Path Planning and Position Control and of an Underactued Electromagnetic Formation Flight Satellite System in the Near Field

by

Alexander James Buck

B.S., United States Naval Academy (2011)

Submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Author ....
Department of Aeronautics and Astronautics
May 23, 2013

Certified by ....
David W. Miller
Professor
Thesis Supervisor

Certified by ....
Alvar Saenz-Otero
Principal Research Scientist
Thesis Supervisor

Accepted by ....
Eytan H. Modiano
Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Path Planning and Position Control and of an Underactuated Electromagnetic Formation Flight Satellite System in the Near Field

by

Alexander James Buck

Submitted to the Department of Aeronautics and Astronautics
on May 23, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Electromagnetic formation flight is the process of using electromagnetic actuators (coils) on multiple spacecraft to produce relative (internal) forces in order to control the relative position and orientation of the spacecraft. This thesis demonstrates the ability to experimentally generate the relative internal electromagnetic forces in a short duration full 6DOF environment. Next the thesis limits itself to a two-satellite system and thus is able to perform a state reduction that constrains the motion to an arbitrary two-dimensional plane in 3-dimensional space showing that this is not actually a constraint on the real system for a two satellite formation. A feedback control law is propsed and simulated in this constrained space demonstrating position control of the underactuated system. Some theoretical guarantees are derived from contraction analysis. Finally time and energy optimal paths for a series of maneuvers are conceived by application of the $\mathbb{GPOPS} - \mathbb{II}$ numerical optimzation software. The results show further that the underactuated system is capable of arbitrary position control with the limitation being that it is unable to simultaneously control attitude and position to desired states because the attitude is used to "steer" the magnetic dipole therefore the desired angle is set by the position controller rather than an external reference. Overall this thesis shows the viability from the controllability perspective of underactuated electromagnetic formation flight for future space missions.

Thesis Supervisor: David W. Miller
Title: Professor

Thesis Supervisor: Alvar Saenz-Otero
Title: Principal Research Scientist

Acknowledgments

There is a long list of people who helped in getting me this far, both in the creation of this Thesis and in the creation of the RINGS system with which it is so tightly coupled. I must thank

- My advisors, Prof. David W. Miller and Alvar Saenz-Otero
- Everyone on the SPHERES team (too many to name)
- Prof. Ray Sedwick, Allison Porter, and Dustin Alinger at UMD for providing me with the opportunity to work on their RINGS project.
- The team at Aurora Flight Sciences who helped make RINGS a success: John, Joanne, Roedolph and Charlie

In particular there are two people I would like to thank

- Jacob Katz, an exceptionally brillia man with whom I have had the pleasure of working with for countless hours on the SPHERES Simulation and SpheresCore. Without his own monumental effort on those fronts I would not be where I am today.
- Greg Eslinger, my partner in crime for the RINGS project. For making the best of a good situation.

And Finally

A Dedication

To my loving wife, Jennifer, who gives me inspiration and who lights up my day when the days grow dim. Without your unending support and help I would never have gotten so far.

Contents

List of Figures

Nomenclature and Symbols

Acronyms & Abbreviations

GPOPS	General Pseudospectral Optimal Control Solver
n.s.d.	Negative Definite
n.s.d.	Negative Semi-definite
p.d.	Postive Definite
p.s.d.	Positive Semi-definite
BVP	Boundary Value Problem
CoM	Center of Mass
EMFF	Electromagnetic Formation Flight
WPT	Wireless Power Transfer
DOF	Degrees of Freedom
HE	Hall Effect
IMU	Inertial Measurement Unit
ISS	International Space Station
KKT	Karush-Kuhn-Tucker optimality conditions
NASA	National Aeronautics and Space Administration
NLOCP	Non-Linear Optimal Control Problem
NLP	Nonlinear Program
RGA	Reduced Gravity Aircraft

RGO	Reduced Gravity Office
SG	Savitsky-Golay
US	Ultrasound

Variables

$\boldsymbol{\tau}$	Torque
$\boldsymbol{d\ell}$	Differential Length Element
$\boldsymbol{F}$	Force
I	Electrical current
ϕ	Phase angle for time varying signal
t	Independent time variable
μ	Magnetic dipole vector
μ_0	Permability of free space or magnetic constant
$\boldsymbol{r}$	General position vector
$\boldsymbol{r}_{ij}$	Relative position vector of i w.r.t. j
λ	Scalar multiplier
θ	Direction of desired force, relative to inertial coordinate frame
$\mathcal{H}$	Hamiltonian
$\boldsymbol{\lambda}$	Lagrange multiplier vector
$\boldsymbol{\nu}$	Lagrange multiplier vector
J_a	Augmented Lagrangian
E	Meyer Cost (Endpoint Cost)
L	Lagrange Cost (Integral Cost)

Mathematical Syntax

The mathematical syntax used within this thesis aims to be highly consistent and clear. To that end the meaning of several formatting choices will be laid out in advance.

