus of Variations [76], the necessary condition under which the augmented cost functional, with the augmented integrand given by Equation (2.14), is minimized is given as [76]

$$\delta J_a(\mathbf{u}^*, \delta\mathbf{u}) \geq 0 \quad \forall \quad \mathbf{u} \in \mathbb{U} \tag{2.15}$$

where $\mathbf{u}^*$ is the optimal control[1] that minimizes the cost functional $J(\cdot)$. Note, for the case of an unbounded control, the inequality relation in Equation (2.15) becomes an equality relation

$$\delta J_a(\mathbf{u}^*, \delta\mathbf{u}) = 0 \quad \forall \quad \mathbf{u} \in \mathbb{U}$$

[1] Components associated with the optimal solution are denoted with an asterisk in the superscript $(\cdot)^*$

since the $\mathbf{u}^*$ lies within the control set bounded by $\pm\infty$ [76]. Evaluating the variation of the cost functional in Equation (2.15), the necessary conditions for optimality are [76],

$$\dot{\mathbf{x}}(t) = \frac{\partial \mathscr{H}}{\partial \boldsymbol{\lambda}}(\cdot) \tag{2.16a}$$

$$\dot{\boldsymbol{\lambda}}(t) = -\frac{\partial \mathscr{H}}{\partial \mathbf{x}}(\cdot) \tag{2.16b}$$

$$\mathscr{H}(\mathbf{x}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*, t^*) \leq \mathscr{H}(\mathbf{x}^*, \mathbf{u}, \boldsymbol{\lambda}^*, t^*) \tag{2.16c}$$

with the boundary condition given as

$$\left[\mathscr{H}(\mathbf{x}^*(t_f), \mathbf{u}^*(t_f), \boldsymbol{\lambda}^*(t_f), t_f) + \frac{\partial h}{\partial t}(\mathbf{x}^*(t_f), t_f)\right] \delta t_f + \left[\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}^*(t_f), t) - \boldsymbol{\lambda}^*(t_f)\right]^{\mathsf{T}} \delta \mathbf{x}_f = 0 \tag{2.17}$$

where δt_f is the variation of the final time and $\delta \mathbf{x}_f$ is the variation of the terminal state. It is important to note, Equation (2.16b) specifies the costate dynamics, Equation (2.16c) is known as Pontryagin's Minimum Principle, and when the boundary conditions are specified, Equation (2.17) reduces to the (terminal) transversality condition [76], [88].

2.4.1 Minimum-Propellant Optimal Control Problem

The minimum-propellant optimal control problem is arguably one of the most prevalent OCPs for RPO. As its moniker implies, the resulting solution from solving this OCP is the one that uses the least amount of fuel. Similar to the development of the generic optimal control problem, it is first assumed that the control input $\mathbf{u} \in \mathbb{U} \subseteq \mathbb{R}^{N_u}$ is bounded. The cost functional associated with the minimum-propellant optimal control problem is the L_1-norm of the control input, $\|\mathbf{u}(t)\|_1$ and can be defined as [76],

$$\|\mathbf{u}(t)\|_1 = \sum_{i=1}^{N_u} |u_i(t)| \tag{2.18}$$

The generic optimal control problem given in Equation (2.11) can be rewritten to form the minimum-propellant optimal control problem:

$$\text{Minimize: } J(\cdot) = \int_{t_0}^{t_f} \|\mathbf{u}(\mathbf{t})\|_1 \, dt \tag{2.19a}$$

$$\begin{aligned} \text{Subject To: } \dot{\mathbf{x}} &= \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) && \text{(2.19b)} \\ \mathbf{x}(t_0) &= \mathbf{x}^0 && \text{(2.19c)} \\ \mathbf{c}_p\left(\mathbf{x}(t), \mathbf{u}(t), t\right) &\leq \mathbf{0} && \text{(2.19d)} \\ \mathbf{e}_f\left(\mathbf{x}_f, t_f\right) &= \mathbf{0} && \text{(2.19e)} \end{aligned}$$

For the purpose of illustration, let the system dynamics be defined as [76],

$$\dot{\mathbf{x}} = \mathbf{a}(\mathbf{x}(t), t) + \mathbf{B}(\mathbf{x}(t), t)\mathbf{u}(t) \tag{2.20}$$

where

$$\mathbf{B}(\cdot) = [\mathbf{b}_1(\cdot), \mathbf{b}_2(\cdot), \ldots, \mathbf{b}_{N_m}(\cdot)] : \mathbb{R}^{N_x} \times \mathbb{R} \mapsto \mathbb{R}^{N_x \times N_u}$$

may depend on the states and time.[1] Note, the assumption of the system dynamics being affine in control is common for dynamical systems [76], [90]. The resulting Hamiltonian (using Equation (2.13)) associated with the minimum-propellant optimal control problem is

$$\mathscr{H}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) = \|\mathbf{u}(t)\|_1 + \boldsymbol{\lambda}(t)^{\mathsf{T}}\mathbf{a}(\mathbf{x}(t), t) + \boldsymbol{\lambda}(t)^{\mathsf{T}}\mathbf{b}(\mathbf{x}(t), t)\mathbf{u}(t) \tag{2.21}$$

Applying Pontryagin's Minimum Principle (given by Equation (2.16c)) to the minimum-propellant OCP yields

$$\begin{aligned} \|\mathbf{u}^*(t)\|_1 + \boldsymbol{\lambda}^*(t)^{\mathsf{T}}\mathbf{a}(\mathbf{x}^*(t), t) + \boldsymbol{\lambda}^*(t)^{\mathsf{T}}\mathbf{B}(\mathbf{x}^*(t), t)\mathbf{u}^*(t) \leq \\ \|\mathbf{u}(t)\|_1 + \boldsymbol{\lambda}^*(t)^{\mathsf{T}}\mathbf{a}(\mathbf{x}^*(t), t) + \boldsymbol{\lambda}^*(t)^{\mathsf{T}}\mathbf{B}(\mathbf{x}^*(t), t)\mathbf{u}(t) \end{aligned} \tag{2.22}$$

Assuming $\mathbf{B}(\cdot)$ is diagonal (implying no interactions between the components of $\mathbf{u}$) Equation (2.22) can be simplified to [76],

$$\begin{aligned} |u_i^*(t)| + u_i^*(t)\boldsymbol{\lambda}^*(t)^{\mathsf{T}}\mathbf{b_i}(\mathbf{x}^*(t), t) \\ \leq |u_i(t)| + u_i(t)\boldsymbol{\lambda}^*(t)^{\mathsf{T}}\mathbf{b_i}(\mathbf{x}^*(t), t) \ , \qquad i = 1, 2, \ldots, N_u \end{aligned} \tag{2.23}$$

[1] When $\mathbf{B}(\cdot)$ is not dependent upon any state or time, it reduces to the control input matrix traditionally associated with linear time-invariant systems.

The conditions under which the Hamiltonian is minimized (and Equation (2.23) is satisfied) can be ascertained by expanding the right-hand side of Equation (2.23) [76]:

$$|u_i(t)| + u_i(t)\boldsymbol{\lambda}^*(t)^\mathsf{T}\mathbf{b_i}(\mathbf{x}^*(t),t) = \begin{cases} \left[1+\boldsymbol{\lambda}^*(t)^\mathsf{T}\mathbf{b_i}(\mathbf{x}^*(t),t)\right]u_i(t) & \text{for } u_i(t)\geq 0 \\ \left[-1+\boldsymbol{\lambda}^*(t)^\mathsf{T}\mathbf{b_i}(\mathbf{x}^*(t),t)\right]u_i(t) & \text{for } u_i(t)\leq 0 \end{cases} \tag{2.24}$$

By inspection, the optimal control law can be found to be [76]:

$$u_i^*(t) = \begin{cases} u_i^+ & \text{for} \quad \boldsymbol{\lambda}^*(t)^\mathsf{T}\mathbf{b}_i(\mathbf{x}^*(t),t) < -1 \\ u_i^- & \text{for} \quad \boldsymbol{\lambda}^*(t)^\mathsf{T}\mathbf{b}_i(\mathbf{x}^*(t),t) > 1 \\ 0 & \text{for} \quad \boldsymbol{\lambda}^*(t)^\mathsf{T}\mathbf{b}_i(\mathbf{x}^*(t),t) \in (-1,1) \\ \text{undetermined} & \text{for} \quad \boldsymbol{\lambda}^*(t)^\mathsf{T}\mathbf{b}_i(\mathbf{x}^*(t),t) = \pm 1 \end{cases} \tag{2.25}$$

where $\bar{\mathbb{R}}_+$ is a non-negative number (i.e., $\mathbb{R} \in [0,\infty)$) and $\mathbb{R}_{<0}$ is a strictly negative number (i.e., $\mathbb{R} \in (-\infty, 0)$). It is worthwhile to note, an interval is said to be singular when $\boldsymbol{\lambda}^*(t)^\mathsf{T}\mathbf{b}(\mathbf{x}^*(t),t) = \pm 1$ for a nonzero duration of time; whereas a crossing of ± 1 merely indicates a control switch from $u^- \to u^+$ or vice versa [76]. The resulting control action specified by the optimal control policy in Equation (2.25) is termed *bang-bang* or *bang-off-bang* control, assuming no singular intervals.

2.4.2 Minimum-Time Optimal Control Problem

Given a bounded control input $\mathbf{u} \in \mathbb{U} \subseteq \mathbb{R}^{N_u}$, the minimum-time optimal control problem is given as:

$$\text{Minimize: } J(\cdot) = \int_{t_0}^{t_f} 1\, dt \tag{2.26a}$$

$$\begin{aligned} \text{Subject To: } \dot{\mathbf{x}} &= \mathbf{f}\left(\mathbf{x}(t),\mathbf{u}(t),t\right) && \text{(2.26b)} \\ \mathbf{x}(t_0) &= \mathbf{x}^0 && \text{(2.26c)} \\ \mathbf{c}_p\left(\mathbf{x}(t),\mathbf{u}(t),t\right) &\leq \mathbf{0} && \text{(2.26d)} \\ \mathbf{e}_f\left(\mathbf{x}_f,t_f\right) &= \mathbf{0} && \text{(2.26e)} \end{aligned}$$

Again, for the purpose of illustration, let the system dynamics $\dot{\mathbf{x}}$ be

$$\dot{\mathbf{x}} = \mathbf{a}(\mathbf{x}(t),t) + \mathbf{B}(\mathbf{x}(t),t)\mathbf{u}(t) \tag{2.27}$$

Note, unlike for the minimum-propellant OCP, the minimum-time OCP requires the final time be free. From Equation (2.13), the Hamiltonian is associated with the minimum-

time OCP is

$$\mathscr{H}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) = 1 + \boldsymbol{\lambda}(t)^{\mathsf{T}}\mathbf{a}(\mathbf{x}(t), t) + \boldsymbol{\lambda}(t)^{\mathsf{T}}\mathbf{b}(\mathbf{x}(t), t)\mathbf{u}(t) \tag{2.28}$$

Applying Pontryagin's Minimum Principle to the minimum-time OCP yields

$$1 + \boldsymbol{\lambda}^*(t)^{\mathsf{T}}\mathbf{a}(\mathbf{x}^*(t), t) + \boldsymbol{\lambda}^*(t)^{\mathsf{T}}\mathbf{B}(\mathbf{x}^*(t), t)\mathbf{u}^*(t) \leq \\ 1 + \boldsymbol{\lambda}^*(t)^{\mathsf{T}}\mathbf{a}(\mathbf{x}^*(t), t) + \boldsymbol{\lambda}^*(t)^{\mathsf{T}}\mathbf{B}(\mathbf{x}^*(t), t)\mathbf{u}(t)$$

which can be simplified to

$$u_i(t)\boldsymbol{\lambda}^*(t)^{\mathsf{T}}\mathbf{b_i}(\mathbf{x}^*(t), t) \;\leq\; u_i(t)\boldsymbol{\lambda}^*(t)^{\mathsf{T}}\mathbf{b_i}(\mathbf{x}^*(t), t) \qquad i = 1, 2, \ldots, N_u \tag{2.29}$$

Following a similar procedure to determine the minimum-propellant optimal control policy, the optimal control policy for the minimum-time problem is

$$u_i^*(t) = \begin{cases} u_i^+ & \text{for} \quad \boldsymbol{\lambda}^*(t)^{\mathsf{T}}\mathbf{b}_i(\mathbf{x}^*(t), t) < -0 \\ u_i^- & \text{for} \quad \boldsymbol{\lambda}^*(t)^{\mathsf{T}}\mathbf{b}_i(\mathbf{x}^*(t), t) > 0 \\ \text{undetermined} & \text{for } \boldsymbol{\lambda}^*(t)^{\mathsf{T}}\mathbf{b}_i(\mathbf{x}^*(t), t) = 0 \end{cases} \tag{2.30}$$

It is worthwhile to note, an interval is said to be singular when $\boldsymbol{\lambda}^*(t)^{\mathsf{T}}\mathbf{b}(\mathbf{x}^*(t), t) = 0$ for a nonzero duration of time; whereas a crossing of ± 1 merely indicates a control switch from $u^- \rightarrow u^+$ or vice versa [76].

It is worthwhile to highlight another property of time-free OCPs by examining the boundary conditions given in Equation (2.17). Since the terminal state is specified, the boundary conditions for minimum-time OCP is given as

$$\left[\mathscr{H}(\mathbf{x}^*(t_f), \mathbf{u}^*(t_f), \boldsymbol{\lambda}^*(t_f), t_f)\right] \delta t_f = 0 \tag{2.31}$$

Since $\delta t_f \neq 0$ since the final time is not specified, $\mathscr{H}(t_f) = 0$ in order to satisfy the boundary condition. Furthermore, since the Hamiltonian is not an explicit function of time, its time-derivative is zero which implies

$$\mathscr{H}(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t) = 0 \quad \forall \quad t \in [t_0, t_f] \tag{2.32}$$

This fact that the Hamiltonian is zero when evaluated on the extremal trajectory can be used in validating the optimal solution [76].

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
The POSEIDYN Test Bed and TRIDENT-GNC Software Architecture

> I constructed a laboratory in the neighborhood of Pike's Peak. The conditions in the pure air of the Colorado Mountains proved extremely favorable for my experiments, and the results were most gratifying to me.
>
> —Nikola Tesla
> "Talking with the planets" [91]

The work presented in this Chapter was originally presented at the American Institute of Aeronautics and Astronautics (AIAA)/American Astronautical Society (AAS) Astrodynamics Specialist Conference 2016 in Long Beach, CA (12–16 September 2016) [33]; a journal version of this work has been published as an article in advance in the AIAA *Journal of Spacecraft and Rockets* [31].

3.1 Motivation for Ground-Based Test Bed

Integration, verification, and validation are some of the greatest challenges which future autonomous GNC systems face [2], [3]. To ensure safe on-orbit operations for RPO, ground testing must be performed to ensure the resulting GNC algorithms meet predefined set of performance requirements. To perform this task, at the recommendation of Quadrelli et al., it is argued that ground-based, system-level, demonstration systems can provide adequate fidelity to verify and validate GNC performance requirements [4]. Air-bearing test beds provide a dynamically representative environment to develop and test GNC algorithms. In these facilities, the test vehicles that represent spacecraft (or resident space objects) operate on top of a planar surface. Air-bearings located on the test vehicles are used to reduce the friction of the vehicles to a create a quasi-frictionless environment. As the planar surface is horizontally leveled, the effects of any in-plane components of gravity on the test vehicles are minimal. The result is a quasi-frictionless and low residual acceleration environment in a plane, thus emulating the drag-free and weightless environment of orbital spaceflight. Additionally, these test beds include various hardware phenomena such as delays, computational constraints, actuator response uncertainty, and sensor noise, which is practically impossible to replicate in a simulated environment. Ground-based GNC test beds are a useful tool to advance the state-of-the-art of these systems and can also be used to

perform end-to-end system-level verification and validation prior to the system's operational deployment.

3.2 Survey of Relevant Test Beds

Although survey papers on air bearing spacecraft simulators for RPO exist in literature [48], [92], new facilities have emerged since these were published. A summary of the characteristics of the examined test beds not found in the survey literature is tabulated in Table 3.1.

Table 3.1. Summary of the relevant test beds characteristics. Adapted from [93]–[97].

Name	Location	Simulator Type	Surface Material	Navigation Sensors
ADAMUS	UF	Translational + Rotational Air-Beaing	Epoxy	PhaseSpace
ASTROS	Georgia Tech	Translational + Rotational Air-Bearing	Epoxy	Vicon SICK Laser sensor IMU
ORION	FIT	Translational + Rotational Air-Bearing	Epoxy	OptiTrack
PINOCCHIO	Sapienza	Translational Air-Bearing	Glass	On-board IMU
POSEIDYN	NPS	Translational Air-Bearing	Granite	Vicon FOG

Both the University of Florida (UF) ADAMUS test bed and the Florida Institute of Technology (FIT) ORION (with its DAWN vehicle) test bed are capable of the full six degree of freedom (DOF) motion [93]–[95]. In these test beds, planar air bearings are used by the test vehicles to translate on top of an epoxy floor [2-DOF] while a spherical air-bearing provides the rotational 3-DOF and a complex counter-balance system is used to provide frictionless out-of-plane motion. The Georgia Institute of Technology ASTROS utilizes a 5-DOF platform, where the ASTROS platform is a 2-DOF rotational air bearing atop a 3-DOF platform that employing translation air bearings [96]. Lastly, the University of Rome-La Sapienza PINOCCHIO test bed is a two-translation and one-rotational DOF test bed utilizing only a translational planar air-bearing [97].

With the exception of the PINOCCHIO test bed at University of Rome-La Sapienza, which is still in development, all the examined test beds utilize a commercial motion capture system, such as Vicon or PhaseSpace, in order to provide an inertial position to the spacecraft simulator. These external position measurements are usually augmented by an on-board inertial measurement unit (IMU) and fused together for more accurate state estimations.

Furthermore, the use of an epoxy surface is pervasive throughout the surveyed air-bearing test beds. Although granite surfaces provide unmatched surface planarity and smoothness, epoxy floors or glass surfaces have a significantly lower procurement cost, thus making them more common. Lastly, it is worthwhile to note that a detailed characterization of the test beds is typically not found in literature, thus hindering the comparison of the experimental results obtained when using different test beds. From an experimental stand point, having such a detailed description and characterization is important in order to understand the system's performance (e.g., navigation performance) and uncertainties (e.g., actuator performance, moment of inertia estimate, residual accelerations) under which a GNC algorithm has been experimentally evaluated.

3.3 Experimental Facility Overview: The POSEIDYN test bed

The Naval Postgraduate School (NPS) Proximity Operation of Spacecraft: Experimental hardware-In-the-loop DYNamic simulator (POSEIDYN) test bed has been developed with the aim to provide a dynamically representative system-level platform upon which to develop, experimentally test, and partially validate GNC algorithms for RPO. Before the installation of the 4-by-4 meter granite monolith in early 2012, the vehicles operated over an epoxy floor since the test bed was initially stood-up in 2004. Over the decade that the test bed has been in operation and in constant upgrade, four different generations of floating spacecraft simulators (FSSs) have been developed. The evolution of these test vehicles is shown in Figure 3.1. Each FSS generation has included a unique capability. The first generation FSS utilized a prototype capture system in order to perform rendezvous and docking [22]. The second generation FSSs featured vectorable thrusters and a miniature control moment gyroscope (CMG) [98], [99]. The third generation FSSs moved away from an aluminum construction in favor of a more lightweight, polycarbonate structure (fabricated using additive manufacturing) and components, such as docking interfaces. Lastly, the fourth-generation FSSs continued the use of polycarbonate components and includes a standardized interface for use with robotic manipulator research [47].

3.4 Hardware Architecture Overview

The POSEIDYN test bed is composed of three main elements: a 4×4 m granite monolith, multiple FSSs, and a laboratory-wide metrology system. An overview of these elements is shown in Fig. 3.2.

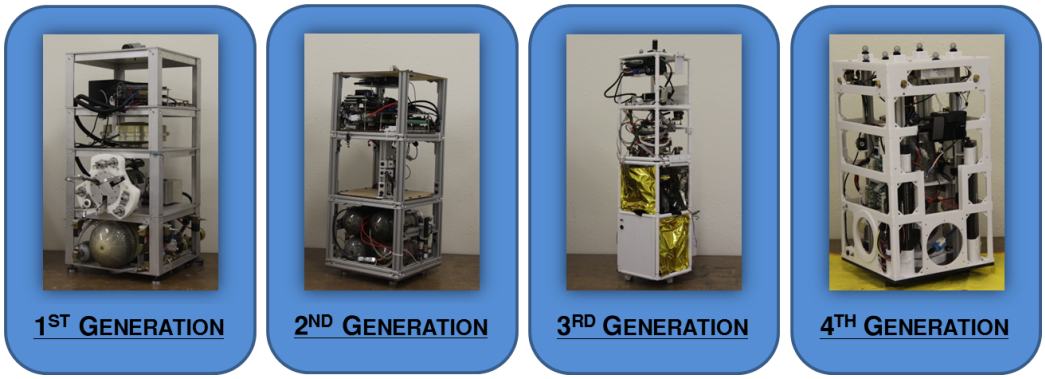


Figure 3.1. Lineup of the first- to fourth-generation FSSs used on the POSEIDYN test bed.

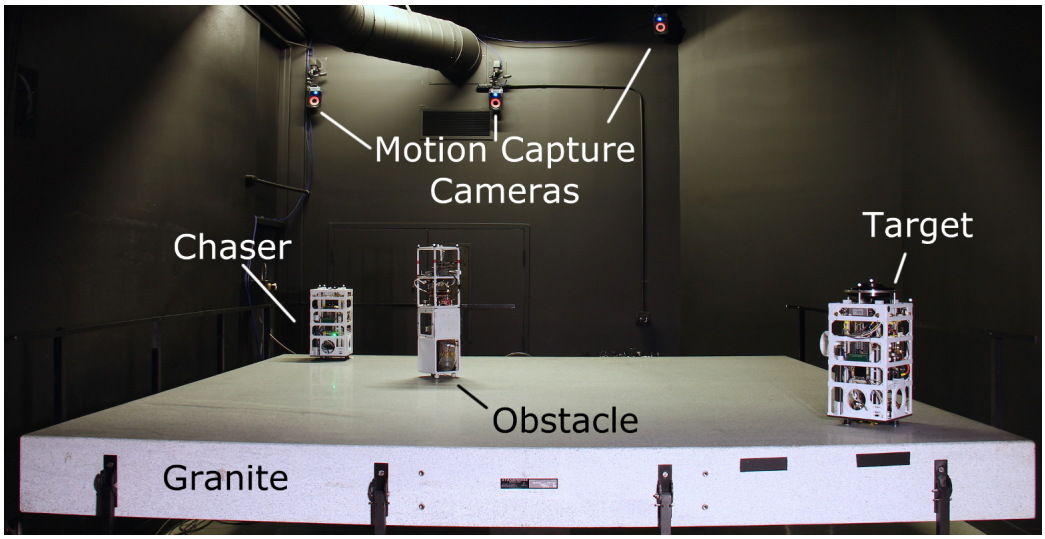


Figure 3.2. Overview of the main elements of the POSEIDYN test bed.

The FSSs are custom-designed vehicles that emulate orbital spacecraft moving in close proximity of another vehicle or object (e.g., another FSS). In this paper, the fourth-generation FSSs will be considered. Three flat 25 mm diameter air-bearings are used by the floating spacecraft simulator to achieve quasi-frictionless motion on top of the granite monolith. The air-bearings use compressed air to lift the FSSs approximately $5\,\mu\mathrm{m}$, creating an air film between the vehicle and the granite surface that acts as a lubricant layer and eliminates their direct contact. The compressed air, supplied at 4.1 bars (60 psi), is delivered from an onboard tank holding 1.87 L of compressed air at 206.8 bars (3000 psi). With a nominal air consumption of 0.1 L/min at 4.1 bars (0.53 normalized liters per minute) per air-bearing, the floating endurance of the FSS is estimated to be at approximately 3.5 h. A solenoid valve controls the flow toward the air-bearings and an air filter prevents contaminants and foreign materials from damaging

the delicate air bearing porous material. A scuba-diving-type compressor is used to refill the onboard tank.

The 15 metric ton, 4-by-4 meter granite monolith is supported by three adjustable pedestals and has a planar accuracy of ± 0.0127 mm. This allows the granite surface to be horizontally leveled to an accuracy of 0.01 deg. Seismic activity and other building structural activity can distort the monolith alignment over time. Periodic checks are performed to ensure that the granite monolith is leveled within the pre-specified tolerance.

The combination of the reduced friction (provided by the air-bearings) with the low residual acceleration (provided by the horizontality and planarity of the granite monolith) allows the POSEIDYN test bed to recreate the drag-free and weightless environment experienced by spacecraft in close proximity maneuvering. A characterization of the residual acceleration experienced by the FSS can be found in Section 3.6. An important limitation of the test bed consists in allowing only 3-DOF motion (two translation and one rotation) compared to 6-DOF motion of a spacecraft on orbit. Furthermore, the applicability of the test bed, as a high-fidelity dynamic simulator, is limited to short duration close proximity operations because the relative orbital mechanics are not recreated. Despite these limitations, the POSEIDYN test bed is critical to experimentally test proximity operations guidance, navigation and control methods in a realistic environment, including sensor and actuator dynamics, as well as real-time computational constraints.

To propel each FSS over the granite monolith the vehicles are equipped with eight cold-gas thrusters, with two thrusters mounted equidistant from the center. Each thruster is composed of a solenoid valve and a custom-made convergent-divergent nozzle [100]. Fed by the on-board tank of compressed air, the thrusters provide a nominal thrust between 0.1–0.15 N of thrust (see Section 3.6 for thruster characterization details). This nominal thrust fluctuates considerably, as the thrust is a function of nozzle inlet pressure, which changes depending on the number of thrusters that are being fired simultaneously. With a nominal mass flow of about 0.3 g/s per thruster, the total firing time is estimated to be 20 min. Red light-emitting diodes paired with each thruster provide a visual indication of which thruster is firing.

Furthermore, one of the fourth generation FSSs has a Ball Aerospace 2.5 N·m·s reaction wheel (RW) (model RW 2.5-A1) mounted atop its structure, as illustrated in Figure 3.3. This particular FSS is used for spacecraft robotics research, and a multilink modular robotic manipulator can be connected to it. The RW was installed to meet the increased torque requirements due to the dynamic coupling (arising from the manipulator motion) and for the significantly higher inertia of the combined system [47]. An Arduino Due microcontroller is used as an interface between the on-board computer and the RW.

This microcontroller converts the output of the three hall sensors on the RW to a rotation rate and provides a suitable analog signal to control the RW motor torque. A switching shunt regulator is used to safely dissipate the back-EMF generated during RW deceleration.

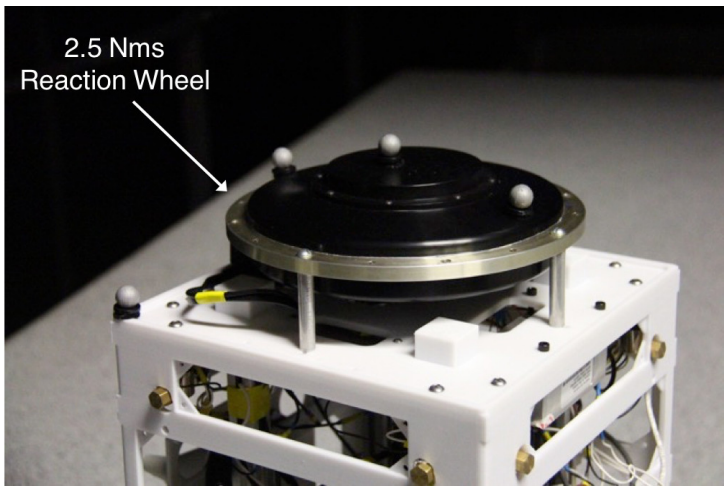


Figure 3.3. 2.5 N·m·s Ball Aerospace reaction wheel assembly mounted atop one of a FSS.

The onboard computational capabilities of the FSS are provided by a commercial off-the-shelf PC-104 form-factor onboard computer. Based on an Intel Atom 1.6 GHz 32-bit processor, the computer has 2 GB of RAM and an 8 GB solid-state drive. Despite the FSS onboard computer not being a space-qualified compute system (such as the BAE RAD750 or Proton 200k compute elements), its computational capabilities could be regarded to be on the same order of magnitude, in terms of the *estimated* MIPS-per-watt,[1] as state-of-the-art space-grade computers, as illustrated in Table 3.2 [101]–[107]. An expansion board with 20 optoisolated relays provides the required switching capability for the thrusters and air bearing solenoid valves. A serial interface is used to communicate with an on-board KVH DSP-3000 fiber-optic gyroscope (FOG), which provides angular velocity measurements at a 100 Hz rate. On the FSS with the RW, another serial port is used to communicate with the microcontroller interfacing with the RW. Two 95 W·h lithium-ion batteries and a battery management module regulate the electrical power to the FSS. Under idle conditions, the batteries can power the FSS for over 5.5 hours; under maximum loading, including the RW, the FSS can operate for just under 3.5 hours, thus making the amount of air inside the onboard tank the limiting factor during experiments.

[1] MIPS: Millions of Instructions Per Second

Table 3.2. Comparison of space-qualified and typical terrestrial processors. Adapted from [101]–[107]

	Name	No. of Cores/Threads	Processor Freq. (MHz)	Peak Power (W)	Est MIPS[1]	MIPS per Watt
Space Qualified Processors	BAE RAD750	1/1	200	12.5	400	32
	Proton200k DSP	1/1	200	5	4,000	800
	Proton200k Lite	1/1	66	1.5	1,200	800
	Tyvak Intrepid (ARM Processor)	1/1	400	0.3	440	1,467
Terrestrial Processors	ARM Cortex A15 (Tegra Processor)	4/4	1,900	2.25	4,000	1,778
	Intel Core i7-4710MQ (Generic Laptop CPU)	4/8	2,500	47	40,000	851
	Intel Atom Processor (FSS Processor)	1/2	1,600	5	3,200	640

[1] *MIPS: Millions of Instructions Per Second*

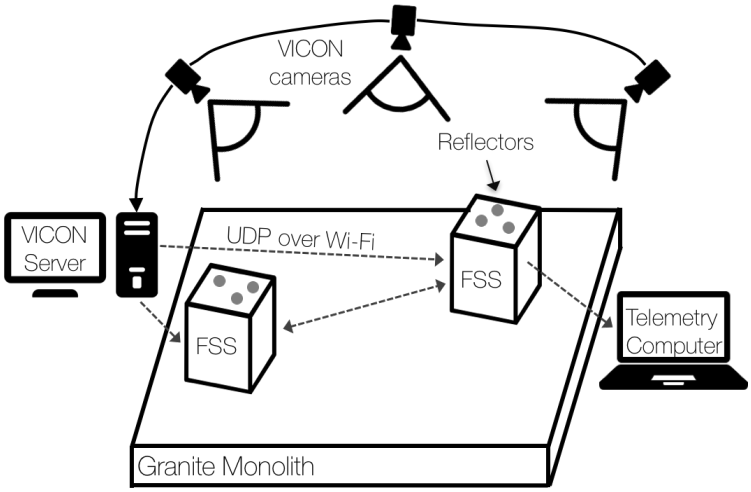


Figure 3.4. Overview of POSEIDYN communications architecture.

A Wi-Fi module provides wireless communication capabilities to each FSS [108]. The POSEIDYN test bed general communications arrangement is shown in Figure 3.4. The Wi-Fi module enables the FSS to communicate with other FSSs or other external computers, such as the laboratory motion-capture system (Vicon) server and ground station for telemetry logging and visualization purposes. To minimize latency the data is transmitted from node to node using User Datagram Protocol (UDP).

A carbon fiber reinforced polymer base plate and four aluminum T-slotted columns make up the core of the FSS structure. A polycarbonate outer shell provides the at-

tachment points for all the FSS components. Fabricated using additive manufacturing, the outer shell can be quickly modified to accommodate any component changes or vehicle upgrades (e.g., this was done with the RW).

The laboratory motion-capture system is a commercial system (Vicon) composed of 10 overhead cameras and an external computer. This system determines the position of objects carrying passive markers (i.e. the FSS) with sub-millimeter level (static) accuracy at rates up to 100 Hz. Once the location of the FSS is determined by the Vicon system, an external computer streams the data to the FSS using the Wi-Fi link via UDP.

3.5 Software Architecture Overview

At the core of the FSS software architecture is a real-time operating system (RTOS) which ensures the overlaying GNC software running on-board responds to sensor inputs and generates the appropriate actuator outputs within a strict and predefined time span.

To achieve the desired real-time requirement, a Ubuntu 10.04, 32-bit server edition operating system (OS) has been chosen and its Linux kernel (v2.6.33) has been patched with the RT-Preempt patch [109], [110]. This particular OS combination used by the FSS will be referred as RT-Linux OS. Running atop the RT-Linux OS is the multi-rate GNC software, which can be broken-up into four main subsystems: navigation, guidance, control, and telemetry.

The navigation subsystem provides a full state estimate of the FSS (position, orientation, and rates). Because most of the research conducted on the POSEIDYN test bed is focused on the guidance and control aspect, the navigation subsystem makes use of the accurate Vicon information to provide an accurate state estimate. As illustrated in Figure 3.5, the navigation block first samples the on-board sensors and the actuator states and fuses them via a Discrete Kalman Filter (DKF) to produce an inertial state estimate. The inertial state estimate is then sent to the guidance subsystem, where the appropriate actuator commands are generated and sent to the control subsystem. The standard control subsystem is comprised of the steering logic and converts the guidance-desired actuator inputs to the required low level signals to drive the different on-board actuators. Typically, only the guidance subsystem is user-defined and the navigation and control subsystems utilize predefined software. However, when research is performed on relative navigation, for example, the navigation block can be used to provide a "ground truth" estimate to benchmark the experimental relative navigation results [22].

Lastly, the telemetry subsystem packages the requested telemetry and sends it to a desired ground station. In addition to the GNC telemetry specified in the model, a

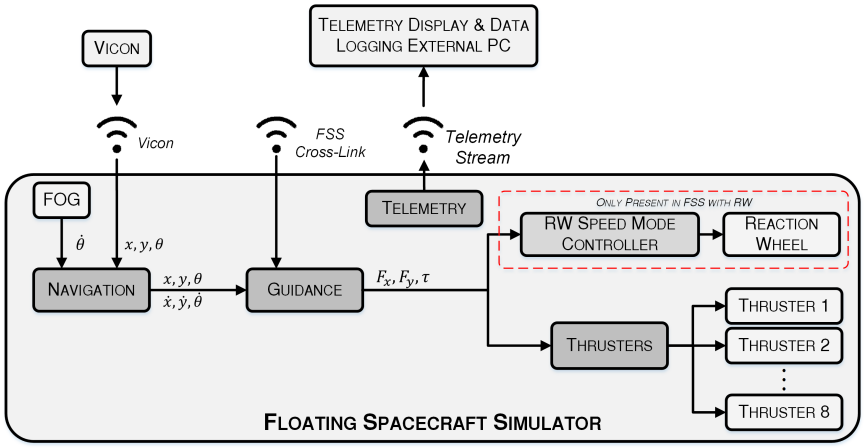


Figure 3.5. Overview of the onboard software architecture of the Floating Spacecraft Simulators.

publicly available system monitoring utility is used to collect metrics for various system-level metrics such as, but not limited to, processor usage, memory consumption, and network bandwidth. In particular, the `sysstat` utility was chosen [111]. It is worthwhile to note, the sampling rate of each subsystem (navigation, guidance and control) is user-specified, allowing for a multi-rate GNC formulation.

The remainder of this section will detail specifics about the GNC software development, navigation formulation, thruster mapping and modulation, reaction wheel control, and the development simulator.

3.5.1 TRIDENT-GNC Development Environment

To simplify the algorithm development and subsequent implementation on the FSSs, a (numerical) development simulator and a FSS software template were created using a common custom library. This library contains common software (navigation and control subsystems) which is used in both the simulator and the FSS autogenerated onboard software. The simulator uses simulated sensors and actuators and also simulates the plant response (i.e., the FSS); whereas the FSS software template uses the interfaces to the onboard sensors and actuators. This commonality between the simulator and FSS onboard software allows for rapid development of the algorithms in a simulation environment and software generation for use onboard the FSS for testing.

The multi-rate GNC software running atop the RT-Linux OS is developed utilizing the MATLAB and Simulink R2015b environment. Once developed, the Simulink models are autocoded into C and compiled (using the `ert_linux` target language compiler [112]). To facilitate the code generation and compilation across multiple OSs and architec-

tures, an Ubuntu virtual machine (VM) has been created the development tool-chain pre-installed. This VM allows the user to easily create hard real-time, multi-rate GNC software. The aggregation of the custom library, (numerical) simulator, FSS software template, Standard Test Framework test scenarios, and development VM and toolchain are collectively referred to as the Toolkit for Real-tIme DevelopmENT of Guidance, Navigation & Control (TRIDENT-GNC) (whose logo is presented in Figure 3.6.

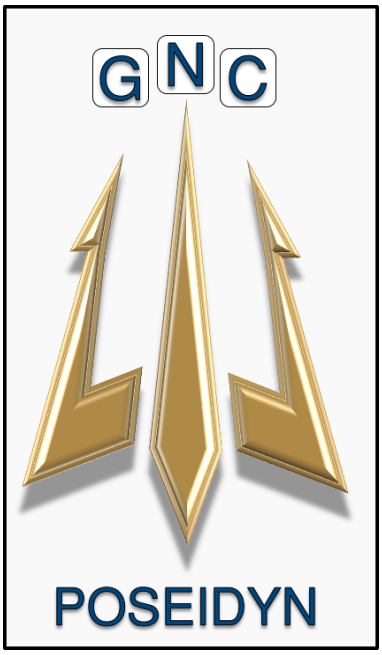
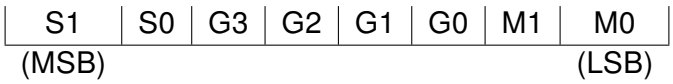


Figure 3.6. TRIDENT-GNC logo.

3.5.2 Navigation

A DKF fuses the Vicon and FOG data as well as an estimate of the actuated force and torque (given the states of the thruster valves and RW torque) to provide an estimate of the FSS inertial state. The FSS DKF is conceptually broken up into an outer-loop consisting of the prediction steps and an inner-loop composed of the correction steps that update the outer-loop while the filter is converged. Furthermore, in order to retain flexibility, the DKF operates at a user-defined time step, Δt_{DKF}. However, dropped or corrupted sensor measurements are inevitable due to the asynchronous sensors, user-defined time step, and lack of error correction in UDP communications. A dynamic construction of the measurement mapping matrix is used to mitigate the effect of dropped sensors measurement and a χ^2-gating rejects corrupted out-of-bound measurements [113]. Lastly, in addition to the standard output of a state estimate and error covariance estimate, additional telemetry indicating the status and “health” of the filter

(e.g., initialization status, convergence status, measurement validity, and measurement gating) is stored in the bit field of a 32-bit unsigned integer, illustrated in Figure 3.7. This section will detail the construction and considerations given in the development of the FSS DKF.



M0: No Vicon Measurement
M1: No FOG Measurement
G0: x Measurement Gated
G1: y Measurement Gated
G2: θ Measurement Gated
G3: $\dot{\theta}$ Measurement Gated
S0: DKF Initialized
S1: DKF Covariance Converged

Figure 3.7. Description of the DKF health telemetry item bit fields.

To formulate the DKF, the system is assumed to take the discrete-time representation

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1} + \mathbf{\Gamma}_{k-1}\mathbf{u}_{k-1} + \mathbf{\Upsilon}_{k-1}\boldsymbol{\omega}_{k-1} \tag{3.1}$$

where the state vector is denoted by $\mathbf{x}_k = \left[x_k, \dot{x}_k, y_k, \dot{y}_k, \theta, \dot{\theta}_k\right]^\mathsf{T} \in \mathbb{R}^{N_x}$ at sample time k; $\mathbf{u}_{k-1} = [f_x, f_y, \tau]^\mathsf{T} \in \mathbb{R}^{N_u}$ is the control input at the previous sample time $k-1$; $\mathbf{\Phi}_k \in \mathbb{R}^{N_x \times N_x}$ is the state transition matrix; $\mathbf{\Gamma}_{k-1} \in \mathbb{R}^{N_x \times N_u}$ is the discrete-time input gain matrix; $\mathbf{\Upsilon}_k \in \mathbb{R}^{N_x \times N_x}$ is the process noise covariance matrix; and $\boldsymbol{\omega}_k \in \mathbb{R}^{N_x}$ denotes the process noise which is assumed to be zero-mean Gaussian white noise. Additionally, measurements acquired at sample time k, $\mathbf{z}_k \in \mathbb{R}^{N_m}$, are assumed to follow

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \boldsymbol{\upsilon}_k \tag{3.2}$$

where $\mathbf{H}_k \in \mathbb{R}^{N_m \times N_x}$ is the measurement mapping matrix and $\boldsymbol{\upsilon}_k \in \mathbb{R}^{N_m}$ is the sensor noise that is assumed to be zero-mean Gaussian white noise. For this particular application, the overhead motion capture system (Vicon) provides position and attitude measurements while the onboard FOG provides attitude velocity information.

The predicted state estimate, $\hat{\mathbf{x}}_{k|k-1}$, and predicted error covariance, $\mathbf{P}_{k|k-1}$, are given as follows:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{\Phi}_k \hat{\mathbf{x}}_{k|k-1} + \mathbf{\Gamma}_k \mathbf{u}_{k-1} \tag{3.3}$$

$$\mathbf{P}_{k|k-1} = \mathbf{\Phi}_k \mathbf{P}_{k|k-1} \mathbf{\Phi}_k^{\mathsf{T}} + \mathbf{Q}_k \tag{3.4}$$

where the process noise covariance $\mathbf{Q}_k \in \mathbb{R}^{N_x \times N_x}$ used in the propagation of the predicted error covariance is assumed to have the following form [113], [114],

$$\mathbf{Q}_k = q_k^2 \begin{bmatrix} \Delta t_{\mathsf{DKF}}^4/4 & \Delta t_{\mathsf{DKF}}^3/2 & 0 & 0 & 0 & 0 \\ \Delta t_{\mathsf{DKF}}^3/2 & \Delta t_{\mathsf{DKF}}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \Delta t_{\mathsf{DKF}}^4/4 & \Delta t_{\mathsf{DKF}}^3/2 & 0 & 0 \\ 0 & 0 & \Delta t_{\mathsf{DKF}}^3/2 & \Delta t_{\mathsf{DKF}}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Delta t_{\mathsf{DKF}}^4/4 & \Delta t_{\mathsf{DKF}}^3/2 \\ 0 & 0 & 0 & 0 & \Delta t_{\mathsf{DKF}}^3/2 & \Delta t_{\mathsf{DKF}}^2 \end{bmatrix} \tag{3.5}$$

For simplicity, it is assumed the output of the thrusters follow a square wave – that is, the output of the thrusters is modeled to be fully ON or fully OFF with no transient response. In order to generalize the implementation of the filter, a forward initialization method is utilized to provide an in-situ initial state estimate and initial error covariance using two complete measurement sets [113]. Furthermore, to compensate for uncertainties associated with thrusting (see Section 3.6), the noise intensity, a strictly positive quantity, $q_k^2 \in \mathbb{R}_+$ associated with the attitude rate process noise diagonal element [see Equation (3.5)] is increased by a factor of 500 from a nominal value of 1×10^{-3}. This is done because the attitude rate, due to the small moment of inertia of the FSS, is more sensitive to thrust uncertainties. Note, the inflation factor was chosen through testing on the hardware.

The next consideration in the construction of the FSS DKF is the dynamic development of the measurement mapping matrix $\mathbf{H}_k \in \mathbb{R}^{N_m \times N_x}$. Nominally, the measurement mapping matrix is defined as follows:

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

The first three rows of the Equation (3.6) are associated with measurements from the Vicon sensor, whereas the last row is associated with measurements from the FOG.

The dynamic construction of measurement mapping matrix involves ensuring each measurement received from each sensor is “valid” before it is processed. The data

validity check is a two-step process; first, the `Data_Available` flag from the Vicon sensor and FOG are checked to be `true`. If either (or both) `Data_Available` flags are `false`, the respective rows in the nominal measurement mapping matrix $\mathbf{H}_k$ in Equation (3.6) are set to a row of zeros and the respective bits in the DKF health telemetry item are set to `0`. It is important to note that the `Data_Available` flags are an output from the implemented sensor blocks. Measurements which pass the first data validity check are then subjected to a χ^2 measurement association check in order to protect against corrupted measurements and filter smugness – a phenomenon which occurs when the estimated error covariance shrinks causing the filter to effectively reject new measurements resulting in a diverging state estimate unless corrected [78]. Therefore, a given measurement is considered to be associated with the FSS to a user-defined probability when [113],

$$\left(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}\right)^{\mathsf{T}}\left(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathsf{T}} + \mathbf{R}_k\right)^{-1}\left(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}\right) \leq \zeta \tag{3.7}$$

where ζ is a χ^2 distributed random variable with one degree of freedom and $\mathbf{R}_k \in \mathbb{R}^{N_m \times N_m}$ is the sensor noise covariance matrix and is defined as follows:

$$\mathbf{R}_k = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{\theta}}^2 \end{bmatrix} \tag{3.8}$$

Note, since each measurement is assumed to be independent, there is only one degree of freedom (DoF). Additionally, the default value chosen is 99.95%, which results in $\zeta_{99.95\%} = 12.116$. Although Equation (3.7) is written in matrical form for compactness, it can be easily expanded into four scalar equations for easier comparison with the association threshold. Since the measurement matrix does not contain any off-diagonal terms, the computations for each measurement are independent, thereby eliminating costly matrix inverses. If any measurement violates Equation (3.7), that specific row in the measurement mapping matrix is set to all zeros and the respective bit in the DKF health telemetry item for gating is set to `1`. Additionally, in an effort to overcome filter smugness, the diagonal term in the error covariance estimate is artificially inflated by a user-defined scalar – whose default value was chosen through testing to be 1.025 – in a similar fashion to a Fading-Memory Kalman Filter [114], [115].

Once the measurement mapping matrix is created, the state estimate correction and error covariance correction are computed in the traditional manner where the Kalman

gain, state estimate, and error covariance estimate are,

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^\mathsf{T}\left(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^\mathsf{T} + \mathbf{R}_k\right)^{-1} \tag{3.9}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}\right) \tag{3.10}$$

$$\mathbf{P}_{k|k} = \left(\mathbf{I}_{6\times 6} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_{k|k-1}\left(\mathbf{I}_{6\times 6} - \mathbf{K}_k\mathbf{H}_k\right)^\mathsf{T} + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^\mathsf{T} \tag{3.11}$$

Lastly, the FSS DKF is considered to be converged when the absolute value of the change in the square of the Frobenius norm of the measured diagonal elements of the error covariance estimate is below some user-defined threshold, ϵ_{DKF}. This condition is shown in Eq. 3.13, where

$$||\mathbf{A}||_\mathsf{F} = \sqrt{\sum_{i=1}^{4}\mathbf{A}_{ii}^2} \tag{3.12}$$

denotes the Frobenius norm of a matrix $\mathbf{A}$:

$$||\mathbf{H}_k\mathbf{P}_{k|k}\mathbf{H}_k^\mathsf{T}||_\mathsf{F}^2 - ||\mathbf{H}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{H}_{k-1}^\mathsf{T}||_\mathsf{F}^2 \leq \epsilon_{\mathsf{DKF}}^2 \tag{3.13}$$

While the filter convergence condition in Equation (3.13) is met, the associated bit field in the DKF health telemetry is set to "converged," or `1`. The filter convergence status is utilized in two places in the FSS GNC. First, on test startup, the FSS air bearings and guidance subsystem are enabled only after the DKF filter is converged. Secondly, the state and error covariance estimates are only passed to the outer-loop for use on the next DKF cycle only when the filter is converged.

3.5.3 Thruster Mapping

The control subsystem of every FSS includes thruster mapping algorithm. The purpose of thruster mapping is to select the appropriate thruster to fire in order to realize the desired control input consisting of forces and torques.

In the development of the thruster mapping algorithm, it is assumed each of the eight FSS thrusters produce identical force. The requested control input can be mapped to the appropriate thruster from:

$$\mathbf{u} = \mathbf{M}\mathbf{f} \tag{3.14}$$

where $\mathbf{u} = [F_x, F_y, \tau]^\mathsf{T} \in \mathbb{R}^{N_u}$ denotes the control input, $\mathbf{f} = [f_1, f_2, \ldots, f_8]^\mathsf{T} \in \mathbb{R}^{N_f}$ the force applied by each thruster, and $\mathbf{M} \in \mathbb{R}^{N_u \times N_f}$ is the thruster to control input mapping matrix. Therefore, the force required by each thruster to realize the control input is

$$\mathbf{f} = 2\mathbf{M}^+\mathbf{u} \tag{3.15}$$

where $\mathbf{M}^+$ denotes the Moore-Penrose pseudoinverse of $\mathbf{M}$. Since the $N_f > N_u$, $\mathbf{M}^+$ is given as follows:

$$\mathbf{M}^+ = \mathbf{M}^\mathsf{T} \left(\mathbf{M}\mathbf{M}^\mathsf{T}\right)^{-1} \tag{3.16}$$

Since the mapping matrix, $\mathbf{M}$, includes both positive and negative assignments, the pseudoinverse results in equal scaling along both the positive and negative directions. As a result, the factor of two in Equation (3.15) compensates for this fact in order to scale the requested force equally across only the applicable thrusters. For instance, if a force is requested along the $+x$ direction, thrusters 2 and 3 would be selected, each applying half of the requested force.

Lastly, unlike other actuators which offer a nearly continuous range of control inputs across their operating envelope (e.g., reaction wheels, control moment gyroscopes, and magnetic actuators), thrusters are discrete and can only take on two states: ON or OFF. As a result, difficulty arises where a continuous control signal has to be modulated in a way to actuate the thrusters in order to realize the control signal. A detailed description of various thruster modulation techniques is presented in Chapter 4. The default modulator, however, is set to be a Sigma-Delta Modulator ($\Sigma\Delta$M).

3.5.4 Reaction Wheel Controller

The RW is controlled though a simple speed-mode controller, illustrated in Figure 3.8. A speed-mode controller can compensate for the RW friction, and it controls the RW such that the correct amount of angular momentum is transferred between the RW and FSS. In such a controller, the requested torque is converted, using the RW inertia, to a required flywheel acceleration and integrated to obtain a desired flywheel velocity. The measured flywheel velocity, ω_{Measured}, is compared with the desired velocity

$$\Delta\omega = \omega_{\text{desired}} - \omega_{\text{Measured}} \tag{3.17}$$

and is scaled by a proportional gain in order to generate the required analog voltage to drive the RW motor torque. The flywheel velocity is sampled and the RW motor input voltage is updated at a 20 Hz rate. To avoid the zero-crossing problem, the RW is spun up to about 1100 RPM – half of its maximum rated flywheel velocity – before the air pads are enabled and the experiment starts.

3.6 POSEIDYN Test Bed Characterization

This section will the detail the system identification of a 4th generation FSS along. The characterization includes OS latency, sensor noise, thruster performance, mass, moment of inertia, and test bed residual acceleration.

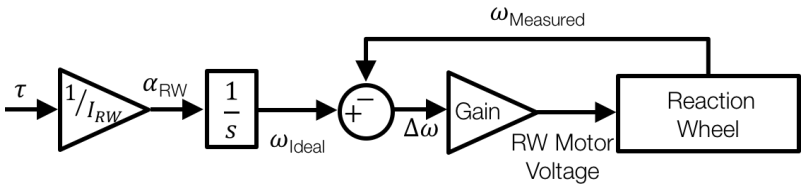
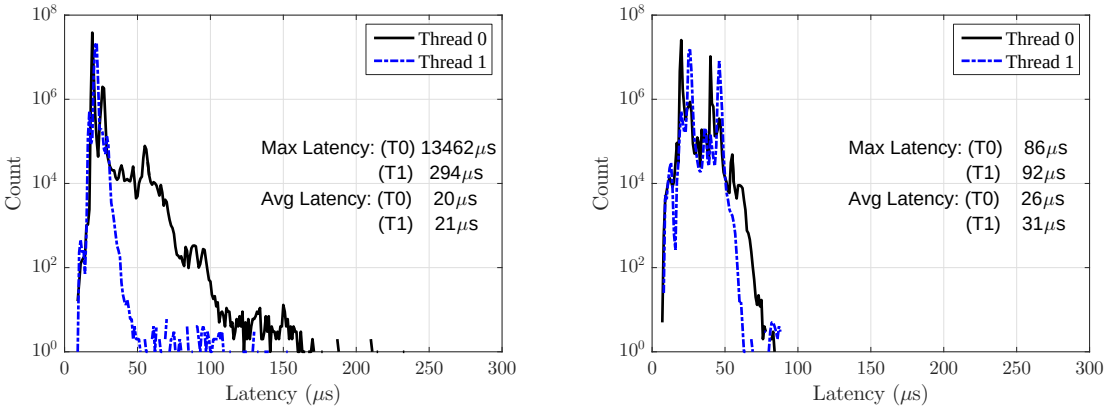


Figure 3.8. Reaction wheel speed mode controller.

3.6.1 OS Latency

By applying the RT-Preempt patch to the Linux kernel, the maximum latency the OS exhibits in response to a stimuli is bounded. To quantify the latency characteristics of both the stock (unpatched) Linux kernel and RT-Preempt patched Linux kernel, the `Cyclictest` was run for 50 million iterations [116]. The results are summarized in Table 3.3 and illustrated in Figure 3.9. Compared to the stock Linux kernel whose maximum observed latency during the `Cyclictest` was over 13 ms, the PREEMPT-RT



(a) Stock, unpatched Linux kernel

(b) RT-Preempt patched Linux kernel

Figure 3.9. Comparison of operating system latencies between an unpatched and patched Linux kernel.

Table 3.3. Summary of the `CyclicTest` latency results.

Kernel Type		Min Latency (μs)	Avg. Latency (μs)	Max Latency (μs)
Stock	Thread 0	9	20	13462
	Thread 1	9	21	294
Patched	Thread 0	7	26	86
	Thread 1	8	31	92

patched kernel bounded the OS latencies to within 100 μs. Furthermore, the maximum latency for both the stock and patched kernel was observed to occur on Thread 0. It is worthwhile to note, while the average latency of the patched kernel is slightly larger in value than the stock kernel, it is of the same order of magnitude and more importantly, the maximum latency is bounded to within 100 μs

Given the observed average latencies, this corresponds to a latency test duration of approximately 17 minutes for the stock Linux kernel and 26 minutes for the patched Linux kernel. Assuming an average upper bound on a candidate GNC algorithm test of five minutes, the duration of the latency test (for the patched Linux kernel) equates to an effective oversampling factor of 5x. Therefore, the TRIDENT-GNC architecture can guarantee that the GNC testing code will respond within 0.1 ms — a claim no other test bed in literature can make.

3.6.2 Sensor Noise Characterization

Vicon Noise Characterization
It is assumed the noise associated with the measurements provided by the Vicon ($x, y,$ and, θ) are independent since only processed measurements are provided. This assumption enables the characterization of each measurement produced by the Vicon to be performed independent of the other measurements. To characterize the translational sensor noise, the FSS is given only a translational velocity across the diagonal of the granite monolith. Its position from the Vicon sensor is then recorded directly, and the resulting time histories for x and y are independently curve fit to both a first-order model, $x_1(t)$ and second-order model, $x_2(t)$, respectively:

$$x_1(t) = a_1 t + a_0 \tag{3.18a}$$

$$x_2(t) = a_2 t^2 + a_1 t + a_0 \tag{3.18b}$$

where $\{a_0, a_1, a_2\}$ are the model coefficients. Assuming the acceleration coefficient in the second-order model, a_2, is sufficiently small, the root mean square error (RMSE) of the first-order error is interpreted as the noise parameter for the Vicon x and y measurements respectively. Furthermore, in order to capture any effects of speed on sensor noise, the FSS was given three different speeds across the diagonal of the granite monolith. Lastly, the geometric mean was used to compute the overall Vicon noise parameter for the x and y measurements. A similar method was followed to characterize the attitude measurement θ, except, instead of a translational-only velocity, the FSS was given only a rotational velocity.

The resulting Vicon measurement noise values across each trial and the overall geometric means are tabulated in Table 3.4. As tabulated, the Vicon sensor noise for the

x measurement is approximately 35% smaller than the y measurement. This is due to the arrangement of the Vicon cameras with respect to the granite monolith. The sensor noise for the θ measurement is over 38% larger than that of the y measurement and 70% larger than the x measurement. This is expected as the attitude measurements are derived from the position of each of the markers located atop the FSS, which are used in forming the dextral orthonormal basis of the Vicon measurement frame for the FSS in the Vicon software.

Table 3.4. Vicon sensor noise characterization results.

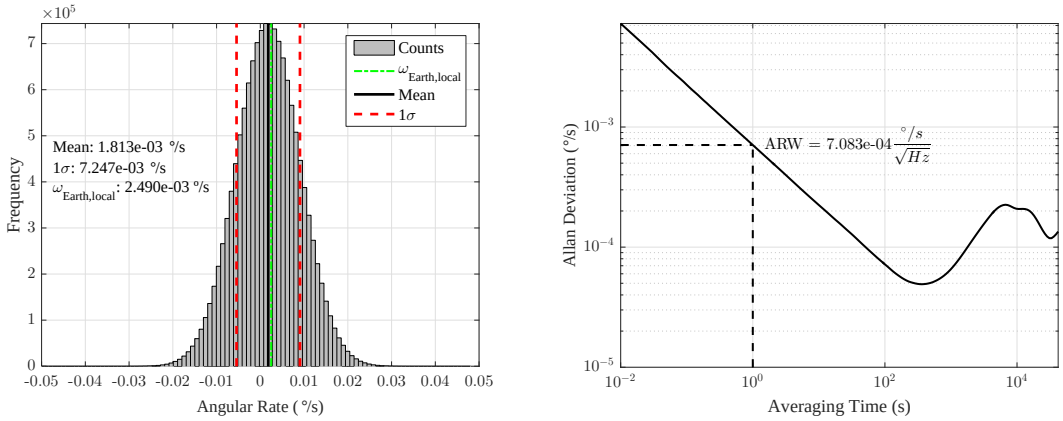
Trial	σ_x (m)	σ_y (m)	σ_θ (rad)
1	0.0113	0.00889	0.0356
2	0.0118	0.0130	0.00255
3	0.00199	0.00662	0.0269
Mean	0.00641	0.00914	0.0135

FOG Noise Characterization

To estimate the FOG noise parameters, an Allan variance analysis was performed following the IEEE Std 952-1997 [117]. The Allan variance method was initially used to investigate the frequency stability of precision oscillators in the time domain. However, this method has been extended to aid in characterizing random noise processes in a wide variety of devices [117], [118]

To perform the static characterization of the FOG, first, over 13.75 million samples were collected at 100 Hz over 27 hours from the FOG, as illustrated in Figure 3.10a. A basic analysis of the of static FOG data yielded the mean output, the standard deviation, as well as the quantization level. While not relevant to the FOG noise characterization, the quantization level is useful for creating the simulated FOG data. It can be seen the FOG is able to detect the rotation of the Earth, whose value at the location of the test bed is 0.002490 deg /s.

Next, performing the Allan variance analysis yields the Allan variance plot for the FOG illustrated in Figure 3.10b. The relevant noise parameter estimated from the Allan variance analysis is the angular random walk (ARW). ARW is identified on the Allan variance plot as having a slope of -1/2. Furthermore, the value for the ARW is obtained by fitting a line through this segment on the plot and reading the value at $\tau = 1$ [117] Given the FOG operates at 100Hz, the equivalent white noise value using the Allan variance method was found to be 7.083×10^{-3} deg /s. It is worthwhile to note that the white noise value derived from the Allan variance analysis is within 2.3 % of the value derived from a statistical analysis of the FOG data. Another useful noise parameter



(a) FOG data histogram

(b) FOG Allan variance Plot

Figure 3.10. FOG data histogram with mean output and 1σ indications and Allan variance plot.

which can be estimated via Allan variance analysis is the bias stability. This parameter would appear on the plot as a segment with a zero slope. However, due to the large averaging times required, estimating this parameter via this method is infeasible. Since this value is typically smaller than the ARW and given the short duration of the tests, this parameter can be neglected.

A summary of relevant FOG parameters from the two analyses is listed in Table 3.5. The white noise value derived from the statistical analysis of the static FOG data to use in the DKF was determined to be 7.247×10^{-3} deg /s, or equivalently, 1.265×10^{-4} rad /s.

Table 3.5. Summary of FOG noise analysis.

Parameter	Value
Mean Output	1.813×10^{-3} deg /s
White Noise (Statistical Analysis)	7.247×10^{-3} deg /s
White Noise (Allan variance)	7.083×10^{-3} deg /s
Quantization Level	1×10^{-6} deg /s

3.6.3 Thruster characterization

Characterizing the performance of the thrusters is essential across several aspects of the POSEIDYN test bed. This includes accurate modeling in the numerical simulator, control input in the DKF, guidance parameter tuning, and moment of inertia (MOI) estimation. A single thruster was characterized in [100]; here, a characterization of the thruster performance when integrated into the FSS is presented. Given the architecture of the system, the force generated by the thruster is highly influenced by the pressure regulator setting and the number of values open at the same time. In an effort to characterize the nominal performance of the thrusters, two thrusters on the same face are fired for several seconds and the maximum force generated by the thrusters is measured by a scale with a resolution of 9.81 mN. Additionally, it is assumed the measured force has equal contributions from the two thrusters. Prior to firing the thrusters, the value reported by the pressure regulator, with a one PSI (0.0689 bar) resolution, is recorded such that a pressure versus thruster force mapping can be made. This process is repeated across a range of operating pressures and several times to ensure repeatability. The results are illustrated in Figure 3.11. As illustrated, several regulator pressures took on similar thrust values due to the resolution of both the scale and pressure regulator.

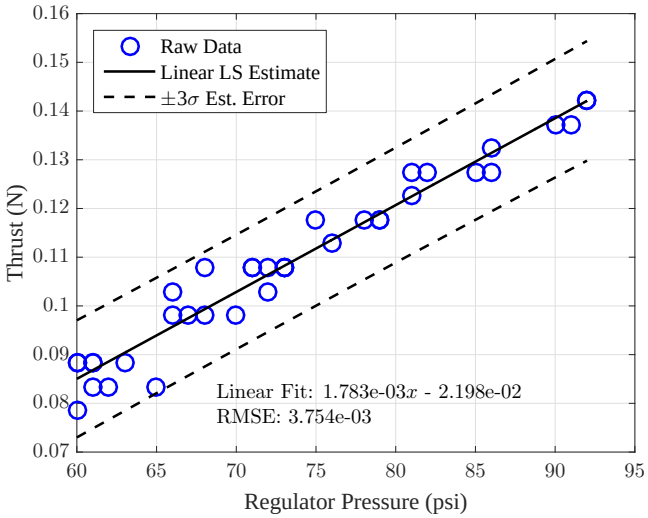


Figure 3.11. FSS thruster characterization data with first-order fit and estimated 3σ bounds.

Since no other data, such as temperature, is available, a linear fit is made between the regulator pressure and thrust generated. This linear model can then be used in the MOI estimation as well as adjusting the thrust parameter for the DKF and simulator. It is important to note, the onboard GNC software does not receive pressure information

from the regulator which precludes the use of the linear model onboard for estimating thrust.

3.6.4 Mass and Moment of Inertia

Estimating the physical properties (i.e., mass and MOI) is critical to not only understanding the system dynamics, but also to creating a realistic simulator and developing and tuning the guidance algorithms. The system masses of a FSS are tabulated in Table 3.6.

To estimate the FSS MOI, a torque was first applied to the stationary vehicle in the $-\theta$ direction by the thrusters, followed by a coast period; lastly, an opposite torque was applied in the $+\theta$ direction by the thrusters to spin the vehicle down. It is worthwhile to note, the value indicated by the thruster pressure regulator was recorded prior to the test in order to use the linear model (from Figure 3.11) to estimate both the thrust output and torque of each thruster. The resulting attitude time history was recorded directly from the Vicon sensor. Next, the portions of the attitude time history under thrusting were fit to the second-order model

$$\theta(t) = a_2 t^2 + a_1 t + a_0 \tag{3.19}$$

where $a_2 = 1/2(\tau/J_{zz})$, from which the MOI (J_{zz}) is estimated given the estimated torque τ. This test was performed five times in order to gather 10 attitude time histories under thrust, refueling the FSS each time prior to running the test. The resulting MOI estimate is tabulated in Table 3.6.

Table 3.6. Summary of relevant FSS physical properties.

Parameter	Value
Mass, wet	9.882 kg ± 0.001 kg
Mass, dry	9.465 kg ± 0.001 kg
Dimensions	0.27 x 0.27 x 0.52 m
Estimated MOI	0.253 kg·m^2
MOI Error (1σ)	0.0115 kg·m^2

3.6.5 Residual Acceleration

From an experimental standpoint, it is important to develop a bounded estimate of the residual accelerations acting on the FSS during a test campaign. In addition to any component of gravity in the plane of motion and residual friction, these unmodeled accelerations are also due to lateral air bearing leakage due to the air bearing not exactly

parallel to the test bed surface and the air flow coming out of the air bearing being not exactly symmetric [92]. In an attempt to quantify these residual accelerations, a FSS was positioned at the nine locations illustrated in Figure 3.12 on the granite monolith via an onboard Linear Quadratic Regulator (LQR) controller. Upon arriving at a predefined grid point, the air pads were closed for 30 seconds, allowing the FSS and the DKF to stabilize, before opening again for 30 seconds to perform the test at the location. The resulting acceleration imposed on the FSS during the 30 second interval was then derived from on-board state estimate information. The average (scaled) accelerations at each location over four tests is illustrated in Figure 3.12. The overall average translational and rotational residual acceleration over the nine locations was found to be 1.871×10^{-4}m /s^2 (19.1μG) and 7.56×10^{-2} deg /s^2, respectively.

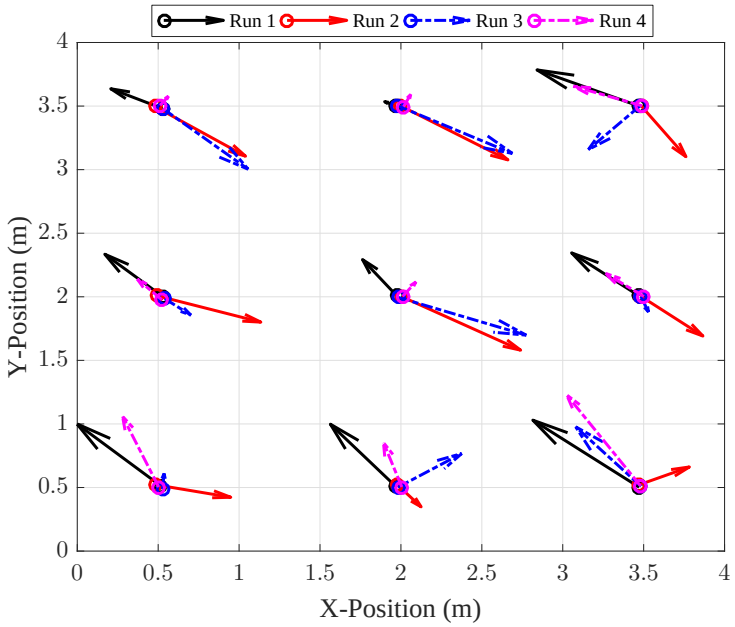


Figure 3.12. Average residual acceleration at nine locations on the granite monolith.

3.7 Case Study

To demonstrate the capabilities of the POSEIDYN test bed, an exemplary case study is performed where the FSS enters into and maintains a circular path with a 0.50 m radius and a tangential velocity for four revolutions, at a constant angular velocity of 3 deg /s about the center, using an APF-based guidance law to control the FSS [19], [20], [28], [32]. The APF-based guidance law was chosen as it is a real-time guidance and control method.

3.7.1 Guidance and Control Development

Assuming some initial configuration of the FSS that is outside of the desired circular reference path, the guidance will initially target a point, $\mathbf{p}_f$, that is tangential to the current position of the FSS, $\mathbf{p}$, and the circular reference trajectory in the direction of the desired rotation around the center:

$$\mathbf{p}_f = r_{\text{ref}} \operatorname{sign}\left(\dot{\psi}_{\text{ref}}\right) \begin{bmatrix} -\sin\left(\phi - \operatorname{sign}\left(\dot{\psi}_{\text{ref}}\right)\beta_t\right) \\ \cos\left(\phi - \operatorname{sign}\left(\dot{\psi}_{\text{ref}}\right)\beta_t\right) \end{bmatrix} + \mathbf{p}_{\text{ref}} \tag{3.20}$$

where r_{ref} is the radius of the circular reference trajectory; $\dot{\psi}_{\text{ref}}$ is the reference angular velocity about the center of the circular trajectory; $\mathbf{p}_{\text{ref}} = [x_c, y_c]^{\mathsf{T}}$ is the center of the reference trajectory; $\beta_t = \sin^{-1}\left((r_{\text{ref}}/D)\right)$ is the angle between the FSS and a point tangent to the reference trajectory and D is the distance from the FSS to the center of the reference trajectory; and $\phi = \tan^{-1}\left((y_c - y)/(x_c - x)\right)$ is the angle of the FSS relative to the center of the reference trajectory in the inertial frame. During this segment, only the target, or desired, position, $\mathbf{p}_f$, is passed to the APF-based control law. Furthermore, the FSS is commanded to point the $+x$ face towards the tangent intercept point. It is worthwhile to note, attitude control is performed via thruster actuation.

Once the FSS is within a user-defined distance, to the targeted point $\mathbf{p}_f$, the guidance then targets the circular reference trajectory and sets the initial angle ψ_0, the angle of the FSS relative to the center of the reference trajectory with respect to the inertial x-direction at this point in time. Subsequent reference positions and velocities are determined by propagating the initial reference angle at the user-specified rate, $\dot{\psi}_{\text{ref}}$:

$$\mathbf{p}_f = \mathbf{p}_{\text{ref}} + r_{\text{ref}} \begin{bmatrix} \cos(\psi) \\ \sin(\psi) \end{bmatrix} \tag{3.21}$$

$$\dot{\mathbf{p}}_f = r_{\text{ref}}\, \dot{\psi}_{\text{ref}} \begin{bmatrix} -\sin(\psi) \\ \cos(\psi) \end{bmatrix} \tag{3.22}$$

During the circumnavigation portion of the maneuver, the $+x$ face of the FSS is pointed inwards towards the fixed reference point

$$\theta_{\text{ref}} = \tan^{-1}\left(\frac{y_c - y}{x_c - x}\right) \tag{3.23}$$

An APF control law is used to generate the desired commands to follow the reference trajectory and reference attitude, $\mathbf{x}_f$. To generate the commands, a potential field is setup utilizing only one quadratic attractive potential located at the reference position

and attitude

$$\mathbf{\Phi}_a = \frac{1}{2}\mathbf{K}_a(\mathbf{x}-\mathbf{x}_f)^{\mathsf{T}}\mathbf{Q}_a(\mathbf{x}-\mathbf{x}_f) \tag{3.24}$$

where $\mathbf{K}_a \in \mathbb{R}^{3\times3}$ is a symmetric, positive-definite gain matrix, and $\mathbf{Q}_a \in \mathbb{R}^{3\times3}$ is the symmetric, attractive potential shaping matrix. It is worthwhile to note, if additional obstacles or keep-out zones were present, repulsive potentials could be placed at these locations to ensure obstacle avoidance and keep-out zone enforcement. The resulting control law is given as the gradient of the attractive potential (and repulsive potential, if used):

$$\mathbf{u} = -\mathbf{B}^{-1}\mathbf{K}_a\left(\mathbf{Q}_a(\mathbf{x}-\mathbf{x}_{\text{ref}}) + (\dot{\mathbf{x}}-\dot{\mathbf{x}}_{\text{ref}})\right. \tag{3.25}$$

where $\mathbf{B} \in \mathbb{R}^{3\times3}$ is the lower half of the continuous-time input gain matrix.

3.7.2 Case Study Setup

A list of the relevant test parameters are reported in Table 3.7. It is worthwhile to note, prior to running the test on the POSEIDYN test bed, the typical development workflow included preliminary testing using the development numerical simulator to tune and debug the guidance algorithm.

Table 3.7. Case study test parameters.

Guidance Parameter	Value	Navigation Parameter	Value	Control Parameter	Value
Sample Time	0.10 s	Sample Time	0.02 s	Sample Time	0.01 s
$(x_{\text{ref}}, y_{\text{ref}})$	(2, 2) m	q_k^2	0.001	SDM Sample Time	0.10 s
r_{ref}	0.5 m	σ_x	6.4123 mm	Thruster Force	0.115 N
$\dot{\psi}_{\text{ref}}$	3 deg /s	σ_y	9.1363 mm	Min. Firing Time	0.06 s
No. of Revs	4	σ_θ	1.3451 mrad	Min. Impulse Bit	0.0115 N·s
$\mathbf{Q}_a$	diag(0.05,0.05,0.05) s^{-2}	$\sigma_{\dot{\theta}}$	0.12649 mrad /s	– –	– –
$\mathbf{K}_a$	diag(1,1,0.1)	ζ	12.116	– –	– –
– –	– –	ϵ^2_{DKF}	5×10^{-8}	– –	– –

3.7.3 Case Study Results and Discussion

As illustrated in Figure 3.13 and Figure 3.14, the FSS starts in the lower-right hand corner of the granite monolith. At the start of the test, the guidance subsystem, targets a point tangent to the circular reference trajectory in the direction of the reference angular velocity. It can be seen that during the tangent point targeting segment the FSS attitude only achieves a near-zero attitude error as it nears the reference trajectory transition (marked by a red asterisk, $*$) due to a small attractive potential weighting on the attitude error. Once the FSS gets sufficiently close to the circular reference trajectory, the reference trajectory switches to the and the FSS is seen to track the desired circular trajectory fairly well. As illustrated in Figure 3.15, the maximum distance from the circular reference trajectory was less than approximately 0.04 m. The maximum difference between the FSS speed and the reference error speed was no larger than approximately 0.003 m/s, except for a brief period of time while the FSS was getting established on the reference trajectory. Furthermore, a relatively constant attitude error throughout the circumnavigation can be seen in Figure 3.13 and Figure 3.15c. This is attributed to the small attractive potential weighting on the attitude error used in the case study. The time, position, speed, and attitude errors across each revolution and the entire maneuver are summarized in Table 3.8. Lastly, as anticipated, frequent actuated control inputs occur throughout the maneuver to maintain the circular reference trajectory as illustrated in Figure 3.16.

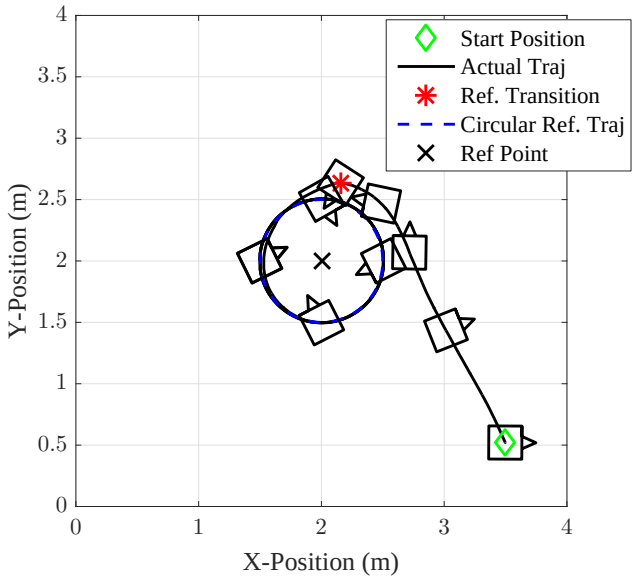
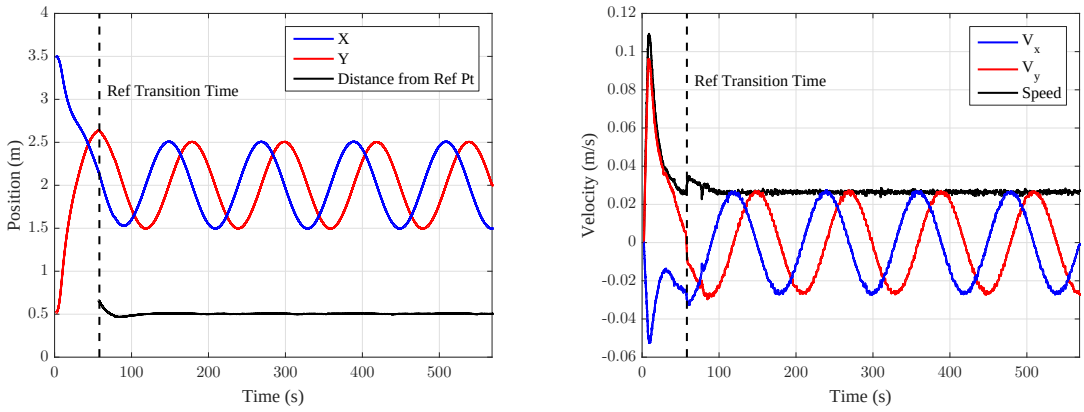


Figure 3.13. Overhead view of the FSS path with four equally-spaced attitude snapshots in time across the tangential intercept guidance segment and a single revolution about the fixed point. (Traj: Trajectory)



(a) Position (b) Velocity

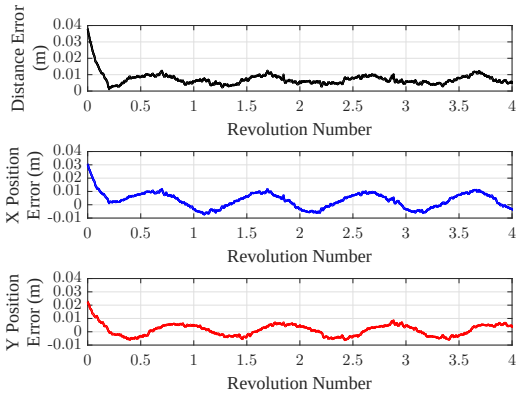
Figure 3.14. Position and velocity of the FSS during the entire maneuver.

Table 3.8. Summary of the circumnavigation time, position, speed, and attitude errors.

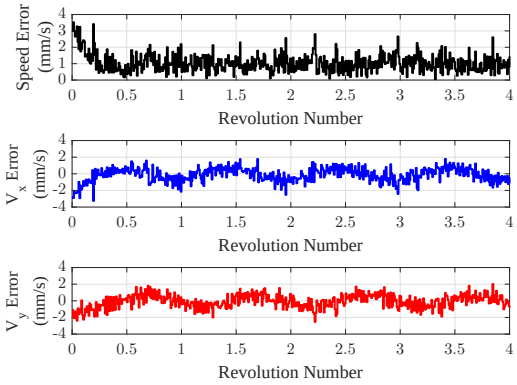
	Revolution				
	1	2	3	4	Overall
Rev. Time, s	120.00	120.00	120.00	119.91	119.98
Mean Pos Error, mm	9.18	6.56	6.64	6.64	7.25
Mean Speed Error, mm/s	1.18	1.01	1.01	0.99	1.04
Mean Att. Error, deg	35.07	26.64	27.62	27.01	29.08
Mean Att. Rate Error, deg /s	0.52	0.34	0.35	0.30	0.38

Table 3.9. Mean estimated error standard deviation over the circumnavigation maneuver.

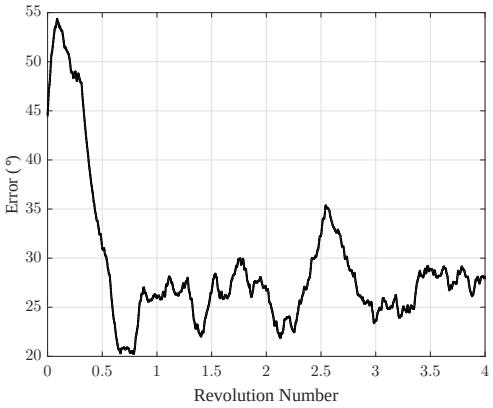
	Revolution				
	1	2	3	4	Overall
x, mm	2.26	2.63	2.64	2.65	2.63
v_x, mm/s	4.19	4.20	4.21	4.21	4.21
y, mm	3.39	3.41	3.43	3.44	3.42
v_y, mm/s	4.58	4.59	4.60	4.60	4.59
θ, deg	0.037	0.038	0.036	0.036	0.037
$\dot{\theta}$, deg /s	0.016	0.016	0.015	0.014	0.015



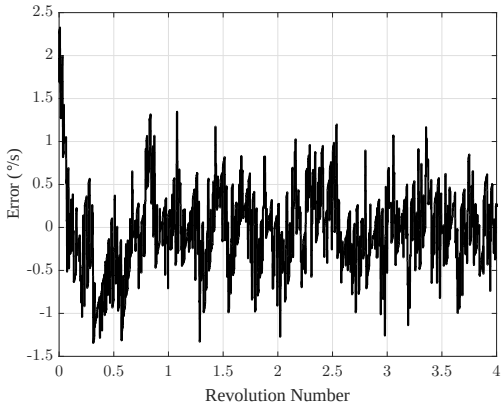
(a) Position Error



(b) Velocity Error

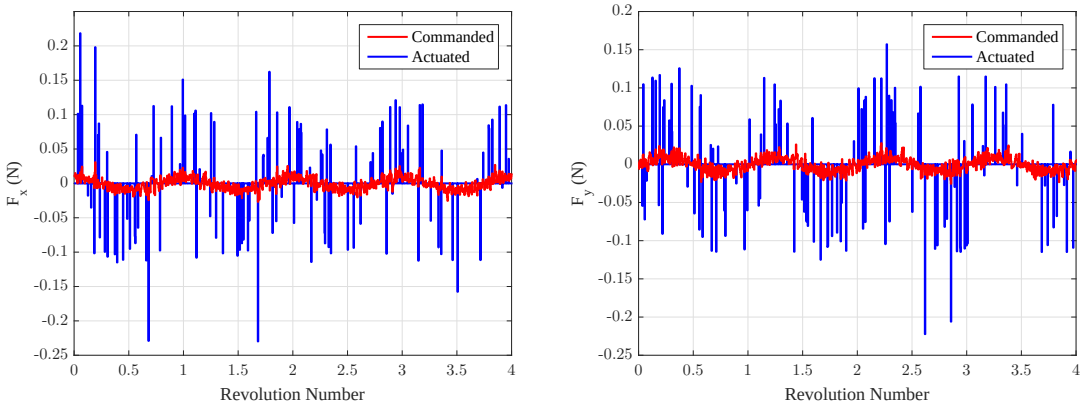


(c) Attitude Error

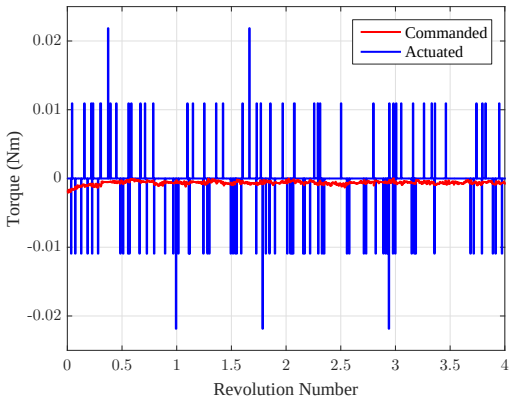


(d) Attitude Rate Error

Figure 3.15. Position and velocity errors of the FSS during circumnavigation.

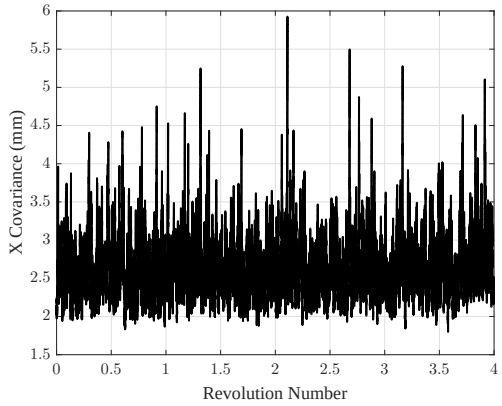


(a) F_x (b) F_y

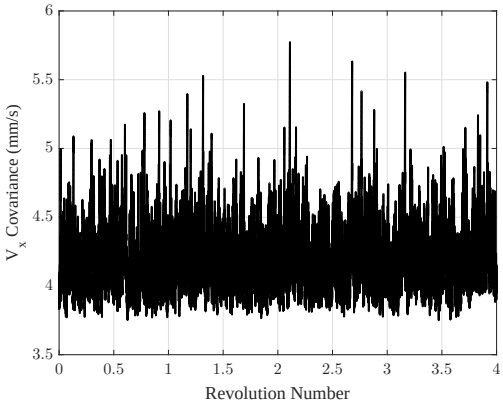


(c) Torque (thruster actuation only)

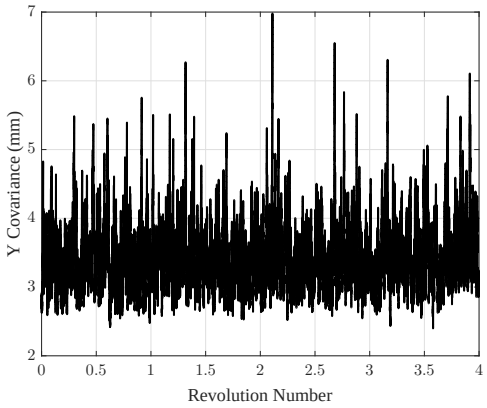
Figure 3.16. Commanded and actuated control inputs (in the inertial frame) throughout the circumnavigation maneuver.



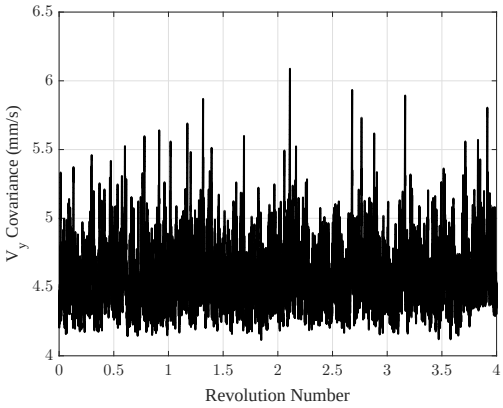
(a) x Covariance



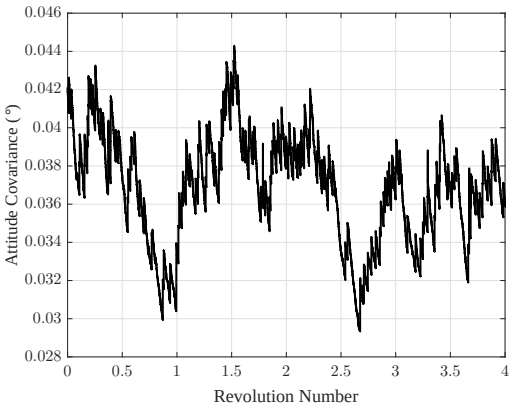
(b) v_x Covariance



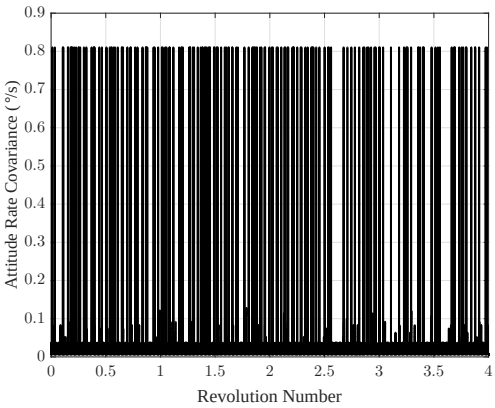
(c) y Covariance



(d) v_y Covariance



(e) Attitude Covariance



(f) Attitude Rate Covariance

Figure 3.17. Covariance time histories throughout the circumnavigation maneuver.

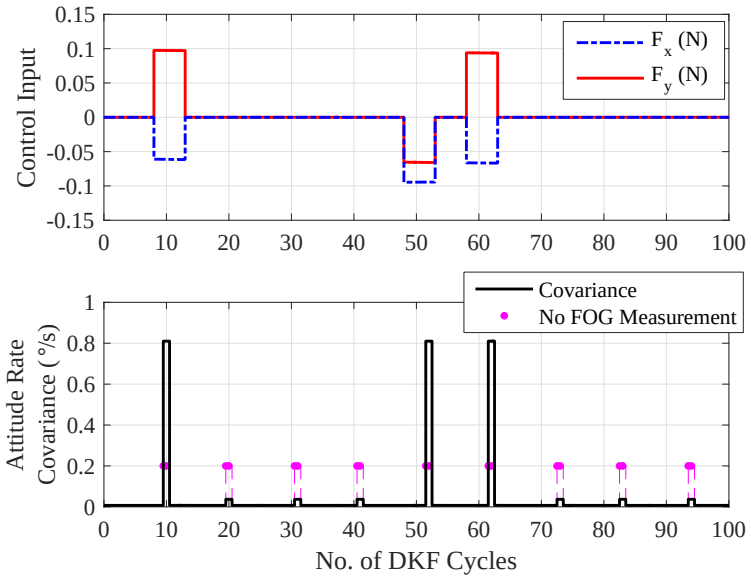


Figure 3.18. Detailed illustration of the Attitude Rate Covariance recovery.

To gauge the performance of the FSS DKF, both the navigation health telemetry and the estimated error covariances must be examined. The estimated error covariances throughout the duration of the maneuver is illustrated in Figure 3.17 and is summarized in Table 3.9. As a result of the frequent thruster firings required to track the desired trajectory, as illustrated in Figure 3.16, this case study tests the performance of the DKF. Specifically, the DKF is given minimal time to filter the noise introduced by the thruster firings. Additionally, the effects of increased process noise on the attitude rate covariance is illustrated in Figure 3.18. Recall, whenever a thruster is actuated, the navigation filter automatically inflates the process noise to account for uncertainties in the thruster model. Despite the process noise associated with the attitude rate state being inflated by a factor of 500, the attitude rate covariance is able to recover quickly (within 1 to 2 DKF cycles) due to the small sensor noise associated with the FOG. Lastly, the small, periodic, increases in the attitude rate covariance in between thruster firings were due to unavailable FOG sensor data, as indicated by the DKF health telemetry.

3.8 Summary

A detailed description of the Naval Postgraduate POSEIDYN test bed has been provided. The quasi-frictionless and low residual acceleration dynamic behavior of the test vehicles operating on the POSEIDYN test bed are limited to two translational and one rotational degree-of-freedom motion. Additionally, multiple vehicles can be operated simultaneously on the POSEIDYN test bed. This capability can be used to conduct re-

search on coordinated control of spacecraft teams. The on-board actuators, composed of eight thrusters and/or a reaction wheel, are equivalent to the actuators found in orbital spacecraft and further enhance the dynamic equivalence. With an available simulator and extensive software development tools, guidance and control algorithms for real-time execution on-board the test vehicles can be quickly and easily developed. Used as a last stage of on-the-ground validation prior to on-orbit deployment or simply as a more realistic development environment for novel guidance and control approaches, this state-of-the-art dynamic hardware-in-the-loop test bed will continue to be fruitful for the advancement of spacecraft proximity operations research.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Spacecraft Thruster Modulation Techniques

> Delta Sigma Converters or Sigma Delta Converters? Mankind does not seem to agree on one notation. Both notations are used equally often when you search via Google. I decided to stay with that guy who told he is living in the Mississippi Delta, so deltas mean something to him - and for him only the Sigma River may have a Sigma Delta...good point. Later I found out that the original name “Delta Sigma” was coined by the inventors Inose and Yasuda and "Sigma Delta is actually not correct. I was lucky...
>
> —*U. Beis*
> Quoted by Hebertt Sira-Ramírez
> *Sliding Mode Control* [119]

A journal version of the work presented in this chapter has been accepted for publication in the AIAA *Journal of Guidance, Control, and Dynamics* [70].

4.1 Overview of Thruster Modulation Techniques

Spacecraft reaction control (RC) thrusters provide a means for attitude control as well as orbital maneuvering. When compared to other actuators, offering a nearly continuous range of control inputs across their operating envelope (e.g., reaction wheels, control moment gyroscopes), RC thrusters take on only two discrete states: ON or OFF. To use the RC thrusters, the often continuous control signal has to be modulated.

One of the simplest thruster control methods is the Schmitt Trigger, commonly described as a relay with hysteresis and a deadband [82]. This control technique, which is not technically a pulse modulator, defines its minimum pulse-width as a function of the changing spacecraft’s inertia [82]. The Pulse-Width Pulse-Frequency Modulator (PWPFM) and Derived-Rate modulator extend the Schmitt Trigger with the addition of a first-order lag filter in either the feed-forward or feedback paths respectively. Another alternative is the Pulse-Width Modulator (PWM), resembling a PWPFM in behavior, but much simpler in its construction. The static characteristics of the PWM, PWPFM, and Derived-Rate modulator are independent of the spacecraft inertia [82], which is an advantage over the Schmitt Trigger. The PWM, PWPFM, and Derived-Rate modulator have been widely used for spacecraft thruster modulation [120]–[122].

Here, the use of the Sigma-Delta Modulator ($\Sigma\Delta$M)[1]is investigated and proposed as an alternative modulation technique for spacecraft thruster modulation. Originally applied to analog-to-digital converters by Inose and Yasuda, the $\Sigma\Delta$M has been employed in numerous applications, ranging from communications, analog-to-digital converters, high-quality audio playback, to high-resolution instrumentation [123]–[125]. Ciarcià et al. briefly considered the use of a $\Sigma\Delta$M for the RC thruster modulation for the translational control of a spacecraft simulator — however, due to its estimated control cost (in an ideal simulation environment), a different actuation strategy was chosen for their application [69]. To the author's best knowledge, the use of a $\Sigma\Delta$M for RC thruster modulation has not been previously used.

Compared to the preceding pulse-control modulators (PCMs), the $\Sigma\Delta$M utilizes a coarse (typically 1-bit) quantizer in combination with feedback to realize a "high resolution conversion of a relatively low bandwidth signal" [126]. Furthermore, this modulation technique provides user-defined noise shaping as well as small-signal actuation [124]–[126]. When compared to other PCMs, the $\Sigma\Delta$M is capable of realizing smaller input signals over smaller sampling periods. The $\Sigma\Delta$M can also be linearized through multiple methods, enabling the application of traditional continuous-time design and analysis techniques [119], [123], [125], [126].

Given its unique features and apparent advantages over traditional PCMs, the use of a a first-order $\Sigma\Delta$M as a thruster modulation technique is examined in detail and experimentally compared to a PWM. A PWM is chosen for the comparison for its simplicity and wide adoption on spacecraft RC control [127], [128].

Linearized models of each modulator are presented first. Next, these models, in combination with a simple PD control law, are used to develop candidate closed-loop models. An experimental campaign is then performed using the NPS POSEIDYN test bed, comparing the PWM and $\Sigma\Delta$M in a relevant environment with hardware-derived noise and uncertainties [31], [33]. Finally, the results from these experiments, and from various numerical simulations, are analyzed to draw conclusions regarding the applicability of the candidate models and of the $\Sigma\Delta$M's merit as a thruster modulation technique.

[1]Despite the anecdotal epigraph at the beginning of the Chapter, the term "Sigma-Delta" will be used throughout this Chapter. According to Gray, the alias "Sigma-Delta" is attributed to "[J.C.] Candy and his colleagues" [123]. A common argument in-favor of the use of the alias is that it is a more accurate description of the modulator, as the summing (Σ) is performed over the differences (Δ) [123], [124]. Additionally, Gray further argues that "the system does not really form a difference of successive samples [such as the Δ modulator] of the input signal as suggested by the $\Delta\Sigma$ name" (p. 60) [123].

4.2 Pulse Modulator Development

The construction of a first-order $\Sigma\Delta$M and PWM and their linearized models is presented in this section. These models are later used to derive candidate models of an attitude control system with a modulator-in-the-loop (MITL). These functional descriptions can either be viewed from either a hardware (e.g., analog-circuit) or digital implementation (e.g., software) perspective, but emphasis is placed on digital implementation, as this is the typical implementation medium.

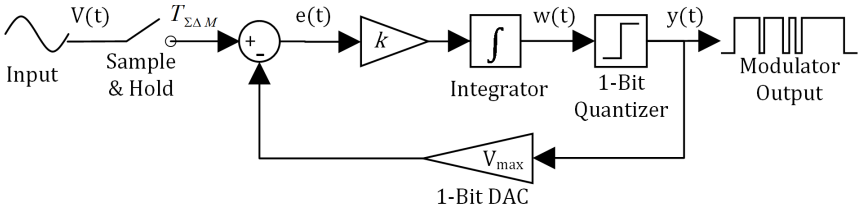
4.2.1 Sigma-Delta Modulator

Overview

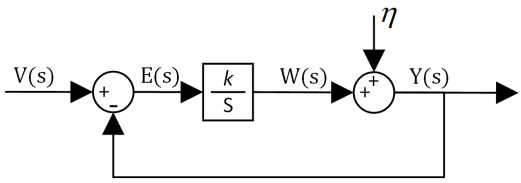
The topology of a first-order $\Sigma\Delta$M, illustrated in Figure 4.1a, takes the form of negative feedback loop. Along the feed-forward path, a low-pass filter, in this case a single integrator, is in cascade with a 1-bit quantizer; the feedback path is composed of a 1-bit digital-to-analog converter [124]–[126]. Without loss of generality, the input to the $\Sigma\Delta$M is a time-varying signal, $v(t)$. In this context, $v(t)$ represents the commanded force to be generated by a particular thruster. The input signal is sampled via a zero-order hold (ZOH) mechanism at a frequency $f_{\Sigma\Delta\text{M}} = 1/T_{\Sigma\Delta\text{M}}$ [129]. The output of the modulator, $y(t)$, is a digital pulse train and is converted back to an analog signal by the 1-bit digital-to-analog converter along the feedback path. In this application, the feedback signal can consist of either a model-driven or sensor-driven response of the thruster. As such, the sampling period of the $\Sigma\Delta$M should be similar to the impulse period of the thruster. The error signal, $e(t)$, is generated by subtracting the output from the input. It is then multiplied by a strictly positive gain $k \in \mathbb{R}_+$ before being integrated over the sampling period $T_{\Sigma\Delta\text{M}}$ to form the intermediate signal $w(t) = k\int e(t)\,\mathrm{d}t$. Lastly, the intermediate signal, $w(t)$, is modulated via a 1-bit quantizer to form the output signal $y(t)$, which is then fed-back. A pulse is triggered when $w(t)$ reaches a desired threshold [123], [125]. In this particular application, the quantizer threshold, ε, is lower-bounded by the thruster minimum impulse bit.