- x, X: regular italic variables denote *scalar quantities* (such as energy or mass)
- $\boldsymbol{x}$, $\boldsymbol{X}$: bold italic variables denote *euclidiean vector quantities* (2 or 3-dimensional vectors)
- $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{X}}$: the $\hat{}$ symbol denotes *unit vectors* (vector 2-norm equals 1)
- $\mathbf{x}$, $\mathbf{X}$: bold upright variables denote *matrix quantities* (MxN matricies, can be column or row matricies). Scalar values and euclidian vectors are allowable. As such this is the most general class of variables.

Capital (X) and lower case (x) letters are considered distinct variables but they still follow the above rules for determining what type of variable they are.

Variables that do not vary with time are referred to as *parameters.* All of the previous notation applies to parameters. Variables that do vary with time are still referred to as variables. As appropriate the following notation convetion applies.

- $\boldsymbol{x}(\cdot)$, $\mathbf{x}(\cdot)$: the $(\cdot)$ denotes the full time series of a variable, i.e. the *function* $\boldsymbol{x}(\cdot)$.
- $\boldsymbol{x}(t)$, $\mathbf{x}(t)$: the (t) denotes the value of $\boldsymbol{x}(\cdot)$ *at time t.*
- $\boldsymbol{x}^0$, $\boldsymbol{x}^f$: $(\cdot)^0$ and $(\cdot)^f$ denote fixed initial or final values of the $\boldsymbol{x}$.
- $\boldsymbol{x}(t_0)$, $\boldsymbol{x}_f$: (t_0) and $(\cdot)_f$ denote free initial or final values of the $\boldsymbol{x}$. Note: $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$
- $\tilde{\mathbf{x}}$: $\tilde{}$ denotes the error value of a particular variable, its value w.r.t. a reference $\mathbf{x}_r$.

Variables with the same time series can be grouped as $[\boldsymbol{x}(\cdot)\, \boldsymbol{u}(\cdot)]$ and done typically to describe a *trajectory,* or an association between the variables at any time step.

Subscripts $_i$ and $_j$ denote the ownership of the variable i.e. x_i denotes the value of x evaluated for object i. Subscript $_n$ denotes the n^{th} value of the variable, typically in reference to some discrete variable i.e. x_n is the n^{th} value of x.

Positive Definite	$(p.d.)$	$V(x) > 0 \forall x \neq 0$
Positive Semi-definite	$(p.s.d.)$	$V(x) \geq 0 \forall x \neq 0$
Negative Semi-definite	$(n.s.d.)$	$V(x) \leq 0 \forall x \neq 0$
Negative Definite	$(n.d.)$	$V(x) < 0 \forall x \neq 0$

Quick Function Properties Definitions

And for all of the above $V(0) = 0$.

Chapter 1

Introduction

To cross the seas, to traverse the roads, and to work machinery by galvanism, or rather electro-magnetism, will certainly, if executed, be the most noble achievement ever performed by man.
– Alfred Smee, *Elements of Electro-Metallurgy*, 1851

1.1 Motivation

The ability to control the relative position of multiple objects in orbit is very useful, and has been done so regularly since the Apollo era and the first orbital rendezvous. The ability to control the relative position of multiple objects *indefinitely, continually, and without refueling* is new and is an enabling ability for a variety of space applications.

1.1.1 Why Formation Flight

Fractionated Space Systems

There are shifting trends in space system design. The predominant single monolithic spacecraft design is losing ground to fractionated system architectures. Distributed space systems, an extension of satellite formations, have numerous advantages over large monolithic satellites. By spreading the functional components of a spacecraft system across several smaller discrete systems, either homogenously or heterogeneously, the overall system is more tolerant of risk and uncertainty. The system is also more flexible and robust that any single point designed spacecraft.

The Value Proposition for Fractionated Space Architectures, Brown and Eremenko 2006

Fractionated systems, as described by Brown[1], have 6 main aspects: networking, wireless communication, distributed computing, distributed payload operations, wireless power transfer, and cluster operations.

The final aspect is of particular importance, where before a satellite perhaps needed to control its attitude for payload operations, for instance, to point an antenna or telescope, under a fractionated architecture it now needs to potentially control the attitude of many spacecraft *and* control the relative positions of many spacecraft (i.e. the cluster). This has created a new requirement for methods of providing linear accelerations (ΔV). Conventionally this requirement is filled by thrusters. However, if the the ΔV requirement for the fractionated system is high enough, the propellant mass requirement can become infeasibly large, leading to either a descoping of the system capabilities and the associated reduction in ΔV requirement or the outright infeasiblity of the system overall.

The Terrestrial Planet Finder (TPF) : a NASA Origins Program to search for habitable planets, Beichman, Woolf, and Lindensmith 1999

One such concept that has a very high ΔV cost is the distributed telescope array[2]: a rotating array of telescopes create a spase telescope array which can image objects with a resolution similar to that of a monolithic aperture of the same length scale as the telescope array size. A rotating array requires constant centripetal accelerations. Relying on consumable propellant inheritly puts an upper limit on the useful life of the telescope array. Once the propellant stores are emptied, the array will cease rotating and drift apart.