Linear Model Development

First, it is assumed the modulator sampling frequency, is sufficiently fast such that the ZOH can be neglected, allowing the $\Sigma\Delta$M to be viewed as a continuous-time device. The only remaining nonlinear element in the $\Sigma\Delta$M is the quantizer, which cannot be directly converted into an equivalent linear system. Multiple approaches can be taken which will linearize this modulator under various assumptions. The first method, which has been developed and used throughout the 20th century, assumes the quantizer error can be adequately described by an additive white noise source, as illustrated in



(a) First-order $\Sigma\Delta$M Topology



(b) Linearized first-order $\Sigma\Delta$M

Figure 4.1. First-order $\Sigma\Delta$M topology and linearized block diagram.

Figure 4.1b. This assumption, commonly known as the *additive white-noise approximation* assumes the quantization error is uniformly distributed over the two quantization levels provided by the 1-bit quantizer [123], [125]. In order to satisfy this assumption, it is necessary that the magnitude of the input to the modulator, $|v(t)|$, remains bounded by the output of the modulator, V_{max} [123], [125]. This results in a two-input, one-output closed-loop linear system given as follows:

$$\mathrm{Y}(s) = \frac{k}{s+k}\,\mathrm{V}(s) + \frac{s}{s+k}\,\mathrm{N}(s) \tag{4.1}$$

Note, the high-pass noise-shaping property of the $\Sigma\Delta$M is highlighted by the noise transfer function (NTF) given by

$$\mathrm{NTF}(s) = \frac{s}{s+k} \tag{4.2}$$

The corresponding magnitude response of the noise transfer function is

$$\mathrm{H}_{\mathrm{NTF}}(j\omega) = \frac{\dfrac{\omega}{k}}{\sqrt{\left(\dfrac{\omega}{k}\right)^2 + 1}} \tag{4.3}$$

which is that of a high-pass filter with a cutoff frequency of $\omega = k$ since $|\mathrm{H}_{\mathrm{NTF}}(jk)| = 1/\sqrt{2}$. Resultantly, the $\Sigma\Delta$M effectively "pushes" the noise to higher frequencies while

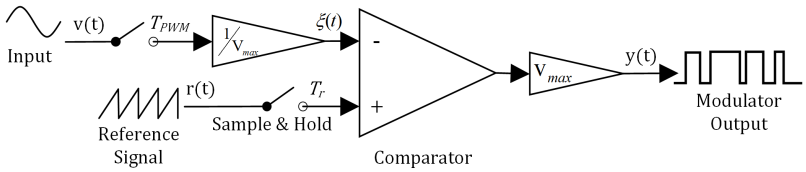
simultaneously increasing the signal-to-noise ratio for frequecies below the cutoff frequency.

Another method used to linearize the $\Sigma\Delta$M views the modulator as a sliding mode controller where the quantizer is represented as an ideal relay [119]. In this formulation, the quantizer is represented as a relay with an infinite switching bandwidth – thus converting the modulator to a continuous-time device. By viewing the $\Sigma\Delta$M as a sliding mode controller, the average output of the modulator is equivalent to its input . However, since the output of this model is a switched-output, a low-pass filter is required to demodulate the signal into a continuous waveform [119].

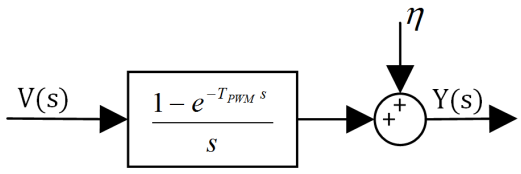
4.2.2 Pulse-Width Modulator

Overview

In contrast to the $\Sigma\Delta$M, the PWM is an open-loop modulator, whose topology is illustrated in Figure 4.2a, that generates a series of varying-width pulses that represent the input signal [130]. Similar to the $\Sigma\Delta$M, the input to the PWM is a time-varying signal, $v(t)$, representing the desired thrust level, which is sampled, via a ZOH, at the PWM frequency, $f_{\text{pwm}} = 1/T_{\text{pwm}}$ [129], [130]. Next, the input signal is normalized by the maximum output of the PWM, in this case, the thruster output, V_{max}, forming the desired pulse width, $\xi(t) = v(t)/V_{\text{max}}$. To realize the pulse-width, a normalized repeating reference waveform $r(t)$, with period T_{r}, is sampled at an integer $n \in \mathbb{R}_+$ frequency greater than the PWM frequency, $f_{\text{r}} = nf_{\text{pwm}}$ [130]. Note, typical reference waveforms used include a Sawtooth, Inverted Sawtooth, and Triangle waveform [130]. During a given PWM period, the output of the PWM, $y(t)$, is a digital pulse whose width which corresponds to the amount of time when $\xi(t) \geq r(t)$.



(a) PWM Topology



(b) Linearized PWM

Figure 4.2. PWM topology and linearized block diagram.

Compared to the $\Sigma\Delta$M which has only two discrete output states (ON or OFF), the number of discrete output levels of the PWM is controlled by the integer ratio

$$n = \frac{f_{\mathsf{r}}}{f_{\mathsf{pwm}}} = \frac{T_{\mathsf{pwm}}}{T_{\mathsf{r}}} \tag{4.4}$$

Note, the sampling period of the normalized repeating reference waveform, T_r determines the pulse width resolution, which establishes a lower bound on the deadband associated with this modulation technique [82]. The upper bound, which can be different from the minimum pulse width, is given by the minimum impulse of the thruster. Note, the lower bound is affected by the implementation of the PWM (e.g., T_{r}), whereas the upper bound is dependent upon the actuator hardware (e.g., thruster solenoid response time).

Linear Model Development

The only nonlinearity present in the PWM is the comparison operation to determine the pulse-width. Since the $f_r >> f_{\mathsf{pwm}}$, the reference signal ZOH can be neglected. However, since the PWM frequency is much smaller than the reference signal frequency, but is still larger than twice the largest frequency of concern in the input signal, the PWM ZOH may not be neglected. Viewing the comparison operation as a quantization process, the same approximation can be made regarding the linearity of the quantization error. Assuming the input $v(t)$ is bounded similarly to the $\Sigma\Delta$M, the comparison operation can be viewed as a quantization process and the same approximation can be made regarding the linearity of the quantization error. The resulting linearized PWM model, as illustrated Figure 4.2a, becomes a ZOH with a sampling period of T_{pwm}, whose transfer function is

$$\mathrm{Y}(s) = \frac{1 - e^{-T_{\mathsf{pwm}}s}}{s}\,\mathrm{V(s)} + \mathrm{N(s)} \tag{4.5}$$

4.3 Closed-Loop Spacecraft ACS with a Modulator-in-the-Loop

A pertinent application of spacecraft thruster actuation is for attitude control. Given the notional spacecraft attitude control system illustrated in Figure 4.3, the commanded actuation from the attitude controller is passed to the steering logic which selects the appropriate thrusters. The resulting desired thruster output signal is then modulated ,which, in-turn, drives the actuator. To gain insight on the effect of each modulation method, a closed-loop model describing the behavior of an attitude control system with a MITL will be developed.

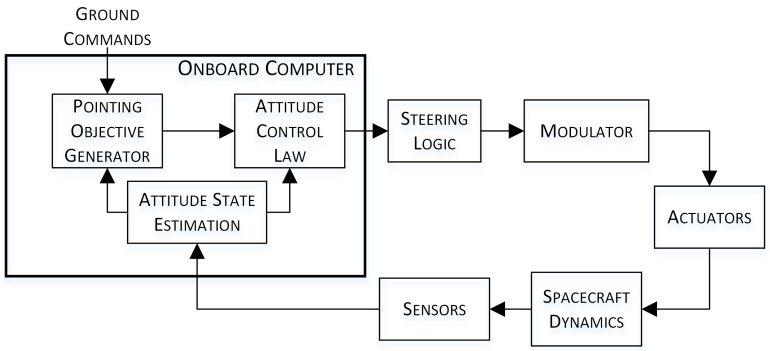


Figure 4.3. Notional spacecraft attitude control system block diagram.

The attitude dynamics of a spacecraft follow Euler's equation

$$\boldsymbol{\tau} = \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega}^{\times}\mathbf{J}\boldsymbol{\omega} \tag{4.6}$$

where $\boldsymbol{\tau} \in \mathbb{R}^3$ is the net torque acting on the spacecraft, $\mathbf{J} \in \mathbb{R}^{3\times3}$ is the inertia matrix with respect to the COM in the spacecraft body Cartesian coordinate system; $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity of the spacecraft in the body frame with respect to the inertial frame expressed in the spacecraft body Cartesian coordinate system, and $\boldsymbol{\omega}^{\times}$ is the matrical representation of the cross product operator [82]. When the inertia is measured in the Principal Axis Cartesian coordinate system, the inertia matrix becomes diagonal. For small angular velocities, the gyroscopic term becomes negligible and Equation (4.6) becomes,

$$\boldsymbol{\tau} = \mathbf{J}\dot{\boldsymbol{\omega}} \tag{4.7}$$

This implies the motion about each axis is decoupled and can be controlled independently. Furthermore, assuming no external disturbances, $\boldsymbol{\tau}$ becomes the control torque. For simplicity, a single-axis, double integrator model representing the spacecraft attitude dynamics is considered here.

Combing the transfer functions for each modulator and the attitude dynamics yields,

$$\mathrm{G}_{\mathsf{pwm}}(s) = \frac{1 - e^{-T_{\mathsf{pwm}}s}}{Js^3} \tag{4.8a}$$

$$\mathrm{G}_{\Sigma\Delta\mathsf{M}}(s) = \frac{1}{J(s^3 + s^2)} \tag{4.8b}$$

Assuming a PD control law of the form

$$\mathrm{C}(s) = k_d s + k_p \tag{4.9}$$

where k_p and k_d are the proportional and derivative gains respectively, the candidate closed-loop model are

$$\mathrm{T}_{\mathsf{pwm}}(s) = \frac{k_p + k_d s(e^{T_{\mathsf{pwm}}s} - 1)}{(Js^3 + k_d s + k_p)\, e^{T_{\mathsf{pwm}}s} - k_d s - k_p} \tag{4.10a}$$

$$\mathrm{T}_{\Sigma\Delta\mathsf{M}}(s) = \frac{(k_d s + k_p)}{Js^3 + Js^2 + k_d s + k_p} \tag{4.10b}$$

The validity of these models will be assessed in Section 4.4.3.

4.4 Experimental Comparison

This section will first discuss the experimental test setup and selection of pertinent experimental parameters. The results of the experimental campaign are then presented and discussed. Lastly, the validity of the linear candidate models are assessed.

4.4.1 Experimental Setup

A first-order $\Sigma\Delta$M was experimentally tested and compared to a PWM to demonstrate its performance and efficacy for spacecraft thruster control. A single-axis, 90 deg, rest-to-rest, slew maneuver was performed using an underdamped PD controller to demonstrate both the signal and steady-state tracking abilities of each modulator. The following metrics were leveraged in comparing the two modulators: thruster ON-time, steady-state attitude error, steady-state attitude rate error, and number of thruster actuations. Relevant experiment and spacecraft simulator system parameters are summarized in Table 4.1. Each experiment was repeated 10 times on the NPS POSEIDYN test bed [31], [33]. The setup for the experimental comparison is illustrated in Figure 4.4.

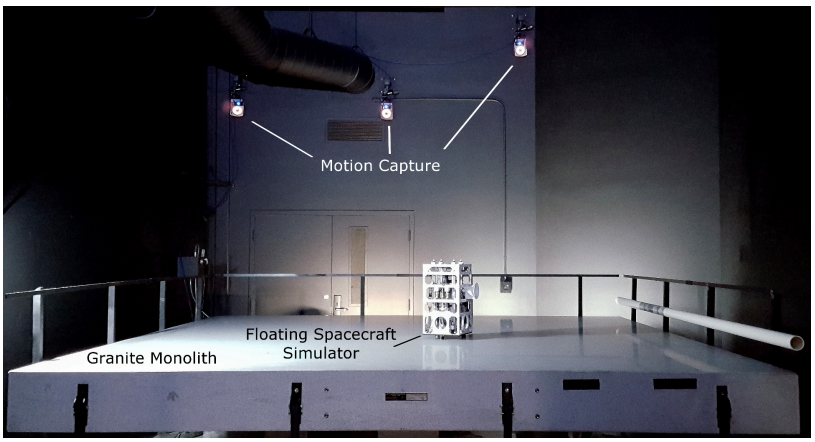


Figure 4.4. Test setup for the comparison between $\Sigma\Delta$M and PWM.

Before continuing, it is worthwhile to comment on the selection of a few of the experimental parameters. The sampling period of the $\Sigma\Delta$M was chosen to be slightly larger than the minimum solenoid response time as to ensure adequate flow development from the thruster, while the quantization level was chosen to be the minimum impulse bit for each thruster. The PWM minimum impulse time – the minimum amount of time for which the thruster must actuate – was chosen to be equivalent to the thruster solenoid response time, while the Pulse-Width resolution, T_r, was chosen to reduce the deadband associated with the attitude control [82]. Using Equation (4.4), the PWM sampling period of 1 s was chosen to allow for 100 discrete output levels.

Table 4.1. Summary of relevant experiment and FSS parameters.

Parameter	Value
Experiment Parameters	
Experiment Duration	120 s
Number of Trials	10
Reference Angle, θ_f	90 deg
Reference Attitude Rate	0 deg /s
Attitude Controller Sample Rate	10 Hz
Proportional Gain, k_p	0.10
Derivative Gain, k_d	0.20
FSS Parameters	
Mass	9.946 kg
Est. Inertia	0.253 kg·m^2
Est. Thruster Force	0.117 N
Est. Total Torque	0.0437 N·m
Thruster Solenoid Response Time	0.06 s
$\Sigma\Delta$M Parameters	
$\Sigma\Delta$M Sampling Period, $T_{\Sigma\Delta\text{M}}$	0.1 s
$\Sigma\Delta$M Error Gain, k	1
$\Sigma\Delta$M Quantizer Threshold, ε	0.0117 N·s
PWM Parameters	
PWM Minimum Impulse Time	0.06 s
PWM Sampling Period, T_{pwm}	1.0 s
Pulse-Width Resolution, T_{r}	0.01 s
Number of Discrete Output Levels	100

4.4.2 Experimental Results & Discussion

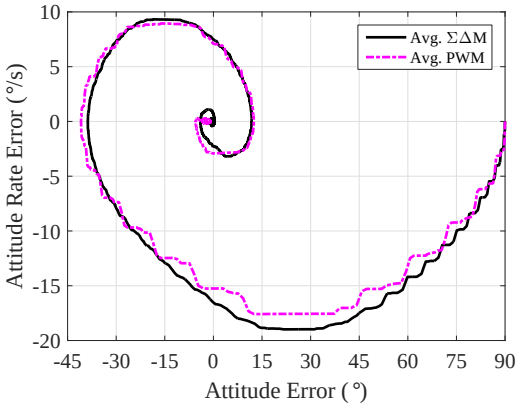
The results presented in Table 4.2 and Figure 4.6 are averaged over 10 trials. For the experimental parameters listed in Table 4.1, the $\Sigma\Delta$M was found to require approximately, on average, an additional 3 s of thruster ON-time when compared to the PWM but achieved a significantly smaller attitude error. This difference in thruster ON-time is attributed to the feedback loop in the $\Sigma\Delta$M, resulting in shorter period limit cycle, when compared to the PWM. After reaching a steady-state (at approximately 25-35 s), the $\Sigma\Delta$M was observed to command, on average, 22.5 thruster pulses (approximately one firing every 3.6 s) achieving an average attitude error of 0.32 deg. The PWM, however, was observed to actuate the thrusters, on average, 6.50 times throughout a similar period (approximately once every 12.3 s) achieving an average attitude error of 2.23 deg. Both modulation methods produced a nearly zero average steady-state attitude rate error; the standard deviation for the PWM was observed to be about half of that of the $\Sigma\Delta$M. As illustrated by Figure 4.5a, both modulators produce a similar (average) phase plane trajectory. In contrast to the smoother trajectory produced by the $\Sigma\Delta$M, the trajectory produced by the PWM has distinct and pronounced coast and actuation segments.

Table 4.2. Comparison of the 10-run averaged experimental metrics (with 1σ error) between the $\Sigma\Delta$M & PWM.

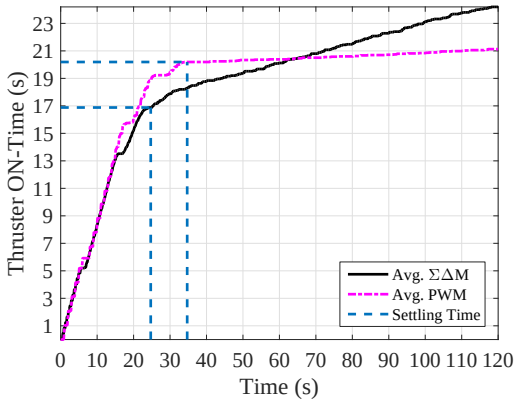
	$\Sigma\Delta$M	PWM
Thruster ON-Time, s	24.20 ± 0.47	21.13 ± 1.13
Steady-State Attitude Error, deg	0.32 ± 1.59	2.23 ± 2.60
Steady-State Attitude Rate Error, deg /s	0.00 ± 0.73	0.01 ± 0.36
Number of Thruster Actuations	100.50 ± 34.85	49.17 ± 17.99

Several important conclusions regarding the performance of the two modulators can be drawn by comparing the thruster ON-time history against time and the attitude and attitude rate errors, illustrated in Figure 4.5b and 4.5c and 4.5d. First, it becomes apparent that the $\Sigma\Delta$M leads the response of the PWM and resultantly settles to within 5 deg of the reference attitude θ_f approximately 10 s faster. As illustrated by the orange asterisks ($*$) in Figure 4.5c, the $\Sigma\Delta$M achieves steady-state using approximately 3 s less thruster ON-time! However, due to the short period of the limit cycle induced by the feedback in the $\Sigma\Delta$M, the thruster ON-Time was observed to linearly increase at a faster rate than the PWM. Given a set of mission requirements, the $\Sigma\Delta$M can be coupled with a deadzone in order to increase the period and size of the limit cycles, thereby

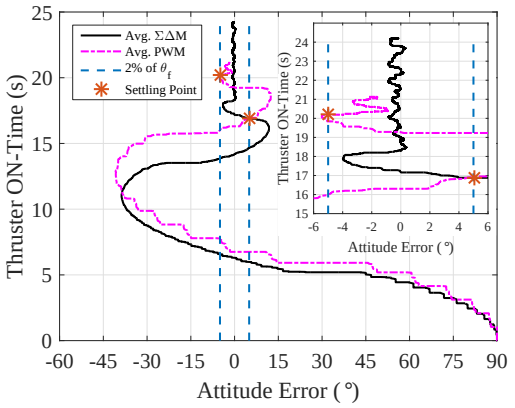
reducing the fuel consumption. Alternatively, one can utilize the $\Sigma\Delta$M throughout the transient period and switch to a lower-bandwidth modulator, such as the PWM, when in the vicinity of the reference state.



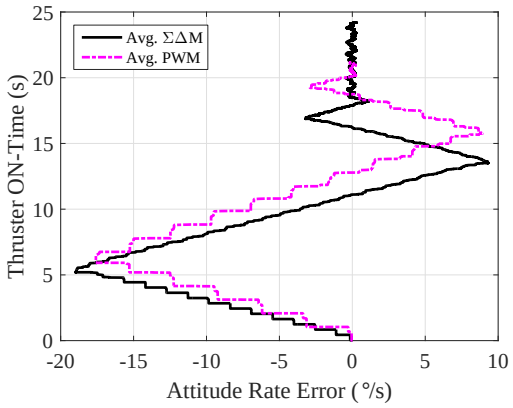
(a) Phase Plane Comparison



(b) Thruster ON-Time Time History

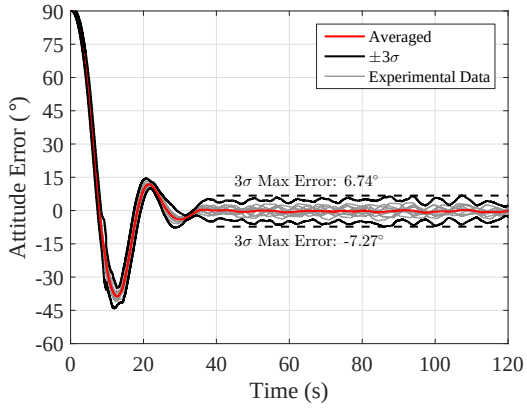


(c) Attitude Error vs. Thruster ON-Time

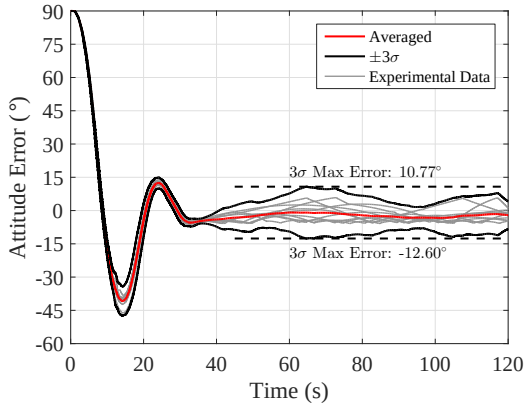


(d) Attitude Rate Error vs. Thruster ON-Time

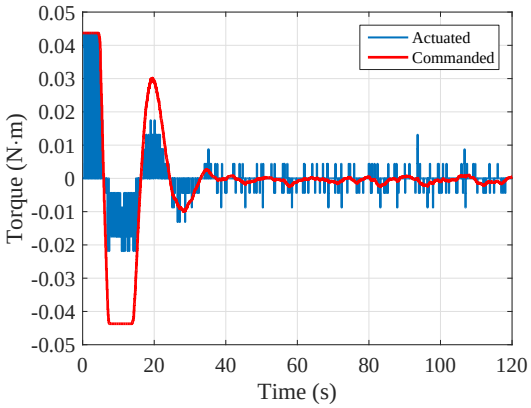
Figure 4.5. Direct comparison of the $\Sigma\Delta$M and PWM trajectories averaged over 10 runs.



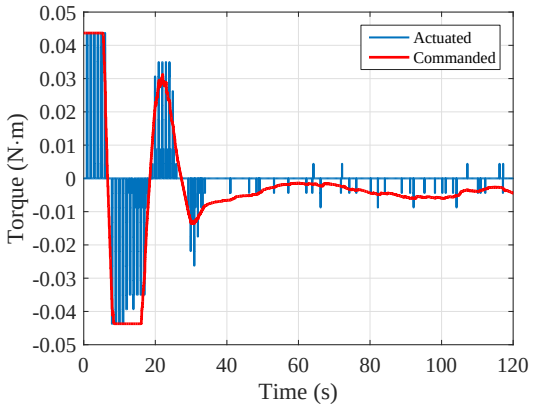
(a) ΣΔM: Attitude Error Response



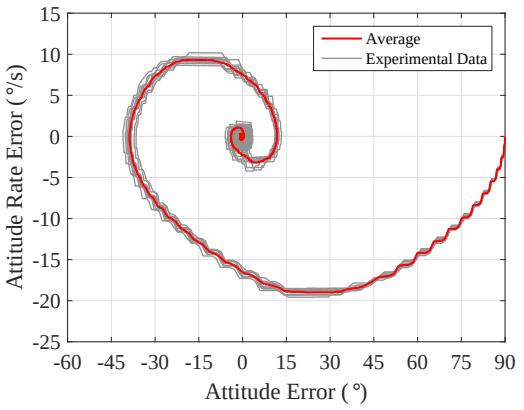
(b) PWM: Attitude Error Response



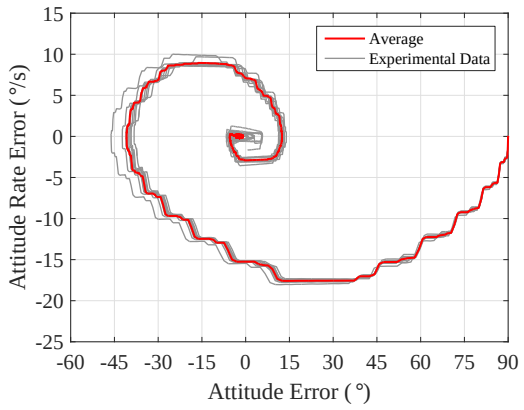
(c) ΣΔM: Control Time History



(d) PWM: Control Time History



(e) ΣΔM: Phase Plane Response



(f) PWM: Phase Plane Response

Figure 4.6. Experimental results averaged over 10 runs comparing the response of the ΣΔM to PWM for a 90 deg slew.

4.4.3 Linear Model Validation

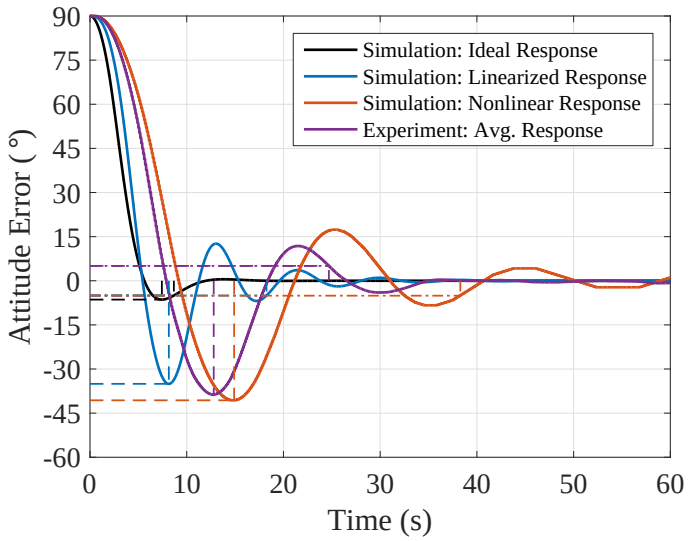
The candidate models developed which describe the closed-loop behavior of a spacecraft attitude control system with a MITL are described by Equations (4.10a) and (4.10b). To validate the candidate models, a comparison was made of their (numerically simulated) attitude error time histories against both the (averaged) experimental response and a (numerically) simulated nonlinear response of FSS [31], [33]. The idealized (linear) response with *no* MITL is presented for reference purposes. When implementing the numerical simulations, the control input was saturated at the maximum total torque listed in Table 4.1 for all systems. The resulting comparison is illustrated in Figure 4.7. Note, only the first 60 seconds are shown in order to increase the clarity of each response. Furthermore, the point of peak overshoot is indicated in Figure 4.7 by a dashed line $(--)$ while the settling time is noted by a dashed-dot line $(- \cdot -)$. Transient and settling characteristics of each system are tabulated in Table 4.3.

Table 4.3. Comparison of transient and settling characteristics.

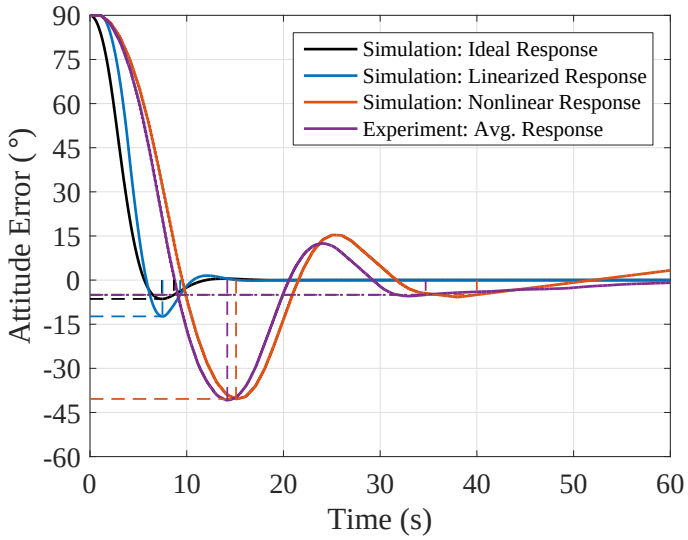
		Ideal Response	Linearized Response	Nonlinear Response	Experimental Response
Percent Overshoot, %	$\Sigma\Delta$M	7.12	38.94	45.15	42.44
	PWM	7.12	13.71	44.89	42.20
Peak Time (T_p), s	$\Sigma\Delta$M	7.43	8.15	14.90	12.79
	PWM	7.43	7.49	15.10	14.19
Settling Time (T_s), s	$\Sigma\Delta$M	8.67	18.25	38.29	24.70
	PWM	8.67	9.32	39.99	34.70
T_p/T_s	$\Sigma\Delta$M	0.86	0.45	0.39	0.52
	PWM	0.86	0.80	0.38	0.41

While the (simulated) attitude error response of the candidate $\Sigma\Delta$M model did not exactly match that of the experimental or simulated nonlinear responses, it did exhibit *similar* characteristics. First, the candidate model was observed to have a percent overshoot of 38.94%, corresponding to a 8.6% difference to the experimental response and a 14.8% difference to the simulated nonlinear response. Furthermore, comparing the ratio of peak time, T_p, to settling time, T_s, the candidate model was found to obtain a value (approximately) half-way in-between the experimental and simulated nonlinear responses, corresponding to a 14.3% difference to the simulated nonlinear response and a 14.4% difference to the experimental response. This has the implication of having similar damping characteristics to the two comparison responses. Given the similarities in the transient and settling characteristics of the candidate $\Sigma\Delta$M model compared to the experimental and simulated nonlinear responses, this model is considered to adequately describe the physical system.

The PWM candidate model, on the other hand, was not found to be in agreement with either the experimental of simulated nonlinear responses. This is apparent due to a significant mis-match between the transient and settling characteristics of the candidate model and the two comparison responses.



(a) ΣΔM



(b) PWM

Figure 4.7. Comparison of the linearized model response to the simulated and experimental responses.

4.5 Summary

Experimentally comparing a Sigma-Delta Modulator ($\Sigma\Delta$M) against a Pulse-Width Modulator (PWM) shows that the $\Sigma\Delta$M achieves a smaller steady-state error in a shorter amount of time, while using less propellant. These results, validated through numerical simulations, suggest that Sigma-Delta modulation is suitable for spacecraft thruster control as it offers better command signal tracking. However, due to the short period of the $\Sigma\Delta$M's limit cycle, the propellant consumption during the steady-state increases at a higher rate, when compared to a PWM. To avoid this drawback, the $\Sigma\Delta$M can be coupled with a deadzone or replaced by a lower-bandwidth modulator when the steady-state is reached, thus increasing the period of the limit cycle and thereby reducing fuel consumption. The linearized closed-loop $\Sigma\Delta$M model is found to exhibit similar response characteristics to both the experimental and simulated nonlinear responses, allowing it to be used, in conjunction with linear analysis techniques, for initial sizing and response estimation.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Experimental Evaluation Methodology

> An experiment is a question which science poses to nature, and a measurement is the recording of nature's answer. Before an experiment can be performed, it must be planned–the question to nature must be formulated before being posed. Before the result of a measurement can be used, it must be interpreted–nature's answer must be understood properly.
>
> —Max Planck
> "The meaning and limits of exact science" [131]

A journal version of the work presented on the *Standard Test Framework* is currently under review for publication in the Institute of Electrical and Electronics Engineers (IEEE) *Transaction on Control Systems Technology* [71]. A journal version of the work presented on the proposed Guidance Comparison Metric is under review for publication to the AIAA *Journal of Guidance, Control, and Dynamics* [75].

5.1 Introduction

With the vast number of RT-G&C methods and emerging number of CG&C algorithms (a term recently coined by Lu [49]) available in literature, the challenge arises of determining the "best" method for a given application. While traditional RT-G&C methods tend to use algebraic control laws, CG&C methods rely heavily on a model- or data-driven iterative process and are typically formulated using optimization-based approaches [49], [50]. The challenge then arises of selecting the appropriate G&C approach that not only enables completion of the mission objectives at hand, but yields an acceptable level of efficiency while simultaneously meeting system-specific requirements and constraints (e.g., algorithm reliability, computational time/responsiveness). Subsequently, a need arises for an equitable methodology to characterize algorithms in a meaningful and useful manner.

The first half of the characterization consists of constructing relevant benchmarking scenarios in an effort to understand the inherent characteristics of a G&C method. To enable an objective evaluation (and validation) of real-time guidance and control algorithms, a standard framework and set of benchmarking tests must be used. The notion of a *Standard Test Framework*, along with various conceptual test scenarios for rendezvous and docking, to evaluate guidance and control algorithms was initially

proposed by Virgili-Llop et al. [52]. The proposed scenarios in Section 5.2.2 build upon this notion and formally defines a *Standard Test Framework* that can be used to experimentally evaluate rendezvous and docking methods.

The second part of this methodology consists of selecting and understanding the measurements and metrics used to characterize a given RT-G&C. Notably, a measurement is the quantification of some feature or attribute of the system (typically expressed as a single number) while a metric can be more “abstract” and is composed of one or more measurements [132]. When reporting on a new G&C method, the majority of measurements presented in literature (e.g., control effort, maneuvering time, computational time) are implementation dependent, thus making it difficult to compare guidance methods. Therefore, it is necessary to combine these measurements in such a way as to form a metric that is agnostic to any specific implementation, denoted here as a “global" metric. To the authors’ best knowledge, no such global comparison metric which characterizes G&C methods has been proposed in literature.

The remainder of this chapter will detail the proposed *Standard Test Framework* which will be utilized as the benchmarking scenarios to evaluate RT-G&C algorithms. A general formulation and interpretation of the Guidance Comparison metric is presented. To illustrate the extensibility of this metric, formulations for both a spacecraft and terrestrial vehicle are given.

5.2 Standard Test Framework

5.2.1 Overview

The proposed test framework specifies the following information for each test:

- Information available to the guidance algorithms;
- Initial state (position, velocity, and orientation) of the Chaser and Target;
- Docking cone corridor parameters;
- Location, size, and movement of any obstacle(s) and associated keep-out zones in the workspace; and
- Tuning objectives

The following proposed scenarios are relevant as they evaluate the ability of the guidance method to handle various combinations of constraints while simultaneously producing safe trajectories and reducing propellant usage.

5.2.2 Test Scenarios

A schematic of the four proposed path-constrained test cases is provided in Figure 5.1 while the associated test parameters are listed in Table 5.1 and Table 5.2. It is worthwhile to comment briefly on the selection of some the parameters. First, the location of both the Chaser and Target were chosen in an effort to make the best use of the POSEIDYN test bed's workspace. Next, the docking cone corridor half-angle was chosen to be similar to the final approach cone corridor of the ISS [18]. To facilitate the tuning of guidance algorithms, a maximum rendezvous time is imposed only on Case 0, which consists of an obstacle-free, straight-in approach. Note, the maximum rendezvous time of 180 seconds corresponds to average speed of just over 0.02 m/s – which is slower than the average speed of the ATV over the final approach segment to the ISS [18]. Additionally, prior to entering the docking corridor, the attitude of the Chaser must be within 10 deg[1] of the final docking attitude, θ_f – thus ensuring a safe and successful docking. Lastly, to add an additional layer of realism to each test case, the only information available to the guidance algorithms about each obstacle is its instantaneous position and velocity.

The first test case, Case 0, is a straight-in case with only the docking cone corridor constraint. A time constraint of 180 seconds is imposed only on this test scenario to facilitate the selection of tuning parameters for an algorithm. This test case serves as the baseline for comparison with other cases.

Case 1 was adopted from [52], [72] which consists of a single obstacle placed directly in-between the Chaser and Target. Note, this is a pathological test case for the a double integrator system as the Chaser will encounter the obstacle along the straight-line path connecting the initial and final positions. This will result result in a non-unique solution to guidance and control problem. Additionally, inclusion of this test case will allow for comparison with previously published results obtained using IDVD and Linear-Quadratic MPC (LQ-MPC) methods.

Case 2 was adopted from Park et al. and incorporates two closely-spaced obstacles that the Chaser must navigate through (or around) in order to reach the Target [61]. Unlike Case 1, the obstacles are not located directly in-between the Chaser and the Target, thus providing insight into the path-planning capability of the guidance and control algorithm. Inclusion of this test case will allow for comparison with previously published results obtained using both a LQ-MPC and a Nonlinear MPC (NMPC) method.

Lastly, Case 3 is composed of an obstacle moving in a circular path that is centered on the straight-line connecting the initial position of the Chaser and Target. The center of this circular is located close to the entry of the docking cone corridor. The rotation rate of the obstacle was chosen such that the obstacle completes one revolution in the

[1] This corresponds to the approximate half-angle of the FSS docking cone.

maximum amount of time allotted to complete Case 0 (180 s). This rotation rate will ensure the obstacle moves into and out-of the path of the Chaser multiple times over the duration of the maneuver. Additionally, this stressing case will provide insight into the ability of the algorithm to adapt to a dynamic environment.

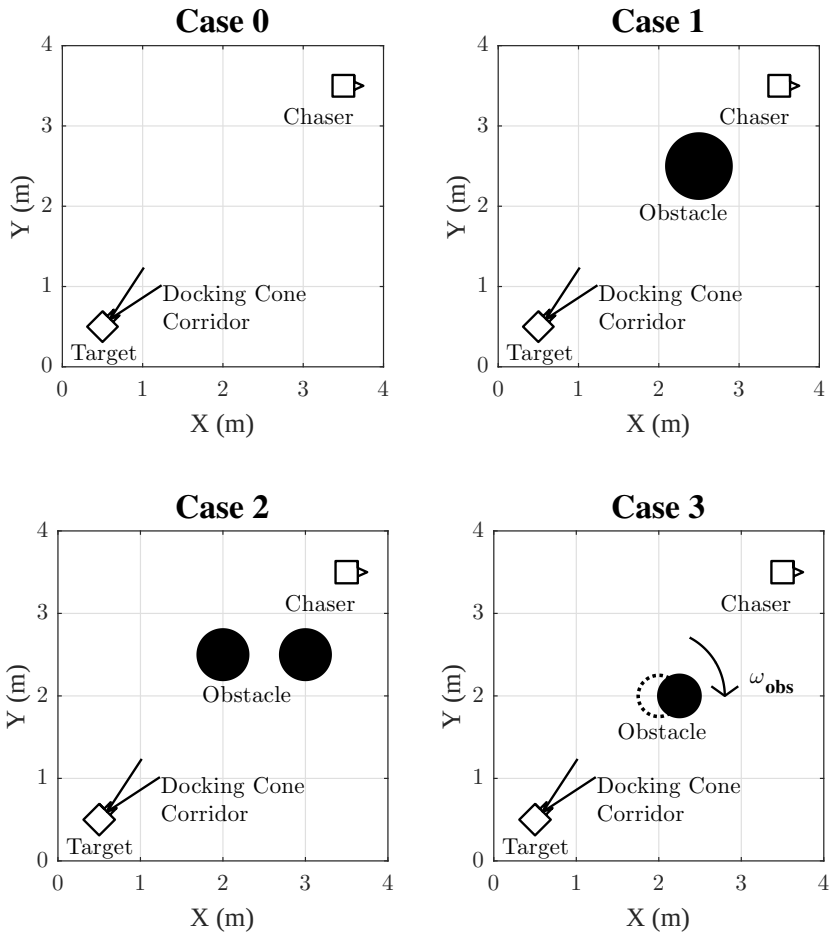


Figure 5.1. Schematic of the Standard Test Framework test cases.

5.2.3 Standard Measurements

The proposed standard measurements associated with this framework are: control effort, rendezvous time, constraint handling (i.e., constraint violations), as well as the computational cost [52], [72]. The control effort, c_e, is defined as

$$c_e = \int_{t_0}^{t_f} \|\mathbf{u}(t)\|_1 \, dt \tag{5.1}$$

Table 5.1. *Standard Test Framework* parameters.

Parameter	Value
Chaser Initial Location, $\mathbf{p}_c$	(3.5, 3.5) m
Chaser Initial Velocity, $\dot{\mathbf{p}}_c$	(0, 0) m/s
Chaser Initial Orientation, θ_c	0 deg
Target Location, $\mathbf{p}_t$	(0.5, 0.5) m
Target Velocity, $\dot{\mathbf{p}}_t$	(0, 0) m/s
Target Orientation, θ_t	45 deg
Target Keep-Out Radius, $r_{t,\text{KOZ}}$	$r_{t,\text{KOZ}} \geq 0.75m$
Docking Cone Corridor Half-Angle, β	10 deg
Docking Cone Corridor Length, ℓ_c	0.75 m
Docking Cone Corridor Entrance Attitude	$\theta_f \pm 10^\circ$
Speed at Docking, $\lvert\mathbf{v}_f\rvert$	$\lvert\mathbf{v}_f\rvert \leq 0.1$ m/s
Maximum Rendezvous Time for Case 0	180 s
Case 3 Obstacle Center of Rotation, $\mathbf{p}_{c,\text{obs}}$	(2.0, 2.0) m
Case 3 Obstacle Radius of Rotation	0.25 m
Case 3 Obstacle Initial Angular Position, $\theta_{0,\text{obs}}$	0 deg
Case 3 Obstacle Angular Rotation Rate about $\mathbf{p}_{c,\text{obs}}$	-2 deg /s

Table 5.2. Obstacle location and size for each test case.

Case	Obstacle Location (m)	Keep-Out Radius (m)
0	– –	– –
1	(2.5, 2.5)	0.40
2	(2.0, 2.5), (3.0, 2.5)	0.30
3	(2.0, 2.0)	0.25

where $\mathbf{u}(t)$ is the control input vector. The L_1-norm of the control input provides a quantitative measure of the efficiency of the RT-G&C method and is measured in N·s. It is worthwhile note, other useful quantities to measure the control effort, such as thruster ON-time, can be derived from this measure [52], [72]. Next, the maneuvering time is measured as the time elapsed from enabling the Chaser onboard guidance to completing the maneuver (e.g., docking with the Target). Note, these two quantities can be used to derive the non-dimensional control effort metric, ν_1 proposed and discussed in Section 5.5.1 Lastly, the computational cost of the guidance algorithm can be estimated by measuring the CPU time taken to derive a control input from the inputs to the algorithm.

5.3 General Formulation of the Guidance Control Metric

To fully describe a G&C algorithm, it is necessary to capture metrics spanning several categories including computational complexity, reliability, correctness (i.e., both numerical and temporal accuracy [24]), and system performance [133]. From these categories, the typical measurements of interest when evaluating candidate algorithms for a particular application is the total (expended) control effort, computational time, and maneuvering time. Steinfeldt et al. proposed a framework utilizing concepts derived from Multidisciplinary Design in an effort to develop a parametric-based comparison index for various guidance and control algorithms with a common application [133]. This methodology, however, is currently limited to a simulation-only environment as it requires a Monte Carlo analysis to produce a parametric index. Additionally, bias is invariably introduced into the comparison index as the weighting factors for each measurement (indicating their relative importance) are selected through heuristics [133], [134]. To the author's best knowledge, no comparison metrics have been proposed in literature which effectively capture the metrics of interest and compares them for various G&C algorithms across various system architectures.

Drawing inspiration from the supercomputing community [134]–[136], a proposed metric to compare different G&C algorithms has the following qualities:

1. *Unbiased*: This property ensures no one system architecture is favored over another architecture [134]. For instance, the metric should be invariant to different system architectures (e.g., different acceleration capabilities) for the same maneuver. As noted by Hsu et al., this property can be rather subjective [134].
2. *Bounded*: The goal of this property is to ensure the metric has either an upper or lower bound. This bound should also indicate either an optimal or desired value of the quantity.
3. *Insightful*: Depending on the interpretation of the metric, its value should provide some insight into how well the system is performing. Additionally, an insightful metric can also be used to motivate design choices [134].

The challenge arises of selecting an appropriate metric which effectively captures the three primary measurements of interest, the total control effort c_e, the maneuvering time Δt_m, and the computational time of the G&C method, T_g. A scalar composite metric, $J \in \mathbb{R}$, can be formulated using either an additive form [133],

$$J = |\alpha| c_e + |\beta| \Delta t_m + |\gamma| T_g \tag{5.2}$$

or a multiplicative form [134]–[136],

$$J = (c_e)^{\alpha}(\Delta t_m)^{\beta}(T_g)^{\gamma} \tag{5.3}$$

where $\alpha, \beta, \gamma \in \mathbb{R}$ are the relative weighting factors for the measurements c_e, Δt_m, and T_g, respectively. As illustrated by these two forms, a scalar composite metric is clearly biased; for a given set of measurements, the value of J can change by adjusting the relative weighting factors [133], [134]. A vector-valued metric, however, mitigates the introduction of bias as it implies each component of the metric is of equal importance [134]. Since the performance of the algorithm (e.g., expended control effort and maneuvering time) are considered to be just as important as the computational performance of the algorithm, the use of a vector-valued metric is appropriate.

The proposed Guidance Comparison metric has the form:

$$\mathbf{z} = [\nu_1, \nu_2]^{\mathsf{T}} \in \mathbb{R}^2 \tag{5.4}$$

The first element of the Guidance Comparison metric, a time-averaged non-dimensional control effort, is defined as The first component of $\mathbf{z}$ is the non-dimensional control effort defined as

$$\nu_1 = \kappa \, \frac{c_e}{m g_0 \Delta t_m} \tag{5.5}$$

where c_e control effort exerted by the system to complete the benchmarking maneuver in Δt_m seconds, $m \in \mathbb{R}_+$ is the mass of the system; g_0 is the gravitational acceleration constant at sea-level, 9.81 m/s^2; and $\kappa \in \mathbb{R}_+$ is a strictly-positive factor which makes ν_1 non-dimensional. Note, the non-dimensionalizing factor, κ, is a system-specific conversion factor as it is dependent upon the dimensionality of the control effort. This will be discussed in Section 5.5. Without loss of generality, the control effort is a system-specific unit of measure needed to complete a benchmark maneuver (e.g., the integral of the L_1-norm of the control input). By having a general definition for ν_1, it becomes a metric which can be applied across several domains. Additionally, to form an unbiased performance metric, any dependencies between the maneuvering time and control are decoupled by using them together to form a scaled time-averaged control. Scaling ν_1 by the system mass attempts to eliminate any bias due to different system designs (e.g., thrust-to-mass ratios), enabling a more global comparison of algorithms. This scaling is discussed in Section 5.5.

The second element, ν_2, of the comparison vector $\mathbf{z}$ is a non-dimensional sampling time

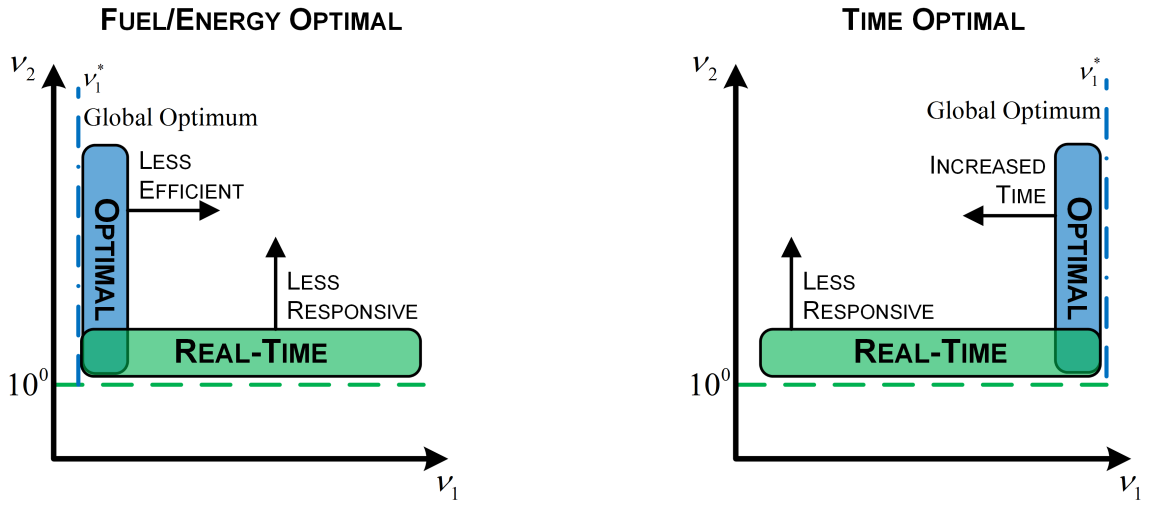
$$\nu_2 = \frac{T_g}{T_b} \tag{5.6}$$

where T_g is the computational period of the guidance method (i.e., the guidance sampling period) and T_b is the duration of the fastest task of the underlying real-time operating system, denoted here as the Executive sampling time. The ideal value and lower bound for this parameter is unity. This implies the guidance algorithm is running at the fastest possible rate, the Executive Rate, for a given (real-time) system. Note, the inverse of this metric, ν_2^{-1}, can be viewed as a measure of the algorithm's "real-time efficiency". Additionally, this metric also provides insight into the computational complexity of the algorithm. A larger guidance sampling period implies the algorithm requires a larger amount of time to solve – thus implying greater computational complexity.

5.4 Interpretation of the Guidance Control Metric

Similar to benchmarking software, comparing several G&C algorithms for a given application requires the use of a standard benchmark in order to form an equitable comparison. For spacecraft AR&D or RPO, the benchmark of choice could be the *Standard Test Framework* proposed in Section 5.2 and [71]. To compare several guidance algorithms, the elements of the comparison vector, $\mathbf{z} = [\nu_1, \nu_2]^\mathsf{T}$, can be treated as a pair of coordinates with ν_1 being the abscissa and $\log_{10}(\nu_2)$ being the ordinate. The resulting graph simultaneously captures both computational complexity and algorithm performance, allowing for comparison with other algorithms. It is worthwhile to note, additional insight can also be gained by plotting ν_1 (as the ordinate) as a function of each the test case for each guidance and control algorithm, as illustrated in Section 6.9.