Phoenix: Initial Technical Elements and Interfaces, Barnhart 2011

Formation Flying: The Future of Remote Sensing from Space, Leitner 2004

Other applications that possibly have high maneuver requirements are satellite inspection, repair, or salvage[3] and remote sensing spacecraft formation control[4]. At present these operations (if done at all) use thrusters to achieve their maneuvering requirements. Electromagnetic formation flight (EMFF) is an emerging technology that can satisfy the cluster operations aspect of fractionated and distributed space systems by enabling relative position control without propellant consumption.

1.1.2 Why Electromagnetic Formation Flight

Propellantless Formation Flight

Mass is at the center of every spacecraft. This is especially true for any vehicle designed to propel itself as the rocket equation is a powerful and unforgiving enemy. If spacecraft propulsion did not entail momentum transfer via particle exhaust then one could simply circumvent the rocket equation. This sounds like cheating but it is precisely what electromagnetic formation flight provides if the system requires relative position control. The electromagnetic propulsive forces are internal to the system so the overall system center of mass cannot be moved, only the relative position of spacecraft within a system can be modified.

No Plume Ejecta

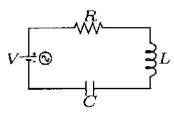
Further benefit of not using conventional thrusters to perform relative maneuvering is that there is no plume ejecta to interfere with sensors or payloads on the source spacecraft or other nearby spacecraft. This is of clear benefit to a distributed telescope system, whose prime payload is a host of sensitive optics. It is also a concern for other satellite systems as plume contamination can deposit exhaust onto other, more common, sensitive spacecraft components such as solar panels and thermal control surfaces and degrade system performance over time[5]. Eliminating the thruster eliminates this concern.

"Interactions Between Spacecraft and Thruster Plumes", Boyd and Ketsdever 2001

Wireless Power Transmission

Finally, and of note as this ability is another of the 6 aspects of fractionated space systems, a system capable of producing a strong, time varying magnetic field, as is used for EMFF, can use that same magnetic field generation system to transmit power wirelessly via resonant inductive coupling[6]. A simple way to generate a time varying magnetic field is to construct an RLC circuit and drive the system at the resonant frequency. This type of oscillating field can still be used for electromagnetic force interaction, though it creates a new requirement: controlling the relative phase between two or more signals. Assuming the frequency of the sinusoidal field is sufficiently fast relative to the rigid body motion, the

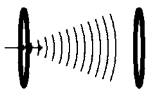
"Wireless Power Transfer via Strongly Coupled Magnetic Resonances", Kurs et al. 2007



The RLC circuit is an electrical circuit with resonant frequency $\frac{1}{\sqrt{LC}}$.

time averaged result is equivilant to a non-oscillating system driven at the RMS level of the sinusoidal system.

Therefore, while still satisfying the requirements for EMFF, if the field generation system is designed to create a time-varying magnetic field the system can perform wireless power transfer (WPT) as well. When actuating one satellite in the system, the transmittere, power can be transferred into the receiving coil of any number of nearby passive satellites via resonant inductive coupling. This could be a further enabling factor in fractionated or distributed space systems, allowing for power to be wirelessly transmitted within the formation.



Wireless power transmission via resonant inductive coupling is when the natural resonance of a secondary coil aligns with the driving frequency of the primary coil such that the induced voltages due to the time varying magnetic field oscillate at its own resonant frequency. The result is a much stronger output signal in the secondary.

1.2 Background

Research into EMFF systems has been conducted in the past several years. This research has been both about how to design an EMFF system and how to control one.

All prior work has only considered a fully actuated EMFF system (i.e. three orthogonal electromagnetic coils). In the fully actuated system the magnetic dipole is steerable independent of the spacecraft attitude. This thesis considers the most underactuated case, where there is only a single electromagnetic coil. In the underactuated system the dipole is no longer steerable independent of spacecraft attitude.

1.2.1 EMFF as a Concept

Elias studied the space interferometer in detail, with one possible solution being a set of free-flying satellites coupled through electromagnetic forces. By application of Kane's method Elias described the equations of motion and then demonstrated fully actuated closed loop controllability[7].

"Dynamics of Multi-Body Space Interferometers Including Reaction Wheel Gyroscopic Stiffening Effects: Structurally Connected and Electromagnetic Formation Flying Architectures", Elias 2004

Neave studied the EMFF and its many subsystems while also validating the model of a small single axis EMFF system operating on the ground to fit well with experimental data from a variety of control methods including linear feedback control and sliding mode control[8].