The ideal G&C algorithm is one that produces the optimal solution to the desired OCP at the fastest rate of the underlying real-time operating system (i.e., the Executive rate). Since both the Minimum-Fuel and Minimum-Energy optimal control problems reduce the amount of control effort at the expense of increased maneuvering time, the ideal value, ν_1^*, is as small as possible, while still completing the maneuver. Maneuvers which are sub-optimal will fall to the right of this boundary and take on larger values of $\nu_1 > \nu_1*$, indicating a larger time-averaged control effort. Similarly, for the Minimum-Time optimal control problem, the maneuver time is minimized at the expense of increased control effort. The resulting ideal and upper-bounding value sets the maximum value, ν_1^*, that can be achieved for this particular objective function. Values of $\nu_1 < \nu_1^*$ indicate sub-optimal solutions which take a larger amount of time compared to the optimal solution. Likewise, the lower bound on the non-dimensional sampling period, ν_2^*, is one. Values of $\nu_2 > 1$ indicate the algorithm will be less responsive than the computational architecture will allow. That is, the algorithm can only respond once every ν_2 Executive sampling periods to any perturbations and disturbances. Notional plots for a Minimum-Fuel/Energy and Minimum-Time optimal control problem illustrating this interpretation of the guidance control metric are illustrated in Figure 5.2.



(a) Fuel and Energy Optimal

(b) Time Optimal

Figure 5.2. Notional illustration on how to interpret the Guidance Comparison Metric.

5.5 System-Specific Formulations

To show the extensibility of the proposed characterization framework, this metric is applied to several systems. A case study focusing on the development of tradespace for the selection of guidance and control algorithms is presented in Chapter 7

5.5.1 Spacecraft

For a spacecraft, the control effort, c_e, to complete an arbitrary maneuver is defined as follows:

$$c_e = \int_{t_0}^{t_f} \left(\|\mathbf{F}(t)\|_1 + \|\boldsymbol{\tau}(t)\|_1 \right) \, dt \tag{5.7}$$

A fundamental issue arises as the control effort consists of forces and torques . To resolve this issue, a torque can be represented by a pair of opposing thrusters of equal strength, F, that have a moment arm, b_0, of 1 m from the center of mass of the spacecraft and generate a pure rotation (i.e., a couple), as illustrated by Figure 5.3. The equivalent force generated by each thruster is

$$F = \frac{1}{2b_0}\tau \tag{5.8}$$

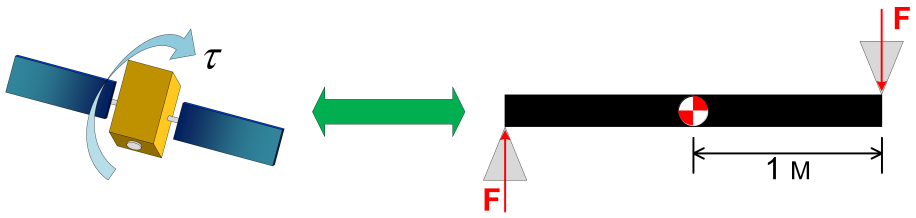


Figure 5.3. Illustration of the conversion of a torque to a force couple.

The control effort in Equation (5.7) can now be expressed a combination of the translational forces (denoted as $\nu_{1,t}$) and an equivalent "rotational force" (denoted as $\nu_{1,r}$):

$$c_e = \int_{t_0}^{t_f} \left(\|\mathbf{F}(t)\|_1 + \frac{1}{b_0} \|\boldsymbol{\tau}(t)\|_1 \right) dt \tag{5.9}$$

Substituting (5.9) into Equation (5.5) and taking κ to be unity, the non-dimensional control effort, ν_1, for a spacecraft becomes

$$\nu_1 = \frac{1}{g_0\, m(t_f - t_0)} \int_{t_0}^{t_f} \left(\|\mathbf{F}(t)\|_1 + \frac{1}{b_0} \|\boldsymbol{\tau}(t)\|_1 \right) dt \tag{5.10}$$

where b_0 is 1 m. It is worthwhile to note that Equation (5.10) takes the form of a thrust-to-mass ratio. This is important as the

Using the Taguchi analysis method[1] [137]–[139], the effect of varying the thrust-to-mass and torque-to-mass ratios on ν_1 was explored by solving for the optimal control (using $\mathbb{GPOPS} - \mathbb{II}$ [89]) to complete Case 0 of the Standard Test Framework (described in Section 5.2). The Taguchi method provides a framework in which to assess the effect on the output of a process by varying selected inputs using a minimal number of experiments [137], [138], [140]. For this study, the control variables (i.e., process inputs) are chosen to be the thrust-to-mass (T/m) and torque-to-inertia (τ/J) ratios; the performance criterion (i.e., process output) was chosen to be the non-dimensional control effort ν_1 defined in Equation (5.10). The magnitude of the range of the control variables was selected to be 100 by surveying relevant RPO and formation flying spacecraft missions [7], [11], [141]–[150]. This value corresponds to the approximate magnitude range of thrust-to-mass ratios surveyed in Table 5.3 and illustrated in Figure 5.4. In order to capture any possible nonlinearities in the performance criterion response, three discrete levels, $\{1/100, 1, 100\}$, for the control variables were used. Resultantly, a $\mathrm{L}_9(3^2)$ (two factors at three levels resulting in nine experiments) Taguchi orthogonal array[2] was used [138].

[1] See Appendix B for more information on the Taguchi analysis method.
[2] See Table B.1 for the structure of the $\mathrm{L}_9(3^2)$ orthogonal array.

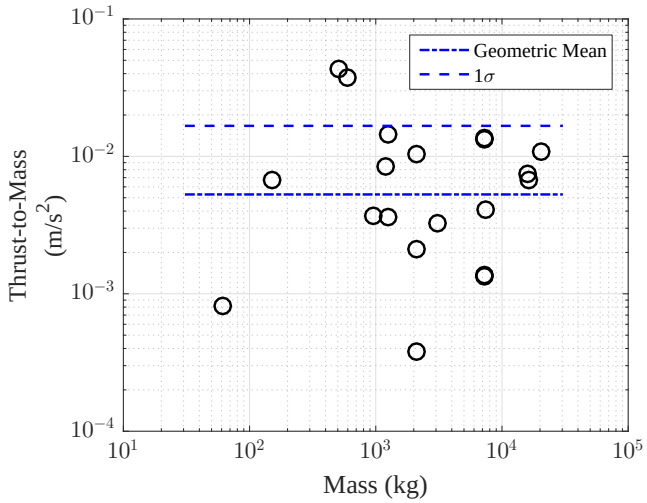
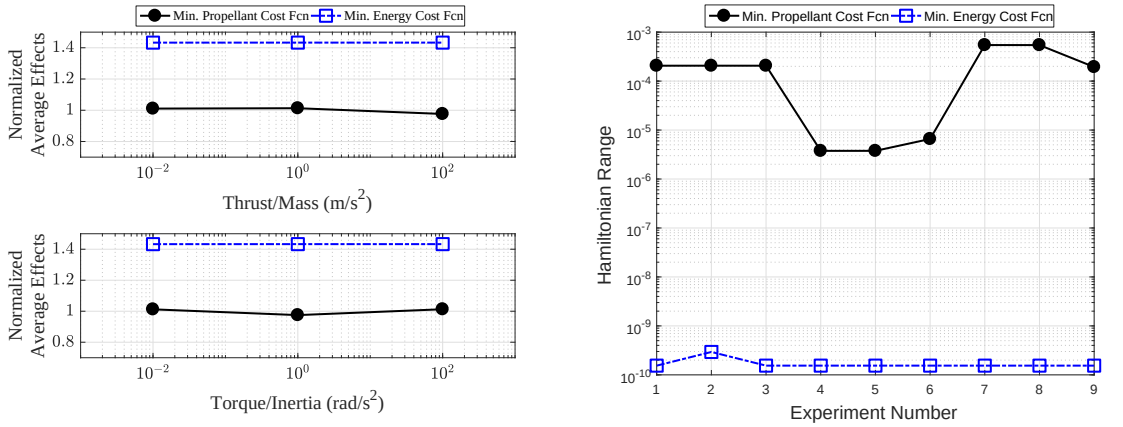


Figure 5.4. Distribution of thrust-to-mass ratios for surveyed spacecraft. Adapted from [7], [11], [141]–[150].



(a) Average effect of scaling on ν_1

(b) Hamiltonian variation for each experiment

Figure 5.5. ν_1 scaling study results showing its scaling invariance.

To simplify the (numerical) experiments, the nominal mass and inertia were chosen to be 1 kg and 0.1 kg·m^2, respectively; the maximum force and torque was then scaled according to this input. The design of experiment (DOE) was performed using both a minimum-propellant and minimum-energy cost functional in the optimizer. This was done in order to validate the results (via comparison) between the computationally tractable minimum-energy and (less tractable) minimum-propellant objective functions. The average effects of the two control variables on the performance criterion (i.e., ν_1) was then computed as described in [137] and normalized by the average output of ν_1

across all experiments using the minimum-propellant cost function. Lastly, to assess the validity of the solutions, the range of values of the Hamiltonian was computed.

As illustrated by Figure 5.5b, the range of the Hamiltonian for each cost function across the nine experiments is small; as such, the Hamiltonian can be considered constant and the resulting solution valid [76]. As illustrated by Figure 5.5a, the non-dimensional control effort, ν_1, given in Equation (5.10), was demonstrated to be *invariant* to the scaling of the maximum rotational and translation accelerations for a relevant magnitude range.

5.5.2 Terrestrial Vehicles

To illustrate the extensibility of this framework, the guidance comparison metric is formulated for autonomous (terrestrial) vehicles (e.g., automobiles and aircraft). A common metric used to measure the control effort of an autonomous vehicle is the power output of the engine, P_{out}, [151]–[155],

$$c_e = \int_0^{t_f} P_{\text{out}}(t)\, dt \tag{5.11}$$

To non-dimensionalize ν_1, κ is chosen to be inversely proportional to the average velocity, $\bar{v}$, of the vehicle throughout the maneuver. The resulting non-dimensional control metric is

$$\nu_1 = \frac{1}{\bar{v}\; g_0\; m(t_f - t_0)} \int_0^{t_f} P_{\text{out}}(t)\, dt\, dt \tag{5.12}$$

Scaling the work performed by the vehicle to complete the maneuver by $\bar{v}^{-1}$ results in Equation (5.12) effectively becoming equivalent to a force-to-weight ratio – resulting in a ν_1 similar to that of a spacecraft, effectively turning a vehicle into a spacecraft!

The mean power-to-weight ratio of 336 automobiles surveyed between 1975-2016 was found to be 0.0696 kW/kg $\pm$ 0.0143 kW/kg with a magnitude range of 2.22 [156]. The distribution of the surveyed automobile power-to-weight ratios are illustrated in Figure 5.6. Compared to a magnitude range of the spacecraft thrust-to-mass ratios of 113.78, the power-to-weight ratio for terrestrial vehicles is significantly smaller. Since the magnitude range of terrestrial vehicles is much smaller than that for spacecraft, the scaling will behave similar to that observed in Figure 5.5a.

5.6 Summary

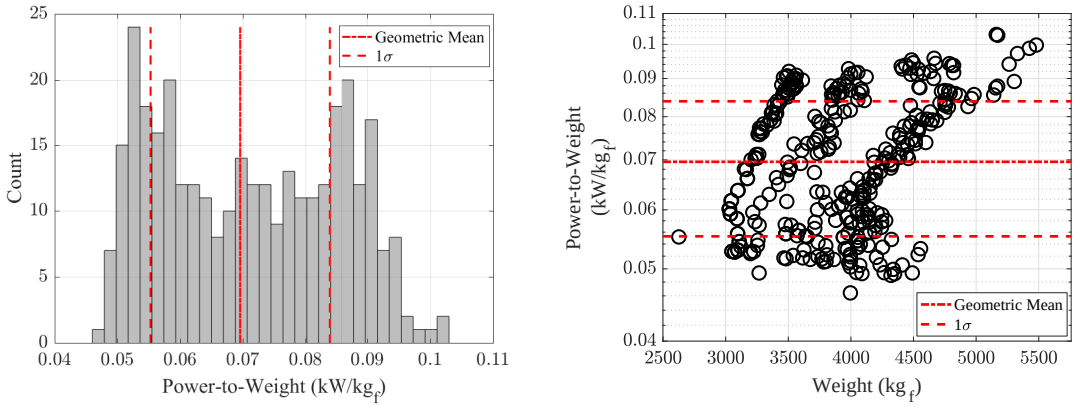
As suggested by Max Planck, before various guidance algorithms can be compared, a (standard) benchmark and comparison framework must be considered [131]. The

proposed *Standard Test Framework* defines relevant scenarios that test various capabilities of candidate scenarios in a relevant and dynamic environment. The *Standard Test Framework* is used in Chapter 6 to evaluate and compare several guidance and control algorithms. In addition, the proposed global Guidance Comparison Metric provides a unique capability to compare the results of different algorithms completing the same maneuver. This proposed metric, never seen before in literature, allows for an

Table 5.3. Summary of surveyed RPO spacecraft design specifications. Adapted from [7], [11], [141]–[150].

Spacecraft	Mass (kg)	Design Mission	Attitude Control Method	RPO Thruster Force, (N)	Thrust-to-Mass Ratio, (mm/s^2)
ASTRO (Orbital Express)	1,089	Formation Flying	Thruster	3.5	3.67
ATV-1	20,252	ISS Resupply	Thruster	220	10.86
Cluster	1,184	Science	Spin-Stabilized	10	8.45
Cygnus Orb-4	7,492	ISS Resupply	Thruster	31	4.14
Hayabusa	510	Science	3-Axis Stabilized	22	43.14
Hayabusa 2	590	Science	3-Axis Stabilized	22	37.29
HTV-2	16,500	ISS Resupply	Thruster	110	6.67
HTV-4	16,000	ISS Resupply	Thruster	120	7.50
Magnestospheric Multiscale (MMS)[1]	1,250	Science	Spin-Stabilized	4.5 18	3.60 14.40
OSIRIS-Rex[1]	2,110	Science	3-Axis Stabilized	0.8 4.5 22	0.38 2.13 10.43
Mango (PRISMA)	150	Formation Flying	3-Axis Stabilized	1	6.67
Progress[1]	7,290	ISS Resupply	Thruster Thruster	9.8 98	1.34 13.44
Prox-1	61.5	Formation Flying	3-Axis Stabilized	0.05	0.81
Rosetta	3,065	Science	Thruster	10	3.26
				Geometric Mean, mm/s^2	5.29
				Standard Deviation, mm/s^2	11.10
				Range,mm/s^2	42.76
				Magnitude Range	113.78

[1] Multiple thrusters used for RPO.



(a) Histogram (b) Distribution

Figure 5.6. Power-to-weight ratios of surveyed automobiles. Adapted from [156].

unbiased comparison of algorithms across multiple system architectures. Due to its generic formulation, the Guidance Comparison Metric can be applied to virtually any type of mechanical system. A case study exploring one possible use of the Guidance Comparison Metric is illustrated in Chapter 7.

CHAPTER 6:
Evaluation of Real-Time Spacecraft Guidance and Control Methods

> The striking difference of what from now on we will succinctly refer to as algorithmic control (AC) with prior practice is the absence of a control law ... that, before even implementation, has decided ... the appropriate resulting control action for each sensory input. ... On the contrary, AC departs from this paradigm and delegates the control action, including the "control law" design itself, to an online algorithm during execution.
>
> —Panagiotis Tsiotras and Mehran Mesbahi
> "Toward an algorithmic control theory" [50]

A journal version of the experimental setup and results for APF and AAPF methods presented in Section 6.3.1 is currently under review for publication in the IEEE *Transaction on Control Systems Technology* [71]. A conference version of the experimental setup and results for IDVD method presented in Section 6.3.2 and the LQ-MPC/RH method presented in Section 6.3.3 was originally presented at the 6th International Conference on Astrodynamics Tools and Techniques (ICATT) in Darmstad, Germany (14–17 March 2016) [52]. A journal version of this work has been published in the CEAS *Space Journal* [45]. A conference version of the experimental setup and results for the LQ-MPC/DH and NMPC methods presented in Section 6.3.3 was originally presented at the AIAA/AAS Astrodynamics Specialist Conference 2016 in Long Beach, CA (12–16 September 2016) [61]. Lastly, a conference version of the experimental setup and results for the proposed closed-loop RRT* method was originally presented at the 27th AAS/AIAA Spaceflight Mechanics Conference in San Antonio, TX (February 2017) [73].

6.1 Introduction

The challenge of efficiently performing a rendezvous and docking with a Target spacecraft in a cluttered and dynamic environment is addressed in this chapter. This particular problem incorporates the need for an algorithm to satisfy both system and path constraints (e.g., actuator saturation limits, obstacle keep-out zone constraints) in addition to reducing the fuel expenditure. To accomplish this, several guidance algorithms will be evaluated and compared using the proposed *Standard Test Framework*. This framework, introduced in Section 5.2, consists of four test scenarios that establish a baseline

and evaluates the path-planning and re-planning abilities of a guidance method in both a static and dynamic environment. Furthermore, the framework establishes several measurements which can be leveraged to compare the guidance algorithms, such as the control effort, maneuver time, computational cost, as well as the proposed Guidance Comparison metric, $\mathbf{z} = [\nu_1, \nu_2]^\mathsf{T}$. Utilizing a facility such as the NPS POSEIDYN test bed allows the algorithm to be tested in a relevant, computationally constrained environment in the presence of actuator response uncertainties and sensor noises.

The remainder of this chapter is devoted to solving the rendezvous and docking problem using several guidance methods which are assessed to be the state of the art in literature [26]. In particular, the APF, AAPF, IDVD, LQ-MPC, and a NMPC methods are compared to a RRT*-based method proposed in this chapter. To serve as a benchmark (and lower bound), the numerical optimal solution (via $\mathbb{GPOPS} - \mathbb{II}$) or the the proposed test scenarios for a FSS is presented. These results, as well as those derived from the experimental evaluation of the above methods will be compared utilizing the Guidance Comparison Metric .

6.2 Optimal Control Solutions to the Standard Test Framework

6.2.1 Optimal Control Problem Formulation

In order to establish a comprehensive benchmark as well as a lower bound for the non-dimension control effort ν_1, the minimum-propellant optimal control problem for the four test scenarios associated with the *Standard Test Framework* was formulated and solved. Defining the state vector to be $\mathbf{x}(t) = \left[x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}\right]^\mathsf{T} \in \mathbb{R}^{N_x}$ and the admissible control input $\mathbf{u}(t) = [f_x, f_y, \tau]^\mathsf{T} \in \mathbb{U} \subseteq \mathbb{R}^{N_u}$, the minimum-fuel optimal control problem for each roto-translational test case can be stated generically as:

$$\text{Minimize: } J = \int_0^{t_f} \|\mathbf{u}(t)\|_1 \, dt \tag{6.1a}$$

$$\begin{aligned}
\text{Subject To: } \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} && \text{(6.1b)}\\
|\mathbf{u}(t)| &\leq \mathbf{u}_{\text{max}} \text{ for } i = 1,2,3 && \text{(6.1c)}\\
\mathbf{x}(t_f) &= \mathbf{x}_f && \text{(6.1d)}\\
t_f &= t^f && \text{(6.1e)}\\
\hat{\mathbf{n}}_{\text{c1}} \cdot \mathbf{p}_{\text{dock}} - \hat{\mathbf{n}}_{\text{c1}} \cdot \mathbf{r}(t) &\leq 0 && \text{(6.1f)}\\
\hat{\mathbf{n}}_{\text{c2}} \cdot \mathbf{p}_{\text{dock}} - \hat{\mathbf{n}}_{\text{c2}} \cdot \mathbf{r}(t) &\leq 0 && \text{(6.1g)}\\
R_i - \|\mathbf{r}_{ci}\|_2 &\leq 0 && \text{(6.1h)}
\end{aligned}$$

where $\mathbf{A} \in \mathbb{R}^{N_x \times N_x}$ and $\mathbf{B} \in \mathbb{R}^{N_x \times N_u}$ are the state and control-input matrices, respectively, and are defined as

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{6.2a}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{3\times3} \\ \text{diag}\,(1/m, 1/m, 1/J_{zz}) \end{bmatrix} \tag{6.2b}$$

where $\mathbf{0}_{3\times3}$ denotes a 3×3 zero matrix, $\mathbf{I}_{3\times3}$ denotes a 3×3 Identity matrix, $\text{diag}(\cdot)$ indicates a square matrix whose diagonal elements are $(\cdot)$, and m and J_{zz} are the mass and inertial about the z axis, respectively. The desired terminal condition is denoted as x_f while t^f is the maximum time allotted to complete the maneuver. The docking cone corridor path constraint, represented by Equations (6.1f) and (6.1g), is active for each test case whereas the obstacle keep-out zone constraint, represented by Equation (6.1h) is only active for Cases 1-3.

To accommodate the docking cone corridor constraint, which is only active during terminal approach, a multi-phase OCP was formulated. The first phase consisted of the trajectory up to the docking cone corridor interface and includes only the obstacle keep-out zone (if applicable); the second phase consisted of the terminal approach inside the docking cone corridor. To ensure proper linkage between the two phases, denoted by $(\cdot)^{(1)}$ for Phase 1 and $(\cdot)^{(2)}$ for Phase 2, the following additional constraints are imposed:

$$\begin{aligned}
\mathbf{x}_0^{(2)} &= \mathbf{x}_f^{(1)} && \text{(6.3a)}\\
t_0^{(2)} &= t_f^{(1)} && \text{(6.3b)}\\
|\theta_f^{(2)} - \theta_f^{(1)}| &\leq 10 \text{ deg} && \text{(6.3c)}
\end{aligned}$$

The constraint specified by Equation (6.3a) ensures the final state of Phase 1, $\mathbf{x}_f^{(1)}$ is equal to the initial state of Phase 2, $\mathbf{x}_0^{(2)}$. Likewise, continuity of time is enforced by Equation (6.3b). The third linkage constraint given by Equation (6.3c) ensures the final attitude of the second phase, $\theta_f^{(2)}$ is smaller than the final attitude of the first phase, $\theta_f^{(1)}$.

6.2.2 Results

A solution for each test case for the *Standard Test Framework* was found via $\mathbb{GPOPS}-\mathbb{II}$ [89]. The resulting trajectories, as well as six equally-spaced attitude snapshots, are illustrated in Figure 6.1. A summary of the control effort for each test scenario is tabulated in Table 6.1. Additionally, the points of closest approach to the obstacle keep-out zone(s) are annotated with a red asterisk ($*$) in Figure 6.1 and presented for each test scenario in Table 6.2.

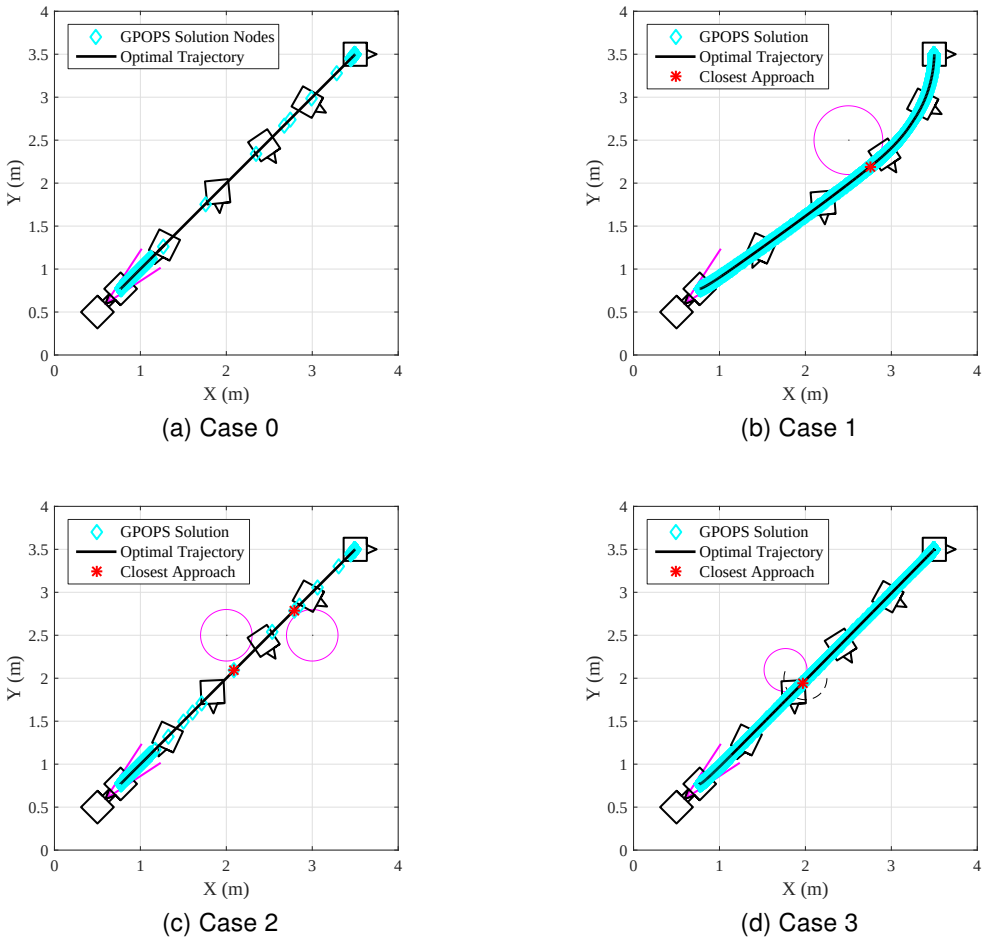
As illustrated by Table 6.1, the rotational control effort is two orders-of-magnitude smaller than the translational control effort. Likewise, the rotational component of ν_1 is two orders-of-magnitude smaller and does not significantly affect the overall ν_1. As a result, only the translational component of ν_1 is reported since it is approximately equal to the overall ν_1.

Table 6.1. Summary of the optimal control solutions.

	Translational Control Effort (N·s)	Rotational Control Effort (mN·m·s)	Translational ν_1^* $(\times 10^{-4})$
Case 0	0.606	7.20	0.345
Case 1	0.749	7.07	0.426
Case 2	0.606	7.20	0.345
Case 3	0.616	7.01	0.382

Table 6.2. Summary of closest approach for each test case.

	Closest Approach Distance (m)	Time of Closest Approach (s)
Case 0	– –	– –
Case 1	0 .000	80.76
Case 2	0.058	46.89
Case 3	0.000	101.41



(a) Case 0

(b) Case 1

(c) Case 2

(d) Case 3

Figure 6.1. Trajectories of the optimal control solution for the Standard Test Framework test cases with six equally-spaced (in time) attitude snapshots.

6.3 Formulation of Evaluated Real-Time Guidance and Control Methods

6.3.1 Artificial Potential Functions (APF)

A journal version of the guidance formulation presented in this sub-section is currently under review for publication in the IEEE *Transaction on Control Systems Technology* [71] as well as presented at the 2016 AAS/AIAA Spaceflight Mechanics Conference in Napa Valley, CA (February 2016) [32].

Constraint Handling

Two types of constraints are considered: a keep-out zone surrounding an obstacle and a docking cone corridor. These path constraints are realized via a repulsive potential, ϕ_r, that creates an area of higher potential in inadmissible regions of the configuration space, such as exclusion-zones surround an obstacle. Two different repulsive potential functions are considered: one for obstacle keep-out zones and one for the Target boundary constraint. A Gaussian function is used to represent a obstacle and its associated keep-out zone as is defined in the next sub-section [19], [28].

The second constraint type, the docking cone corridor, not only ensures a safe, collision-free final trajectory to the Target, but also guarantees the docking action occurs parallel to the docking axis [28]. The docking cone corridor is anchored at the Target docking interface and defined by two parameters, the cone half-angle, β, and the length of the corridor, ℓ_c, as illustrated in Figure 6.2. To enforce the docking cone constraint, a cardioid-like function consisting of the portions of four ellipses[1] whose boundary is defined by the distance, d, from the center of the Target spacecraft, $\mathbf{x}_t$, is given in polar coordinates as [28], [32],

$$d(\alpha) = \begin{cases} \dfrac{2b_1 a_1^2 \cos(\alpha)}{a_1^2 \cos^2(\alpha) + b_1^2 \sin^2(\alpha)}, & \alpha \in \left[0,\ \frac{\pi}{2}\right) \\ \dfrac{-2b_2 a_2^2 \cos(\alpha)}{a_2^2 \cos^2(\alpha) + b_2^2 \sin^2(\alpha)}, & \alpha \in \left[\frac{\pi}{2},\ \pi\right) \\ \dfrac{2b_2 a_3}{\sqrt{a_3^2 \cos^2(\alpha) + 4b_2^2 \sin^2(\alpha)}}, & \alpha \in \left[\pi,\ \frac{3\pi}{2}\right) \\ \dfrac{2b_1 a_3}{\sqrt{a_3^2 \cos^2(\alpha) + 4b_2^2 \sin^2(\alpha)}}, & \alpha \in \left[\frac{3\pi}{2},\ 2\pi\right) \end{cases} \tag{6.4}$$

[1] When $b_1 = b_2$, the resulting cardiod is composed of three distinct ellipses

where the parameters a_1, a_2, a_3, b_1, b_2, the distance d, and the angle α are defined in Figure 6.2. To satisfy docking cone corridor constraint, the semi-major axes a_1 and a_2 are chosen such that the Target keep-out zone constraint intersects with the docking cone corridor at $d(\pi/2 \pm \beta) = \ell_c / \cos(\beta)$. Therefore, the length of the semi-major axes, a_1 and a_2 which satisfy the docking cone corridor constraint:

$$a_1 = \sqrt{\frac{b_1^2 \sin^2(\alpha_1)(\ell_c + d_{\text{off}})}{2b_1 \cos(\alpha_1)\cos(\beta) - (\ell_c + d_{\text{off}})\cos^2(\alpha_1)}} \tag{6.5a}$$

$$a_2 = \sqrt{\frac{-b_2^2 \sin^2(\alpha_2)(\ell_c + d_{\text{off}})}{2b_2 \cos(\alpha_2)\cos(\beta) + (\ell_c + d_{\text{off}})\cos^2(\alpha_2)}} \tag{6.5b}$$

where d_{off} is the distance from the Target center of mass to the docking cone interface, $\alpha_1 = \pi/2 - \beta$ and $\alpha_2 = \pi/2 + \beta$. It is worthwhile to note, this approach can also be used to realize an asymmetric keep-out zone around the Target spacecraft due to both static and dynamic appendages such as antennae, articulated solar arrays, and robotic manipulators. This is done by selecting values for $b_1, b_2,$ and a_3 which encompass the appendage(s) and their respective workspaces [32].

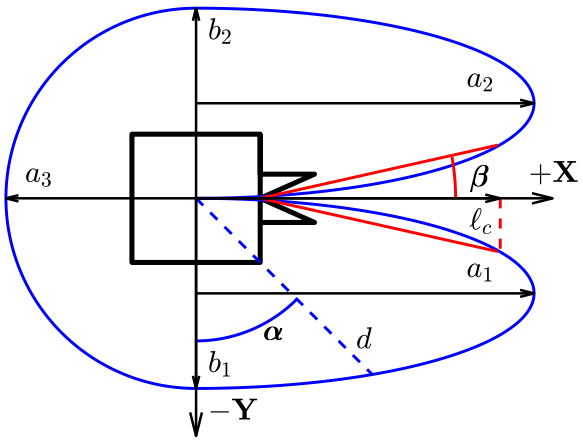


Figure 6.2. Notional Target keep-out zone and docking cone path constraints (blue) with overlaid docking cone constraint (red) in the Target body frame.

Artificial Potential Function Formulation

The Artificial Potential Function (APF) method utilizes the gradient of a potential field, which is composed of both attractive and repulsive potentials, to derive the necessary control inputs to reach the desired or goal position. The following development of the APF guidance method is adapted from [19], [28], [32].

The attractive potential, ϕ_a, establishes a global minimum in the workspace at the desired (terminal) configuration, $\mathbf{p}_f = [x_f, y_f, \theta_f]^\mathsf{T}$. The attractive potential was chosen to be a quadratic function given as

$$\phi_a = \frac{k_a}{2}\mathbf{r}_{cf}^\mathsf{T}\mathbf{Q}_a\mathbf{r}_{cf} \tag{6.6}$$

where $\mathbf{r}_{cf} = \mathbf{p}_c - \mathbf{p}_f$ is the relative difference in the Chaser configuration, $\mathbf{p}_c = [x, y, \theta]^\mathsf{T}$, and the desired configuration, $k_a \in \mathbb{R}_+$ is a strictly positive constant, and $\mathbf{Q}_a \in \mathbb{R}^{3\times3}$ is a symmetric, positive-definite, attractive potential shaping matrix. In order to avoid certain portions of the configuration space, the repulsive potential, ϕ_r, creates an area of higher potential effectively "pushing" the Chaser away from these regions [19]. Two different repulsive potential functions are considered — one for an obstacle keep-out zone and one for the Target boundary constraint. Keep-out zones centered around an obstacle are represented using a Gaussian function [19], [20], [28],

$$\phi_r = \sum_{i=1}^{N_o} \psi_i \exp\left[-\frac{\mathbf{r}_{co_i}^\mathsf{T}\mathbf{N}_i\mathbf{r}_{co_i}}{\sigma_i}\right] \tag{6.7}$$

where N_o is the number of obstacles, $\mathbf{r}_{co_i}$ is the relative position of the Chaser with respect to the center point of i^{th} obstacle, σ_i and ψ_i are the width and height parameters for the i^{th} obstacle, and $\mathbf{N}_i \in \mathbb{R}^{3\times3}$ is a positive definite shaping matrix for the i^{th} repulsive potential. For this application, the repulsive potential shaping matrix is taken to be the Identity matrix, $\mathbf{I}_{3\times3}$, implying a circular obstacle keep-out zone.

In order to ensure the Chaser appropriately enters the docking cone constraint and remains clear of any appendages attached to the Target, the following repulsive potential function was selected as [32],

$$\phi_b = \frac{k_r}{2}\frac{\mathbf{r}_{cf}^\mathsf{T}\mathbf{Q}_b\mathbf{r}_{cf}}{\exp\left[\mathbf{r}_{cb}^\mathsf{T}\mathbf{P}_b\mathbf{r}_{cb} - 1\right]} \tag{6.8}$$

where $k_r \in \mathbb{R}_+$, $\mathbf{Q}_b, \mathbf{P}_b \in \mathbb{R}^{3\times3}$ are positive definite shaping matrices for the Target boundary constraint, and $\mathbf{r}_{cb}$ is the relative position of the Chaser with respect to the Target boundary constraint inertial position, $\mathbf{x}_b$. Note, explicit knowledge of $\mathbf{x}_b$ is not necessary as the relative position vector $\mathbf{r}_{cb}$ can be expressed as

$$\mathbf{r}_{cb} = d(\alpha)\frac{\mathbf{r}_{ct}}{\|\mathbf{r}_{ct}\|_2} - \mathbf{r}_{ct} \tag{6.9}$$

where $\mathbf{r}_{ct}$ is the relative position between the Chaser and Target, $d(\alpha)$ is given in Equation (6.4) and $\mathbf{x}_t$ is the Target state. From a geometric prospective, the shaping matrices

$\mathbf{Q}$ and $\mathbf{P}$ can be viewed as shaping the *height* and *width* of the potential function, respectively. The resulting total potential function, ϕ_{tot}, is defined as the superposition of the attractive and all repulsive potential functions:

$$\phi_{\text{tot}} = \phi_a + \phi_b + \sum_{i=1}^{N_o} \phi_{r_i} \tag{6.10}$$

Since finer control is desired when in close proximity of the Target and desired terminal state, a continuous feedback control law was chosen and is given as [32], [35],

$$\mathbf{u}(\mathbf{p}_c, \mathbf{p}_f, \dot{\mathbf{p}}_c) = -\mathbf{B}^{-1}\mathbf{K}_a \left(\dot{\mathbf{p}}_c + \nabla_x \phi_{\text{tot}}\right) \tag{6.11}$$

where $\mathbf{B} = \text{diag}[{}^1/_m,\, {}^1/_m,\, {}^1/_{I_z}]$, $\mathbf{K}_a \in \mathbb{R}^{3\times 3}$ is a positive gain matrix and $\nabla_x \phi_{\text{tot}}$ is given as

$$\begin{aligned}\nabla_x \phi_{\text{tot}} = k_a \mathbf{Q}_a \mathbf{r}_{cf} + k_r \exp\left[1 - \mathbf{r}_{cb}^{\mathsf{T}} \mathbf{P}_b \mathbf{r}_{cb}\right] \left(\mathbf{Q}_b \mathbf{r}_{cf} - \left(\mathbf{r}_{cf}^{\mathsf{T}} \mathbf{Q}_b \mathbf{r}_{cf}\right) \mathbf{P}_b \mathbf{r}_{cb}\right) \\ - \sum_{i=1}^{N_o} \frac{2\psi_i}{\sigma_i} \exp\left[-\frac{\mathbf{r}_{co_i}^{\mathsf{T}} \mathbf{N}_i \mathbf{r}_{co_i}}{\sigma_i}\right] \mathbf{N}_i \mathbf{r}_{co}\end{aligned} \tag{6.12}$$

Stability analysis of the APF control law showing $\mathbf{x}_c$ approaches $\mathbf{x}_f$ asymptotically can be shown via a Lyapunov analysis and is presented in [28], [35].

Adaptive Artificial Potential Function Formulation

The Adaptive APF (AAPF) method [19], [20] was developed in an effort to reduce fuel consumption while still providing effective and robust real-time obstacle avoidance. To achieve this, a time-varying attractive potential shaping matrix, $\mathbf{Q}_a(t)$, is updated according to an adaptive update law in order to follow a given reference trajectory. The upper bound on the optimality the AAPF method can achieve is bounded by the optimality of the reference trajectory. Note, the following development of the adaptive update laws are adapted from [19], [20].

To begin, the reference trajectory was generated by considering a rest-to-rest, obstacle-free, fixed-time, minimum-fuel OCP for a double-integrator system with a bounded control input. The resulting optimal control can be described as a *bang-coast-bang* maneuver, whose desired control input direction is anti-parallel to the $\hat{\mathbf{r}}_{cf}$ [76]. Therefore, the resulting reference trajectory follows the path described by $\mathbf{p}_{\text{ref}}(t) = -\mathbf{r}_{cf}(t)$ and a reference velocity, $\dot{\mathbf{p}}_{\text{ref}}(t) = -v_{\text{ref}}\hat{\mathbf{r}}_{cf}(t)$. Notably, for this application, the reference speed, v_{ref}, can be chosen arbitrarily within the capabilities of the system. For this study, the

reference speed was selected to be constant by specifying a final time as described in more detail in Section 6.6.1.

Since the translational and rotational motions of a spacecraft can be considered uncoupled, two independent adaptive laws can be developed. Consider first the translational motion of the Chaser with the configuration variable being $\mathbf{p} = [x, y]^{\mathsf{T}}$. The Cholesky factorization $\mathbf{Q}_a(t) = \mathbf{R}^{\mathsf{T}}(t)\mathbf{R}(t)$, where $\mathbf{R}(t)$ is the upper triangular matrix [19], [20],

$$\mathbf{R}(t) = \begin{bmatrix} \rho_{11}(t) & \rho_{12}(t) \\ 0 & \rho_{22}(t) \end{bmatrix} \tag{6.13}$$

is utilized to enforce the symmetric positive-definite condition of the time-varying attractive potential shaping matrix.

The weighting factors composing $\mathbf{R}(t)$ are then chosen such that increased efficiency is achieved while using the same continuous control law given in Equation (6.11) [19], [20]. This is achieved by first defining an error function as the difference between the reference velocity, $\dot{\mathbf{p}}_{\text{ref}}$ and the negative gradient of the attractive potential

$$\begin{aligned} \mathbf{e} &= \dot{\mathbf{p}}_{\text{ref}} - (-\nabla_x \phi_a) \\ &= -v_{\text{ref}} \frac{\mathbf{r}_{cf}}{\|\mathbf{r}_{cf}\|_2} + k_a \mathbf{R}^{\mathsf{T}} \mathbf{R} \mathbf{r}_{cf} \end{aligned} \tag{6.14}$$

The time derivative of the error function for a stationary terminal configuration is

$$\dot{\mathbf{e}} = -v_{\text{ref}} \left(\frac{\dot{\mathbf{p}}_c}{\|\mathbf{r}_{cf}\|_2} - \frac{\mathbf{r}_{cf} \mathbf{r}_{cf}^{\mathsf{T}} \dot{\mathbf{p}}_c}{\|\mathbf{r}_{cf}\|_2^3} \right) + k_a \left(\dot{\mathbf{R}}^{\mathsf{T}} \mathbf{R} + \mathbf{R}^{\mathsf{T}} \dot{\mathbf{R}} \right) \dot{\mathbf{p}}_c \tag{6.15}$$

Letting $\boldsymbol{\rho} = [\rho_{11},\ \rho_{12},\ \rho_{22}]^{\mathsf{T}}$ the second term of Equation (6.15) can be parameterized as,

$$k_a \left(\dot{\mathbf{R}}^{\mathsf{T}} \mathbf{R} + \mathbf{R}^{\mathsf{T}} \dot{\mathbf{R}} \right) \dot{\mathbf{x}}_c = k_a \mathbf{K} \dot{\boldsymbol{\rho}}$$

where $\mathbf{K} \in \mathbb{R}^{2 \times 3}$ is

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \end{bmatrix}$$

and whose elements are given as

$$\begin{array}{ccc} k_{11} = 2\rho_{11}\tilde{x} + \rho_{12}\tilde{y}, & k_{12} = \rho_{11}\tilde{y}, & k_{13} = 0 \\ k_{21} = \rho_{12}\tilde{x}, & k_{22} = \rho_{11}\tilde{x} + 2\rho_{12}\tilde{y}, & k_{23} = 2\rho_{22}\tilde{y} \end{array}$$

where $\tilde{x} = x_c - x_f$ and $\tilde{y} = y_c - y_f$.

In order for the error to converge to zero asymptotically, it is desired to have $\dot{\mathbf{e}} = -k\mathbf{e}$ where $k \in \mathbb{R}_+$ [19], [20]. The resultant adaptive update law for specifying the attractive potential weights is given as follows:

$$\dot{\boldsymbol{\rho}} = v_{\text{ref}}\mathbf{K}\left(\mathbf{K}\mathbf{K}^{\mathsf{T}}\right)^{-1}\left(\frac{\dot{\mathbf{p}}_c}{\|\mathbf{r}_{cf}\|_2} - \frac{\mathbf{r}_{cf}\mathbf{r}_{cf}^{\mathsf{T}}\dot{\mathbf{p}}_c}{\|\mathbf{r}_{cf}\|_2^3}\right) - \mathbf{K}\left(\mathbf{K}\mathbf{K}^{\mathsf{T}}\right)^{-1}\left(k_a\mathbf{R}^{\mathsf{T}}\mathbf{R}\dot{\mathbf{p}}_c - k\mathbf{e}\right) \quad (6.16)$$

which will drive $\mathbf{e} \to 0$ asymptotically.

Following a similar procedure, an adaptive update law for the rotational motion of the Chaser can be derived. Since the Chaser rotational motion is constrained to a single axis of rotation, $R = \rho_{33}$. The error function is given as

$$\begin{aligned} e_\theta &= \omega_{\text{ref}} - (-\nabla_\theta \phi_a) \\ &= -\omega_{\text{ref}}\frac{\theta_c - \theta_f}{|\theta_c - \theta_f|} + k_a\rho_{33}^2(\theta_c - \theta_f) \end{aligned} \quad (6.17)$$

where ω_{ref} is the reference angular speed. The resulting adaptive update law that will drive $e_\theta \to 0$ asymptotically is given as follows:

$$\dot{\rho}_{33} = \omega_{\text{ref}}\frac{(|\theta_c - \theta_f|^{-1} - (\theta_c - \theta_f)^2||\theta_c - \theta_f|^{-3})}{2k_a\rho_{33}(\theta_c - \theta_f)} - \frac{k_a\rho_{33}^2\dot{\theta}_c - ke_\theta}{2k_a\rho_{33}(\theta_c - \theta_f)} \quad (6.18)$$

The adaptive update law given in Equation (6.16) and Equation (6.18) has several critical implications [20]. First, a nonzero initial value for $\boldsymbol{\rho}$ is required for an initial positive-definite shaping matrix. Secondly, the matrix $\mathbf{K}$ is non-singular for all $\mathbf{x} \neq \mathbf{x}_f$. As the Chaser approaches its terminal state, the matrix $\mathbf{K}$ becomes ill-conditioned causing the adaptive estimates to diverge and tend to infinity [20]. Resultantly, the adaptive estimates are bounded by first defining a convex set for each estimate [20]

$$\Lambda_i = \left\{\mu \in \mathbb{R} \mid \rho_i^- \leq \mu \leq \rho_i^+\right\}$$

given an upper and lower bound ρ^+, ρ^- for each estimate and letting

$$\tilde{\boldsymbol{\rho}} = [\rho_{11},\ \rho_{12},\ \rho_{13},\ \rho_{22},\ \rho_{23},\ \rho_{33}]^{\mathsf{T}}$$

be the augmented vector of adaptive estimates. Next, a projection algorithm is used to ensure the adaptive estimates remain bounded [20],

$$\dot{\hat{\rho}} = \mathrm{Proj}\left(\dot{\rho}\right) = \begin{cases} \dot{\rho}_i & \text{if } \rho_i \in \Lambda_i \\ \dot{\rho}_i & \text{if } \rho_i = \rho_i^- \text{ and } \dot{\rho}_i \geq 0 \\ \dot{\rho}_i & \text{if } \rho_i = \rho_i^+ \text{ and } \dot{\rho}_i \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{6.19}$$

Bounding the adaptive estimates allows the algorithm to be used for close-proximity and rendezvous maneuvers.

6.3.2 Inverse Dynamics in the Virtual Domain (IDVD)

A journal version of the guidance formulation presented in this sub-section has been accepted for publication in the CEAS Space Journal [72] as well as presented at the 6th International Conference on Astrodynamics Tools and Techniques (ICATT) in Darmstad, Germany (14–17 March 2016) [52].

The IDVD method, a quasi-optimal, direct guidance method, solves the optimal control by parameterizing both the trajectory and time using basis functions [54], [72], [157]. The resulting parameter optimization determines the set number of coefficients while minimizing a user-specified cost functional and satisfying system and path constraints. The following development of the IDVD method is adapted from [52], [72].

The polynomials describing the time are defined as a function of a virtual time, $\kappa_t \in [0, \kappa_{t,f}]$ as [52], [72],

$$t_x(\kappa) = \sum_{i=0}^{n_{t_x}} d_{a_i} \kappa_t^i \tag{6.20a}$$

$$t_y(\kappa) = \sum_{i=0}^{n_{t_y}} d_{b_i} \kappa_t^i \tag{6.20b}$$

where d_{a_i} and d_{b_i} are the coefficients for the polynomial describing time in the respectively x and y directions; n_{t_x} and n_{t_y} are the number of coefficients for the polynomials describing the t_x and t_y trajectories, respectively. Note, no constraints are placed on n_{t_x} and n_{t_y} being the same. Likewise, the polynomials describing the x and y trajectories

are defined as [52], [72],

$$x(\kappa) = \sum_{i=0}^{n_x} a_i \kappa_t^i \tag{6.21a}$$

$$y(\kappa) = \sum_{i=0}^{n_y} b_i \kappa_t^i \tag{6.21b}$$

where a_i and b_i are the coefficients for the x and y trajectories, respectively; n_x and n_y are the number of coefficients for the polynomials describing the x and y trajectories respectively. Similar to the time polynomials, n_x and n_y need not be the same. Thus, the OCP is converted into a parameter optimization where the coefficients $a_i, b_i, d_{a_i},$ and d_{b_i} must be determined.

Before continuing, it is important to note the three constraints that are placed on the time polynomials [52], [72]. The first constraint is the placed on the terminal condition of the time polynomials, specifically, the polynomial final time, $t(\kappa_{t,f}t)$, is less than the specified maximum time t_f:

$$t(\kappa_{t,f}) < t_f \tag{6.22}$$

The second constraint only applies when different time polynomials are used. In this case, d_{b_1} is chosen such that $t_x(\kappa_{t,f}) = t_y(\kappa_{t,f})$. The last constraint is placed on the derivative of the time polynomials such that it must be strictly positive over the entire range of the virtual times

$$t_x'(\kappa_t) = \sum_{i=0}^{n_{t_x}} i d_{a_i} \kappa^{i-1} > 0 \quad \forall \quad \kappa \tag{6.23a}$$

$$t_y'(\kappa_t) = \sum_{i=0}^{n_{t_y}} i d_{b_i} \kappa^{i-1} > 0 \quad \forall \quad \kappa \tag{6.23b}$$

Next, the polynomials describing the time derivatives of the x and y trajectories with respect to the time polynomials are given as follows [52], [72]:

$$\dot{x} = \frac{x'}{t'_x} \qquad \dot{y} = \frac{y'}{t'_y} \tag{6.24a}$$

$$x'(\kappa_t) = \sum_{i=0}^{n_{t_x}} i a_i \kappa_t^{i-1} \qquad y'(\kappa_t) = \sum_{i=0}^{n_{t_x}} i b_i \kappa_t^{i-1} \tag{6.24b}$$

$$\ddot{x} = \frac{x''}{(t'_x)^2} - \frac{\dot{x}}{(t'_x)^2} t''_x \qquad \ddot{y} = \frac{y''}{(t'_y)^2} - \frac{\dot{y}}{(t'_y)^2} t''_y \tag{6.24c}$$

$$x'' = \sum_{i=0}^{n_{t_x}} i(i-1) a_i \kappa_t^{i-2} \qquad y'' = \sum_{i=0}^{n_{t_x}} i(i-1) b_i \kappa_t^{i-2} \tag{6.24d}$$

$$t''_x(\kappa_t) = \sum_{i=0}^{n_{t_x}} i(i-1) d_{a_i} \kappa_t^{i-2} \qquad y''(\kappa_t) = \sum_{i=0}^{n_{t_y}} i(i-1) d_{b_i} \kappa_t^{i-2} \tag{6.24e}$$

It is worthwhile to note, the number of polynomial coefficients can be reduced given the (estimated) state of the Chaser and the desired terminal state [52], [72] – resulting in a smaller parameter optimization process. Specifically, the initial and terminus conditions of the trajectory-describing polynomials for the rendezvous and docking scenario are

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{x}_c, & \dot{\mathbf{x}}(0) &= \dot{\mathbf{x}}_c \\ \mathbf{x}(t_f) &= \mathbf{x}_f, & \dot{\mathbf{x}}(t_f) &= 0, \quad \ddot{\mathbf{x}}(t_f) = 0 \end{aligned} \tag{6.25}$$

As a result of these initial and terminal constraints, the minimum order for the trajectory polynomials while still meeting system and path constraints is $n_{x,y} \geq 5$. Additionally, the minimum order of the time polynomial is $n_{t_x,t_y} \geq 1$.