"Dynamics and Thermal Control of an Electromagnetic Formation Flight Testbed", Neave and Sedwick 2005

Kwon took the first in-depth look at EMFF. In his thesis Kwon laid out a baseline system design for an EMFF system, considering multi-satellite formations and formations with vehicles of disparate size[9]. An

"Electromagnetic Formation Flight of Satellite Arrays", Kwon and Miller 2005 example EMFF system for the Terrestrial Planet Finder mission is compared to various micro-thruster system designs. This work does not address the control algorithms for utilizing an EMFF system.

Schweighart demonstrated the controllability of an N-satellite formation of fully actuated satellites[10]. His thesis also develops three models for electromagnetic force and torque: far-field, mid-field and near-field. Schweighart provides analytic solutions for the far- and mid-field models. Finally this work provides methods for determining the necessary magnetic dipoles to achieve arbitrary set of forces for an N-satellite array, some consideration is given to angular-momentum management maneuvers.

"Electromagnetic Formation Flight Dipole Solution Planning", Schweighart 2005

Ahsun[11] presented two trajectory generation methods and two trajectory tracking methods for the N-satellite fully actuated formation flight problem. The trajectory generation methods are artificial potential function motion planning with dipole inversion and pseudospectral optimal control methods. The tracking algorithms demonstrated are an application of adaptive control and receding horizing optimal contol.

"Dynamics and Control of Electromagnetic Satellite Formations", Ahsun and Miller 2007

Wawrzazek showed how to apply linear control theory to the restricted case of a spun-up fully actuated two satellite formation[12]. He shows how the uncontrolled system, using only feed forward centripetal forces is unstable and that while the linear controller developed stabilizes the system, it does so insufficiently for interferometric mission requirements.

"Control and reconfiguration of satellite formations by electromagnetic forces", Wawrzazek and Banaszkiewicz 2007

1.2.2 Contraction Theory

Created

1.2.3 Optimal Control

Optimal control theory deals with the class of problems that ask to find the control law or history of control inputs that satisfy an optimality requirement, that is, they minimize some cost function, while subject to arbitrary state and input path and boundary constraints. There only exist analytic solutions for a very small set of problems, and most real-world problems require numerical methods.

The sufficient conditions for optimality provide a boundary value problem (BVP) that often is very difficult to solve numerically. Methods that attempt to find approximate solutions to the BVP are called indirect methods.

In recent years alternative methods for finding solutions to optimal control problems have been developed. In particular, and of recent popularity, are direct pseudospectral methods. Conceptually this class of solution method is finding the coeffients for a special set of polynomial basis functions that approximate the state and control histories and explicity enforcing the optimality criteria (Karush-Kuhn-Tucker, KKT[13]) at a set of points called collocation nodes.

"Nonlinear Programming", Kuhn and Tucker 1951

Pseudospectral methods have been applied to electromagnetic formation flight before[11] however prior work has been limited to the fully actuated case when the system has as many electromagnetic actuators as translational degrees of freedom.

"Dynamics and Control of Electromagnetic Satellite Formations", Ahsun and Miller 2007

1.3 RINGS EMFF Testbed

It will be useful for the reader to be familiar with the basic functionality and geometry of an EMFF system. To that end a brief overview of the RINGS EMFF system is presented.

1.3.1 Appearance

An electromagnetic formation flight system is at its core simply an electromagnet. To have a basic level of functionality it also requires independent attitude control, such as reaction wheels or thruster pairs. The magnetic dipole is a function of the area of the electromagnet, the number of turns or windings in the electromagnet and the amount of electrical current flowing in the wires of the electromagnetic coil, $\boldsymbol{\mu} = \hat{\boldsymbol{n}}NIA$. The direction of the dipole is determined by the vector $\hat{\boldsymbol{n}}$ normal to the plane defined by the coil of wire. To a basic level, it is thus desireable to have a large area, many turns and high current as stronger dipoles allow for stronger interaction and larger forces or torques. The geometry that maximizes the area for a given length of wire is a circle, thus when free of other restrictions an electromagnet used for EMFF should look

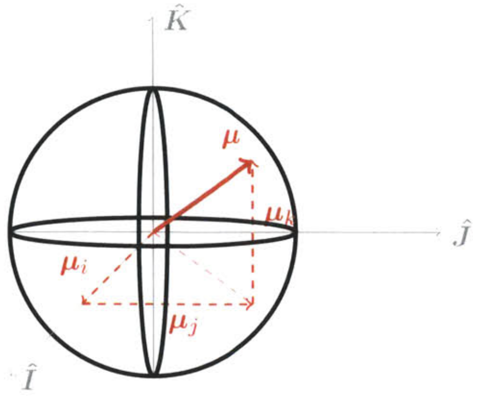


Figure 1-1: The fully actuated EMFF system has three coils of wire that produce magnetic dipoles $\hat{\boldsymbol{\mu}}_i$, $\hat{\boldsymbol{\mu}}_j$, and $\hat{\boldsymbol{\mu}}_k$. These unit vectors form a basis set that spans the space $\mathbb{R}^3$. Therefore any vector $\boldsymbol{\mu} \in \mathbb{R}^3$ can be created in component parts by this configuration.

something like a large bundle of wire coiled into a circular shape.

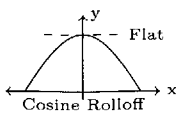
To be able to point the dipole in any direction, three coils are required whose normal vectors $\hat{\boldsymbol{n}}_i$ are linearly independent, that is the basis set $[\hat{\boldsymbol{n}}_i, \hat{\boldsymbol{n}}_j, \hat{\boldsymbol{n}}_k]$ spans the space of $\mathbb{R}^3$. The obvious choice if free from other restrictions is to orient the three coils in mutually orthogonal directions. Thus a fully actuated electromagnetic formation flight system might look something like Figure 1-1.

1.3.2 Electrical Operation

The system can operate using either direct or alternating current. In the case of direct current there will be a coupling with Earth's magnetic field (and any other static magnetic fields in the area) that must be dealt with. In the case of alternating current, both the phase and frequency of the signal must be matched across all units to be coupled. Units driving at a non-identical frequency ($f \neq g$) will net to zero average force.

$$\int_0^{2\pi N} \sin(2\pi f t)\sin(2\pi g t)dt = 0$$

Regardless of the frequencies there will be beat-frequency vibrations of $|f \pm g|$ Hz. For $f = g$ the beat frequencies shift to 0Hz and $2f$Hz. The "0Hz vibration" is the bulk motion of the satellites. Units driving with some non-zero phase angle between them ($\Delta\phi \neq 0$) will have a force



Cosine Rolloff
The cosine waveform is flat in the viscinity of 0.

response of $\boldsymbol{F}_{actual} = \boldsymbol{F}_{\Delta\phi=0}\cos(\Delta\phi)$. The force rolls off as the cosine of the phase error therefore very precise phase control is not important because the curve is flat ($\frac{\partial F}{\partial \Delta\phi} = 0$) around zero phase angle difference which means. However lack of phase control altogether results in the system drifting randomly in phase relative to each other, with zero average net force.

1.4 Overview

This thesis discusses the prior electromagnetic force models developed and reproduces the results in the near-field. A limit is found to how discritized the model needs to be before losing precision as a function of range. Then experimental data is collected on a Reduced Gravity Aircraft (RGA) with the goal of validating the electromagnetic force models. Then the underactuated system is examined in a linear control sense and feedback position control is achieved. Finally optimal control theory is used to generate time or energy optimal paths for more complex maneuvers.

In Chapter 2 the electromagnetic force and torque near-field model are described. The far-field model is stated for comparison, having been thoroughly developed in prior work. A limit is found for the discretization level in the near-field model such that machine precision is reached. Also common geometries are introduced alongside small glyphs that may appear in parts of the remainder of the text.

Chapter 3 attempts to verify and validate the near-field model. The model is compared to far-field for verification, with the expectation of the two converging at roughly 6.67 radii. Validation is conducted experimentally with data collected from step responses and the response is compared to near-field model. Difficulties with data collection and trajectory reconstruction have hampered this effort. The process described should be sufficient with a more complete data set.

Chapter 4 takes an underactauted system with only a single electromagnetic actuator and applies Lyapunov stability analysis and convergence theory to produce a dipole-steering feedback control law. This control law is then simulated and evaluated within the near-field region.

This method proves the ability of a single actuator system to achieve position control, though it cannot simultaneously achieve arbitrary attitude control due to the nature of the position control.

Chapter 5 uses the same underactuated system from Chapter 4 and applies optimal control theory along with a numerical optimzation tool, $\mathbb{GPOPS} - \mathbb{II}$, to find time and energy optimal trajectories for a series of maneuvers. The maneuvers tested are: axial step, lateral step (slew), a collision-avoidance maneuver, and a formation spin-up maneuver. These maneuvers are selected for their gradual increase in complexity (in order to assist in developing tools for generating optimal trajectories) and for their application to real scenarios, the axial and lateral step test basic position control, the collision-avoidance demonstrates a crucial ability when operating a multi-satellite formation in close proximity and the formation spin-up is an essential manevuer for a system such as a sparese telescope array.

Chapter 2

Force & Torque Modelling

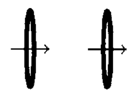
One of the foremost contributions this thesis makes is examining the underactuated EMFF system, that is, when there are fewer electromagnetic coils than translational degrees of freedom. The transfer function from electrical current (dipole strength) to electromagnetic force and torque is the same, but in the underactuated model only the strength of the dipole is controllable, while its orientation is defined by the spacecraft attitude vector. This means the control input is not the electromagnetic dipole vector, but rather is the intensity of the magnetic field and torque inputs to rotate the field direction. We begin by a careful examination of the approximate near field force model.