Selecting the cost functional to be the L_1-norm of the control inputs, the resulting instantiation of the IDVD method is a minimum-propellant, bounded-time optimal control problem:

$$\text{Minimize: } J = \int_0^{t_f} \|\mathbf{u}(t)\|_1 \, dt \tag{6.26a}$$

$$\begin{aligned}
\text{Subject To: } \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} && \text{(6.26b)}\\
|u(t)_1| &\leq u_{\text{max}} && \text{(6.26c)}\\
|u(t)_2| &\leq u_{\text{max}} && \text{(6.26d)}\\
\mathbf{x}(t_f) &= \mathbf{x}_f && \text{(6.26e)}\\
t_{x,y}(\kappa_{t,f}) &< t^f && \text{(6.26f)}\\
d_{a_i,b_i} &> 0 && \text{(6.26g)}\\
t_x(\kappa_{t,f}) &= t_y(\kappa_{t,f}) && \text{(6.26h)}\\
R_i - \|\mathbf{r}_{ci}\|_2 &\leq 0 && \text{(6.26i)}\\
\hat{\mathbf{n}}_{\text{c1}} \cdot \mathbf{p}_{\text{dock}} - \hat{\mathbf{n}}_{\text{c1}} \cdot \mathbf{r}(t) &\leq 0 && \text{(6.26j)}\\
\hat{\mathbf{n}}_{\text{c2}} \cdot \mathbf{p}_{\text{dock}} - \hat{\mathbf{n}}_{\text{c2}} \cdot \mathbf{r}(t) &\leq 0 && \text{(6.26k)}
\end{aligned}$$

where $\mathbf{x} = [x, y]^{\mathsf{T}}$; $\mathbf{u} = [f_x, f_y]^{\mathsf{T}}$; t_f is the maximum time allowed to complete the maneuver; and $\mathbf{r}_{co}$ is the relative position of the Chaser with respect to the obstacle. The docking cone corridor constraint is implemented as two hyperplanes defined by a point $\mathbf{p}_{(\cdot)}$ with normal vectors $\hat{\mathbf{n}}_{(\cdot)}$ while the obstacle keep-out zone is kept as a nonlinear, non-convex path constraint. To solve the resulting nonlinear programming problem, the open-source Interior Point OPTimizer (IPOPT) solver [158] is implemented onboard the FSS at a rate of 5 Hz in order to produce a feedback action. It is important to note, the constraints specified in Equation (6.26) are only satisfied at the equally-spaced discrete nodes considered by the IPOPT solver.

6.3.3 Model Predictive Control (MPC)

A journal version of the guidance formulation presented in this sub-section has been published in the CEAS Space Journal [72] as well as presented at the 6th International Conference on Astrodynamics Tools and Techniques (ICATT) in Darmstad, Germany (14–17 March 2016) [52] and the 2016 AIAA/AAS Astrodynamics Specialist Conference in Long Beach, CA [61].

While based primarily on the LQR, the MPC framework is solves the constrained optimal control problem over a finite horizon subject to both system and path constraints [55], [72] – compared to the IDVD method which optimizes over the entire trajectory . Feedback action is obtained by implementing the first control input and resolving the problem using the current estimated state of the system [52], [55], [61], [72]. The following development of the LQ-MPC and NMPC formulations are adapted from [52], [61], [72]. The MPC guidance method, as implemented, is based on the

following optimal control problem:

$$\begin{aligned}\text{Minimize: } J &= (\mathbf{x}_N - \mathbf{x}_f)^\mathsf{T} \mathbf{P} (\mathbf{x}_N - \mathbf{x}_f) \\ &+ \sum_{i=0}^{N-1} (\mathbf{x}_{k+i} - \mathbf{x}_f)^\mathsf{T} \mathbf{Q} (\mathbf{x}_{k+i} - \mathbf{x}_f) \\ &+ \sum_{i=0}^{N-1} \mathbf{u}_{k+i}^\mathsf{T} \mathbf{R} \mathbf{u}_{k+i} \end{aligned} \tag{6.27a}$$

$$\begin{aligned}\text{Subject To: } \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k && (6.27b)\\ |\mathbf{u}_{k_i}| &\leq \mathbf{u}_{\max} \text{ for } i = 1, 2, \ldots, N_u && (6.27c)\\ R_i - \|\mathbf{r}_{ci}\|_2 &\leq 0 && (6.27d)\\ \hat{\mathbf{n}}_{\text{c1}} \cdot \mathbf{p}_{\text{dock}} - \hat{\mathbf{n}}_{\text{c1}} \cdot \mathbf{r}(t) &\leq 0 && (6.27e)\\ \hat{\mathbf{n}}_{\text{c2}} \cdot \mathbf{p}_{\text{dock}} - \hat{\mathbf{n}}_{\text{c2}} \cdot \mathbf{r}(t) &\leq 0 && (6.27f)\end{aligned}$$

where state and control vectors at sampling instant k are denoted by $\mathbf{x}_k \in \mathbb{R}^{N_x}$ and $\mathbf{u}_k \in \mathbb{R}^{N_u}$ respectively; $\mathbf{A} \in \mathbb{R}^{N_x \times N_x}$ and $\mathbf{B} \in \mathbb{R}^{N_x \times N_u}$ are the respective discrete state and control input matrices; $\mathbf{r} = [x, y]^\mathsf{T}$ is the position of the Chaser; $\mathbf{x}_f$ is the terminal state; and N is the length of the horizon. The matrices $\mathbf{Q} \in \mathbb{R}^{N_x \times N_x}$ and $\mathbf{R} \in \mathbb{R}^{N_u \times N_u}$ define the positive definite relative weighting on the path and control effort, respectively, while $\mathbf{P} \in \mathbb{R}^{N_x \times N_x}$ is the solution to the discrete Riccati equation [55].

Two instantiations of the MPC method are considered, a Linear-Quadratic MPC (LQ-MPC) and a Nonlinear MPC (NMPC). Compared to the nonlinear variant, the LQ-based approach requires both linear dynamics as well as linear inequality constraints [72]. The nonlinear obstacle keep-out zone constraint is linearized through two methods, a rotating hyperplane method [52], [61], [72] as well as a novel dual-hyperplane method [61]. Compared to the LQ-based method, NMPC directly handle both nonlinear dynamics and constraints. Additionally, both implementations of the MPC method handle the docking cone corridor constraint in a similar fashion to the IDVD method. As implemented, the LQ-MPC method uses a publicly-available quadratic-programming solver [159] and resolves the optimal control problem every 3-5 s while NMPC uses IPOPT [158] and resolves the optimal control problem every 3 s.

6.4 Closed-Loop Rapidly-Expanding Random Trees

A conference version of this work was presented at the 2017 Spaceflight Mechanics Conference in San Antonio, TX (February 2017) [73].

6.4.1 Overview of Sampling-Based Methods

The RRT Algorithm

The Rapidly-Exploring Random Tree, or RRT, algorithm was introduced by LaValle in 1998 [160] as a kinodynamic path-planning method to quickly search high-dimensional, non-convex spaces for a feasible trajectory [64], [66]. This sampling-based path planner addressed the need for a general-purpose path-planner that scales well with the dimension of the problem.

Before continuing, it is worthwhile to note some key concepts and nomenclature. For the planar[1] translational guidance problem, the configuration space is denoted as $\mathbb{X} \subseteq \mathbb{R}^2$ and the configuration variable be denoted as $\mathbf{q} \in \mathbb{R}^2$. A tree, denoted as $\mathcal{T}$, has a single root node q_0 that is its initial configuration [161] and contains no cycles [161]. That is, each node, not including the root, has only one incoming edge, termed the parent node and denoted as q_{par}, but can be the parent to numerous other nodes, termed child nodes. This results in a unique path from any node belonging to the tree, denoted as $q_i \in \mathcal{T}$, to the root node. Lastly, a newly added node, denoted as q_{new} is commonly referred to as a child node. [161].

The RRT algorithm, illustrated in Algorithms 1 and 2, begins by initializing a tree, $\mathcal{T}$ with root node, q_0, which is the initial configuration of the system, in this application the Chaser, at the start of the algorithm. The objective of the RRT algorithm is find a *feasible*, if one exists, from q_0 to the goal configuration, $q_f \in \mathbb{X}$ given any fixed obstacles, $\mathbb{X}_{\text{obs}} \in \mathbb{X}$ [64], [66], [160].

The tree is first initialized using the $\texttt{Initialize_Tree}(\cdot)$ function where the algorithm attempts to connect the root node to the goal node with a collision free path using a user-defined, system-specific steering function $\texttt{Steer}(\cdot)$ [160]. If such a connection is possible, the goal node is added to the tree and the algorithm is completed. In the event this is not possible, the obstacle-free configuration space,

$$\mathbb{X}_{\text{free}} = \mathbb{X} \setminus \mathbb{X}_{\text{obs}} \tag{6.28}$$

is uniformly sampled via the $\texttt{Sample}(\cdot)$ function in line 3 to produce a random configuration, q_{rand}. Given some user-selected metric (e.g., Euclidean distance), the nearest node, q_{nearest}, to q_{rand} is determined using the $\texttt{Nearest}(\cdot)$ function. For the RRT algorithm, the parent node, q_{par}, of q_{rand} is chosen to be q_{nearest} [160]. The control input, $\mathbf{u}_i$, required to take the system from q_{nearest} to q_{rand} is determined via $\texttt{Steer}(q_{\texttt{nearest}}, q_{\texttt{rand}})$ function (line 3). If a collision-free path between q_{nearest} and q_{new} exists, the resulting

[1] Note, for the three-dimensional case, the configuration space is denoted as $\mathbb{X} \subseteq \mathbb{R}^3$ and the configuration variable be denoted as $\mathbf{q} \in \mathbb{R}^3$.

configuration from applying the steering law to the system, q_{new}, is added to the tree; otherwise, the tree reamins unchanged and q_{rand} is discarded.

This process is completed until the maximum number of iterations is obtained [160].

Algorithm 1 The RRT Algorithm [160].

$\mathcal{T} \leftarrow \texttt{Initialize_Tree}(q_0)$
for $i = 1$ **do** $N_{q_{\text{max}}}$
  $q_{\text{rand}} \leftarrow \texttt{Sample}(\mathbb{X}, \mathbb{X}_{\text{obs}})$
  $\mathcal{T} \leftarrow \texttt{Extend}(\mathcal{T}, q_{\text{rand}})$ ▷ q_{nearest} is the parent node of q_{new}
end for

Algorithm 2 The RRT `Extend` function [160].

function EXTEND_RRT($\mathcal{T}, q_{\text{rand}}$)
  $q_{\text{nearest}} \leftarrow \texttt{Nearest}(\mathcal{T}_i, x_{\texttt{rand}})$
  $(q_{\text{new}}, \mathbf{u}_i) \leftarrow \texttt{Steer}(q_{\texttt{nearest}}, q_{\texttt{rand}})$ ▷ q_{nearest} is the parent node of q_{new}
  if $\texttt{Obstacle_Free}(q_{\texttt{nearest}}, q_{\texttt{new}})$ **then**
    $\mathcal{T} \leftarrow \texttt{AddNode}(\mathcal{T}, q_{\texttt{new}}, q_{\texttt{nearest}}, u_i)$
  end if
  return $\mathcal{T}$ ▷ Return updated tree
end function

It is worthwhile to note, the fidelity of collision-checking is problem specific. Collision-checking can range from a simple point-mass model to three-dimensional collision detection algorithms [162]–[165].

This method has several key advantages including excellent exploration of the search space, probabilistic completeness, and computational simplicity. Additionally, due to the construction methodology of the tree, the RRT algorithm tends to, on average, partition largely unexplored areas into smaller regions [64], [160]. However, the resulting path from the root to the goal the algorithm converges to will never be optimal [65]. In terms of computational complexity, the RRT algorithm is bounded by the computational efficiency of the nearest neighbor search [64], [66], [160]. Karaman and Frazzoli showed the computational complexity of the RRT algorithm at each iteration is at least on the order $\mathcal{O}(\log(N_{v_i})$, where N_{v_i} is the number of nodes at the end of the i^{th} iteration [65].

This framework was later extended by Frazzoli et al. by exploring a new technique to connect the newly sampled nodes to the tree and on a method to handle *moving* obstacles. Rather than connecting the sampled node to its nearest neighbor, Frazzoli proposed choosing the parent based on an optimization heuristic [161]. Given an optimal steering law, each node in the tree, q_i, is sorted based on the obstacle-free cost to

from q_i to q_{new} plus the cost accumulated at node q_i. Starting with the lowest obstacle-free cost node, the trajectory between q_{new} and each node in the tree is checked to ensure the trajectory is collision-free for some duration of time [161]. The parent node is selected as the first node which satisfies this condition. This method was then applied to several systems including the path planning for a small autonomous helicopter – which was the motivation for the development of this algorithm – in addition to the attitude slew planning for a spacecraft with multiple constraints [83], [161].

The RRT* Algorithm

Until Karaman and Frazzoli introduced an optimality extension to the RRT framework, no RRT algorithm was shown in literature to converge to an optimal solution. This extension initially introduced in 2010, called RRT* [65], [166], fuses the asymptotic optimality of rapidly-exploring random graphs with the tree-like structure of the RRT algorithm. The RRT* algorithm achieves the improved performance through the way it extends the tree given a new node. Specifically, the parent node is chosen from a set of $k-$nearest nodes, denoted as $\mathbb{X}_{\text{near}}$, that minimizes a user-defined cost functional, as illustrated in Algorithm 3.

It is worthwhile to note, there are typically three different costs are associated with each node (x_i) [44]. First, the total cost is the cost accumulated from the root q_0 to node q_i and is represented by the function $\texttt{Total_Cost}(q_i)$. The incremental cost is the cost associated with going from the parent of node q_i to node q_i, represented by the function $\texttt{Delta_Cost}(q_i)$. Lastly, the cost-to-go is a lower-bound on the cost associated with moving the system from the current node to goal configuration, q_f and is typically taken to be the (unconstrained) Euclidean distance [44].

In an attempt to further optimize the tree by reducing the overall cost of the resulting path, the cost of going from the newly added node q_{new} to each nearby node $q_{\text{near}} \in \mathbb{X}_{\text{near}}$ is evaluated. When the total cost from the root node q_0 to the node q_{near} ($\texttt{Total_Cost}(q_{\texttt{near}})$) is found to be greater than the total cost from q_0 to the q_{new} plus the incremental cost go to q_{near} ($\texttt{Delta_Cost}(q_{\texttt{near}})$), q_{new} is chosen to be the new parent node of q_{near}. When this occurs, the tree is said to be *rewired* [44], [65], [166], [167]. The rewiring process is illustrated in Algorithm 5.

Example path planning solutions of the RRT* algorithm are presented in Figure 6.3 utilizing a minimum fuel (L_1-norm) cost function, 150 nodes, for various numbers of nearest neighbors, k. Note, for $k = 1$, the RRT* algorithm becomes the RRT algorithm as it does not consider any additional nodes other than the closest. It can be observed, as the k increases, the resulting graph becomes more directed from the start point towards the goal and more efficient paths are found. Increasing the value of k from one

Algorithm 3 The RRT* `Extend` function.

function EXTEND_RRT*($\mathcal{T}, x_{\text{rand}}$)
  $q_{\text{nearest}} \leftarrow \texttt{Nearest}(\mathcal{T}, q_{\texttt{rand}})$
  $(q_{\text{new}}, \mathbf{u}_i) \leftarrow \texttt{Steer}(q_{\texttt{nearest}}, q_{\texttt{rand}})$
  if $\texttt{Obstacle_Free}(q_{\texttt{nearest}}, q_{\texttt{new}})$ **then**
    $\mathbb{X}_{\text{near}} \leftarrow \texttt{Near}(\mathcal{T}, q_{\texttt{new}}, k)$
    $(q_{\text{min}}, \mathbf{u}_i) \leftarrow \texttt{Choose_Parent}(\mathbb{X}_{\texttt{near}}, q_{\texttt{nearest}}, q_{\texttt{new}}, u_i)$
    $\mathcal{T} \leftarrow \texttt{AddNode}(\mathcal{T}, q_{\texttt{min}}, q_{\texttt{new}}, u_i)$
    $\mathcal{T} \leftarrow \texttt{Rewire}(\mathcal{T}, \mathbb{X}_{\texttt{near}}, q_{\texttt{min}}, q_{\texttt{new}})$
  end if
  return $\mathcal{T}$
end function

Algorithm 4 The RRT* `Choose_Parent` function.

function CHOOSE_PARENT($\mathbb{X}_{\text{near}}, q_{\text{nearest}}, q_{\text{new}}$)
  $q_{\text{min}} \leftarrow q_{\text{nearest}}$ ▷ Initialize q_{min} and c_{min}
  $c_{\text{min}} \leftarrow \texttt{Total_Cost}(x_{\texttt{nearest}}) + \texttt{Delta_Cost}(x_{\texttt{new}})$
  for all $q_{\text{near}} \in \mathbb{X}_{\text{near}}$ **do** ▷ Select parent which minimizes the cost
    $(q', \mathbf{u}') \leftarrow \texttt{Steer}(q_{\texttt{near}}, q_{\texttt{new}})$
    if $\texttt{Obstacle_Free}(x', x_{\texttt{new}})$ **then**
      $c' \leftarrow \texttt{Total_Cost}(q_{\texttt{near}}) + \texttt{Delta_Cost}(q')$
      if $c' < c_{\text{min}}$ **then**
        $c_{\text{min}} \leftarrow c'$; $q_{\text{min}} \leftarrow q'$; $\mathbf{u}_{\text{min}} \leftarrow \mathbf{u}'$ ▷ Update best parent node
      end if
    end if
  end for
  return $q_{\text{min}}, \mathbf{u}_{\text{min}}$
end function

to five results in 77.1% difference between the RRT and RRT* methods; letting k be 25 results in a 142.6% difference between the RRT and RRT* algorithms.

Extensions and Applications of the RRT* Method

In an effort to increase the convergence rate, Qureshi et al. modified the RRT* algorithm by biasing the randomly sampled points with the gradient of an APFs by placing an attractive potential at the goal point and repulsive potentials on the obstacles [67]. This method was shown to reduce the time, number of iterations, and cost associated with the first solution that was found. A similar approach proposed by Pharpatara et al. creates a convex cone from which the randomly sampled points are drawn [68]. This

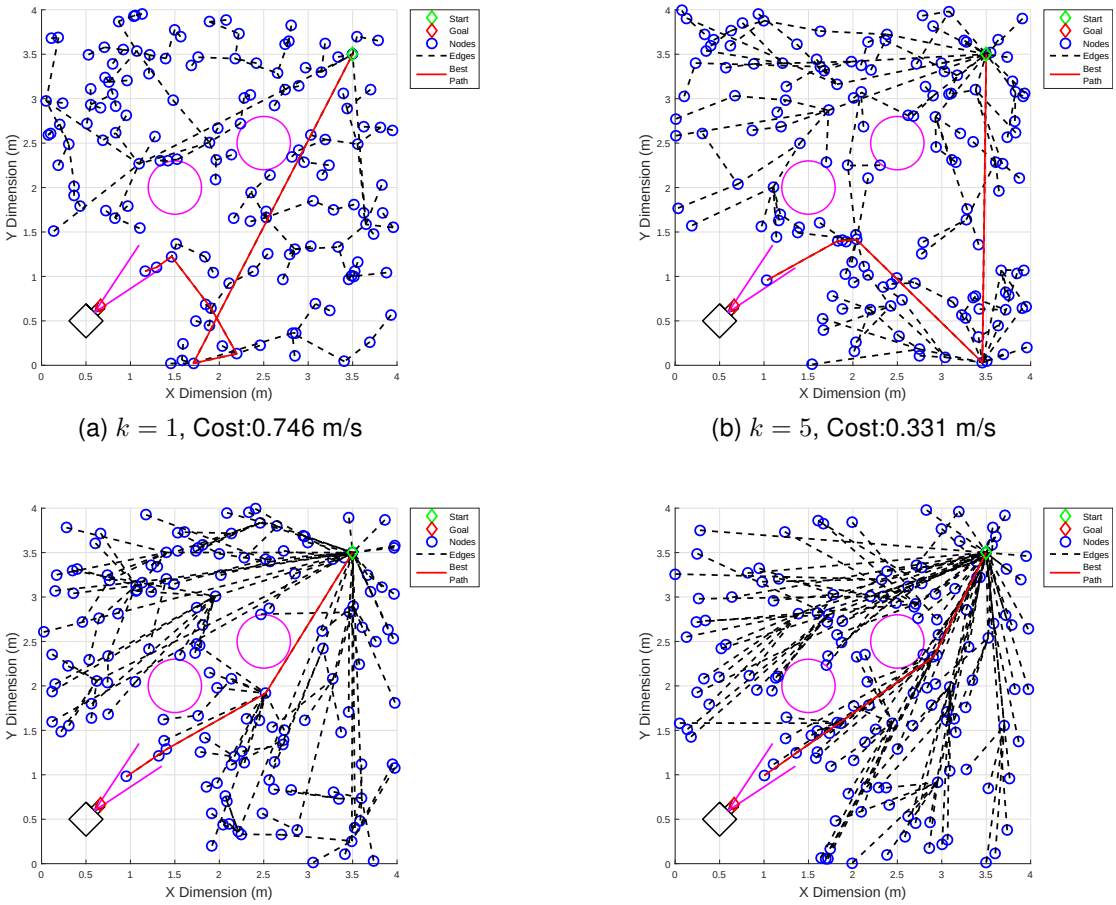
Algorithm 5 The RRT* `Rewire` function.

function REWIRE($mathcalT, \mathbb{X}_{\text{near}}, x_{\text{min}}, x_{\text{new}}$)
  for $x_{\text{near}} \in \mathbb{X}_{\text{near}} \setminus \{x_{\text{min}}\}$ **do**
    $(x', u') \leftarrow \texttt{Steer}(x_{\texttt{new}}, x_{\texttt{near}})$
    if $\texttt{Obstacle_Free}(x_{\texttt{new}}, x')$ **then**
      if $\texttt{Total_Cost}(x_{\texttt{new}}) + \texttt{Delta_Cost}(x') < \texttt{Total_Cost}(x_{\texttt{near}})$ **then**
        $\mathcal{T} \leftarrow \texttt{Reconnect}(\mathcal{T}, x_{\texttt{new}}, x_{\texttt{near}})$
      end if
    end if
  end for
  return $\mathcal{T}$
end function

method was found to reduce the average number of iterations by half (compared to the RRT* method); the average path length was reduced by approximately 15.3%. Several other extensions to the RRT* algorithm are detailed in [168].

The RRT* algorithm has been applied to several real-time systems. Karaman et al. extended this algorithm for use with autonomous wheeled robot by first selecting and storing part of the generated path, called the committed trajectory, at the end of every planning cycle [44]. While the robot is executing the committed trajectory, the algorithm re-roots the tree at the end of the committed trajectory and continues to improve the uncommitted portion of the path. When the committed trajectory is completed, the algorithm selects a new committed trajectory and repeats until the maneuver is finished. Additionally, this algorithm was applied to the *Talos* vehicle, MIT's entry to the DARPA Urban Challenge [42], [169]. In this application, the connection between existing nodes and a newly-sampled node is performed through a nonlinear, closed-loop simulation. This closed-loop simulation ensures vehicle stability throughout the challenge.

it is worthwhile to note, the systems discussed thus far utilized the RRT-based method strictly as a path planner. In each implementation discussed, there was a lower-level controller which would guides the system from one configuration to the next. It is the intention of this work to utilize a RRT*-based method coupled with Harmonic Potential Functions as a feedback controller for the translational guidance and control of a spacecraft in the presence of (a) dynamic obstacle(s). The difficultly arises as a spacecraft is not a kinematic vehicle (i.e., not controlled by its velocity) and must manage and take into account its linear momentum to ensure collision avoidance.



(a) $k = 1$, Cost:0.746 m/s

(b) $k = 5$, Cost:0.331 m/s

(c) $k = 10$, Cost:0.150 m/s

(d) $k = 25$, Cost:0.125 m/s

Figure 6.3. Example path planning solutions using the RRT* algorithm for various numbers of nearest neighbors, k, and 150 nodes.

6.4.2 Harmonic Potential Functions (HPF) Applied to the RRT

In this work, Harmonic Potential Functions (HPFs) are utilized to generate a Navigation Function which biases the generated random states towards the goal point and obstacle-free space. Paths are generated by following the negative gradient of the Navigation Function – which is commonly referred to as the velocity potential [36], [39], [40]. To achieve this, elementary solutions to the Laplace equation, $\nabla^2\phi = 0$, are considered [36], [38], [39]. While the primary development is for two dimensions, extension to three dimensions is trivial through the use of spherical coordinates [36]. Thus, only elementary elements which can be extended to three-dimensions are considered. Since these elementary solutions have strong ties to incompressible fluid flow[1], the

[1] It is worthwhile to note, solutions to Laplace's equation shares strong ties to Electrostatics [170].

subsequent development of the potential field leverages nomenclature and notation from Aerodynamics adopted from Reference [36].

To formulate the underlying navigation function (at each sampling instance), elements are added to the workspace incrementally, exploiting the linearity of the Laplace equation and its solutions [36], thus simplifying the selection of the strengths associated with each element.

First, a uniform flow that is directed towards the Chaser spacecraft from the goal position is added to the workspace [39], [40], [43],

$$\begin{aligned} \phi_{\text{uf}} &= v_{\text{ref}} \left\|\mathbf{r}_{cf}\right\|_2 \cos\left(\alpha_f\right) \\ &= v_{\text{ref}} \left(x\cos\left(\alpha_f\right) + y\sin\left(\alpha_f\right)\right) \end{aligned} \tag{6.29}$$

where v_{ref} is the reference speed of the chaser; $\mathbf{r}_{cf}$ is the relative position Chaser with respect to the desired terminal position; and α_f is the angle between the Chaser and the terminal position given as

$$\alpha_f = \tan^{-1}\left(\frac{y - y_f}{x - x_f}\right)$$

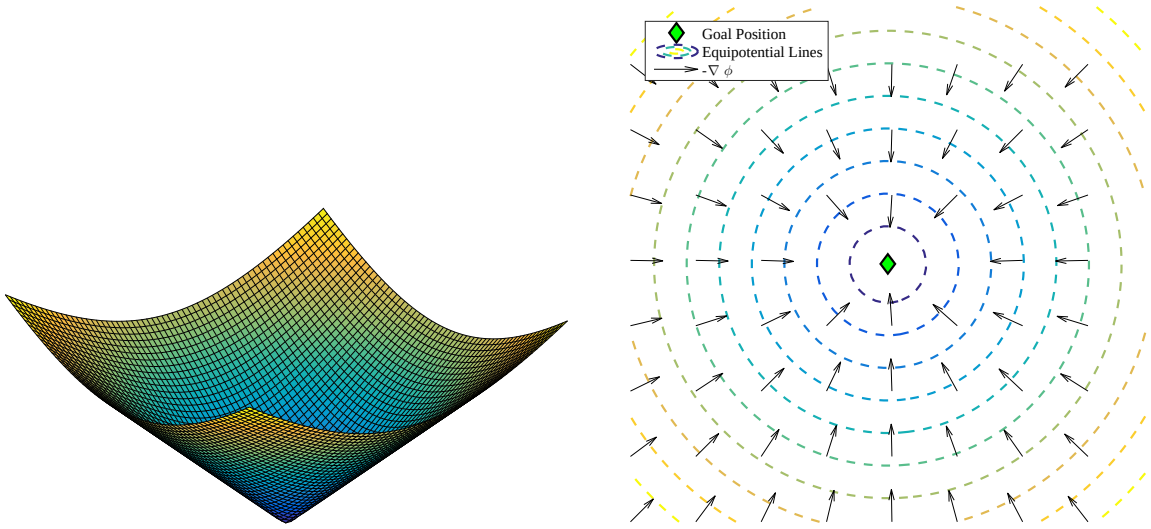
The resulting velocity potential due to the uniform flow is

$$-\nabla\phi_{x,\text{uf}} = -v_{\text{ref}} \begin{bmatrix} \cos\left(\alpha_f\right) \\ \sin\left(\alpha_f\right) \end{bmatrix} \tag{6.30}$$

The resulting potential field, illustrated in Figure 6.4, has zero potential at the goal position and increases radially-outward form the goal position. From Equation (6.30), the uniform flow will bring the Chaser from any point in the potential field to the desired terminal position — but it does not take into account any path constraints such as the docking cone corridor constraint or obstacle keep-out zone. The inclusion of the uniform flow eliminates numerical noise due to the small gradients associated with source, sink, and doublet elements at large distances [39], [40]. As it will be illustrated shortly for the case of a doublet, the radial component of the velocity potential of these elements in a polar coordinate system is inversely proportional to the distance $\mathbf{r}$ to the [36],

$$\nabla\phi_r = \frac{\partial\phi}{\partial r} \propto \frac{1}{\left\|\mathbf{r}\right\|_2} \tag{6.31}$$

Next, to account for the docking cone corridor path constraint, a doublet – the resulting elementary solution arising from a source and sink of equal strength which are made to approach each other [36] – is added to the workspace (that already contains the uniform) its central influx (anti-efflux direction, see Figure 6.5) pointed towards the



(a) 3-D Representation of the Uniform Flow Potential (b) 2-D Uniform Flow Potential Contours and Gradient

Figure 6.4. Elementary 2-D uniform flow centered at the goal position.

goal position and parallel to the docking axis. This creates a global minimum in the workspace in the vicinity of the goal position at $-\infty$. Note, as long as the goal position is sufficiently close to the Target position, this is a sufficient approximation of locating the global minimum at the goal position as the HPF is not being directly used for navigation, but rather to influence the random selection of a configuration. Additionally, this approximation allows the use of a doublet to enforce the docking cone corridor path constraint in a similar fashion the cardioid function enfroces this constraint for the APF-based guidance methods (illustrated in Figure 6.2) [32], [71].

The general expression for the potential due to a doublet is

$$\phi_d = \frac{\kappa_{d_i}}{\|\mathbf{r}_{cd_i}\|_2} \cos\left(\gamma_{d_i}\right) \tag{6.32}$$

where $\mathbf{r}_{cd_i}$ is the relative position of the Chaser in the potential field with respect to the i^{th} doublet; $\kappa_{d_i} \in \mathbb{R}$ is the strength of the doublet; γ_{d_i} is the angle between the Chaser and the i^{th} doublet given as

$$\gamma_{d_i} = \tan^{-1}\left(\frac{y - y_{d_i}}{x - x_{d_i}}\right)$$

Note, a positive value of κ_d indicates the flow proceeds outward from source-side of the doublet in the negative direction [36]. For simplicity, the flow direction into (influx) or out of (efflux) is with respect to the negative gradient of the doublet potential, as this is the

trajectory the Chaser will follow. That is, the directions of each flux follows the direction of the negative gradient. The resulting potential field and velocity potential field of a doublet is illustrated in Figure 6.5.

The target doublet strength, κ_{tar}, is selected by first considering the total potential of the field given the two elements added thus far:

$$\phi = v_{\text{ref}} \left\|\mathbf{r}_{cf}\right\|_2 \cos\left(\alpha_g\right) + \frac{\kappa_{\text{tar}}}{\left\|\mathbf{r}_{cf}\right\|_2} \cos\left(\gamma_t\right) \tag{6.33}$$

where $\mathbf{r}_{cf}$ is the relative position of the Chaser with respect to the desired terminal position. Applying the *von Neumann* boundary condition – which specifies the flow must be tangential to the boundary of a solid obstacle [40] – to the target doublet implies the radial velocity potential along the boundary of the target doublet is zero, $v_r = \partial\phi/\partial r = 0$, yields:

$$v_r = v_{\text{ref}} \cos\left(\alpha_g\right) - \frac{\kappa_{tar}}{\left\|\mathbf{r}\right\|_2^2} \cos\left(\gamma_t\right) = 0 \tag{6.34}$$

Note, at the boundary of the target doublet, $\left\|\mathbf{r}\right\|_2 = R$, where R is the radius of the equipotential needed to realize the docking cone corridor. Treating each equipotential lobe of the doublet as an ellipse (with similar semi-major and semi-minor axes), the necessary radius to satisfy the docking cone corridor can be found by applying Equations (6.5a) and (6.5b):

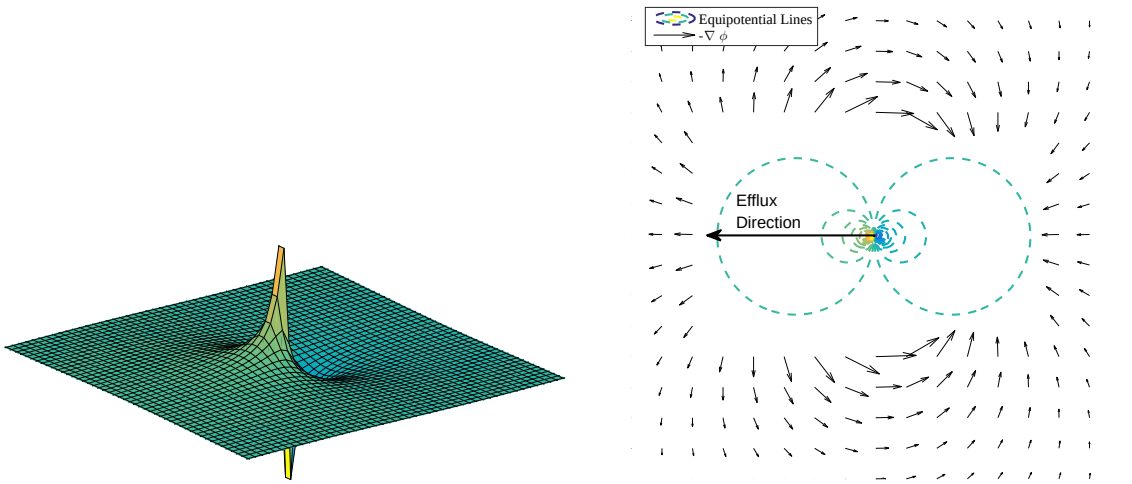
$$R = \frac{\ell_c + d_{\text{off}}}{2\cos\left(\beta\right)\cos\left(\theta_t\right)} \tag{6.35}$$

given the docking cone half-angle, β, corridor length,ℓ_c, and offset d_{off} and the attitude of the docking axis, θ_t. Note, for the potential field consisting of these only the uniform flow centered on the terminal position and the doublet centered on the Target, $\alpha_g = -\gamma_t$. Substituting Equation (6.35) into Equation (6.34) and solving for target doublet strength yields:

$$\kappa_{\text{tar}} = -v_{\text{ref}} \left(\frac{\ell_c + d_{\text{off}}}{2\cos\left(\beta\right)\cos\left(\theta_t\right)}\right)^2 \tag{6.36}$$

Lastly, doublets can be added throughout the workspace to represent circular obstacle keep-out zones with radius R_i [38], [40], [43]. To enable greater safety, each doublet is free to rotate such that the negative velocity potential (i.e., the efflux from the doublet) "pushes" the Chaser away from the obstacle. The resulting potential due to an obstacle doublet is

$$\phi_{r,i} = -\frac{\kappa_i}{\left\|\mathbf{r}_{co_i}\right\|_2} \cos\left(\gamma_i\right) \tag{6.37}$$



(a) 3-D Representation of a Doublet Potential (b) 2-D Doublet Potential Contours and Gradient

Figure 6.5. Elementary 2-D Doublet Potential Flow.

where γ_i is the angle between the Chaser and the i^{th} obstacle given as

$$\gamma_i = \tan^{-1}\left({}^{y-y_i}/_{x-x_i}\right)$$

and $\mathbf{r}_{co_i}$ is the relative position of the Chaser with respect to the i^{th} obstacle. The last remaining step is to determine the doublet strength associated with each obstacle.

Assuming the spacing between multiple objects is sufficiently large such that there is little near-field interaction between two or more doublets, each doublet in the workspace can be treated independently [40]. Due to the linearity of Laplace's Equation and subsequent elementary solutions, the strength of obstacle doublet is the superposition of the doublet strengths associated with the individual interactions between the obstacle doublet and other elementary solutions present in the workspace. That is, the total strength of the i^{th} doublet is the sum of the individual doublets strengths when considering all the other elements independently and can be written as

$$\kappa_i = \kappa_{\text{uf}} + \kappa_{\text{tar}} \tag{6.38}$$

where $\kappa_{i,uf}$ is the doublet strength due to the interaction of the i^{th} obstacle doublet and the uniform flow while $\kappa_{i,tar}$ is the doublet strength due to the interaction of the i^{th} obstacle doublet and the Target doublet. By applying the *von Neumann* boundary condition to each obstacle doublet, the strength of the doublet can be found as a function of the radius associated with the i^{th} obstacle keep-out zone.

First, consider the interaction between the obstacle doublet and the uniform flow. Similar to as before, at the boundary of the circular obstacle, the radial velocity is zero and $\mathbf{r}_{cd} = R_i$, yielding an obstacle doublet strength due to the interaction with the uniform flow of

$$\kappa_{i,\text{uf}} = v_{\text{ref}} R_i^2 \cos(\alpha_g) \tag{6.39}$$

Similarly, the obstacle doublet strength due to the interaction with the Target doublet is

$$\kappa_{i,\text{tar}} = \kappa_{\text{tar}} \frac{R_i^2}{\left(\|\mathbf{r}_{o_i t}\|_2 - R_i\right)^2} \frac{\cos(\gamma_t)}{\cos(\gamma_i)} \tag{6.40}$$

where $\mathbf{r}_{o_i t}$ is the relative position of the obstacle with respect to the Target spacecraft. Substituting Equation (6.39) and Equation (6.40) into Equation (6.38), the resultant obstacle doublet strength is given as

$$\kappa_i = v_{\text{ref}} R_i^2 \cos(\alpha_g) + \kappa_{\text{tar}} \frac{R_i^2}{\left(\|\mathbf{r}_{it}\|_2 - R_i\right)^2} \frac{\cos(\gamma_t)}{\cos(\gamma_i)} \tag{6.41}$$

The resulting potential field given the elementary solutions, docking cone corridor constraint, and any obstacles is

$$\phi_{\text{tot}} = v_{\text{ref}} \|\mathbf{r}_{cf}\|_2 \cos(\alpha_g) + \frac{\kappa_{\text{tar}}}{\|\mathbf{r}_{ct}\|_2} \cos(\gamma_t) - \sum_{i=1}^{N_o} \frac{\kappa_i}{\|\mathbf{r}_{co_i}\|_2} \cos(\gamma_i) \tag{6.42}$$

The negative gradient of the total potential field in polar coordinates, $\nabla\phi(r,\theta) = \left(\frac{\partial\phi}{\partial r}, \frac{1}{r}\frac{\partial\phi}{\partial\theta}\right)$, is given as follows:

$$-v_r = -\frac{\partial\phi}{\partial r} = -v_{\text{ref}} \cos(\alpha_g) + \frac{\kappa_{\text{tar}}}{\|\mathbf{r}_{ct}\|_2 r} \cos(\gamma_t) - \frac{\kappa_i}{\|\mathbf{r}_{co_i}\|_2^2} \cos(\gamma_i) \tag{6.43a}$$

$$-v_\theta = -\frac{1}{r}\frac{\partial\phi}{\partial\theta} = v_{\text{ref}} \sin(\alpha_g) + \frac{\kappa_{\text{tar}}}{\|\mathbf{r}_{ct}\|_2 r} \sin(\gamma_t) + \frac{\kappa_i}{\|\mathbf{r}_{co_i}\|_2^2} \sin(\gamma_i) \tag{6.43b}$$

In order to convert the negative velocity potential from a Polar coordinate system in Equation (6.43) to a Cartesian coordinate system, the following transformation is utilized:

$$-\nabla\phi_x = \mathbf{R}(\alpha_g)\nabla\phi_{\text{uf}}^{\mathsf{T}}(r,\theta) + \mathbf{R}(\gamma_t)\nabla\phi_{\text{tar}}^{\mathsf{T}}(r,\theta) + \sum_{i=1}^{N_o} \mathbf{R}(\gamma_i - \theta_t)\nabla\phi_{\text{r}_i}^{\mathsf{T}}(r,\theta) \tag{6.44}$$

where the rotation $\mathbf{R}(\cdot)$ is defined as

$$\mathbf{R}(\cdot) = \begin{bmatrix} \cos(\cdot) & -\sin(\cdot) \\ \sin(\cdot) & \cos(\cdot) \end{bmatrix} \tag{6.45}$$

given the angle the Chaser makes to each elementary solution.

6.4.3 CL-RRT*–HPF for Spacecraft Guidance and Control

The proposed sampling-based guidance method, which is given the prefix CL for closed-loop, is predicated on the RRT* algorithm (described in Section 6.4.1) and includes the capability to bias the configuration space using HPFs as the underlying Navigation function. The subtle nuances to this baseline algorithm is the biasing of the random samples from the configuration space with a HPF (in a manner similar to [67]) in addition to utilizing this algorithm in feedback form — thus achieving a "closed-loop".

Feedback action is achieved in a manner similar to the IDVD [52], [72] and MPC [52], [61] methods by implementing the control to transfer the system from the current (root) node to the next node in the resulting path and resolving the path-planning problem. Instead of completely rebuilding the tree every guidance sampling period, the proposed closed-loop RRT* (CL-RRT*) guidance methods reuse the previous solution, if one was found. In a similar fashion to the Dynamic RRT introduced by Fergurson et al. [171], the proposed CL-RRT* checks the previous nodes and trajectories for a collision with obstacle space $\mathbb{X}_{\text{obs}}$ and removes them from the previous solution. The verified nodes are then used to initialize a new tree that is rooted at the current position of the Chaser.

For spacecraft proximity maneuvering, the "default" choice of a cost functional is to minimize propellant usage – implying a minimum-propellant optimal control problem (described in Section 2.4.1 and 6.2). As such, a minimum ΔV cost functional was chosen as the metric by which to assign the parent node to the randomly sampled configuration. That is, the connection between an existing node, q_i, and the candidate node, q_{new} which minimizes the L_1-norm of the change in velocity required to go from x_q to q_{new} *and* from q_{new} to the goal configuration q_f while satisfying the constraints

$$\|v_{\text{ref}}\hat{\mathbf{r}}_{\text{new},i} - \mathbf{v}_i\|_1 + \|v_{\text{ref}}\left(\hat{\mathbf{r}}_{f,\text{new}} + \hat{\mathbf{r}}_{i,\text{new}}\right)\|_1 \tag{6.46}$$

where $\hat{\mathbf{r}}_{\text{new},i}$ is the unit vector parallel to the line segment connecting q_i to q_{new} and $\hat{\mathbf{r}}_{f,\text{new}}$ is the unit vector parallel to the line segment connecting q_{new} to q_f.

In this proposed guidance strategy, a two-fold approach for the avoidance of a dynamic obstacle is utilized. Without fitting the motion of obstacle some predefined models, straight-line motion of the obstacle is assumed since the underlying motion is rectilinear. However, since this assumption may not necessarily hold, it is desired to adapt a

strategy which does not depend upon the velocity of the obstacle. The implication of not using this information is to propagate the position of the obstacle is a sufficiently fast guidance sampling rate.

The premise of this two-point strategy is adapted from [38] where the radius of the obstacle is artificially increased. In its original implementation, the virtual radius of the obstacle, R_i', was chosen to be

$$R_i' = R_i + V_a \Delta t \tag{6.47}$$

where V_a is the maximum approaching speed and Δt is the total time delay associated with the system. Since RRT-based methods were originally developed for kinematically-controlled and driftless vehicles[1], it is necessary to adapt it for a dynamically-controlled vehicle.

For a dynamically-controlled vehicle, such as a spacecraft, the smallest control input in terms of impulse is governed by the minimum (acceleration) impulse bit of the thruster

$$\Delta V = a_{\text{thr}} \Delta t_{\text{min}} \tag{6.48}$$

where a_{thr} is the acceleration due to the control input (i.e., thruster). Rearranging Equation (6.48) for the minimum time, Δt_{min}, and substituting into Equation (6.47) yields the virtual radius as a function of the minimum impulse bit of the Chaser:

$$R_i' = R_i + \frac{V_a^2}{a_{\text{thr}}} \tag{6.49}$$

Furthermore, the maximum approach velocity, V_a, is chosen to be

$$V_a^2 = |\mathbf{v}_c^{\mathsf{T}}||\mathbf{v}_i| \tag{6.50}$$

where $\mathbf{v}_c$ is the velocity of the Chaser and $\mathbf{v}_i$ is the instantaneous velocity of the i^{th} obstacle. It is important to note, defining the approach velocity in the traditional sense – as the relative velocity between the Chaser and the obstacle – can lead to temporary conditions where the relative velocity between the Chaser and obstacle are zero, thereby causing $R_i' \to R_i$. This can result in safety concerns where the Chaser moves too close to the obstacle causing a possible violation of the obstacle keep-out zone. Defining the approach velocity to be Equation (6.50) implies the approach velocity will only be zero when either the Chaser or obstacle is stationary.

[1] Recall a kinematically-controlled vehicle is one whose control input is its velocity. A driftless vehicle is a vehicle that "can be stopped instantaneously by setting the control input to zero" [42] – that is, “momentum-less systems”. Examples of kinematically-controlled and driftless vehicles include autonomous vehicles [38], [43], [44] as well as robotic manipulators under kinematic control [45]–[48].

The second point obstacle avoidance is integrated into the proposed strategy is into the generation of the HPF gradient. First, the virtual obstacle radius (defined by Equation (6.49)) can be used to determine the obstacle doublet strength given by Equation (6.41). Additionally, the effect of the obstacle's velocity on the gradient is computed as

$$\nabla\phi_{x,\text{tot}} = \nabla\phi_x + \sum_{i=1}^{N_o} \frac{R_i'}{\|\mathbf{r}_{co_i}\|_2}\mathbf{v}_i \tag{6.51}$$

where $\nabla\phi_x$ is the gradient of the static HPF (in the Cartesian coordinate system). The gradient in Equation (6.51) can then be directly applied to bias the randomly sampled configuration

$$q_{\text{rand, biased}} = q_{\text{rand}} + (-\nabla\phi_{x,\text{tot}}) \tag{6.52}$$

Lastly, when the algorithm terminates (either by successfully completing the path to the goal configuration or exceeding the number of iterations) the "solution node", denoted as q_s, is chosen as the node that minimizes the distance to the goal configuration q_f [44]. The solution path is then determined by traversing the tree from the q_s to the root node, q_0. The resulting commanded control input is the one control input associated with transferring the system from q_0 to the next node along the solution path – similar to how feedback action is achieved for the IDVD [52], [72] and MPC [52], [61] methods. The resulting path is then fed-back and the process repeated.

The remainder of this sub-section will incorporate (dynamic) obstacle avoidance into proposed guidance method.

6.5 Attitude Control

While each of the algorithms described in Section 6.3 are capable of controlling the attitude of the FSS, they are utilized for translation guidance and control only – with the exception of the APF and AAPF methods, which also provide attitude guidance and control. This was chosen since the single rotational DOF motion of the Chaser is largely unconstrained and not coupled to the translational dynamics of the FSS. Two forms of attitude control are implemented: a Minimum Fuel Slew plus PD control and a saturation controller [172]. A summary of the attitude control methods is listed in Table 6.3.

6.5.1 Saturation Attitude Controller

Compared to a wide range of controllers, including linear time (in)variant, nonlinear time invariance, discontinuous, and optimal controllers, the saturation controller was shown by Rao and Bernstein to have a lower maximum under 'nominal' conditions

Table 6.3. Summary of attitude control methods.

	APF	AAPF	IDVD	LQ-MPC	NMPC	RRT*	RRT*-HPF
APF-Based	×	×					
Min. Fuel + PD			×	×	×		
Saturation Controller						×	×

[172]. Note, by 'nominal', it is meant the conditions under which it was tuned. For off-nominal conditions, such as variations in mass, pole perturbations, measurement delays, and unmodeled dynamics, the saturation controller was observed to exhibit "Good" performance (on a scale of "Good", "Fair", "Poor", and "Very Poor") [172]. The resulting saturation controller implemented is given as,

$$u = -\operatorname{sat}_{u_{\max}}\left[s_b J_{zz}\omega_{\text{err}} + \operatorname{sat}_{\varepsilon}\left(s_a J_{zz}\theta_{\text{err}} + \left(\frac{s_a}{s_b}\right) J_{zz}\omega_{\text{err}}\right)\right] \tag{6.53}$$

where $s_a, s_b \in \mathbb{R}_+$; $J_{zz} \in \mathbb{R}_+$ is the moment of inertia of the FSS about its vertical axis; $\theta_{\text{err}} \in \mathbb{R}$ is the error attitude of the FSS; and $\omega_{\text{err}} \in \mathbb{R}$ is the error angular velocity of the FSS. The saturation controller was tuned (per Reference [172]) with the aid of the TRIDENT-GNC nonlinear numerical simulator (described in Section 3.5.1) [31]. The parameters chosen for the saturation controller is listed in Table 6.4.