2.1 Electromagnetic Force Model Development

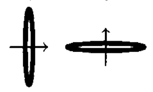
The force and torque interaction exploited by an EMFF system derive from the effect known as the Lorentz force. This is the interaction of moving charged particles and a magnetic field resulting in a force orthogonal to both the field and velocity vectors. When the charged particles are constrained to a conductor (a wire) this is sometimes known as the Laplace force, and when the external magnetic field is itself caused by electrical current in a wire (thus the interaction between two conductive wires) this is often known as Ampère's force law. The forces and torques created by an EMFF system all derive from this effect.

2.1.1 Common Geometries

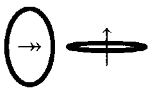
There are a few geometries that continually show up when talking about near-field electromagnetic forces/torques. Between these maneuvers almost any more complex maneuver can be composed so it will help to properly define them before using them. Often there will be small glyphs shown in the margin to make clear which geometry is relevant to the discussion at hand.



1. **Axial**: When the coil plane normal vectors are co-axial



2. **Shear**: When the coil plane normal vectors are orthogonal



3. **Skew**: When the coil plane normal vectors are *not* coplanar

The first two orientations span the range of *planar* motion: the axial case causes motion along the relative position vector (attraction or repulsion), the shear case causes transverse motion orthogonal to the relative position vector and a strong rotation around the planar normal axis. Importantly, any torques produced by these first two orientations do not cause the spacecraft dipole to leave the plane. That feature is what sets the planar orientations apart from the skew orientation.

2.2 Near-Field

Ampère's force law is expressed as the infinite summation of forces on infinitesmal lengths of conductor $\boldsymbol{d\ell}_i$ with current I_i due to the magnetic field generated by infinitesimal lengths of conductor $\boldsymbol{d\ell}_j$ with current I_j. It is assumed that the structure of each conductor is rigid meaning forces between lengths of conductor within the same object are ignored. When the conductors are circular in shape the result is as shown in Figure 2-1 on the next page. Extending this by noting that $\boldsymbol{\tau} = \boldsymbol{r} \times \boldsymbol{F}$ defines the near-field torque equation as well, where $\boldsymbol{R}_i$ is the position vector of element $\boldsymbol{d\ell}_i$ relative to the center of mass (CoM) of object i.

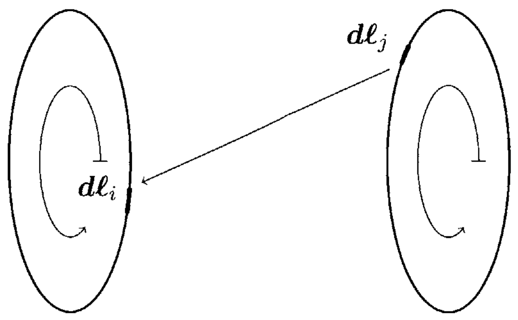


Figure 2-1: The Ampère Force Law computes the total force on a current carrying wire due to the magnetic field generated by electrical current in a nearby wire.

$$\boldsymbol{F}_{ij} = \frac{\mu_0}{4\pi} \int_i \int_j \frac{I_i d\boldsymbol{\ell}_i \times (I_j d\boldsymbol{\ell}_j \times \hat{\boldsymbol{r}}_{ji})}{\|\boldsymbol{r}\|^2} \tag{2.1a}$$

$$\boldsymbol{\tau}_{ij} = \frac{\mu_0}{4\pi} \int_i \int_j \boldsymbol{R}_i \times \frac{I_i d\boldsymbol{\ell}_i \times (I_j d\boldsymbol{\ell}_j \times \hat{\boldsymbol{r}}_{ji})}{\|\boldsymbol{r}\|^2} \tag{2.1b}$$

This integral has no known analytic solution. It is however useful to use this as the truth model for simulations, as it is a very accurate description of the physical system. In order to implement Equation (2.1) on the facing page numerically it must be discretized first.

$$\boldsymbol{F}_{ij} = \frac{\mu_0}{4\pi} \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} \frac{I_i d\boldsymbol{\ell}_i \times (I_j d\boldsymbol{\ell}_j \times \hat{\boldsymbol{r}}_{ji})}{\|\boldsymbol{r}\|^2} \tag{2.2a}$$

$$\boldsymbol{\tau}_{ij} = \frac{\mu_0}{4\pi} \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} \boldsymbol{R}_i \times \frac{I_i d\boldsymbol{\ell}_i \times (I_j d\boldsymbol{\ell}_j \times \hat{\boldsymbol{r}}_{ji})}{\|\boldsymbol{r}\|^2} \tag{2.2b}$$

There are two important parameters in this model, N_i and N_j. These parameters determine how discritized the problem is. Specifically they determine how many segments to approximate each circular coil as. Using more coil segments should result in a more accurate result, but it comes at the cost of increased computation.

There are three distinct levels of detail for electromagnetic force/-

torque models.

1. **Far Field**: Ignore geometry of conductor, consider as a point dipole, $\mathcal{O}(1)$
2. **Near Field**: Consider simple geometry of conductor, $\mathcal{O}(N_iN_j)$
3. **Nearest Field**: Consider full geometry of conductor, $\mathcal{O}(N_iN_jM_iM_j)$

The decision reduces to a trade of computational cost for numerical accuracy. The cost is $\mathcal{O}(N_iN_jM_iM_j)$ for N segments per turn and M loops per vehicle i, j. Therefore both the number of turns per coil and the number of segments per turn are very important to the computational intensity of the problem and should be carefully set to appropriate values.

2.2.1 Turns per coil reduction

Applying the second level requires integration of the full geometry. If the system consists of two circular coils with 10 turns each, that requires integration from $[0, 20\pi]$ on both integrals, which is a *100x increase* in computation cost! While this is the most precise way of computing the electromagnetic forces, it only becomes relevant in the extremely near-field. The first level allows for an equivilance in amp-turns which eliminates the quadratic increase in computational cost from adding turns. By assuming all turns occupy the same physical space the integration can again be over a single turn with a larger multiplier out front.

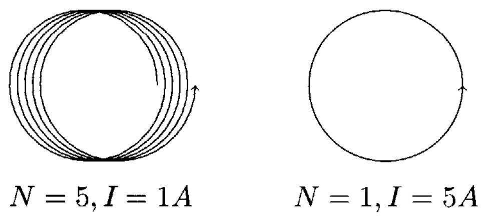


Figure 2-2: Amp-turn equivilance

2.2.2 Segments per turn convergence

Let us next look at how many of the infinitesemal differential length elements in the near field model we actually need to use. The near field model is very accurate and for that it is desirable to use, either as ground truth for a simulation or other analysis or as the process model used for real hardware. The primary reason to *not* use the near field model is the computational intensity of doing so. However, if it were possible to smoothly and dynamically switch between low cost computation at large ranges and higher cost but arbitrarily accurate results at very close ranges then this would perhaps be the best of both worlds. This proposed method offers precisely that best-of-both-worlds outcome.

Arbitrarily accurate in the range $\epsilon_{des} \in [\epsilon_m \infty)$. This thesis does not claim to be magical in any way, the physical limitations of the machine being used still apply.

Intelligent Discretization

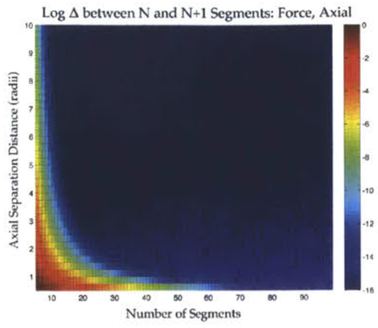
There are four steps involved in the process of creating an intelligent discretization strategy. The final step will vary depending on the application specific details of the hardware and software thus it will not be described in detail. The first three steps are more generalizable and thus the process through these steps is described.

1. Evaluate the magnitude, $M_F = ||\mathbf{F}||$ and $M_\tau = ||\boldsymbol{\tau}||$, of the near-field model electromagnetic force and torque over various distances normalized radial r and various coil discritization levels $N = N_i = N_j$.
2. Perform a first order central difference to determine the slope $\frac{\partial M}{\partial N}$
3. Approximate the levelcurve of $\frac{\partial M}{\partial N} = \epsilon_{des}$ with some function $N(r)$
4. Modify near-field model to use $N(r)$ segment discretization instead of a predified fixed number.

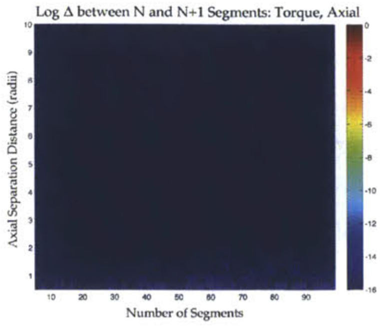
There is no consideration in this method for different discritzation levels N_i for each satellite. The method explicitly assumes $N_i = N_j$ etc... Note: While unique discretizations are dissallowed the coils may geometrically be unique. The size and shape of each coil is irrelevant as the final result is based only on the output of the model, i.e. magnitude of force and torque.

The first two steps were completed and the resulting graph of $\frac{\partial M}{\partial N}$ for both force and torque is show in Figure 2-3 on the next page for two different orientations (resulting in 4 charts). Looking at the four results we can see behavior that is dependent both on range and orientation. Across these dimensions the bounds imposed by force convergence (charts a,c) is stricter than those from torque convergence (b,d). Furthermore, the shear case imposes stricter bounds than the axial case (again, for both force and torque), therefore we can conservatively pick the number of coil segments based off the worst case scenario, which is the shear force orientation (c). With futher analysis the effect of orientation could be quantified to take advantage of looser bounds in the axial orientation.