Table 6.4. Saturation Controller Parameter Selection.

Parameter	Value
s_a	0.05 s^{-2}
s_b	0.0625 s^{-3}
J_{zz}	0.253 kg·m^2
$u_{\max}$	0.0437 N·m

6.6 Experimental Setup

To conduct the experimental campaign, two FSSs are used – one is the set as the Target while the other is the Chaser. The test setup of the POSEIDYN test bed for Case 1 is illustrated in Figure 6.6 which used a third, third-generation FSS as the obstacle. The remainder of the section lists the various guidance parameters used throughout the test campaign.

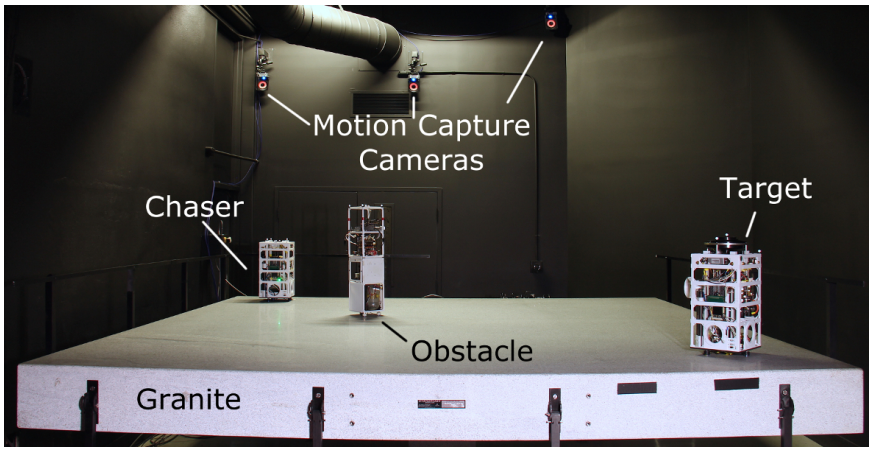


Figure 6.6. Test setup for the real-time guidance and control algorithm experimental campaign.

6.6.1 APF and AAPF Parameters

Relevant guidance parameters are summarized in Table 6.5. Guidance algorithm parameters were tuned with the aid of TRIDENT-GNC, nonlinear numerical simulator. These parameter were then held constant throughout the test campaign in an effort to enable greater comparison between the algorithms. It is worthwhile to comment on the selection of some of the parameters. First, the potential field shaping matrices, $\mathbf{Q}_a, \mathbf{Q}_d,$ and $\mathbf{P}_d$, were selected to produce minimal chattering upon encountering an obstacle boundary constraint using the APF method. Next, the AAPF reference translational speed, v_{ref}, was chosen to produce a similar rendezvous time compared to the Case 0 APF experimental results. The AAPF reference rotational angular speed, ω_{ref}, was chosen in order to ensure proper alignment prior to entering the docking cone corridor. Lastly, the adaptive parameter projection limits, ρ_i^-, ρ_i^+, were chosen through the use of a nonlinear numerical simulator to reduce chattering about the target configuration [31].

6.6.2 IDVD Parameters

Relevant guidance parameters for the IDVD method is presented in Table 6.6 [52], [72].

6.6.3 LQ-MPC and NMPC Parameters

Relevant guidance parameters for the LQ-MPC and NMPC guidance methods are presented in Tables 6.7 and 6.8, where $\bar{\mathbf{P}}$ is the solution to the algebraic Riccati equation associated with the discrete LQR problem [52], [61], [72]. Since the data presented are from two different test campaigns prior to the adoption of the proposed *Standard Test Framework* (Refs. [52], [72] and [61]), different tunings are presented for the LQ-MPC

Table 6.5. Summary of APF and AAPF guidance parameters.

Parameter	Value
Number of Runs	5
Guidance Sample Time, T_g	0.1 s
b_1	0.400 m
a_1	1.121 m
b_1	0.600 m
a_2	1.320 m
a_3	1.600 m
k_a, k_r	1.0
$\mathbf{Q}_a$	diag([0.025, 0.025, 0.075]) s^{-2}
$\mathbf{Q}_d$	diag([0.0125, 0.0125, 0.075]) s^{-2}
$\mathbf{P}_d$	diag([20, 20, 0]) s^{-2}
$\mathbf{N}_i$	$\mathbf{I}_{3\times3}$
σ_i	$^9/_{16}\, r^2_{\text{obs,i}} m^2$
ψ_i	0.3
v_{ref}	0.025 m/s
ω_{ref}	$2\,^\circ/s$
$\boldsymbol{\rho}_0^*$	$[0.1581, 0, 0, 1.1581, 0, 0.2739]^\mathsf{T}$
ρ_i^+, ρ_i^-	± 0.75 for $i = 1, 2, \ldots, 5$
	± 0.25 for $i = 6$

Table 6.6. Summary of IDVD guidance parameters.

Parameter	Value
Number of Runs	1
Guidance Sample Time, T_g	0.2 s
Translation Polynomial Order, $n_{x,y}$	5
Time Polynomial Order, $n_{t_{x,y}}$	2
Maximum Force, T_{max}	0.15 N
Maximum Time, t_f	100 s
Maximum Number of IPOPT Iterations	15

method. As previously mentioned, experimental results for the rotating hyperplane (rotating hyperplane (RH)) and dual hyperplane (dual hyperplane (DH)) methods will be presented.

Table 6.7. Summary of LQ-MPC guidance parameters for Case 1.

Parameter	Value
Number of Runs	1
Guidance Sample Time, T_g	5 s
$\mathbf{P}$	$\bar{\mathbf{P}}$
$\mathbf{Q}$	diag$(10^2, 10^2, 10^5, 10^5)$
$\mathbf{R}$	diag$(10^3, 10^3)$
Maximum Force, T_{max}	0.15 N
Horizon Length, N	20
Max. Number of Iterations	100

Table 6.8. Summary of LQ-MPC and NMPC guidance parameters for Case 2.

Parameter	Value
Number of Runs	3
Guidance Sample Time, T_g	3 s
$\mathbf{P}$	$\bar{\mathbf{P}}$
$\mathbf{Q}$	diag$(1, 1, 3 \times 10^3, 3 \times 10^3)$
$\mathbf{R}$	diag$(10^2, 10^2)$
Maximum Force, T_{max}	0.15 N
Horizon Length, N	30
Max. Number of Iterations	100

Table 6.9. Summary of RRT* and RRT*-HPF guidance parameters.

Parameter	Value
Number of Runs	1
Guidance Sample Time, T_g	0.1 s
No. of Nearest Neighbors, k	20
Max. Number of Iterations	250
v_{ref}	0.025 m/s

6.6.4 RRT* and RRT*-HPF Parameters

Relevant guidance parameters for the RRT* and RRT*-HPF guidance methods is listed in Table 6.9. In the proposed closed-loop RRT* guidance method, candidate nodes select the best parent node which minimizes a Minimum ΔV cost function (defined in Section 6.4.3).

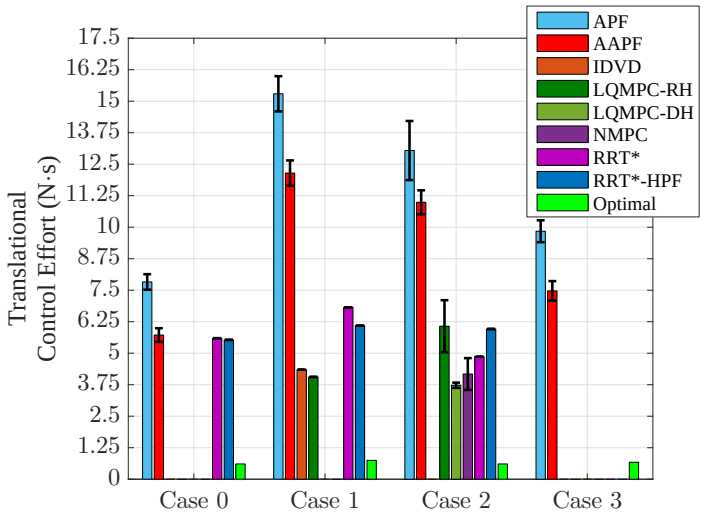
6.7 Experimental Results and Discussion

6.7.1 Results using the Standard Test Framework

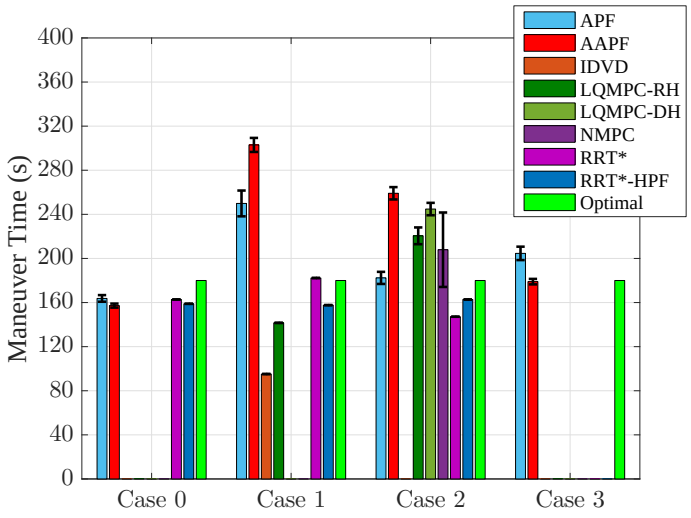
A summary of the averaged translational control effort and maneuver time is presented in Figure 6.7. A detailed summary of the averaged translational (and rotational) control effort, maneuver time, and the translational component of the non-dimensional control effort, ν_1 for Test Cases 0-3 is presented in Tables 6.11, 6.12, 6.13, and 6.14, respectively. Furthermore, the averaged trajectories, distance to the goal time history, and (translational) control effort versus the distance to the goal for Test Cases 0-3 is presented in Figures 6.9, 6.10, 6.11, 6.12, respectively. A comparison of the attitude time history for each test case is presented in Figure 6.7. Lastly, all methods evaluated successfully avoided the obstacle keep-out zone in addition to meeting the attitude error requirement prior to entering the docking cone corridor as illustrated in Figure 6.8.

For the baseline case, Case 0, the two RRT*-based methods performed in a similar manner with the AAPF guidance algorithm. These methods were found to exhibit a similar linear increase in the translational control effort, as observed in Figure 6.9b. In fact, during any obstacle-free portion, this characteristic was observed to hold across all test cases for not only this algorithm but also the IDVD method. In contrast, the nominal distance time history for the APF resembles that of an exponential response, illustrated in Figure 6.9b as well. This is evident as the Chaser crosses the DCC in approximately 70 s and spends the remaining 90 s traversing 0.75 m to dock with the Target. It is worthwhile to note, the MPC methods exhibit a similar exponential decrease in velocity once they are sufficiently close, but to a lesser extent then the APF method, as illustrated in Figure 6.10b and 6.11b.

The ratios of ν_1 to the optimal ν_1^* are tabulated in Table 6.10. Additionally, for Case 1, both the IDVD and closed-loop RRT*–HPF (CL-RRT*–HPF) methods were found to produce optimal trajectories (compared to the $\mathbb{GPOPS} - \mathbb{II}$ solution), as illustrated by Figure 6.10a. However, while the LQ-MPC/RH method did not produce an optimal trajectory, it did utilize the smallest amount of control effort amongst the evaluated methods. For Case 2, the RRT*-based methods, NMPC, and LQ-MPC/DH were found to produce optimal trajectories. For this case, the LQ-MPC method using the DH method developed in [61] achieved the smallest expended control effort. Overall, the implemented RT-G&C methods were observed to use approximately $5\times - 15\times$ greater control effort compared to the optimal ($\mathbb{GPOPS} - \mathbb{II}$) solution.



(a) Translational Control Effort



(b) Maneuver times

Figure 6.7. Comparison of the translational control effort and maneuver time.

Table 6.10. Ratio of the non-dimensional control effort ν_1 for each evaluated method to the optimal non-dimensional control effort ν_1^*.

	Case 0	Case 1	Case 2	Case 3
APF	14.17	14.72	21.25	12.88
AAPF	10.78	9.62	12.38	11.18
IDVD	– –	10.93	– –	– –
LQ-MPC/RH	– –	6.88	8.20	– –
LQ-MPC/DH	– –	– –	4.52	– –
NMPC	– –	– –	6.06	– –
RRT*	10.20	9.01	9.83	– –
RRT*-HPF	10.35	9.30	10.87	– –

Table 6.11. Case 0 comparison of guidance algorithm performance metrics.

	Avg. Translational Control Effort (N·s)	Avg. Rotational Control Effort (N·m·s)	Avg. Maneuver Time (s)	Avg. Translational ν_1 $\left(\times 10^{-4}\right)$
APF	7.83 ± 0.31	0.47 ± 0.026	163.74 ± 3.01	4.89 ± 0.26
AAPF	5.72 ± 0.27	0.38 ± 0.015	157.23 ± 1.82	3.72 ± 0.14
CL-RRT*	5.60	0.39	162.77	3.52
CL-RRT*-HPF	5.53	0.38	158.86	3.57

Table 6.12. Case 1 comparison of guidance algorithm performance metrics.

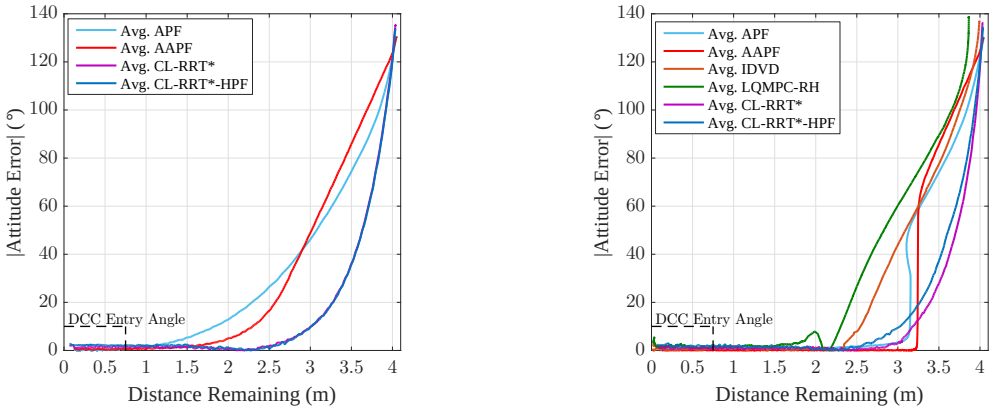
	Avg. Translational Control Effort (N·s)	Avg. Rotational Control Effort (N·m·s)	Avg. Maneuver Time (s)	Avg. Translational ν_1 $\left(\times 10^{-4}\right)$
APF	15.29 ± 0.70	0.76 ± 0.048	249.89 ± 11.69	6.27 ± 0.29
AAPF	12.15 ± 0.50	0.81 ± 0.067	302.96 ± 6.36	4.10 ± 0.19
IDVD [52], [72]	4.35	0.20	95.16	4.67
LQ-MPC/RH [52], [72]	4.06	0.22	141.56	2.93
CL-RRT*	6.82	0.41	182.26	3.84
CL-RRT*-HPF	6.10	0.39	157.66	3.96

Table 6.13. Case 2 comparison of guidance algorithm performance metrics.

	Avg. Translational Control Effort (N·s)	Avg. Rotational Control Effort (N·m·s)	Avg. Maneuver Time (s)	Avg. Translational ν_1 $\left(\times 10^{-4}\right)$
APF	13.04 ± 1.17	0.61 ± 0.011	182.36 ± 5.49	7.33 ± 0.83
AAPF	10.99 ± 0.47	0.73 ± 0.056	178.96 ± 2.49	4.27 ± 0.23
LQ-MPC/RH [61]	6.07 ± 1.03	0.36 ± 0.013	220.52 ± 7.60	2.83 ± 0.55
LQ-MPC/DH [61]	3.73 ± 0.10	0.39 ± 0.044	244.81 ± 5.62	1.56 ± 0.033
NMPC [61]	4.17 ± 0.63	0.33 ± 0.038	207.91 ± 33.76	2.09 ± 0.49
CL-RRT*	4.86	0.33	147.13	3.39
CL-RRT*-HPF	5.96	0.40	162.78	3.75

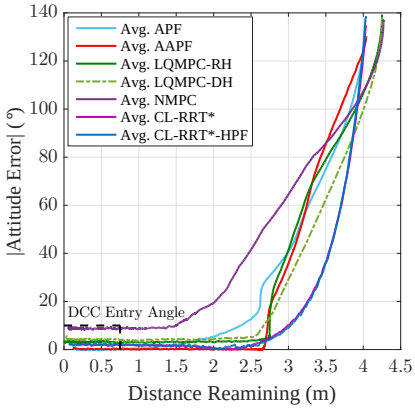
Table 6.14. Case 3 comparison of guidance algorithm performance metrics.

	Avg. Translational Control Effort (N·s)	Avg. Rotational Control Effort (N·m·s)	Avg. Maneuver Time (s)	Avg. Translational ν_1 $\left(\times 10^{-4}\right)$
APF	9.84 ± 0.43	0.57 ± 0.058	204.60 ± 6.09	4.92 ± 0.23
AAPF	7.47 ± 0.39	0.47 ± 0.06	178.96 ± 2.49	4.27 ± 0.27

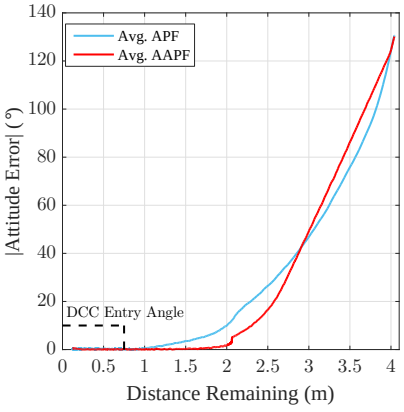


(a) Case 0

(b) Case 1

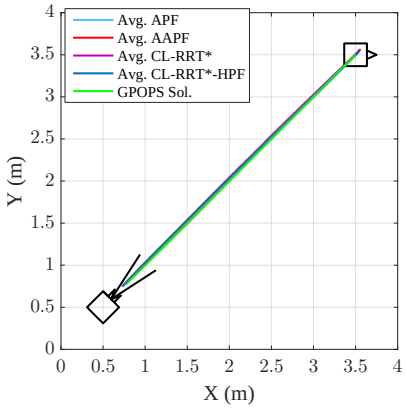


(c) Case 2

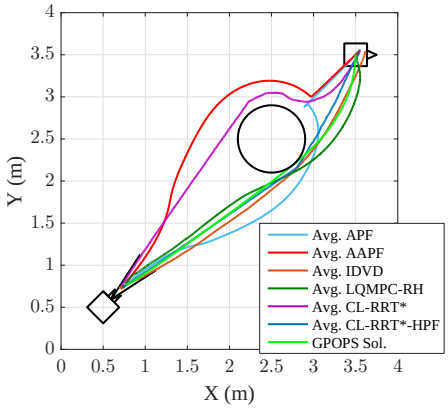


(d) Case 3

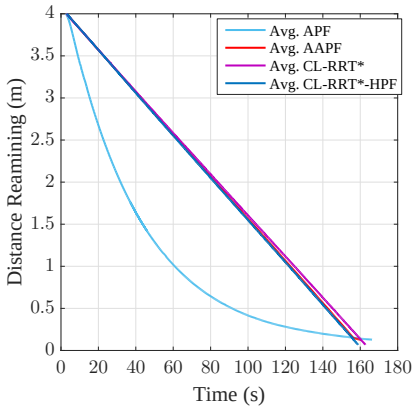
Figure 6.8. Comparison of the attitude error versus distance remaining.



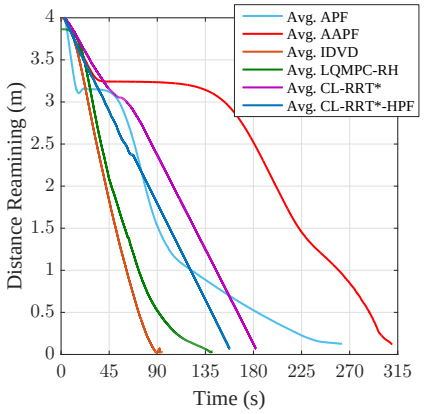
(a) Case 0: Attitude Error Time History



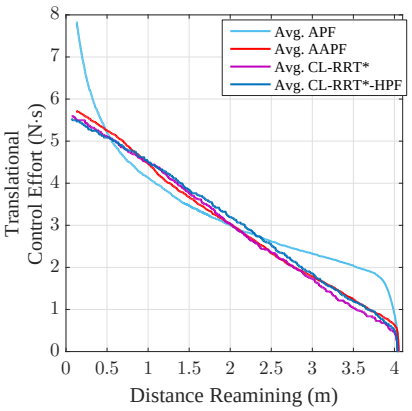
(a) Case 1: Trajectory



(b) Case 0: Distance to Goal Time History

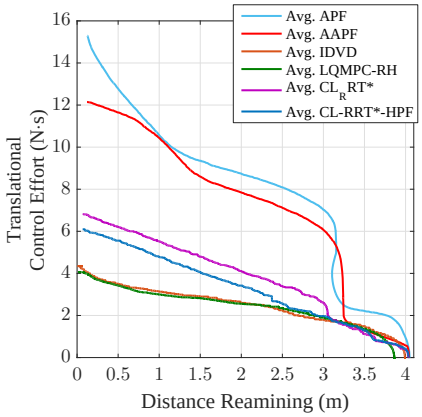


(b) Case 1: Distance to Goal Time History



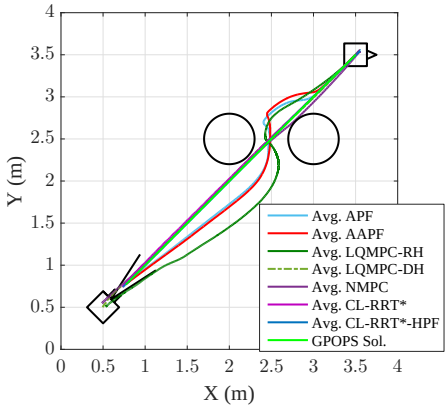
(c) Case 0: Thruster ON-Time Profile

Figure 6.9. Case 0 experimental results.

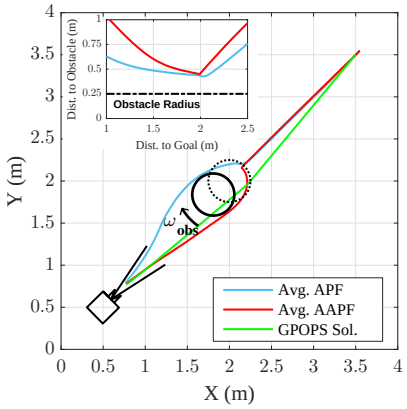


(c) Case 1: Thruster ON-Time Profile

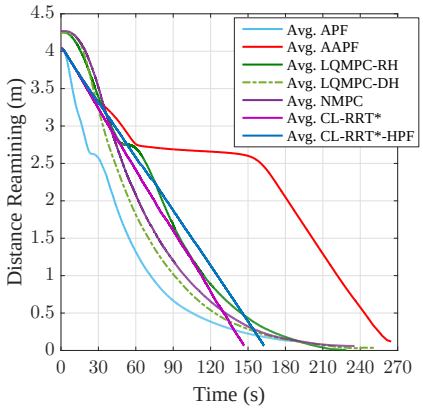
Figure 6.10. Case 1 experimental results.



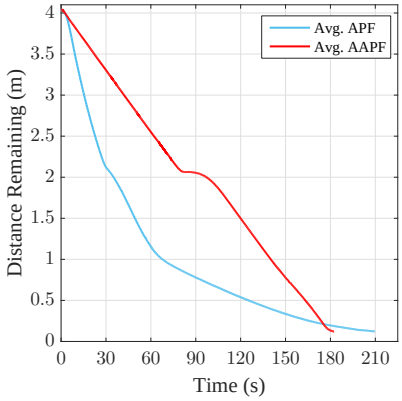
(a) Case 2: Trajectory



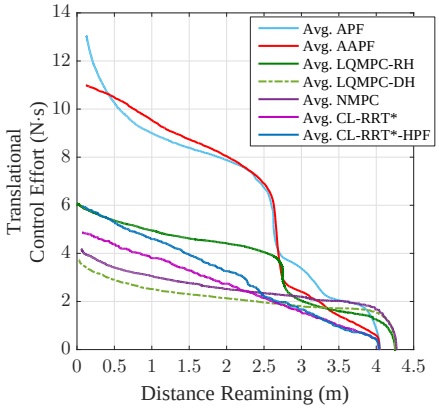
(a) Case 3: Trajectory (with Distance to obstacle inset)



(b) Case 2: Distance to Goal Time History

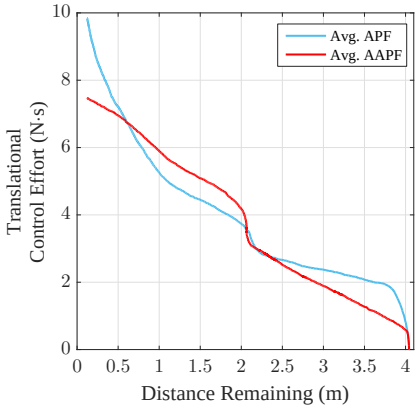


(b) Case 3: Distance to Goal Time History



(c) Case 2: Thruster ON-Time Profile

Figure 6.11. Case 2 experimental results.



(c) Case 3: Thruster ON-Time Profile

Figure 6.12. Case 3 experimental results.

6.8 Simulation-Based Case Study Evaluating the Effect of Dynamic Obstacles on G&C Methods

A simulation-based Taguchi Analysis[1] was performed in order to assess the effect of a dynamic environment on the non-dimensional control effort $\nu 1$ for various classes of RT-G&C methods. The three algorithms considered — the APF, AAPF, and IDVD guidance methods — span the range of RT-G&C from analytical methods (APF) to hybrid (AAPF) and optimization-based (IDVD) methods. To assess the effect of a dynamic environment, the rotation radius and radius of rotation of the obstacle in Case 3 are varied per Table 6.15. A $L_9(3^2)$ Taguchi array[2] was used for the two-level, three-factor design of experiment (DOE), resulting in nine experiments for each guidance method [138]. Each experiment was performed using the TRIDENT-GNC numerical simulator [31] (described in Section 3.5.1). The normalized average effect of each factor on the performance criterion ν_1 is presented in Figure 6.13. As one would expect, the smallest value of ν_1 occurs when both the rotation rate and radius of rotation are small. Note, as the these factors go to zero, Case 3 effectively becomes Case 0 with a point-obstacle.

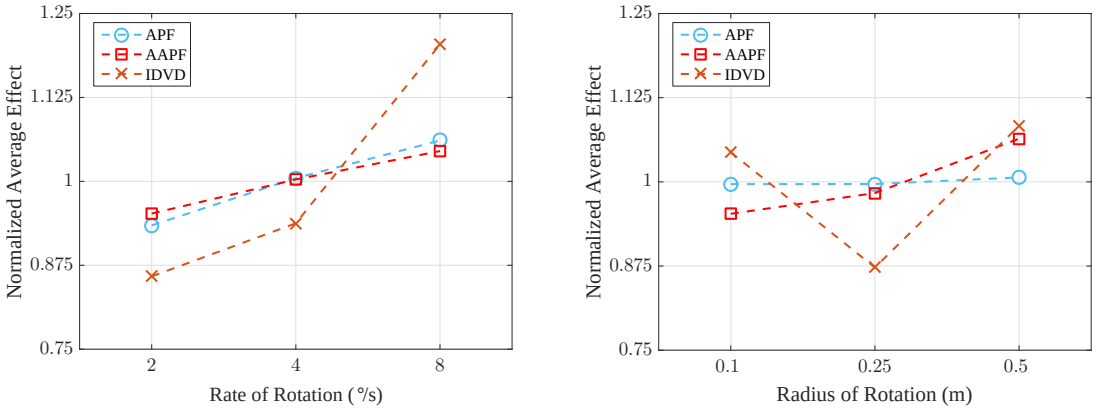
Table 6.15. Summary of the values for the factors considered in the case study.

Level	Rotation Rate (deg /s)	Radius of Rotation (m)
1	2	0.1
2	4	0.25
3	8	0.5

As illustrated, each of the three methods (representing each class of RT-G&C algorithms) were observed to exhibit an effect on ν_1 proportional to the rotation rate of the obstacle. Both the analytical and hybrid methods, APF and AAPF respectively, exhibited similar sensitivities while the optimization-based method, IDVD, was observed to have a greater sensitivity. Furthermore, unlike the hybrid and optimization-based methods, the analytical method was found to be insensitive to the radius of rotation of the obstacle. For the hybrid method, the radius of rotation was found to exhibit an effect on ν_1 proportional to the radius of rotation. However, for the optimization-based method, the effect of the radius of rotation on ν_1 is more complex. This complexity may arise from possible interactions between the rotation rate and radius of rotation. Additionally, the optimization-based method exhibited a larger range of effect on ν_1 compared to the other analytical and hybrid methods.

[1] See apprefappendix:taguchMethod for more information on the Taguchi Analysis.

[2] See Table B.1 for the structure of the $L_9(3^2)$ orthogonal array.



(a) Average Effect of Rotation Rate (b) Average Effect of Radius of Rotation

Figure 6.13. Average effect of varying the rotation rate and the radius of rotation of the obstacle for an analytical (APF), hybrid (AAPF), and an optimization-based (IDVD) guidance and control methods.

It is worthwhile to comment on the effect of the sampling rate of the guidance method as well as knowledge of obstacle's motion. If the motion of the obstacle is known and can be predicted (with high confidence), this information can be successfully integrated into the path-planning process in addition to a larger sampling period can be used. When the motion of obstacle is unknown, or predicted with low confidence, a smaller sampling period must be used to compensate for the lack of information. A similar trend was found to hold for the MPC-based methods. However. since significant tunning of the various MPC parameters (e.g., sampling period, $\mathbf{Q}$, $\mathbf{R}$) was found to be required, these methods were not considered for this study.

6.9 Comparison of Guidance and Control Methods

In order to fully assess the guidance and control algorithms using the Guidance Comparison Metric proposed in Chapter 5, two separate plots are required. The first plot compares the non-dimensional control effort ν_1 across each test case, illustrated in Figure 6.14 and 6.15. The second plot than compares the non-dimensional control effort ν_1 to the non-dimensional sampling period ν_2, illustrated in Figure 6.16 and 6.17.

6.9.1 Overall Performance Assessment

The translational component of the non-dimensional control effort, ν_1, for each method across each test case is illustrated in Figure 6.14 and 6.15. Note, the optimal control

solutions found in Section 6.2 are included for reference purposes. Additionally, the numerical values for ν_1 are provided in Tables 6.11, 6.12, 6.13, and 6.14.

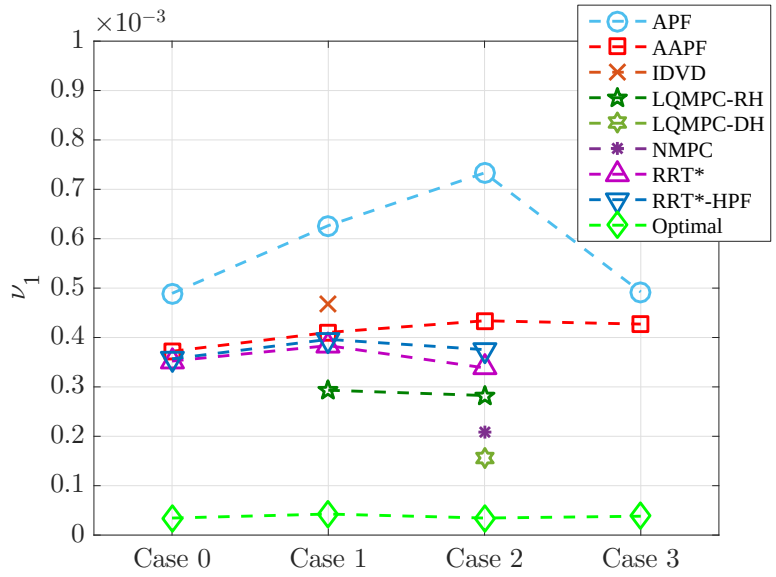


Figure 6.14. Comparison of the APF and AAPF to other guidance and control methods tested using the *Standard Test Framework*.

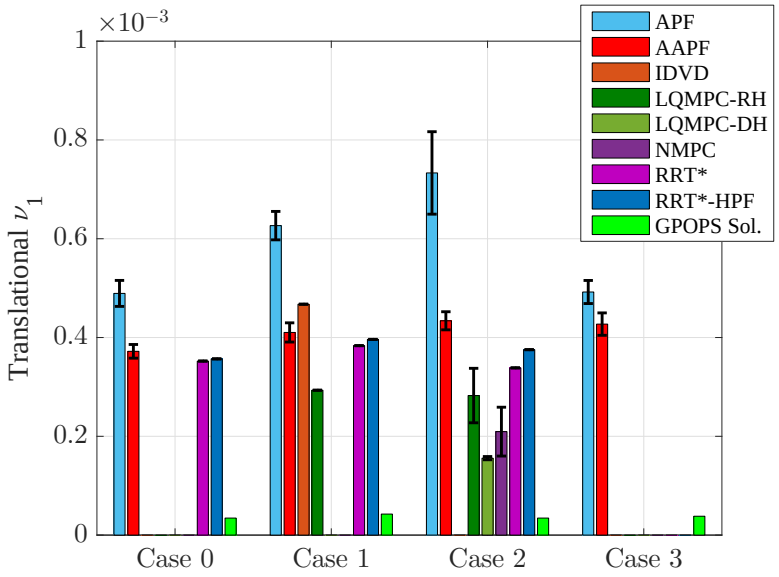


Figure 6.15. Comparison of guidance and control methods using the Guidance Comparison metric.

Of the algorithms evaluated, the APF method was observed to have the largest range of values, with its largest value being associated with Case 2. In contrast, the AAPF, CL-RRT*, CL-RRT*–HPF, and MPC methods were observed to produce smaller variations across the test cases for which they were evaluated. Additionally, the ν_1 values for the optimal control solutions were found to be relatively consistent across all the test cases. However, Cases 1 and 3 are larger by approximately 21.0% and 10.2% respectively.

While the AAPF was observed to have a smaller variation across the four test cases, ν_1 was observed to trend in a similar fashion to the APF method. This trend is indicative of the sensitivity to the number and geometry of the obstacles for the APF-based methods. However, despite this observed characteristic of APF-based methods, the use of an adaptive feedback law was observed to mitigate this sensitivity significantly for a small penalty in the increased complexity of the AAPF method. Similarly, the ν_1 values for both of the RRT-based methods were found to trend in a similar fashion. The increase in ν_1 from the baseline (Case 0) to Case 1 for the CL-RRT* method was observed to be approximately 8.7% while the CL-RRT*–HPF method was observed to have a 10.4% increase. While this increase was smaller than that of the optimal control solution, the value of ν_1 is roughly an order of magnitude larger. Compared to the APF-based methods, the value of ν_1 was observed to decrease between Case 1 and Case 2 rather than increasing. These observations regarding the trending of the APF-based and

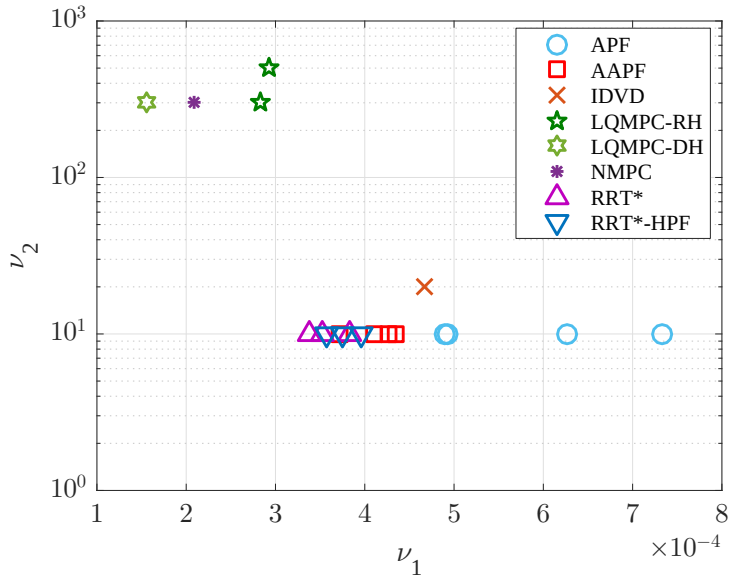
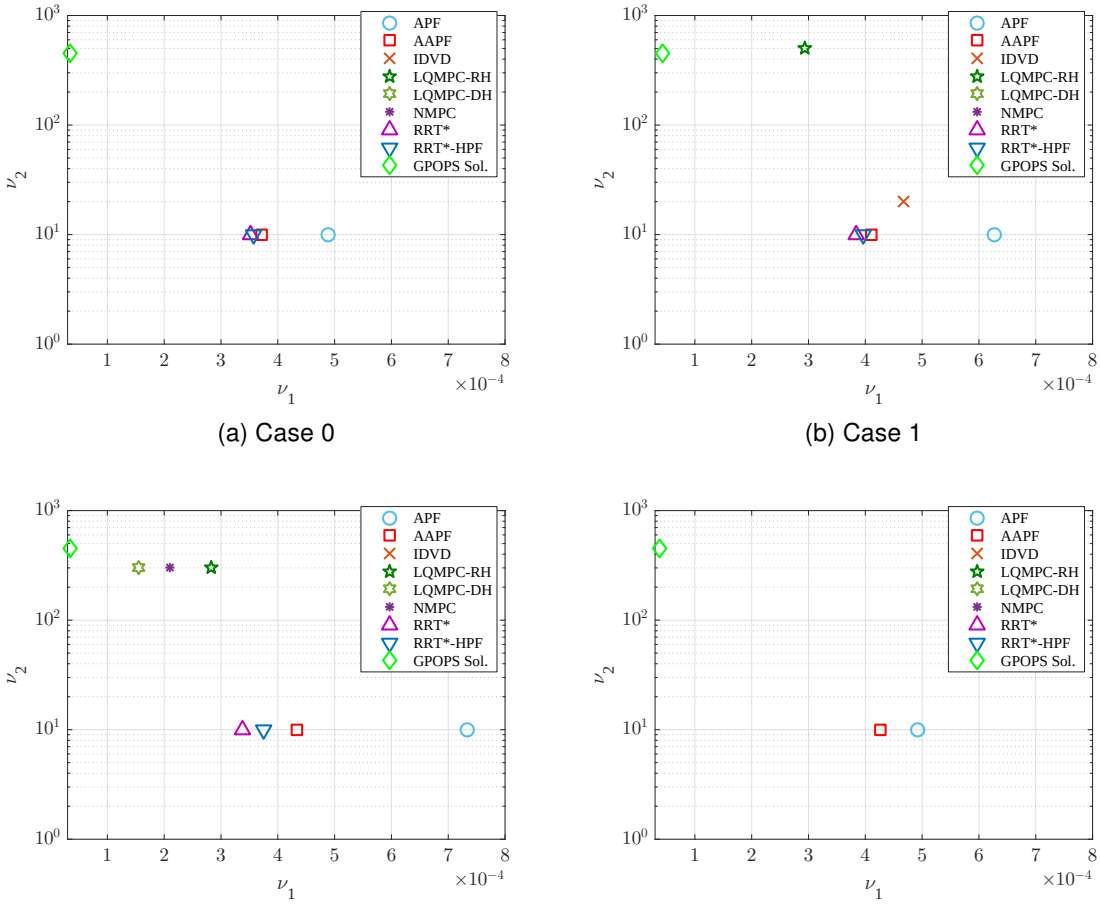


Figure 6.16. Comparison of guidance and control methods using the Guidance Comparison metric.



(a) Case 0

(b) Case 1

(c) Case 2

(d) Case 3

Figure 6.17. Comparison of guidance and control methods using the Guidance Comparison metric.

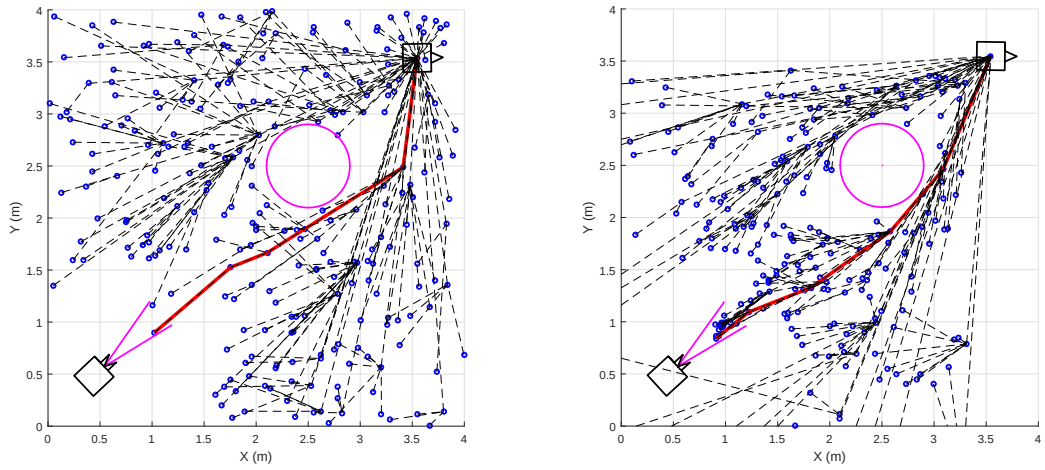
RRT-based methods support the claim regarding the ability of this metric to characterize guidance and control methods.

Lastly, the proposed Guidance Comparison metric, $\mathbf{z} = [\nu_1, \nu_2]^\mathsf{T}$ is illustrated in Figure 6.16 and 6.17. The sensitivity of each method to the number and geometry of obstacles can be observed by the tightness of the grouping it forms. That is, the APF method was observed to generate a loose grouping (i.e. a larger range) of values for ν_1 – indicating a larger sensitivity to number and location of the obstacles – while the AAPF and RRT-based methods were observed to form a tight grouping. Additionally, while MPC-based methods were observed to attain better performance (i.e. a smaller ν_1 value) than the AAPF and RRT-based methods, it can be observed that both of these methods have the ability to react to a more dynamic environment (due to their

lower computational time) while maintaining a comparable level of performance to the MPC-based methods.

6.9.2 Assessment of Path-Planning Capability

From these test cases, one can draw conclusions regarding the path-planning capabilities of each algorithm. Despite the pathological nature of Case 1, both the APF and AAPF methods were able to reach the goal position despite their susceptibility to local minima. For navigating multiple obstacles, the AAPF method was observed to produce a smoother, more efficient trajectory, compared to the APF method. This was attributed to a slower changing gradient in the vicinity of the obstacles, as illustrated by the slowly decreasing average distance to the goal time history in Figure 6.11b. Both the APF and AAPF methods were observed to successfully avoid the moving obstacle, as illustrated by the inset in Figure 6.12a. Note, these methods encountered the obstacle at different times and positions. The average closest approach of the Chaser to the moving obstacle using the APF method was 0.424 meters and occurred at a distance of just over 2 meters from the goal – corresponding to an average obstacle angular position of approximately -64° from the initial angular position. Likewise, the average closest approach using the AAPF method was 0.446 meters occurred at a distance of just under 2 meters from the goal; however, due to the speed difference of the Chaser, the average angular position of the obstacle was approximately -158° from the initial angular position. However, despite the position differences of the obstacle when it was encountered, the AAPF method was able to generate a more efficient trajectory compared to the APF method.



(a) CL-RRT* Initial Tree for Case 1

(b) CL-RRT*–HPF Initial Tree for Case 1

Figure 6.18. Comparison of the Initial Trees Generated for Case 1.

For the RRT*-based methods, the effect of biasing the randomly sampled configuration space can be observed by comparing the initial trees for Case 1 in Figure 6.18. The primary effect of biasing results in increased directionality of the tree, reducing the exploration tendency associated with the construction of the tree [160]. Compared to the observed final trajectories, the CL-RRT*–HPF was able to generate a path similar to the IDVD method.

Lastly, a large difference in the path-planning capability of the MPC methods was observed for Case 2, as illustrated by Figure 6.11a. Compared to the LQ-MPC method using a rotating hyperplane (LQ-MPC/RH) convexification method, both the LQ-MPC using the dual hyperplane (LQ-MPC/DH) method as well as the NMPC method were able to resolve the straight-line path. In fact, the LQ-MPC/DH was observed to use 11.14% less control effort than the NMPC method.

6.9.3 Effect of the Modulation Technique on ν_1 and ν_2

As illustrated in Chapter 4, the use of the $\Sigma\Delta$M provided superior signal tracking capability compared to the PWM. This has important implications on the guidance and control method and ν_1 and ν_2. First, the $\Sigma\Delta$M provides more accurate realization of the commanded input as a result of its feedback loop. Consequently, a slight higher control effort is expended – when compared to the PWM – resulting in a larger value of ν_1. As a direct result of this more accurate command input realization by the $\Sigma\Delta$M, a larger guidance sampling period (and larger ν_2 value) can possibly used. This implies that more computationally intensive algorithms could be used due to the availability of a larger sampling period.

6.10 Summary

Several guidance and control methodologies were experimentally evaluated using the proposed *Standard Test Framework* for a rendezvous and docking scenario. The optimal solution to maneuvering a spacecraft in a cluttered and constrained environment was found and presented for each of the test cases associated with this framework. Next, several algorithms representing the state of the art, including Artificial Potential Functions (APFs), Adaptive Artificial Potential Functions (AAPFs), Linear-Quadratic (LQ-MPC) and Nonlinear Model Predictive Control (NMPC), as well as Inverse Dynamics in the Virtual Domain (IDVD), were experimentally evaluated and compared to the proposed Closed-Loop RRT*–HPF method. The resulting comparison was facilitated through the use of the proposed Guidance Comparison metric. Compared to the current state of the art, the proposed RRT-based guidance method was found to produce near-optimal trajectories and was found to be comparable, in terms of performance, to the current state of the art. Furthermore, several key characteristics were demonstrated

for the Guidance Comparison metric, including its ability to characterize guidance and control methods in addition to its efficacy in comparing guidance and control methods.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 7:
Development of the Guidance and Control Trade Space

> The greatest results are simple and may even look trivial
>
> —Dennis S. Bernstein
> "A student's guide to research" [173]

A journal version of the work presented on the proposed Guidance Comparison Metric is under review for publication to the AIAA *Journal of Guidance, Control, and Dynamics* [75]. The work presented in Section 7.3.1 was originally presented at the 27th AAS/AIAA Spaceflight Mechanics Conference in San Antonio, TX (February 2017) [74].

7.1 Case Study Overview

To illustrate the extensibility of the proposed characterization framework, this chapter presents a case study illustrating the development of a never-before-seen guidance and control trade space to aid in the selection of a G&C method for a notional mission. Knowledge of the impact of the guidance and control method on the system can inform the design earlier in the design process. To aid in this selection, a new and unique perspective on the age-old problem of selecting the minimum sampling rate for real-time guidance and control algorithms is proposed. Furthermore, several bounding methods are presented in an effort to define the boundaries of the trade-space.

The parameters for this example case study are given in Table 7.1. Note, the reference orbit is that of the ISS [77]; the spacecraft parameters are based on a FSS [31], [100]; and a typical driving mission requirement, the ΔV budget, is arbitrarily chosen to be within the capability of the system.

7.2 Bounding the Control Effort

A driving requirement for autonomous RPO missions is the ΔV budget. Specifically, as the efficiency of the guidance increases, the useful life of the spacecraft is larger compared to a less efficient algorithm. Using the parameters in Table 7.1, an upper bound on ν_1 can be estimated given one additional parameter, the maximum accumulated thruster ON-time, Δt_b. Note, in the case that this parameter is not available, it can be estimated given these parameters.

Table 7.1. Summary of parameters for the example case study.

Parameter	Value
Reference Orbit	400 km
(Wet) Mass, m_0	10 kg
Maximum Thruster Force, T_max	0.20 N
Specific Impulse, I_{sp}	20 s
Propellant Mass, m_p	0.5 kg
$\Delta V_{\text{required}}$	2.5 m/s
Executive Time, T_b	0.1 s

The change in velocity associated with the linear momentum imparted by a thruster is given as

$$\Delta V = \frac{T_{\text{max}} \Delta t_b}{m} \tag{7.1}$$

Given that for a spacecraft, the associated control effort considered in the defining ν_1 is the linear momentum impulse, Equation (7.1) can be rewritten as

$$\Delta V = g_0 \Delta t_m \nu_1 \tag{7.2}$$

where g_0 is the gravitational acceleration constant and Δt_m is the maneuvering time. For the purposes of upper-bounding ν_1, it can be assumed $\Delta t_m \equiv \Delta t_b$. This assumption allows ν_1 to be redefined in terms of the required ΔV and maximum accumulated thruster ON-time:

$$\nu_{1,u} = \frac{\Delta V_{\text{required}}}{\Delta t_b g_0} \tag{7.3}$$

Lastly, if Δt_b is not known, it can be approximately by manipulating the equation defining the specific impulse

$$\Delta t_b = \frac{m_p g_0 I_{sp}}{T_{\text{max}}} \tag{7.4}$$

Substituting Equation (7.3) into Equation (7.3) yields the upper bound for ν_1 in terms system parameters:

$$\nu_{1_u} = \frac{T_{\text{max}} \Delta V_{\text{required}}}{m_p g_0^2 I_{sp}} \tag{7.5}$$

7.3 Bounding the Sampling Time

This section will detail several considerations in an effort to bound the sampling time and aid in defining the trade space. Section 7.3.1 details a new method on choosing a sampling period for a guidance or control task based on first-principles. Section 7.3.2

discusses an approach on specifying a time constant associated with the motion of a resident space object.

7.3.1 Minimum Sampling Rate for Real-Time Guidance and Control

Metric Formulation
The work in this sub-section was originally presented at the 27th AAS/AIAA Spaceflight Mechanics Conference in San Antonio, TX (February 2017) [73]. Modern controllers are designed and analyzed in the continuous-time domain and implemented, however, in a discrete manner via a digital computer, typically at a fixed interval, T_c. The use of a digital computer fundamentally implies a non-zero delay to computational delay, Δt_c, associated with sensor acquisition and computation of the control input. This further implies the control output is held constant over the next computational period. Furthermore, actuators typically realize the control input through the conversion of a digital commanded input to an analog signal via a digital-to-analog conversion. As a result of this quantization process, the output of the actuator is limited to discrete levels between zero to it maximum output. Exploiting these inherent features of a physical system, a minimum sampling rate metric can be derived from first principles.

Consider first the generic model for a nonlinear system

$$\dot{x}(t) = f(x(t), u(t), t) \tag{7.6}$$

Over short the period over which the actuator output remains constant, the motion of a mechanical system is assumed to be sufficiently described as rectilinear motion (i.e., double integrator dynamics). As a result, the time-averaged acceleration can be written as

$$\frac{\dot{x}\left(t + \Delta t_c\right) - \dot{x}(t)}{\Delta t_c} = \frac{1}{\Delta t_c} \int_t^{t+\Delta t_c} u(\tau)\, d\tau \tag{7.7}$$

where $u(\tau)$ is the control input. Note, the first order difference equation in Equation (7.7) approaches the derivative ${}^{d\dot{x}}/_{dt} = u$ in the limit as $\Delta t_c \to 0$. Rewriting Equation (7.7) as

$$\Delta \dot{x} = u \Delta t_c \tag{7.8}$$

implies a minimum acceleration impulse which can be imparted onto the system.

Before continuing, it is worth noting, in the context of defining real-time for a system, the acceleration term can be interpreted as a *dynamic bandwidth*, ζ_{BW}. The dynamic bandwidth is defined as

$$\zeta_{\text{BW}} = \frac{\text{Minimum Control Resolution}}{\text{Appropriate Physical Quantity}} \tag{7.9}$$

Analogous to a system bandwidth, ω_{BW}, the dynamic bandwidth is a system-level metric which can be used to provide insight into the responsiveness of the system. However, unlike the system bandwidth which is typically considered to be inversely proportional to the maximum responsiveness of the system, the dynamic bandwidth captures the minimum responsiveness of the system [174]. Additionally, while the system bandwidth is only formally defined for linear systems, the dynamic bandwidth is defined for both linear and nonlinear systems [129], [174].

Continuing, Equation (7.8) implies a corresponding change in the state of the system which can be written in terms of the dynamic bandwidth as

$$x\left(t+\Delta t_c\right)-x(t)=\Delta x=\frac{1}{2}\zeta_{\text{BW}}\Delta t_c^2 \tag{7.10}$$

Furthermore, given an acceptable change in state (i.e. error) and dynamic bandwidth for a particular application, the maximum sampling time, T_{RT}, which captures the dynamics of the system can be estimated as

$$T_{\text{RT}}=\frac{1}{2}\sqrt{\frac{2\Delta x}{\zeta_{\text{BW}}}} \tag{7.11}$$

Therefore, the resulting minimum sampling rate, f_{RT} for a control system to be considered real-time is

$$f_{\text{RT}}=2\sqrt{\frac{\zeta_{\text{BW}}}{2\Delta x}} \tag{7.12}$$

Both the maximum sampling period and minimum sampling rate metrics proposed in Equation (7.11) and Equation (7.12) respectively have several implications. First, both metrics are bounded by the physical system (i.e. mass, inertial , actuator strength) as well as the control system implementation. The proposed metrics provide insights into the system which are easily mapped to the physical system. Furthermore, these metrics provide the designer with 'knobs' they can adjust in order to estimate the affects of changes on various aspects of the system.

Validation of this metric is presented in Appendix A in an effort to retain the focus of this Chapter.

Guidance Path Re-Planning Rates

The applicability of this metric is not only for the low-level controller as demonstrated in Appendix A; it can also be applied to the guidance, or path-planning, task as well. Since optimal control is predicated on a predictive model, uncertainty in the navigation solution, actuator dynamics, and unmodeled disturbances can degrade the quality of

the solution over time and require an update to the problem. As a result, this metric can be used to estimate the rate at which it is necessary to generate a solution. To do so, however, requires the estimated error be bounded by some (error) function, $e(t)$,

$$e(t) \geq |f - \hat{f}| \tag{7.13}$$

where f is the dynamics of the system and $\hat{f}$ is the estimated dynamics of the system. Next, differentiating Equation (7.13) with respect to time yields the error dynamics, $\dot{e}$. Note, the change in error over the real-time sampling period is

$$\Delta e(t) = \dot{e}(t) T_{\mathsf{RT}} \tag{7.14}$$

Substituting Equation (7.14) into Equation (7.11) as the change in position, one can solve for the time necessary to replan given the bounded error function $e(t)$ as

$$T_{\mathsf{RT}} = \frac{1}{2} \frac{\dot{e}(t)}{\zeta_{\mathsf{BW}}} \tag{7.15}$$

7.3.2 Time Constant of a Resident Space Object

The challenges associated with specifying a time constant of a RSO, T_{rso}, is that it requires knowledge of the RSO and the trajectory relative to the the RSO – which may be not known *a priori*. Specifically, two cases arise for specifying a time constant associated with a RSO: force and unforced motion. For the case of unforced motion, the relative velocity, is proportional to the mean motion,n, of the RSO [78]. Therefore, the time constant for the unforced case is can be taken to be the inverse of the mean motion

$$T_{\mathsf{rso,u}} = \frac{1}{n} \tag{7.16}$$

The time constant associated with the forced motion, $T_{\mathsf{rso,f}}$, can be specified by

$$T_{\mathsf{rso,f}} = \frac{V_{\mathsf{rel,max}}}{a_{\mathsf{max}}} \tag{7.17}$$

where $V_{\mathsf{rel,max}}$ is the maximum relative velocity between the Chaser and the RSO, and a_{max} is the (estimated) maximum acceleration of the RSO. However, it is unlikely the maximum relative velocity will be known *a priori*. To provide an estimate, $V_{\mathsf{rel,max}}$ can be chosen to be 1 m/s. This simplification allows the time constant to be inversely proportional to the estimated maximum acceleration. If the acceleration is unknown, the time constant of the RSO can be bounded by using an acceleration from Table 5.3 or Figure 5.4.

7.4 Analysis and Discussion

Assuming no *a priori* information is available regarding the forced motion of a spacecraft, the parameters listed in Table 7.1, and a soft-constraint on the acceptable control error of 2 m, the resulting time constants and bounds on ν_1 and ν_2 are tabulated in Table 7.2. The bounds on the acceleration were chosen to be the maximum acceleration, a_{max}, and the acceleration given by a_{bar}

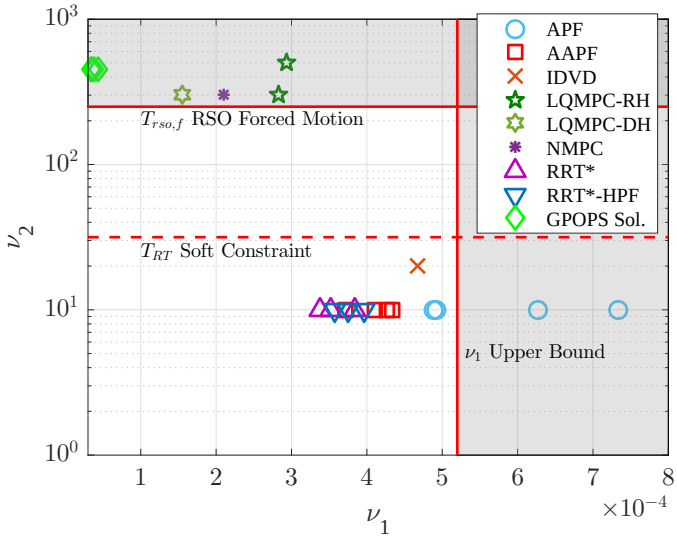
$$a_{bar} = a_{avg} + a_{1\sigma} \tag{7.18}$$

where a_{avg} is the average acceleration and $a_{1\sigma}$ is the 1σ standard deviation listed in Table 5.3. The time constants specified in Section 7.3 are normalized by dividing by the Executive time (T_b). The resulting bounds are overlaid on the Guidance Comparison plot, illustrated in Figure 7.1 to create a guidance and control trade space. Note, the grayed-out areas are the infeasible areas based on the bounds on ν_1 and ν_2.

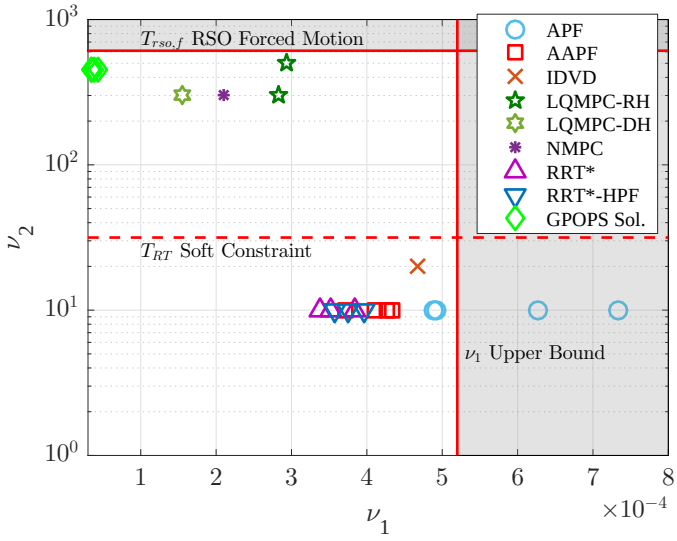
Table 7.2. Summary of time constants and bounds on ν_1 and ν_2.

Parameter	Value
T_{RT}	3.16 s
$T_{\text{RSO,u}}$	909 s
a_{bar}	0.0164 m/s^2
a_{max}	0.0431 m/s^2
$T_{\text{RSO,f}}$	$\{23.2, 61\}$ s
ν_{1_u}	5.20×10^{-4}
ν_2	$\{31.6, 250, 9090, 12340\}$

As illustrated, the upper bound on ν_1 eliminates the APF method as a possible guidance choice due to its high averaged control effort required. For the RSO time constant associated with a_{max}, the relatively large acceleration of Hayabusa eliminates the MPC methods as a possible guidance choice, as illustrated in Figure 7.1a. If the acceleration of the RSO is taken to be a_{bar}, $T_{RSO,u}$ becomes larger and the MPC methods become a feasible guidance method illustrated in Figure 7.1b. Regardless of the bounds chosen for $T_{RSO,u}$, the RRT-based methods and AAPF methods – and possibly the IDVD method – are feasible methods for this particular application. Given these results, it is clear the RRT-based methods yield a slightly smaller value for ν_1 compared to the AAPF methods. However, it is worthwhile to note, this methodology does not take into account any implementation complexities – which is more or less a qualitative assessment – associated with the -potential methods.



(a) Trade Space Associated with Using A RSO Time Constant Associated with a_{max}



(b) Trade Space Associated with Using A RSO Time Constant Associated with a_{bar}

Figure 7.1. Annotated guidance and control trade space based on the Guidance Comparison metric.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 8:
Summary and Future Work

> When we are engaged in research, especially when we are working on our dissertations, we think that research is everything, but it is not. There is joy in discovering a new result, but I think research (in Engineering especially) is most satisfying when it serves some immediate practical purpose as well.
>
> Research should be a source of joy, of exhilaration, and, in many ways, an act of love. If it isn't, then it may be difficult to endure the hardships that research entails.
>
> —Malcolm D. Shuster
> "Advice to Young Researchers" [175]

8.1 Summary of Contributions

The work presented in this dissertation focuses on the advancement of the state of the art in the evaluation of Real-Time Guidance & Control (RT-G&C) and Computational Guidance & Control (CG&C) methods. This work is fundamental to the emerging field of CG&C algorithms as it provides the necessary methodology and tools to evaluate their performance in an equitable manner. This was accomplished by: significantly improving the TRIDENT-GNC (Toolkit for Real-tIme DevelopmENT of Guidance, Navigation & Control) software architecture for the Naval Postgraduate School (NPS) POSEIDYN (Proximity Operation of Spacecraft: Experimental hardware-In-the-loop DYNamic simulator) test bed; performing an extensive characterization of the POSEIDYN test bed; an experimental evaluation framework and benchmarking test scenarios were established; and a global guidance comparison metric was developed and validated. Additionally, the challenge of efficiently modulating the guidance control signal into a sequence of discrete, binary signals which control spacecraft thrusters was explored – effectively solving the low-level actuator control problem. The fully characterized test bed was then utilized to compare a newly proposed RT-G&C algorithm based on the Rapidly-Exploring Random Tree Star (RRT*) method to existing algorithms in literature using the evaluation framework. Lastly, the global Guidance Comparison Metric was then utilized to compare the evaluated algorithms.

8.1.1 Extensive Characterization of the NPS POSEIDYN Testbed

A detailed description of the Naval Postgraduate POSEIDYN test bed has been provided. The quasi-frictionless and low residual acceleration dynamic behavior of the test vehicles operating on the POSEIDYN test bed are limited to two translational and one rotational degree-of-freedom motion. Multiple vehicles can be operated simultaneously on the POSEIDYN test bed allowing research on coordinated control of spacecraft teams. The on-board actuators, composed of eight thrusters and/or a reaction wheel, are equivalent to the actuators found in orbital spacecraft and further enhance the dynamic equivalence. With an available simulator and extensive software development tools, guidance and control algorithms for real-time execution on-board the test vehicles can be quickly and easily developed. From the characterization performed of the POSEIDYN test bed, the real-time operating system was shown to have bounded latencies and low-residual acceleration of 19 μg — a claim no other test bed can make. Used as a last stage of on-the-ground validation prior to on-orbit deployment or simply as a more realistic development environment for novel guidance and control approaches, this state-of-the-art dynamic hardware-in-the-loop test bed will continue to be fruitful for the advancement of spacecraft proximity operations research.

8.1.2 Development of the TRIDENT-GNC Software Architecture

The TRIDENT-GNC (Toolkit for Real-tIme DevelopmENT of Guidance, Navigation & Control) software architecture is composed of a (numerical) development simulator of the floating spacecraft simulator (FSS), a software template, and custom Simulink library. This library contains common software, such as the navigation and control subsystems which is used in both the simulator and the FSS autogenerated onboard software. This commonality between the simulator and FSS onboard software allows for rapid development of multi-rate Guidance, Navigation, and Control (GNC) algorithms in a simulation environment and software generation for use onboard the FSS for testing.

8.1.3 Development of a Standardized Test Framework to Evaluate Autonomous G&C Algorithms

The proposed *Standard Test Framework* defines relevant scenarios that test various capabilities of candidate algorithms in a relevant and dynamic environment. This framework is composed of four test cases. The first case is a straight-in approach with only the docking cone corridor (DCC) and aids in establishing the baseline. The next two cases test the path planning capabilities of the candidate algorithm with a static obstacle. The last two cases test the ability of the algorithm to adapt to a dynamic environment with one and three obstacles rotating about some predefined point in the workspace. The *Standard Test Framework* is used in Chapter 6 to evaluate and compare several guidance and control algorithms.

8.1.4 Development and Validation of a Global Guidance Comparison Metric

The proposed global Guidance Comparison Metric provides a unique capability to compare different algorithms across different architectures. This proposed metric, never seen before in the literature, allows for an unbiased comparison of algorithms across multiple system architectures through the use of a vector-valued metric, $\mathbf{z} = [\nu_1, \nu_2]^\mathsf{T}$. Due to its generic formulation, the Guidance Comparison Metric can be extended to virtually any mechanical system. A case study using the Guidance Comparison Metric to develop a never-before-seen guidance and control trade space illustrated in Chapter 7.

8.1.5 Experimental Evaluation of the Sigma-Delta Modulator as a Thruster Modulation Technique

Motivated by the inherent properties of a Sigma-Delta Modulator ($\Sigma\Delta$M), an experimental campaign was conducted to examine its viability as a spacecraft thruster modulation technique. The experimental campaign consisted of a rest-to-rest, 90 deg slew maneuver using the reaction control jets onboard a floating spacecraft simulator. In this campaign, the $\Sigma\Delta$M was compared against a traditional spacecraft thruster modulation technique, the Pulse-Width Modulator (PWM). The $\Sigma\Delta$M was found to achieve a smaller steady-state error in a shorter amount of time, while using less propellant. However, due to the short period of the limit cycle induced by the feedback in the $\Sigma\Delta$M, the propellant consumption during steady was observed to linearly increase at a higher rate than the PWM. Given a set of mission requirements, the $\Sigma\Delta$M can be coupled with a deadzone in order to increase the period of the limit cycles, thereby reducing the fuel consumption. As an alternative to a deadband, one could alternatively utilize the $\Sigma\Delta$M throughout the transient of a response and switch to a lower-bandwidth modulator, such as the PWM, when in the vicinity of the reference state. Additionally, candidate closed-loop transfer functions describing the behavior of spacecraft attitude control system with a modulator-in-the-loop was developed for both modulation techniques. The candidate $\Sigma\Delta$M model was found to exhibit similar characteristics to both the experimental and a simulated nonlinear response. Resultantly, this model, which is considered to be validated, can be used in conjunction with linear analysis techniques to estimate the response of system.

8.1.6 Application of a RRT-Based Algorithm as a Feedback Controller for the G&C of a Spacecraft in a Dynamic Environment

A new guidance and control strategy leveraging the path-generating capabilities of Rapid-exploring Random Trees (RRT) as well as the obstacle avoidance capabilities of Harmonic Potential Functions (HPF) has been developed and experimentally tested

in a rendezvous and docking scenario. To accomplish this, the potential field was developed using elementary solutions to the Laplace Equation which have strong ties to incompressible and irrotational flow. The strengths of the various elements composing the potential field were then autonomously assigned given the capabilities of the spacecraft. Next, the gradient of the resulting fields was used to bias a randomly sampled state in the workspace towards the goal while remaining in the obstacle-free portion of the workspace. Lastly, the path-generating capabilities were utilized to develop a path which minimizes the fuel consumption of the spacecraft while reducing computational complexity compared to traditional methods to solve optimal control problems. The Naval Postgraduate School (NPS) POSEIDYN was then used to validate the real-time capabilities of the proposed guidance and control methodology in a computationally constrained environment and assess its robustness to realistic noise, delays and uncertainties. While the two methods provided similar results the straight-path solution, the advantage of biasing the random samples from the configuration space is illustrated in the single static obstacle case where the chaser must circumnavigate an obstacle.

8.1.7 Development and Validation of a Metric to Estimate the Minimum Sampling Rate

When implementing a controller for a physical system, the rate at which the system is sampled and controlled must be determined. Often times, this rate is chosen through various “Rules of Thumbs” which are rooted in the frequency domain. However, a new approach to determine the minimum sampling rate rooted in first principles for the real-time control of a spacecraft was presented and validated via numerical case studies. This approach exploits the discrete nature associated with the digital implementation of a modern control system along with a unique view on the underlying dynamics to derive an estimate for the minimum sampling rate necessary for a control system to be considered real-time. This metric, unlike other methodologies, is valid for both linear and nonlinear systems. Additionally, the resulting formulation easily maps to the physical system. Several case studies were performed using relevant linear and nonlinear systems including a spacecraft undergoing a 3-axis reorientation maneuver using an Eigenaxis-Quaternion feedback controller to a double integrator system under a minimum-time control policy. Each system considered utilized a realistic actuator with discrete output levels. From these case studies, the proposed metric was found to sufficiently estimate the average steady-state output of the system. Additionally, by exercising the flexibility of the definition *dynamic bandwidth*, an upper bound on the minimum sampling rate can be found by estimating transient tracking capabilities of the system. Furthermore, an extension of this metric was presented to estimate the rate at which the guidance path re-planning task must be performed. This metric, while does not exactly predict the necessary sampling rate, provides an estimate to well within an order of magnitude with few assumptions under ideal conditions.

8.1.8 Development and Formulation of a Guidance and Control Trade Space for Spacecraft Design and GNC Selection

Using the experimental evaluation framework and the Guidance Comparison Metric, $\mathbf{z} = [\nu_1, \nu_2]^\mathsf{T}$, a never-before-seen guidance and control trade space was formulated. Knowledge of the impact of the guidance and control method on the system can inform the design earlier in the design process. This trade space was developed using parameters typically available early in the design process. Using these parameters, bounds were placed on the non-dimensional control effort., ν_1, and non-dimensional sampling period, ν_2, to narrow down the trade space to select a candidate algorithm.

8.2 Contribution Traceability

The supporting evidence for each contribution described in Section 8.1 is cross-referenced with the corresponding chapter in Table 8.1.

Table 8.1. Mapping of the contribution to the supporting chapters.

Contribution	Dissertation Chapter				
	3	4	5	6	7
Extensive characterization of the NPS POSEIDYN testbed	✓	– –	– –	– –	– –
Development and implementation of the TRIDENT-GNC software architecture	✓	– –	– –	– –	– –
Development of a standardized test framework to evaluate autonomous G&C algorithms	– –	– –	✓	– –	– –
Development and validation of a global Guidance Comparison metric	– –	– –	✓	✓	✓
Experimental evaluation of the Sigma-Delta modulator as a thruster modulation technique	– –	✓	– –	– –	– –
Application of a RRT-based algorithm as a feedback controller for the G&C of a spacecraft in a dynamic environment	– –	– –	– –	✓	✓
Development and validation of a metric to estimate the minimum sampling rate	– –	– –	– –	– –	✓
Development and formulation of a G&C trade space for spacecraft design and GNC selection	– –	– –	✓	✓	✓

8.3 Future Work

In this section, recommendations for future work is suggested.

8.3.1 Simulation-Based Standard Test Framework

The *Standard Test Framework* proposed in this dissertation provides a framework for the experimental evaluation of RT-G&C algorithms. However, test facilities such as the POSEIDYN testbed involve a large amount of resources in terms time and capital. To make the evaluation methodology described in this dissertation more accessible, a simulation-based framework needs to be developed with relevant scenarios and constraints. The resulting framework and evaluation methodology can then be adapted for more widespread use.

8.3.2 Spacecraft Control

While the Sigma-Delta Modulator ($\Sigma\Delta$M) was shown to achieve a smaller steady-state error while using less propellant when compared to the Pulse-Width Modulator (PWM), the propellant consumption after achieving steady-state was observed to increased at a higher rate due to the shorter period of the limit cycle. Future work can explore various techniques to reduce the propellant consumption during steady-state while maintaining smaller tracking errors (compared to the PWM). Potential avenues of research include the incorporation of a deadzone to the $\Sigma\Delta$M, switching to a lower-bandwidth modulator, and even exploring more advanced modulators, such as an adaptive $\Sigma\Delta$M [176].

8.3.3 Evaluation of G&C Methods

In addition to evaluating different G&C methods to create a larger repository of data points, the G&C methods should be run at various sampling periods. While the RRT*-based was shown to produce a near-optimal path, the normalized control effort, ν_1, was observed to achieve a similar value as the AAPF. Therefore, it would be beneficial to determine if there exists any dependency on the sampling period and ν_1. If a dependency is found to exist, then an optimal value for the sampling period can be found which will further reduce the fuel consumption while maintaining a desired level of responsiveness.

8.3.4 Increasing the Computational Performance of G&C Methods

The ideal optimal G&C method solves the constrained optimal control problem (OCP) sufficiently fast. “Sufficiently fast” can be achieved through multiple means including computational architectures, speed of the compute element(s), efficient solvers, or better predictive models. The first method which should be explored is the use of custom programming solvers to create more-efficient solvers, such as the those presented in [177]–[179]. The use of a custom solver can result in speedup 10x-1000x by taking advantage of the specific structure of the problem [177].

The second avenue of research to increase the computational performance of G&C methods is the use of new compute architectures. Current compute architectures typically consist of a single serial processor that is shared with other processes. One possible compute architecture (loosely based on the NVIDIA Tegra series of processors and embedded systems, such as the NVIDIA Jetson TK1 [105]), illustrated in Figure 8.1, includes the use of a multi-core CPU, co-processor (or secondary CPU), and a graphic processing unit (GPU). The multi-core CPU could for a multi-threaded flight software design where the GNC algorithms and other housekeeping software run on individual threads.

Lastly, the third avenue of research is the use of better predictive models that can be incorporated into the guidance method. This area of research is considered to be an extension of the Navigation subset of algorithms of the GNC architecture. It is believed this can be achieved by using an Interacting Multiple Model filter approach [180]. The resulting model probabilities can then be used to inform the guidance algorithm of the appropriate model to use. However, the challenge arises of performing this motion association in an efficient manner as the Interacting Multiple Model approach is a resource-intensive task (in terms of computational processing and memory requirements) as there n number of filters being run updated at each measurement update. The resulting model probabilities can then be used to inform the guidance algorithm of the appropriate model to use.

8.3.5 Extension of the CL-RRT* to RPO

From the work performed in this dissertation, the advantages of sampling-based methods are clearly illustrated. Given the higher-dimensionality and nonlinear dynamics associated RPO, it would be advantageous to formally extend this method to this domain. While this may seem like a daunting task, only two functions in the algorithm need to be updated: the steering function which connects candidate nodes to existing nodes in the tree and the collision detection routine.

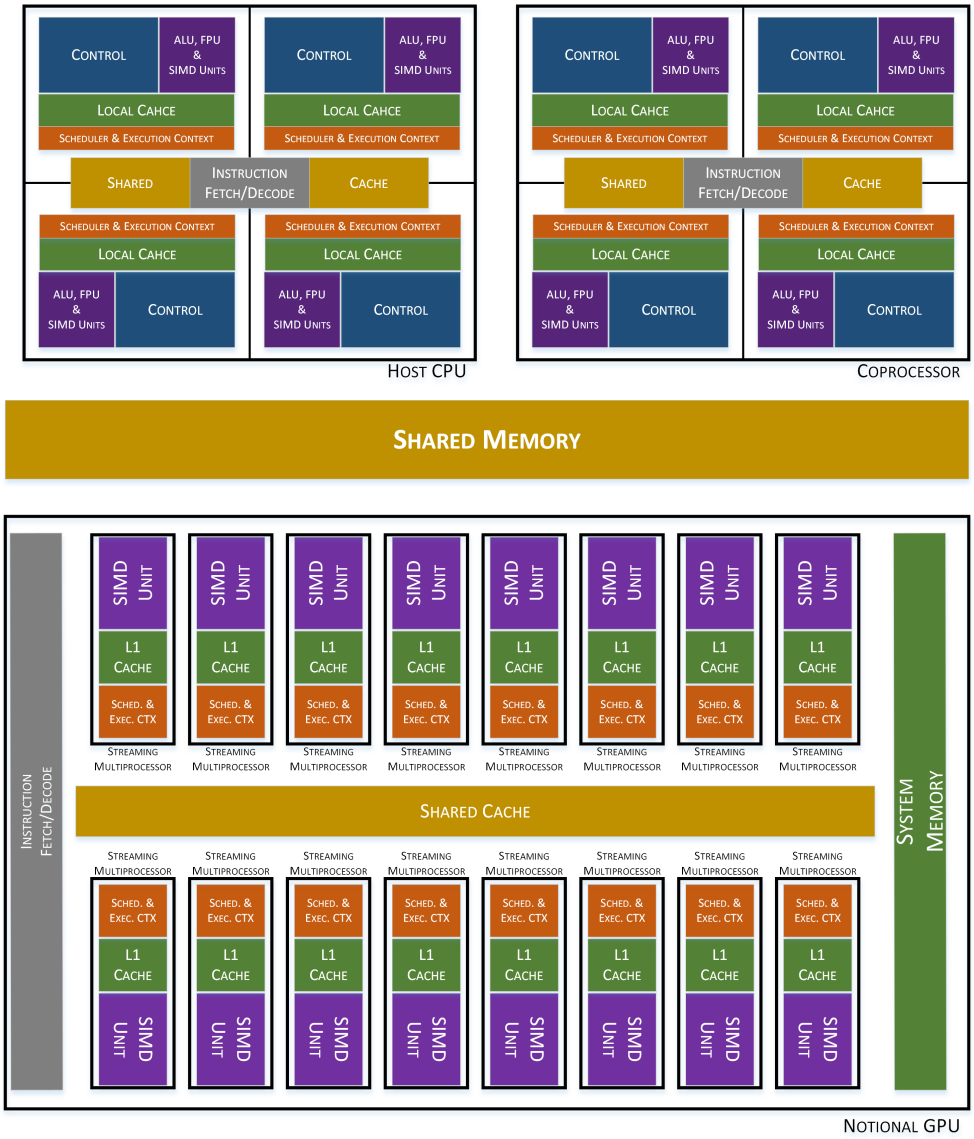


Figure 8.1. Notional computational architecture consisting of a CPU, Co-processor, and GPU.

APPENDIX A:
Validation of the Minimum Sampling Rate Metric

The work presented in this appendix was originally presented at the 27th AAS/AIAA Spaceflight Mechanics Conference in San Antonio, TX (February 2017) [74].

A.1 Numerical Case Study Overview

To aid in the validation of this metric, several numerical case studies are performed using representative mechanical systems, controllers, and actuators with discrete output levels. The systems considered include a three-axis reorientation maneuver of a spacecraft using reaction wheels and an Eigenaxis-Quaternion feedback controller [82]; a simple harmonic oscillator with a PD controller [129], [174]; and a double integrator (DI) system with reaction control jets using the minimum-time optimal control policy [76].

Unless otherwise specified, the mass of the system was chosen to be unity for simplicity. Each actuator is assumed to have a maximum control output of of unity and a resolution of 0.001, except for the reaction jets which is a purely discrete actuator with two states: ON or OFF. Additionally, each system was tuned to settle within 60 seconds and achieve stability with a delay of 1 second. Furthermore, each system utilizes continuous dynamics propagated using a variable step solver and a controller which is sampled with a fixed period, T_s. The controller sampling period was varied between 0.001 seconds (1 kHz) to 1 second (1 Hz). Each scenario was run for 120 seconds and the average steady-state error in x was recorded over the last 60 seconds. The results of each case study were then compared with the predicted minimum change in state given by Equation (7.10). To further aid in comparison, the dynamic bandwidth in Equation (7.10) is replaced with the second-order equivalent system bandwidth, $\omega_{\text{BW}}^2 = {}^{k}/_{m}$ and the result is presented.

A.1.1 Overall System Responses

The combined responses of each system are illustrated together in Figure A.1. Each system was found to exhibit a log-linear response in the average minimum change in the state x compared to the sampling rate. Additionally, in order to reduce the numerical noise and increase clarity, each response was curve-fit to a log-linear response. This response, and its estimated 1σ error are illustrated in subsequent figures to further highlight the trends of various responses.

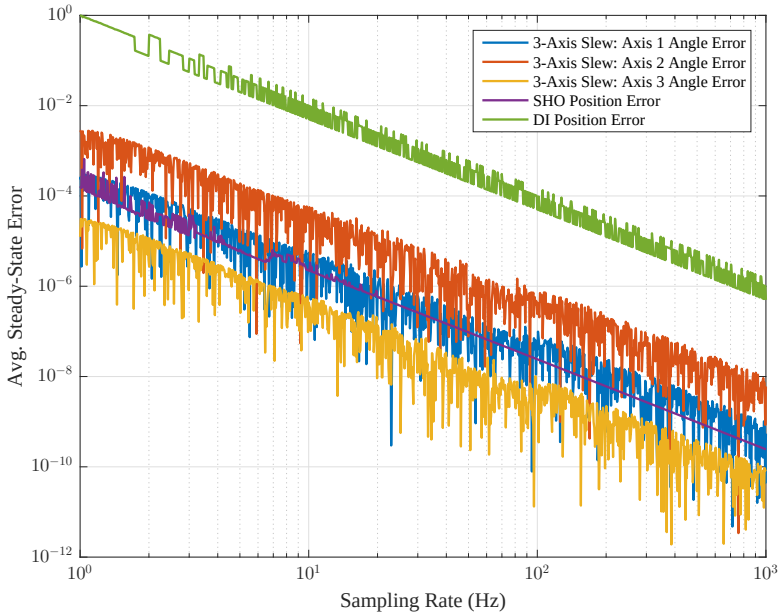


Figure A.1. Average steady-state error versus controller sample rate.

A.1.2 Eigenaxis-Quaternion Feedback Controller

The first case considered is a three-axis reorientation maneuver of spacecraft using an Eigenaxis-Quaternion feedback controller. Additionally, to produce the torque, a set of 3 orthogonal reaction wheels were modeled and placed in the loop This system is relevant as it is nonlinear in both the system dynamics and control. The system dynamics are governed by Euler's Equation and given as [82],

$$\boldsymbol{\tau} = \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega}^{\times}\mathbf{J}\boldsymbol{\omega} \tag{A.1}$$

where $\boldsymbol{\tau}$ is the control torque, $\mathbf{J}$ is the inertia matrix about the center of mass in the spacecraft body Cartesian coordinate system, $\boldsymbol{\omega}$ is the angular velocity of the spacecraft body frame with respect to the inertial frame, and $\boldsymbol{\omega}^{\times}$ is the matrical representation of the cross product operator [82]. Additionally, the Eigenaxis-Quaternion feedback controller is given as

$$\mathbf{u}(t) = -2k\eta_{\text{err}}\mathbf{J}\boldsymbol{\varepsilon}_{\text{err}} - c\mathbf{J}\boldsymbol{\omega}_{\text{err}} + \boldsymbol{\omega}_{\text{err}}^{\times}\mathbf{J}\boldsymbol{\omega}_{\text{err}} \tag{A.2}$$

where $\boldsymbol{\varepsilon}_{\text{error}} \in \mathbb{R}^3$ is a column vector containing the vector components of the error quaternion , $\eta_{\text{error}} \in \mathbb{R}$ is the scalar component of the error quaternion, and the scalar gains $c, k \in \mathbb{R}_+$. The stability of this controller was proven via Lyanpunov's Direct method [82], [90]. Lastly, the error angle was defined as [82]:

$$\theta_{\text{error}} = 2\cos^{-1}\left(\eta_{\text{error}}\right) \tag{A.3}$$

In order to examine the effect of various physical properties, the inertia matrix was chosen to be $\mathbf{J} = \mathrm{diag}(1, 0.1, 10)$. Additionally, to meet the 60-second settling time, the scalar gains were chosen to be $k = 0.3$ and $c = 1.0$. The results from the case study are presented in Figure A.2. When compared to the response of the system, the dynamic bandwidth closely matches the first-order fitted response of the numerical experiment. Additionally, the dynamic bandwidth was observed to maintain a similar level of error as the inertia was varied between the three axes. Lastly, the predicted minimum sampling rate associated with using the system bandwidth and dynamic bandwidth both over predict the minimum sampling rate in order to achieve a specific error level. The prediction associated with the dynamic bandwidth estimate is within an order of magnitude as the numerical response compared to using the system bandwidth metric. For this particular case, the predicted error for a given sample time agrees very well with the numerical experiment.

A.1.3 Simple Harmonic Oscillator with PD Control

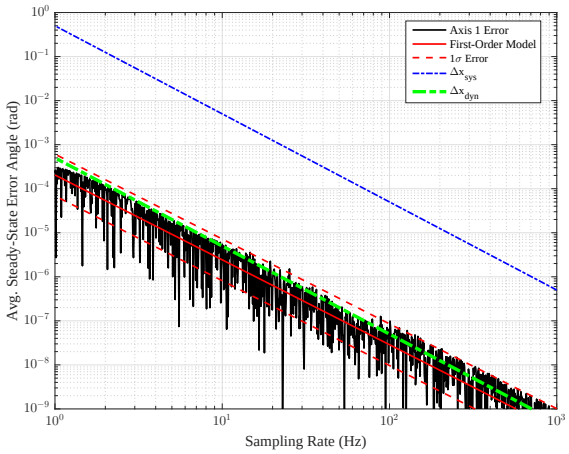
The second case considered is a simple harmonic oscillator (SHO) with a PD controller. The dynamics of a SHO are given as,

$$\ddot{x} = -\frac{k}{m}x \tag{A.4}$$

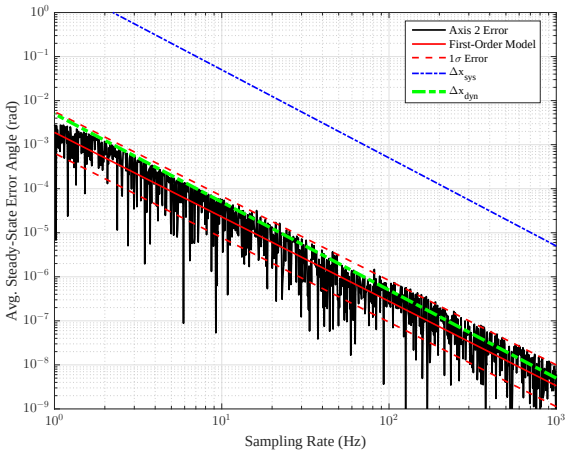
where $k \in \mathbb{R}_+$ and is typically referred to as the spring constant for a mechanical system [129], [174]. This is a representative second-order system with a common controller. As illustrated in Figure A.3 the dynamic bandwidth sufficiently estimates the sample time required to achieve a desired level of system performance. Similar to the first case, both metrics over-predict the sampling rate necessary to achieve a given level of steady-state error. Likewise, the prediction associated with the dynamic bandwidth is smaller than that associated with the system bandwidth and is within an order of magnitude of the average numerical response.

A.1.4 Minimum-Time Optimal Control for a Double Integrator System

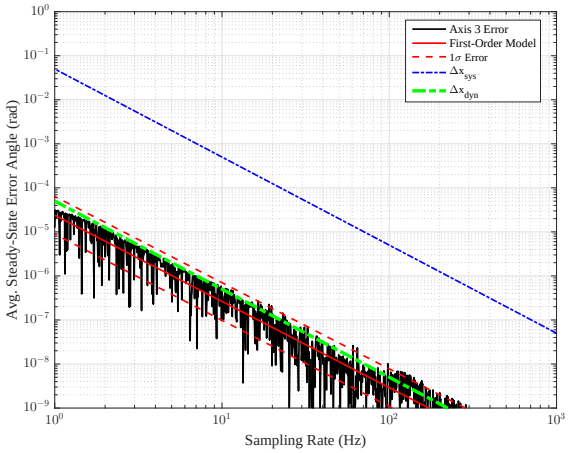
The third case study considered is a double integrator (DI) system under a minimum-time optimal control policy using discrete actuators, such as reaction jets. There are two primary methods of implementing this particular control policy. The first method is a clock-based method which assumes knowledge of the switching time. The second method (the method implemented for this case study) involves the use of a full-state



(a) Axis 1:($J_{11} = 1\,\mathrm{kg\,m^2}$)



(b) Axis 2: ($J_{22} = 0.1\,\mathrm{kg\,m^2}$)



(c) Axis 3: ($J_{33} = 10\,\mathrm{kg\,m^2}$)

Figure A.2. Average steady-state error versus controller sample rate for a 3-axis spacecraft reorientation maneuver.

switching curve to determine the switching point given as [76]

$$s\left(x(t), \dot{x}(t)\right) = x(t) + \frac{m}{2|u|}|\dot{x}(t)|\dot{x}(t) \tag{A.5}$$

where $m \in \mathbb{R}_+$ is the mass and $u \in \mathbb{R}$ is the control input. As illustrated by Figure A.4, the dynamic bandwidth sufficiently estimates the predicted steady-state error given a sampling time. Note, since the system bandwidth and dynamic bandwidth are the same numerically, both methods produce the same prediction. Lastly, compared to the SHO, the DI system with discrete actuators produces substantially more variation in the steady-state error.

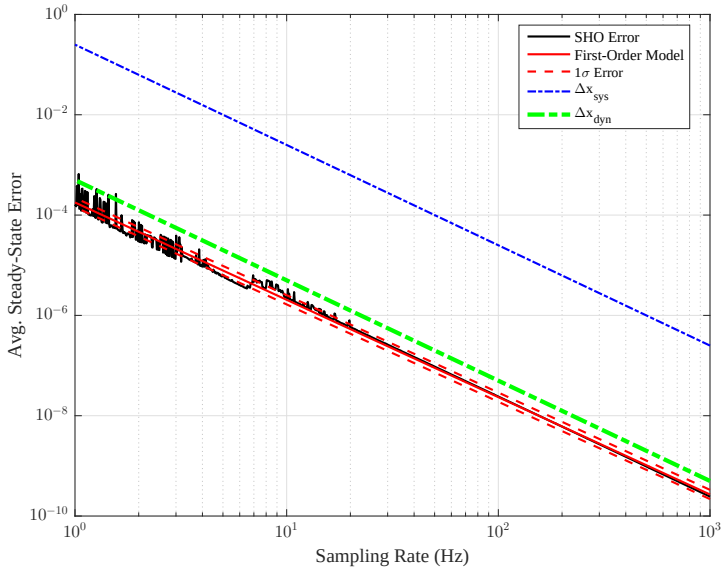


Figure A.3. Average steady-state error versus controller sample rate for a simple harmonic oscillator.

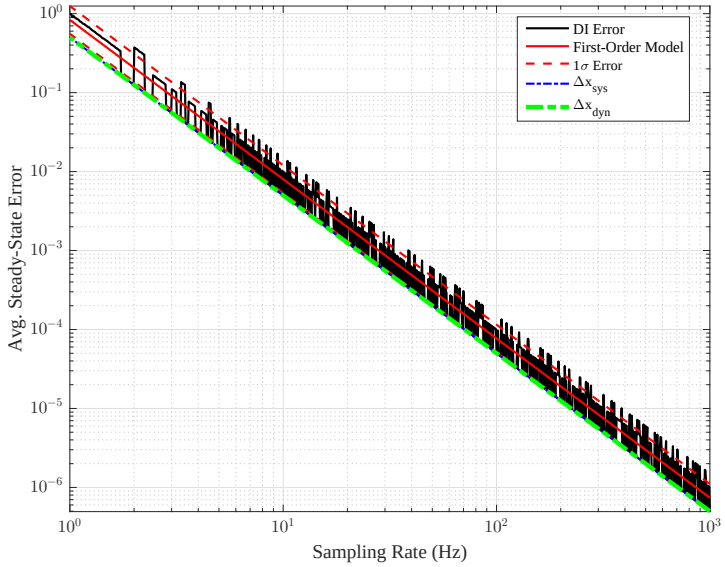


Figure A.4. Average steady-state error versus controller sample rate for a double-integrator system under the minimum-time optimal control policy.

A.2 Minimum Sampling Rate for Tracking Transient Responses

As demonstrated, this metric provides an order of magnitude estimate for the sampling rate required to track a control signal to within a specified level of error. The dynamic

bandwidth as given in Equation (7.9) provides the lower-bound on the steady-state tracking capability of the system. Redefining the dynamic bandwidth to be equivalent to the maximum acceleration of the system

$$\zeta_{\text{BW}} = \frac{\text{Maximum Control Effort}}{\text{Appropriate Physical Quantity}} \tag{A.6}$$

one can upper bound the transient tracking capability of the system. Rewriting Equation (7.8) and substituting in Equation (A.6), the maximum sampling time T_{RT} and minimum sampling rate f_{RT} can be upper-bounded respectively as

$$T_{\text{RT}} = \frac{1}{f_{\text{RT}}} = \frac{\Delta \dot{x}_{\max}}{\zeta_{\text{BW}}} \tag{A.7}$$

For example, consider the maximum change in error, ${}^{\Delta x_{err}}/_{\Delta t_c}$, of the previously examined SHO system during its transient period consisting of the first 60 seconds. As illustrated by Figure A.5, the dynamic bandwidth provides an estimated sampling rate to within 36% of the response of the system, while the system bandwidth significantly underestimates the required sampling rate to achieve a desired level of tracking error.

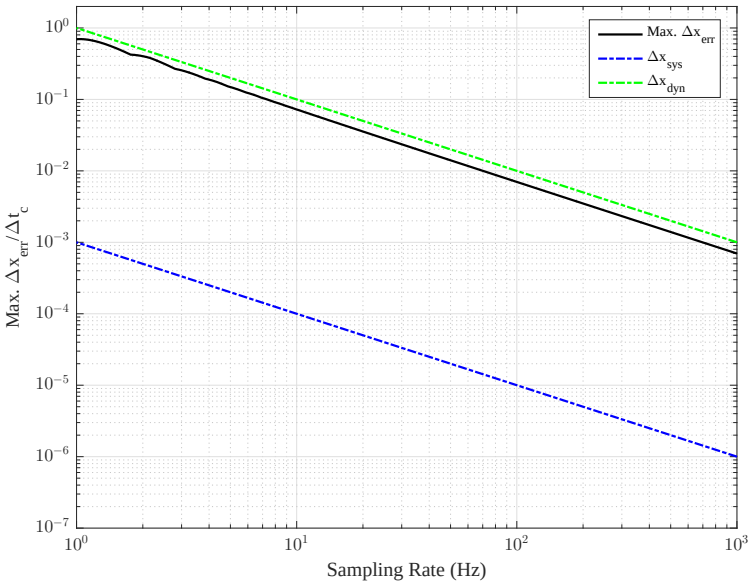


Figure A.5. Maximum change in error during the transient phase for a simple harmonic oscillator.

APPENDIX B:
Taguchi Analysis Method

B.1 Overview

The Taguchi analysis method (also referred to as the Taguchi parameter design approach by some authors [140]) was initially designed for use in process control product quality improvements [138], [139]. According to Roy, Taguchi's most significant contribution was the underlying framework for designing experiments and subsequent analysis of their results [138]. Due to this wide-reaching contribution, this method has been adapted to other applications, such as robust design optimization and analysis [137], [140], [181] and experiments involving agriculture and pharmaceuticals [138].

The Taguchi method, as it applies to analysis, will be described in this appendix.

B.2 Methodology

Roy defines a design of experiment (DOE) as the "technique of defining and investigating all possible conditionals in an experiment involving multiple factors" [138]. It is important to note, an input to the DOE is referred to as a factor while the value it takes on is referred to as its "level" [137], [138] Conventional approaches have utilized either a full-factorial or fractional-factorial unstructured DOE where the number of experiments (consisting of a combination of factors at various levels) grows exponentially and the contributions of each factor cannot be identified [137], [138], [140]. While the Taguchi method does not completely solve the dimensionality issue associated with a DOE with many factors, it standardizes both the implementation of the DOE and computation and interpretation of the results [138].

Compared to the brute-force approach of "think, try this, think, try that", the Taguchi method is applied in four distinct steps: brainstorming, experimentation, analysis, and confirmation [138]. To apply the Taguchi method, the experiment must be developed using a standard set of orthogonal arrays, referred to as Taguchi orthogonal arrays, that specify the minimum number of experiments given the number of factors and the levels of each factor [137], [138]. If determining interactions between factors is important, the standard Taguchi arrays must be modified. Since interactions are not considered in the applications presented in this dissertation, the reader is referred to [137]–[139] for further information on interactions and how to modify the standard methodology. Since the Taguchi method was formed around measuring deviations from a nominal

value [138], [139], the desired output from each experiment is a measurement. This measurement, denoted here as the performance criterion (Roy refers to it as an "Overall Evaluation Criteria" [138]) and is chosen as the variable which is affected by the level of the factors [138]. Lastly, the appropriate level of analysis must be determined.

As the original application of the Taguchi method was for quality engineering, the analysis associated with the standard Taguchi method allows the user to identify the impact of each factor on the performance criterion, but also identify any interactions (if the experiment was chosen to include interactions) as well as pick the set of levels for each factor which optimize a certain performance criterion [137], [138]. However, for this application, it is only necessary to determine the average effect of each factor on the output. As such, the reader is encouraged to refer to [137]–[139] for further analysis using the Taguchi method. Once the factors, levels, the performance criterion are selected, the brainstorming phase is completed [138] and the experiment and data collection phase can begin.

B.3 Taguchi Orthogonal Arrays

The experimentation phase of the Taguchi method is predicated upon the use of a Taguchi orthogonal array. Each Taguchi array is denoted using the following nomenclature:

$$\mathsf{L}_{N_{exper}}(N_{lvl}^{N_{fac}})$$

where N_{exper} is the total number of experiments and N_{lvl} is the number of levels for each of the number of factors, N_{fac}. In this dissertation, a two-level, three-factor, nine-experiment array was selected for the studies performed in Section 5.5.1 and 6.8. The $\mathsf{L}_9(3^2)$ standard orthogonal array used throughout this dissertation is given in Table B.1. In this array, the first column corresponds to each experiment, or run. The columns in the middle array represent the level for each factor, where each factor is denoted using a capitalized letter, beginning with "A" for the first factor. The last column is composed of the result, or performance criterion, from nine individual experiments, denoted as X_i, for $i = 1, 2, \ldots, 9$. When conducting non-virtual (i.e., physical) experiments, Roy suggests running the experiments in a random order to remove any effects due to the experiment setup [138]. Once the experiments have been performed and the data recorded, the average effects of each factor on the performance criterion can be assessed.

Table B.1. $L_9(3^2)$ Taguchi orthogonal array.

	Factor A	Factor B	X
1	1	1	X_1
2	1	2	X_2
3	1	3	X_3
4	2	1	X_4
5	2	2	X_5
6	2	3	X_6
7	3	1	X_7
8	3	2	X_8
9	3	3	X_9

B.4 Computing the Average Effects

In order to assess the effect, or contribution, of a various factor on the performance criterion, the average effect can be simply computed. The average effect of each factor at each level can be computed by summing all the elements X_i that contain that factor at the level [137], [138]. For example, using Table B.1, the average effect of the factor A at level 1 can be computed as follows:

$$\bar{A_1} = \frac{X_1 + X_2 + X_3}{3} \tag{B.1}$$

Likewise, the computation of the average effect of the factor B at level is

$$\bar{B_1} = \frac{X_1 + X_4 + X_7}{3} \tag{B.2}$$

Lastly, each of the average effects may be normalized by the grand average X_{avg} [137],

$$X_{\text{avg}} = \frac{1}{N_{exper}} \sum_{i=1}^{N_{exper}} X_i \tag{B.3}$$

Once these values are computed, they can be plotted for visual inspection and interpretation using a quality characteristic (“nominal is best”, “bigger is better”, or “smaller is better”) [137], [138].

The utility of the Taguchi method arises from the fact that vast insight into the underlying sensitivities of the system being analyzed can be made despite the small number of experiments performed [137]. To illustrate this, an average effect plot for a notional DOE performed using a $L_9(3^2)$ standard orthogonal array is presented in Figure B.1. As illustrated, as factor B is varied from B_1 to B_3, the performance criterion was found to

decrease while the opposite holds for factor A as it is varied from A_1 to A_3. In contrast, no effect on the performance criterion is observed when factor A is varied between A_2 and A_3 as the line connecting the two points is horizontal. Furthermore, a *qualitative* assessment on the presence of any interactions can be determined from this plot simply by examining the relatively slopes of the lines for each factor. If they placed on top of each other, such as in Figure B.2, one can assess that an interaction between the two factors does exist since they are not (nearly) parallel [137], [138]. Further analysis can be performed to determine the extent of any interaction between the two factors and the average effect on the performance criterion [137], [138].

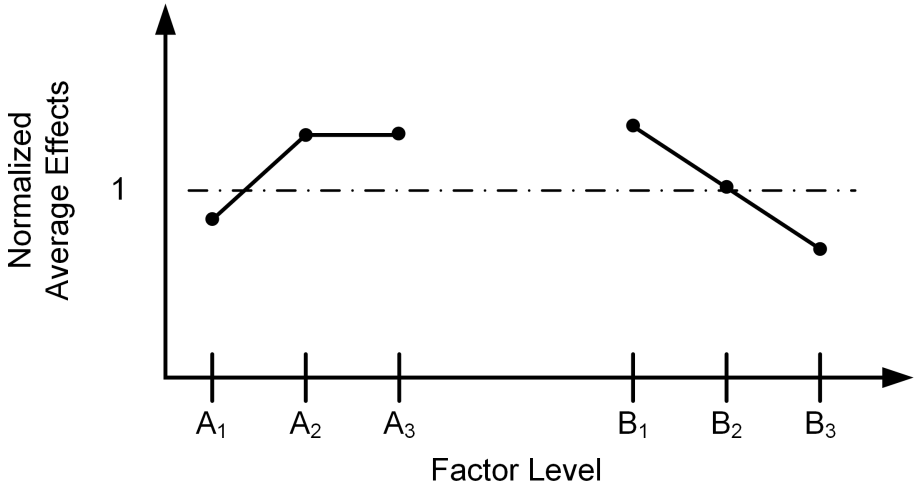


Figure B.1. Notional average effect plot for a $L_9(3^2)$ DOE.

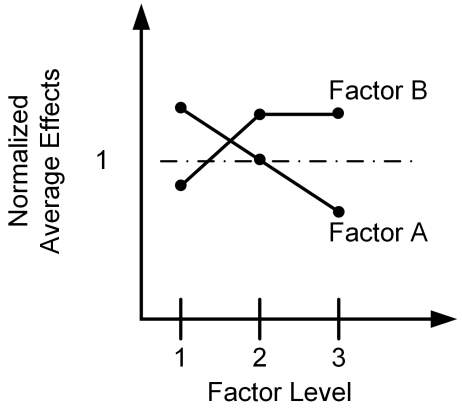


Figure B.2. Factor A and B average effects overlaid for the notional $L_9(3^2)$ DOE.

APPENDIX C:
Publication History

C.1 Patents

J. Virgili–Llop, R. Zappulla II, and M. Romano, "Dynamically tilting flat table to impart a time-varying gravity-induced acceleration on a floating spacecraft simulator," U.S. Patent Pending 62 455 775, Feb. 07, 2017

C.2 Supplemental Media

Demonstrations of select maneuvers are available on the NPS Spacecraft Robotics Laboratory YouTube Channel:
https://www.youtube.com/channel/UCyPOVZ2Xrcz_onUMs578aCw.

A MathWorks user story about the POSEIDYN test bed and the TRIDENT-GNC software architecture was published on May 19, 2017 and can be found by clicking here.

C.3 Journal Publications

C.3.1 Published

R. Zappulla II, J. Virgili–Llop, H. Park, C. Zagaris, and M. Romano, "Dynamic air-bearing hardware-in-the-loop testbed to experimentally evaluate autonomous spacecraft proximity maneuvers," *Journal of Spacecraft and Rockets*, 2017, DOI: 10.2514/1.A33769 (In Press)

J. Virgili-Llop, C. Zagaris, H. Park, R. Zappulla II, and M. Romano, "Experimental evaluation of model predictive control and inverse dynamics control for spacecraft proximity and docking maneuvers," *CEAS Space Journal*, 2017, (In Press, 22 May 2017)

C.3.2 Accepted

R. Zappulla II, J. Virgili-Llop, and M. Romano, "Spacecraft thruster control via sigma-delta modulation," *Journal of Guidance, Control, and Dynamics*, 2017, (Accepted 13 May 2017)

C.3.3 Under Review

R. Zappulla II, H. Park, J. Virgili-Llop, and M. Romano, "Real-time autonomous spacecraft proximity maneuvers and docking using an adaptive artificial potential field," *IEEE Transactions on Control Systems Technology*, pp. 1–12, 2017, (Under Review)

R. Zappulla II, H. Park, C. Zagaris, J. Virgili-Llop, and M. Romano, "A metric for the equitable comparison of guidance and control methods," *Journal of Guidance, Control, and Dynamics*, 2017, (Under Review)

J. Virgili-Llop, J. V. Drew, R. Zappulla II, and M. Romano, "Autonomous capture of a resident space object by a spacecraft with a robotic manipulator: Analysis, simulation and experiments," *Aerospace Science and Technology*, 2017, (Under Review)

C.4 Conference Proceedings

R. Zappulla II, H. Park, J. Virgili-Llop, and M. Romano, "Experiments on autonomous spacecraft rendezvous and docking using an adaptive artificial potential field approach," in *26th AAS/AIAA SpaceFlight Mechanics Meeting*. San Diego, CA: Univelt Inc., 2016, paper AAS 16-459, pp. 4461–4478. Available: http://hdl.handle.net/10945/50864

[Best Paper Award] J. Virgili-Llop, C. Zagaris, H. Park, R. Zappulla II, and M. Romano, "Experimental evaluation of model predictive control and inverse dynamics control for spacecraft proximity and docking maneuvers," *6th International Conference on Astrodynamics Tools and Techniques*, Mar.14–17 2016, Darmstad, Germany. Available: http://hdl.handle.net/10945/50868

R. Zappulla II, J. Virgili-Llop, H. Park, C. Zagaris, A. Sharp, and M. Romano, "Floating spacecraft simulator test bed for the experimental testing of autonomous guidance, navigation, & control of spacecraft proximity maneuvers and operations," *AIAA/AAS Astrodynamics Specialist Conference, AIAA SPACE Forum*, Sep.13–16 2016, AIAA Paper 2016-5268. Available: http://dx.doi.org/10.2514/6.2016-5268

H. Park, C. Zagaris, J. Virgili–Llop, R. Zappulla II, I. Kolmanovsky, and M. Romano, "Analysis and experimentation of model predictive control for spacecraft operations with multiple obstacle avoidance," *AIAA/AAS Astrodynamics Specialist Conference, AIAA SPACE Forum*, Sep.13–16 2016, AIAA 2016-5273. Available: http://dx.doi.org/10.2514/6.2016-5273

J. Virgili-Llop, J. V. Drew, R. Zappulla II, and M. Romano, "Autonomous capture of a resident space object by a spacecraft with a robotic manipulator: Analysis, simulation and

experiments," *AIAA/AAS Astrodynamics Specialist Conference, AIAA SPACE Forum*, Sep.13–16 2016. Available: http://dx.doi.org/10.2514/6.2016-5269

R. Zappulla II, J. Virgili–Llop, and M. Romano, "Near-optimal real-time spacecraft guidance and control using harmonic potential functions and modified RRT*," *27th AAS/AIAA Spaceflight Mechanics Meeting*, Feb. 2017, AAS Paper 17-420. Available: http://hdl.handle.net/10945/51981

R. Zappulla II and M. Romano, "A systematic approach to determining the minimum sampling rate for real-time spacecraft control," *27th AAS/AIAA Spaceflight Mechanics Meeting*, Feb. 2017, AAS Paper 17-424. Available: http://hdl.handle.net/10945/51982

H. Park, R. Zappulla II, C. Zagaris, J. Virgili–Llop, and M. Romano, "Nonlinear model predictive control for spacecraft rendezvous and docking with a rotating target," *27th AAS/AIAA Spaceflight Mechanics Meeting*, Feb. 2017, AAS Paper 17-496. Available: http://hdl.handle.net/10945/51981

J. Virgili-Llop, C. Zagaris, R. Zappulla II, A. Bradstreet, and M. Romano, "Convex optimization for proximity maneuvering of a spacecraft with a robitc manipulator," *27th AAS/AIAA Spaceflight Mechanics Meeting*, Feb. 2017, AAS Paper 17-418. Available: http://hdl.handle.net/10945/51983

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] A. Einstein and L. Infeld, *The Evolution of Phyiscs from Early Concepts to Relativity and Quanta*. Simon & Schuster, 1938, p. 95.

[2] *Vision and Voyages for Planetary Science in the Decade 2013–2022*. Washington, DC: The National Academies Press, 2011. Available: http://nap.edu/13117

[3] NASA Technology Roadmaps TA4: Robotics and Autonomous Systems. (2015, July). National Aeronautics and Space Administration. [Online]. Available: https://www.nasa.gov/offices/oct/home/roadmaps/index.html. (Retrieved March 21, 2017).

[4] M. B. Quadrelli, L. J. Wood, J. E. Riedel, M. C. McHenry, M. Aung, L. A. Cangahuala, R. A. Volpe, P. M. Beauchamp, and J. A. Cutts, "Guidance, navigation, and control technology assessment for future planetary science missions," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 7, pp. 1165–1186, Feb. 2015.

[5] D. C. Woffinden and D. K. Geller, "Navigating the road to autonomous orbtial rendezvous," *Journal of Spacecraft and Rockets*, vol. 44, no. 4, pp. 898–909, July-August 2007.

[6] L. Blackmore, *Autonomous Precision Landing of Space Rockets*. Washington,DC: National Academy of Engineering, 2016, vol. 46, no. 4, pp. 16–20.

[7] D. A. Spencer, S. B. Chait, P. Z. Schulte, K. J. Okseniuk, and M. Veto, "Prox-1 university-class mission to demonstrate automated proximity operations," *Journal of Spacecraft and Rockets*, vol. 53, no. 5, Special Section on Advancements in Small Satellite Technology and Applications, pp. 847–863, 2016.

[8] P. Z. Schulte and D. A. Spencer, "Development of an integrated spacecraft guidance, navigation, & control subsystem for automated proximity operations," *Acta Astronautica*, vol. 118, pp. 168 – 186, 2016. Available: http://www.sciencedirect.com/science/article/pii/S0094576515003811

[9] "Department of Defense Space Technology Guide:FY 2000-01," [Online], Office of the Secretary of Defense, pp. 12–8, 2001. Available: handle.dtic.mil/100.2/ADA393837

[10] "Defense science and technology success stories," [Online], Defense Research & Engineering, pp. 134–135, 2007. Available: http://www.dtic.mil/dtic/tr/fulltext/u2/a568949.pdf

[11] S. D'Amico, J.-S. Ardaens, and R. Larsson, "Spaceborne autonomous formation-flying experiment on the prisma mission," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 3, pp. 834–850, 2012.

[12] Soyuz. National Aeronautics and Space Administration. [Online Image]. Available: https://www.nasa.gov/images/content/113447main_iss010e24875hires.jpg. (Retrieved 27 April 2017).

[13] DART. National Aeronautics and Space Administration. [Online Image]. Available: https://www.nasa.gov/images/content/67032main_dart1b_800x600.jpg. (Retrieved 27 April 2017).

[14] ATV. National Aeronautics and Space Administration. [Online Image]. Available: http://www.nasa.gov/images/content/715872main_iss030e175078_226.jpg. (Retrieved 27 April 2017).

[15] Defense Advanced Research Projects Agency. [Online Image]. Available: http://archive.darpa.mil/orbitalexpress/images/FamilyPortrait_Final.jpg. (Retrieved 27 April 2017).

[16] CRS-8 Landing. SpaceX. [Online Image]. Available: https://www.flickr.com/photos/spacex/26265214251/. (Retrieved 27 April 2017).

[17] Space System Design Laboratory. Prox-1 Spacecraft. Georgia Institute of Technology. [Online Image]. Available: http://prox-1.gatech.edu/index.html. (Retrieved 27 April 2017).

[18] D. Pinard, S. Reynaud, P. Delpy, and S. E. Strandmoe, "Accurate and autonomous navigation for the ATV," *Aerospace Science and Technology*, vol. 11, no. 6, pp. 490 – 498, 2007.

[19] J. D. Muñoz and N. G. Fitz-Coy, "Rapid path-planning options for autonomous proximity operations of spacecraft," *AIAA/AAS Astrodynamics Specialist Conference*, 2010.

[20] J. D. Muñoz, "Rapid path-planning algorithms for autonomous proximity operations of satellites," Ph.D. dissertation, University of Florida, 2011. Available: http://etd.fcla.edu/UF/UFE0042882/munoz_j.pdf

[21] D. Cornier, D. Berthelier, H. Requiston, E. Zekri, and R. Chase, "Automated transfer vehicle proximity flight safety overview," in *1st IAASS Conference Space Safety, A New Beginning*. Noordwijk, The Netherlands: European Space Agency Publ. Division, ESTEC, 2005, vol. 599, pp. 89–95.

[22] M. Romano, D. A. Friedman, and T. J. Shay, "Laboratory experimentation of autonomous spacecraft approach and docking to a collaborative target," *Journal of Spacecraft and Rockets*, vol. 44, no. 1, pp. 164–173, 2007.

[23] I. M. Ross and F. Fahroo, "Issues in the real-time computation of optimal control," *Mathematical and Computer Modelling*, vol. 43, no. 9–10, pp. 1172 – 1188, 2006. Available: http://www.sciencedirect.com/science/article/pii/S089571770500542X

[24] P. A. Laplante, *Real-Time Systems Design and Analysis*, 3rd ed. Piscataway, NJ: IEEE Press, 2004, pp. xvii–xx, 1–21, 73–74, 225–231, 445–473.

[25] A. S. Keys, J. H. Adams, D. O. Frazier, M. C. Patrick, M. D. Watson, M. A. Johnson, J. D. Cressler, and E. A. Kolawa, "Developments in radiation-hardened electronics applicable to the vision for space exploration," *AIAA SPACE 2007 Conference & Exposition*, Sep. 18–20 2007, AIAA Paper 2007-6269.

[26] J. A. Starek, B. Açıkmeşe, I. A. Nesnas, and M. Pavone, *Spacecraft Autonomy Challenges for Next-Generation Space Missions*. Springer, 2016, pp. 1–48.

[27] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[28] I. Lopez and C. R. McInnes, "Autonomous rendezvous using artificial potential function guidance," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 2, pp. 237 – 241, March–April 1995.

[29] S. B. McCamish, M. Romano, S. Nolet, C. M. Edwards, and D. W. Miller, "Flight testing of multiple-spacecraft control on SPHERES during close-proximity operations," *Journal of Spacecraft and Rockets*, vol. 46, no. 6, pp. 1202–1213, 2009.

[30] R. Bevilacqua, T. Lehmann, and M. Romano, "Development and experimentation of LQR/APF guidance and control for autonomous proximity maneuvers of multiple spacecraft," *Acta Astronautica*, vol. 68, no. 7, pp. 1260–1275, 2011.

[31] R. Zappulla II, J. Virgili–Llop, H. Park, C. Zagaris, and M. Romano, "Dynamic air-bearing hardware-in-the-loop testbed to experimentally evaluate autonomous spacecraft proximity maneuvers," *Journal of Spacecraft and Rockets*, 2017, DOI: 10.2514/1.A33769 (In Press).

[32] R. Zappulla II, H. Park, J. Virgili-Llop, and M. Romano, "Experiments on autonomous spacecraft rendezvous and docking using an adaptive artificial potential field approach," in *26th AAS/AIAA SpaceFlight Mechanics Meeting*. San Diego, CA: Univelt Inc., 2016, paper AAS 16-459, pp. 4461–4478. Available: http://hdl.handle.net/10945/50864

[33] R. Zappulla II, J. Virgili-Llop, H. Park, C. Zagaris, A. Sharp, and M. Romano, "Floating spacecraft simulator test bed for the experimental testing of autonomous guidance, navigation, & control of spacecraft proximity maneuvers and operations," *AIAA/AAS Astrodynamics Specialist Conference, AIAA SPACE Forum*, Sep.13–16 2016, AIAA Paper 2016-5268. Available: http://dx.doi.org/10.2514/6.2016-5268

[34] G. Li, Y. Tamura, A. Yamashita, and H. Asama, "Effective improved artificial potential field-based regression search method for autonomous mobile robot path planning," *International Journal of Mechatronics and Automation*, vol. 3, no. 3, pp. 141–170, 2013.

[35] R. A. Fields, "Continuous control artificial potential function methods and optimal control," Master's thesis, Air Force Institute of Technology, March 2014. Available: http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA599786

[36] J. D. Anderson, *Fundamentals of Aerodyanmics*, 4th ed. Boston, MA: McGraw-Hill, 2006, ch. 2 – 3, 6.

[37] S. Akishita, S. Kawamura, and K. Hayashi, "New navigation function utilizing hydrodynamic potential for mobile robot," in *Intelligent Motion Control, 1990. Proceedings of the IEEE International Workshop on*, Aug 1990, vol. 2, pp. 413 – 417.

[38] S. Akishita, S. Kawamura, and T. Hisanobu, "Velocity potential approach to path planning for avoiding moving obstacles," *Advanced robotics*, vol. 7, no. 5, pp. 463 – 478, 1992.

[39] J. O. Kim and P. K. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 338 – 349, June 1992.

[40] H. J. S. Feder and J. J. E. Slotine, "Real-time path planning using harmonic potentials in dynamic environments," in *Proceedings of International Conference on Robotics and Automation*, Apr 1997, vol. 1, pp. 874–881 vol.1.

[41] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, Oct 1992.

[42] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, Sept 2009.

[43] S. Waydo and R. M. Murray, "Vehicle motion planning using stream functions," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, September 2003, vol. 2, pp. 2484–2491 vol.2.

[44] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT*," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 1478–1483.

[45] J. Virgili-Llop, J. V. Drew, R. Zappulla II, and M. Romano, "Autonomous capture of a resident space object by a spacecraft with a robotic manipulator: Analysis, simulation and experiments," *Aerospace Science and Technology*, 2017, (Under Review).

[46] J. Virgili-Llop, J. V. Drew, R. Zappulla II, and M. Romano, "Autonomous capture of a resident space object by a spacecraft with a robotic manipulator: Analysis, simulation and experiments," *AIAA/AAS Astrodynamics Specialist Conference, AIAA SPACE Forum*, Sep.13–16 2016. Available: http://dx.doi.org/10.2514/6.2016-5269

[47] J. Virgili-Llop, J. Drew, and M. Romano, "Design and parameter identification by laboratory experiments of a prototype modular robotic arm for orbiting spacecraft applications," in *6th International Conference on Astrodynamics Tools and Techniques (ICATT)*, Darmstad, Germany, Mar.14–17 2016.

[48] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Progress in Aerospace Sciences*, vol. 68, pp. 1 – 26, 2014. Available: http://www.sciencedirect.com/science/article/pii/S0376042114000347

[49] P. Lu, "Introducing computational guidance and control," *Journal of Guidance, Control, and Dynamics*, no. 2, 2017.

[50] P. Tsiotras and M. Mesbahi, "Toward an algorithmic control theory," *Journal of Guidance, Control, and Dynamics*, no. 2, pp. 194–196, Feb. 2017.

[51] M. Ciarcià, A. Grompone, and M. Romano, "A near-optimal guidance for cooperative docking maneuvers," *Acta Astronautica*, vol. 102, pp. 367 – 377, 2014. Available: http://www.sciencedirect.com/science/article/pii/S0094576514000046

[52] J. Virgili-Llop, C. Zagaris, H. Park, R. Zappulla II, and M. Romano, "Experimental evaluation of model predictive control and inverse dynamics control for spacecraft proximity and docking maneuvers," *6th International Conference on Astrodynamics Tools and Techniques*, Mar.14–17 2016, Darmstad, Germany. Available: http://hdl.handle.net/10945/50868

[53] M. Wilde, M. Ciarcià, A. Grompone, and M. Romano, "Experimental characterization of inverse dynamics guidance in docking with a rotating target," *Journal of Guidance, Control, and Dynamics*, no. 6, 2016.

[54] G. A. Boyarko, "Spacecraft guidance strategies for proximity maneuvering and close approach with a tumbling object," Ph.D. dissertation, Naval Postgraduate School, Monterey, CA, Mar. 2010.

[55] C. Zagaris, M. Baldwin, C. Jewison, and C. Petersen, "Survey of spacecraft rendezvous and proximity guidance algorithms for on-board implementation," in *25th AAS/AIAA Space Flight Mechanics Meeting, AAS 15-334*, Williamsburg, VA, Jan.11–15 2015.

[56] H. Park, R. Zappulla II, C. Zagaris, J. Virgili–Llop, and M. Romano, "Nonlinear model predictive control for spacecraft rendezvous and docking with a rotating target," *27th AAS/AIAA Spaceflight Mechanics Meeting*, Feb. 2017, AAS Paper 17-496. Available: http://hdl.handle.net/10945/51981

[57] E. N. Hartley, P. A. Trodden, A. G. Richards, and J. M. Maciejowski, "Model predictive control system design and implementation for spacecraft rendezvous," *Control Engineering Practice*, vol. 20, no. 7, pp. 695–713, 2012.

[58] 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*, vol. 22, no. 12, pp. 1398–1427, 2012.

[59] C. Petersen, A. Jaunzemis, M. Baldwin, M. Holzinger, and I. Kolmanovsky, "Model predictive control and extended command governor for improving robustness of relative motion guidance and control," in *Advances in The Astronautical Sciences*, Santa Fe, NM, 2014, vol. 152.

[60] A. Weiss, M. Baldwin, R. S. Erwin, and I. Kolmanovsky, "Model predictive control for spacecraft rendezvous and docking: Strategies for handling constraints and case studies," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1638–1647, 2015.

[61] H. Park, C. Zagaris, J. Virgili–Llop, R. Zappulla II, I. Kolmanovsky, and M. Romano, "Analysis and experimentation of model predictive control for spacecraft operations with multiple obstacle avoidance," *AIAA/AAS Astrodynamics Specialist Conference, AIAA SPACE Forum*, Sep.13–16 2016, AIAA 2016-5273. Available: http://dx.doi.org/10.2514/6.2016-5273

[62] H. Park, S. D. Cairano, and I. Kolmanovsky, "Model predictive control of spacecraft docking with a non-rotating platform," *IFAC Proceedings Volumes*, vol. 44,

no. 1, pp. 8485 – 8490, 2011. Available: http://www.sciencedirect.com/science/article/pii/S1474667016449736

[63] H. Park, S. D. Cairano, and I. Kolmanovsky, "Model predictive control for spacecraft rendezvous and docking with a rotating/tumbling platform and for debris avoidance," in *Proceedings of the 2011 American Control Conference*, June 2011, pp. 1922–1927.

[64] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, 1999, vol. 1, pp. 473–479 vol.1.

[65] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," *Robotics Science and Systems VI*, vol. 104, June 2010.

[66] S. M. LaValle and J. James J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001. Available: http://dx.doi.org/10.1177/02783640122067453

[67] A. H. Qureshi, K. F. Iqbal, S. M. Qamar, F. Islam, Y. Ayaz, and N. Muhammad, "Potential guided directional-RRT* for accelerated motion planning in cluttered environments," in *2013 IEEE International Conference on Mechatronics and Automation*, Aug 2013, pp. 519–524.

[68] P. Pharpatara, B. Hérissé, and Y. Bestaoui, "3-D trajectory planning of aerial vehicles using RRT*," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 1116–1123, May 2017.

[69] M. Ciarcià, R. Cristi, and M. Romano, "Experimental emulation of the scaled Clohessy-Wiltshire dynamics on a flat air-bearing testbed," *AIAA Guidance, Navigation, and Control Conference, AIAA SciTech Forum*, Jan.9–13 2017.

[70] R. Zappulla II, J. Virgili-Llop, and M. Romano, "Spacecraft thruster control via sigma-delta modulation," *Journal of Guidance, Control, and Dynamics*, 2017, (Accepted 13 May 2017).

[71] R. Zappulla II, H. Park, J. Virgili-Llop, and M. Romano, "Real-time autonomous spacecraft proximity maneuvers and docking using an adaptive artificial potential field," *IEEE Transactions on Control Systems Technology*, pp. 1–12, 2017, (Under Review).

[72] J. Virgili-Llop, C. Zagaris, H. Park, R. Zappulla II, and M. Romano, "Experimental evaluation of model predictive control and inverse dynamics control for spacecraft proximity and docking maneuvers," *CEAS Space Journal*, 2017, (In Press, 22 May 2017).

[73] R. Zappulla II, J. Virgili–Llop, and M. Romano, "Near-optimal real-time spacecraft guidance and control using harmonic potential functions and modified RRT*," *27th AAS/AIAA Spaceflight Mechanics Meeting*, Feb. 2017, AAS Paper 17-420. Available: http://hdl.handle.net/10945/51981

[74] R. Zappulla II and M. Romano, "A systematic approach to determining the minimum sampling rate for real-time spacecraft control," *27th AAS/AIAA Spaceflight Mechanics Meeting*, Feb. 2017, AAS Paper 17-424. Available: http://hdl.handle.net/10945/51982

[75] R. Zappulla II, H. Park, C. Zagaris, J. Virgili-Llop, and M. Romano, "A metric for the equitable comparison of guidance and control methods," *Journal of Guidance, Control, and Dynamics*, 2017, (Under Review).

[76] D. E. Kirk, *Optimal Control Theory* An Introduction. Englewood Cliffs, NJ: Prentice-Hall, 2004, pp. 1–7, 10–16, 115–123, 184–188, 227–233, 259–280.

[77] ISS Trajectory Data. National Aeronautics and Space Administration. [Online]. Available: https://spaceflight.nasa.gov/realdata/sightings/SSapplications/Post/JavaSSOP/orbit/ISS/SVPOST.html. (Retrieved 21 March 2016).

[78] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*, 4th ed. Hawthorne, CA: Microcosm Press, 2013, pp. 155–157, 388–422, 777–833.

[79] R. R. Bate, D. D. Mueller, and J. E. White, *Fundementals of Astrodynamics*. Mineola, NY: Dover, 1971, ch. 1, 2.

[80] The Blue Marble: Land Surface, Ocean Color, Sea Ice and Clouds. National Aeronautics and Space Administration. [Online]. Available: https://visibleearth.nasa.gov/view.php?id=57735. (Retrieved 29 March 2017).

[81] D. P. Scharf, F. Y. Hadaegh, and B. H. Kang, "On the validity of the double integrator approximation in deep space formation flying," *International Symposium on Formation Flying Missions and Technologies*, 2002, toulouse, France. Available: http://hdl.handle.net/2014/37196

[82] B. Wie, *Space Vehicle Dynamics and Control*, 2nd ed. Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 2008, pp. 77–78, 350–353, 359–361, 425–444, 478–482, 558–574.

[83] E. Frazzoli, M. Dahleh, E. Feron, and R. Kornfeld, "A randomized attitude slew planning algorithm for autonomous spacecraft," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2001, aIAA Paper 2001-4155.

[84] STS-134 Flight Day 8. National Aeronautics and Space Administration. [Online Image]. Available: https://www.nasa.gov/images/content/559819main_iss027e036636_hires.jpg. (Retrieved 27 April 2017).

[85] J. Virgili-Llop, C. Zagaris, R. Zappulla II, A. Bradstreet, and M. Romano, "Convex optimization for proximity maneuvering of a spacecraft with a robitc manipulator," *27th AAS/AIAA Spaceflight Mechanics Meeting*, Feb. 2017, AAS Paper 17-418. Available: http://hdl.handle.net/10945/51983

[86] J. A. Starek, E. Schmerling, G. D. Maher, D. W. barbee, and M. Pavone, "Fast, safe, propellant-efficient spacecraft motion planning under clohessy-wiltshire-hill dynamics," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 418–438, 2017.

[87] P. Lu and X. Liu, "Autonomous trajectory planning for rendezvous and proximity operations by conic optimization," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 2, pp. 375–389, March–April 2013.

[88] I. M. Ross, *A Primer on Pontryagin's Principle in Optimal Control*, 2nd ed. Carmel,CA: Collegiate Publishers, 2015, pp. 85–169.

[89] M. A. Patterson and A. V. Rao, "$\mathbb{GPOPS} - \mathbb{II}$: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming," *ACM Trans. Math. Softw.*, vol. 41, no. 1, pp. 1:1–1:37, Oct. 2014. Available: http://doi.acm.org/10.1145/2558904

[90] J. E. Slotine and W. Li, *Applied Nonlinear Control*. Upper Saddle River, NJ: Prentice-Hall, 1991, pp. 40–96.

[91] N. Telsla, "Talking with the planets," in *Collier's Weekly*, vol. XXVI, Feb. 9 1901, no. 19, pp. 4–5.

[92] T. Rybus and K. Seweryn, "Planar air-bearing microgravity simulators: Review of applications, existing solutions and design parameters," *Acta Astronautica*, vol. 120, pp. 239 – 259, 2016. Available: http://www.sciencedirect.com/science/article/pii/S0094576515004634

[93] D. Gallardo, R. Bevilacqua, and R. E. Rasmussen, "Advances on a 6 degrees of freedom testbed for autonomous satellites operations," in *AIAA Guidance, Dynamics and Control Conference*, 2011.

[94] K. Saulnier, D. Pérez, R. Huang, D. Gallardo, G. Tilton, and R. Bevilacqua, "A six-degree-of-freedom hardware-in-the-loop simulator for small spacecraft," *Acta Astronautica*, vol. 105, no. 2, pp. 444–462, 2014.

[95] M. Wilde, B. Kaplinger, T. Go, H. Gutierrez, and D. Kirk, "ORION: A simulation environment for spacecraft formation flight, capture, and orbital robotics," in *2016 IEEE Aerospace Conference*, March 2016, pp. 1–14.

[96] P. Tsiotras, "ASTROS: A 5DOF experimental facility for research in space proximity operations," in *37th AAS Guidance and Control Conference, Breckenridge*, Breckenridge, CO, 2014, vol. 104, paper AAS 14-114. Available: http://hdl.handle.net/1853/53259

[97] M. Sabatini, M. Farnocchia, and G. B. Palmerini, "Design and tests of a frictionless 2D platform for studying space navigation and control subsystems," in *2012 IEEE Aerospace Conference*, March 2012, pp. 1–12.

[98] J. S. Hall and M. Romano, *Laboratory experimentation of guidance and control of spacecraft during on-orbit proximity maneuvers*. INTECH Open Access Publisher, Mar.01 2010, ch. 11, pp. 187–222. Available: http://hdl.handle.net/10945/50869

[99] J. Hall and M. Romano, "Novel robotic spacecraft simulator with mini-control moment gyroscopes and rotating thrusters," in *2007 IEEE/ASME international conference on advanced intelligent mechatronics*, Sept 2007, pp. 1–6.

[100] C. Lugini and M. Romano, "A ballistic-pendulum test stand to characterize small cold-gas thruster nozzles," *Acta Astronautica*, vol. 64, no. 5–6, pp. 615–625, 2009. Available: http://www.sciencedirect.com/science/article/pii/S0094576508003603

[101] R. W. Berger, D. Bayles, R. Brown, S. Doyle, A. Kazemzadeh, K. Knowles, D. Moser, J. Rodgers, B. Saari, D. Stanley, and B. Grant, "The RAD750 - a radiation hardened powerPC processor for high performance spaceborne applications," in *2001 IEEE Aerospace Conference Proceedings (Cat. No.01TH8542)*, 10–17 March 2001, vol. 5, pp. 2263–2272 vol.5.

[102] *Proton200k DSP processor board datasheet*, Space Micro Inc., Sand Diego, CA, May 9 2014. Available: http://www.spacemicro.com/assets/datasheets/digital/slices/proton200k-dsp.pdf

[103] *Proton200k Lite processor board datasheet*, Space Micro Incorporated, Apr. 9 2015. Available: http://www.spacemicro.com/assets/datasheets/digital/slices/proton200k-L.pdf

[104] J. Kolmas, P. Banazadeh, A. W. Koenig, B. Macintosh, and S. D'Amico, *System design of a miniaturized distributed occulter/telescope for direct imaging of star vicinity*, March 2016, pp. 1–11.

[105] *NVIDIA Tegra K1 series processors with Kepler mobile GPU for embedded applications data sheet*, NVIDIA, Santa Clara, CA, May 2016, version 04.

[106] Intel Core i7-4710MQ Processor. Intel Corporation. [Online]. Available: http://ark.intel.com/products/78931/Intel-Core-i7-4710MQ-Processor-6M-Cache-up-to-3_50-GHz. [Retrieved 22 December 2016].

[107] Intel Atom Processor Z530. Intel Corporation. [Online]. Available: http://ark.intel.com/products/35463/Intel-Atom-Processor-Z530-512K-Cache-1_60-GHz-533-MHz-FSB. (Retrieved 22 December 2016).

[108] R. Bevilacqua, J. S. Hall, J. Horning, and M. Romano, "Ad hoc wireless networking and shared computation for autonomous multirobot systems," *Journal of Aerospace Computing, Information, and Communication*, vol. 6, no. 5, pp. 328–353, 2009. Available: http://dx.doi.org/10.2514/1.40734

[109] S. Arthur, C. Emde, and N. Mc Guire, "Assessment of the realtime preemption patches (rt-preempt) and their impact on the general purpose performance of the system," *Proceedings of the 9th Real-Time Linux Workshop*, 2007.

[110] F. Cerqueira and B. B. Brandenburg, "A comparison of scheduling latency in linux, preempt_rt, and LITMUS^{RT}," *9th Annual Workshop on Operating Systems Platforms for Embedded Real-Time Applications*, pp. 19–29, July 2013.

[111] S. Godard. (2016). `sysstat`-System performance tools for the Linux operating system. GitHub Repository. [Online Repository]. Available: https://github.com/sysstat/sysstat. (Retrieved 29 April 2016).

[112] M. Sojka, P. Píša, R. Lisový, L. Waszniowski, and L. Hamáček. (2016, Oct.14). Linux Target for Simulink Embedded Coder. [Online Repository]. Available: http://lintarget.sourceforge.net. (Retrieved 21 March 2017).

[113] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: Theory algorithms and software*. New York, NY: Wiley, 2004, ch. 1.4, 5–7.

[114] D. Simon, *Optimal state estimation: Kalman, H-infinity, and nonlinear approaches*. New York, NY: Wiley, 2006, ch. 4–7.

[115] H. Sorenson and J. Sacks, "Recursive fading memory filtering," *Information Sciences*, vol. 3, no. 2, pp. 101 – 119, 1971. Available: http://www.sciencedirect.com/science/article/pii/S0020025571800014

[116] T. G. C. Williams and J. Kacur. (2016). `Cyclictest` . [Online Database]. Available: https://rt.wiki.kernel.org/index.php/Cyclictest. (Retrieved 29 April 2016).

[117] "IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros," *IEEE Std 952-1997*, pp. 1–84, Feb 1998.

[118] L. C. Ng and D. J. Pines, "Characterization of ring laser gyro performance using the Allan variance method," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 1, pp. 211–214, 1997.

[119] H. Sira-Ramírez, *Sliding Mode Control: The Delta-Sigma Modulation Approach*. Switzerland: Springer International Publishing, 2015, ch. 3, pp. 89–125.

[120] F. Bernelli-Zazzera, P. Mantegazza, and V. Nurzia, "Multi-pulse-width modulated control of linear systems," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 1, pp. 64–70, 1998.

[121] F. Curti, M. Romano, and R. Bevilacqua, "Lyapunov-based thrusters' selection for spacecraft control: Anlaysis and experimentation," *Journal of Guidance, Control, and Dyanmics*, vol. 33, no. 4, pp. 1143–1160, 2010.

[122] G. Song and B. N. Agrawal, "Vibration suppression of flexible spacecraft during attitude control," *Acta Astronautica*, vol. 49, no. 2, pp. 73 – 83, 2001. Available: http://www.sciencedirect.com/science/article/pii/S0094576500001636

[123] S. R. Norsworthy, R. Schreier, and G. C. Temes, Eds., *Quantization Noise in $\Delta\Sigma$ A/D Converters*. Hoboken, NJ: Wiley, 1997, ch. Introduction,1,2.

[124] E. Janssen and A. van Roermund, *Basics of Sigma-Delta Modulation*. Dordrecht: Springer Netherlands, 2011, pp. 5–28. Available: http://dx.doi.org/10.1007/978-94-007-1387-1_2

[125] J. M. de la Rosa, "Sigma-delta modulators: Tutorial overview, design guide, and state-of-the-art survey," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 1, pp. 1–21, Jan 2011.

[126] P. M. Aziz, H. V. Sorensen, and J. vn der Spiegel, "An overview of sigma-delta converters," *IEEE Signal Processing Magazine*, vol. 13, no. 1, pp. 61–84, Jan 1996.

[127] D. B. DeBra. (1990). 16.338: Pulse Modulators. [Online]. Available: https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-30-estimation-and-control-of-aerospace-systems-spring-2004/lecture-notes/intro_nnl.pdf

[128] S. W. Jeon and S. Jung, "Hardware-in-the-loop simulation for the reaction control system using PWM-based limit cycle analysis," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 2, pp. 538–545, March 2012.

[129] N. S. Nise, *Control Systems Engineering*, 5th ed. Hoboken, NJ: Wiley, 2008, ch. 4, 13.

[130] J. Sun, *Pulse-Width Modulation*. London: Springer London, 2012, pp. 25–61. Available: http://dx.doi.org/10.1007/978-1-4471-2885-4_2

[131] M. Planck, "The meaning and limits of exact science," *Science*, vol. 110, no. 2857, pp. 319–327, 1949.

[132] P. E. Black, K. Scarfone, and M. Souppaya, "Cyber security metric and measures," in *Wiley Handbook of Science and Technology for Homeland Security*, J. G. Voeller, Ed., 2009. Available: https://www.nist.gov/publications/cyber-security-metrics-and-measures?pub_id=51292

[133] B. A. Steinfeldt, R. D. Braun, and S. C. Paschall II, "Guidance and control algorithm robustness baseline indexing," *AIAA Guidance, Navigation, and Control Conference*, Aug.2–5 2010, aIAA Paper 2010-8787.

[134] C.-H. Hsu, J. A. Kuehn, and S. W. Poole, "Towards efficient supercomputing: Searching for the right efficiency metric," in *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering* (ICPE '12). New York, NY, USA: ACM, 2012, pp. 157–162. Available: http://doi.acm.org/10.1145/2188286.2188309

[135] C.-H. Hsu, W.-c. Feng, and J. S. Archuleta, "Towards efficient supercomputing: A quest for the right metric," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) - Workshop 11 - Volume 12* (IPDPS '05). Washington, DC, USA: IEEE Computer Society, 2005, pp. 230.1–. Available: http://dx.doi.org/10.1109/IPDPS.2005.440

[136] C. H. Hsu and W. C. Feng, "The right metric for efficient supercomputing: A ten-year retrospective," in *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, May 2016, pp. 1090–1093.

[137] R. Yurkovich, "The use of taguchi techniques with the ASTROS code for optimum wing structural design," in *AIAA/ASME/ASCE/AHS/ASC 35th Structures, Structural Dynamics, and Materials Conference*. Washington, DC: AIAA, 1994, paper AIAA-94-1484-CP, pp. 1334–1342.

[138] R. K. Roy, *A Primer on the Taguchi Method*. New York, NY: Van Nostrand, 1990, pp. 1–5,7–10,29–31,40–50,224,243–244.

[139] P. J. Ross, *Taguchi Techniques for Quality Engineering: Loss Function, Orthogonal Experiments, Parameter and Tolerance Design*. McGraw-Hill, 1998.

[140] D. N. Mavris, O. Bandte, and D. A. DeLaurentis, “Robust design simulation: A probabilistic approach to multidisciplinary design,” *Journal of Aircraft*, vol. 36, no. 1, pp. 298–307, 1999.

[141] Orbital Express (ASTRO/NextSat). (2013, Jan. 15). Jane’s Space Systems and Industry. [Online Database]. Available: https://janes.ihs.com/Space/Display/1384658

[142] Automated Transfer Vehicle (ATV). (2016, June 10). Jane’s Space Systems and Industry. [Online Database]. Available: https://janes.ihs.com/Space/Display/1384687

[143] Cluster. (2016, Dec. 29). Jane’s Space Systems and Industry. [Online Database]. Available: https://janes.ihs.com/Space/Display/1383046

[144] Cygnus. (2016, Feb. 08). Jane’s Space Systems and Industry. [Online Database]. Available: http://janes.ihs.com/Janes/Display/1592754

[145] Hayabusa (MUSES-C)/Hayabusa 2. (2015, Dec. 22). Jane’s Space Systems and Industry. [Online Database]. Available: https://janes.ihs.com/Space/Display/1384610

[146] HTV2 (KOUNTORI 4) Mission Press Kit. (2011, Jan. 20). Japan Aerospace Exploration Agency. [Online]. Available: http://iss.jaxa.jp/en/htv/mission/htv-2/library/presskit/htv2_presskit_en.pdf. Revision A.

[147] HTV4 (KOUNTORI 4) Mission Press Kit. (2013, Aug. 02). Japan Aerospace Exploration Agency. [Online]. Available: http://iss.jaxa.jp/en/htv/mission/htv-4/presskit/htv4_presskit_a.pdf. Revision A.

[148] Magnetospheric Multiscale. (2015, Apr. 02). Jane’s Space Systems and Industry. [Online Database]. Available: https://janes.ihs.com/Space/Display/1739774

[149] Progress series. (2015, May 22). Jane’s Space Systems and Industry. [Online Database]. Available: https://janes.ihs.com/Space/Display/1383292

[150] Rosetta. (2015, Feb. 02). Jane’s Space Systems and Industry. [Online Database]. Available: https://janes.ihs.com/Space/Display/1384597

[151] M. H. Tayarani-N., X. Yao, and H. Xu, “Meta-heuristic algorithms in car engine design: A literature survey,” *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 609–629, Oct 2015.

[152] A. A. Menezes and I. V. Kolmanovsky, “Energy and power management in a series hybrid electric vehicle using selective evolutionary generation,” in *53rd IEEE Conference on Decision and Control*, Dec 2014, pp. 3310–3315.

[153] X. Wei, L. Guzzella, V. I. Utkin, and G. Rizzoni, "Model-based fuel optimal control of hybrid electric vehicle using variable structure control systems," *Journal of Dynamic Systems, Measurement, and Contorl*, vol. 129, pp. 13–19, Jan. 2007.

[154] J. Jeong, D. Lee, N. Kim, Y. i. Park, and S. W. Cha, "Fuel economy analysis of a parallel hybrid bus using the optimal control theory," in *2011 IEEE Vehicle Power and Propulsion Conference*, Sept 2011, pp. 1–5.

[155] A. Sciarretta, M. Back, and L. Guzzella, "Optimal control of parallel hybrid electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 3, pp. 352–363, May 2004.

[156] Office of Transportation and Air Quality, "Light-duty automotive technology, carbon dioxide emissions, and fuel economy trends: 1975 through 2016," U.S. Environmental Protection Agency, Ann Arbor, MI, Tech. Rep. EPA-420-R-16-010, Appendix D, Nov. 2016.

[157] G. A. Boyarko, M. Romano, and O. A. Yakimenko, "Time-optimal reorientation of a spacecraft using an inverse dynamics optimization method," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 1197–1208, 2011.

[158] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006. Available: http://dx.doi.org/10.1007/s10107-004-0559-y

[159] M. Brand, V. Shilpiekandula, C. Yao, S. A. Bortoff, T. Nishiyama, S. Yoshikawa, and T. Iwasaki, "A parallel quadratic programming algorithm for model predictive control," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1031 – 1039, 2011. Available: http://www.sciencedirect.com/science/article/pii/S1474667016437444

[160] S. M. LaValle. (1998). Rapidly-Exploring Random Trees: A New Tool for Path Planning. Department of Computer Science, Iowa State University. Ames,IA. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.1853

[161] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 116–129, 2002.

[162] S. Quinlan, "Efficient distance computation between non-convex objects," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, May 1994, pp. 3324–3329 vol.4.

[163] B. Mirtich, “V-Clip: Fast and robust polyhedral collision detection,” *ACM Trans. Graph.*, vol. 17, no. 3, pp. 177–208, July 1998. Available: http://doi.acm.org/10.1145/285857.285860

[164] P. Jiménez, F. Thomas, and C. Torras, “3D collision detection: a survey,” *Computers & Graphics*, vol. 25, no. 2, pp. 269 – 285, 2001. Available: http://www.sciencedirect.com/science/article/pii/S0097849300001308

[165] N. K. Govindaraju, S. Redon, M. C. Lin, and D. Manocha, “CULLIDE: Interactive collision detection between complex models in large environments using graphics hardware,” in *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware* (HWWS ’03). Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2003, pp. 25–32. Available: http://dl.acm.org/citation.cfm?id=844174.844178

[166] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011. Available: http://dx.doi.org/10.1177/0278364911406761

[167] S. Karaman and E. Frazzoli, “Optimal kinodynamic motion planning using incremental sampling-based methods,” in *49th IEEE Conference on Decision and Control (CDC)*, Dec 2010, pp. 7681–7687.

[168] A. Short, Z. Pan, N. Larkin, and S. van Duin, “Recent progress on sampling based dynamic motion planning algorithms,” in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2016, pp. 1305–1311.

[169] Y. Kuwata, J. Teo, S. Karaman, G. Fiore, E. Frazzoli, and J. P. How, “Motion planning in complex environments using closed-loop prediction,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Aug. 18–21 2008, AIAA Paper 2008-7166.

[170] V. C. Poor, *Electricity and Magnetism: The Mathematical Theory*. New York,NY: Wiley, 1931, pp. 15–18.

[171] D. Ferguson, N. Kalra, and A. Stentz, “Replanning with RRTs,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, May 2006, pp. 1243–1248.

[172] V. G. Rao and D. S. Bernstein, “Naive control of the double integrator,” *IEEE Control Systems*, vol. 21, no. 5, pp. 86–97, Oct 2001.

[173] D. S. Bernstein, “A student’s guide to research,” *IEEE Control Systems*, vol. 19, no. 1, pp. 102–108, Feb 1999.

[174] K. Ogata, *Modern Control Engineering*, 1st ed. Englewood Cliffs, NJ: Prentice-Hall, 1970, pp. 69–82, 93–117, 430—440.

[175] M. D. Shuster, "Advice to young researchers," *IEEE Control Systems*, vol. 28, no. 5, pp. 113–148, October 2008.

[176] C. M. Zierhofer, "Adaptive sigma-delta modulation with one-bit quantization," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 5, pp. 408–415, May 2000.

[177] D. Dueri, B. Açıkmeşe, D. P. Scharf, and M. W. Harris, "Customized real-time interior-point methods for onboard powered-descent guidance," *Journal of Guidance, Control, and Dynamics*, pp. 197–212, 2016.

[178] M. Ciarcià and M. Romano, "Spacecraft proximity maneuver guidance based on inverse dynamic and sequential gradient-restoration algorithm," in *AAS/AIAA Astrodynamics Specialist Conference*, 2011, pp. 3679–3697.

[179] M. Ciarcià and M. Romano, "Suboptimal guidance for orbital proximity maneuver with path constraints capability," *AIAA/AAS Guidance, Navigation, and Control Conference*, Aug.13–16 2012, AIAA 2012-4920.

[180] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: A survey," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 103–123, Jan 1998.

[181] C. Zang, M. Friswell, and J. Mottershead, "A review of robust optimal design and its application in dynamics," *Computers & Structures*, vol. 83, no. 4–5, pp. 315 – 326, 2005. Available: http://www.sciencedirect.com/science/article/pii/S0045794904004079

[182] J. Virgili–Llop, R. Zappulla II, and M. Romano, "Dynamically tilting flat table to impart a time-varying gravity-induced acceleration on a floating spacecraft simulator," U.S. Patent Pending 62 455 775, Feb. 07, 2017.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California