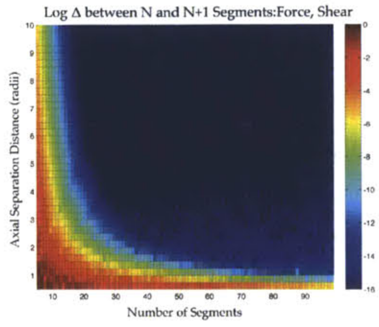
The first order central difference is defined as: $\left.\frac{dx}{dy}\right|_i \approx \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}}$



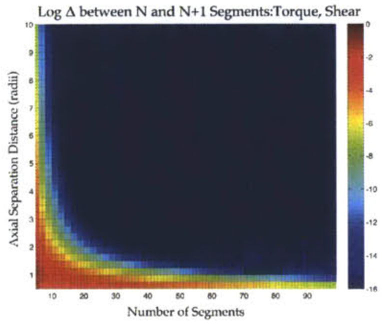
(a) Axial Force



(b) Axial Torque



(c) Shear Force



(d) Shear Torque

Figure 2-3: Change in vector norm with with increasing segments. Columns: Force vs Torque. Rows: Axial vs Shear.

The normalized range is the absolute range r_a divided by the coil radius R_{coil}. The coil radius provides a scale factor that helps discuss these electromagnetic systems in a more general fashion. Therefore the normalized range r is $\frac{r_a}{R_{coil}}$

The final step is to approximate the levelcurve $\frac{\partial M}{\partial N} = \epsilon_{des}$ as a function of the normalized range r, by an inverse quadratic polynomial

$$N(r) = \frac{1}{ar^2 + br + c} \tag{2.3}$$

where ϵ_{des} is a precision level from $+\infty$ to machine precision inclusive. The particular application may not require full machine precision and thus this method can be used to find the levelcurve that satisfies some different precision value.

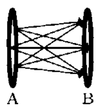
The main conclusion is that in practice for numerical computations there can be quantified a boundary beyond which there is sufficient discretization that the computation does not suffer loss of precision due to coil discretization. Once there are enough segments to reach the desired precision, using any more is wasted computation. The method to find this boundary is a numerical method and given the initial computation cost to find the desired precision level curve, this method is only advised if a large number of calculations are to be done such that the initial computational investment can be recouped subsequently by the more efficient discretization.

The remainder of this thesis leverages this result when computing near-field forces and torques. In this way, when in the far-field the computational cost of the near-field model is the same order of magnitude as the far-field model and when in the near-field the full force precision of the near-field model is readily available.

2.2.3 Parallelization

The near-field model was not parellelized but it is possible to peform massive parallelization of Equation (2.1) on page 26. The near field force equation is an all-to-all type problem where every segment on coil A must be evaluated against every segment on coil B. Thus the top level can be parallelized, evaluate every segment of coil A in parallel. Given that the number of segments is essentially always greater than the number of hardware cores available there should be considerable speed improvements seen proportional to the how parallelized the hardware is (i.e. how many computing cores are available).

All to all:



2.3 Far-Field

Prior work in EMFF has developed and used far field models[11,10]. Referencing this prior work we show the far-field electromagnetic force and torque models in Equation (2.4). These equations assume each object is a point magnetic dipole. All physical dimension of each object is ignored. When sufficiently far away this is a valid assumption.

"Dynamics and Control of Electromagnetic Satellite Formations", Ahsun and Miller 2007

"Electromagnetic Formation Flight Dipole Solution Planning", Schweighart 2005

$$\boldsymbol{F}_{ij} = 3C_m \frac{\mu_i \mu_j}{r_{ij}^4} \Big(5\left(\hat{\boldsymbol{\mu}}_i \cdot \hat{\boldsymbol{r}}_{ij}\right)\left(\hat{\boldsymbol{\mu}}_j \cdot \hat{\boldsymbol{r}}_{ij}\right)\hat{\boldsymbol{r}}_{ij} - \left(\hat{\boldsymbol{\mu}}_i \cdot \hat{\boldsymbol{\mu}}_j\right)\hat{\boldsymbol{r}}_{ij} - \left(\hat{\boldsymbol{\mu}}_i \cdot \hat{\boldsymbol{r}}_{ij}\right)\hat{\boldsymbol{\mu}}_j - \left(\hat{\boldsymbol{\mu}}_j \cdot \hat{\boldsymbol{r}}_{ij}\right)\hat{\boldsymbol{\mu}}_i \Big) \tag{2.4a}$$

$$\boldsymbol{\tau}_{ij} = C_m \frac{\mu_i \mu_j}{r_{ij}^3} \Big(\hat{\boldsymbol{\mu}}_i \times \left[3\hat{\boldsymbol{r}}_{ij}\left(\hat{\boldsymbol{\mu}}_j \cdot \hat{\boldsymbol{r}}_{ij}\right) - \hat{\boldsymbol{\mu}}_j \right] \Big) \tag{2.4b}$$

The expression $\frac{\mu_0}{4\pi}$ occurs in essentially all equations relating to electromagnetic force and torque. As such we give this expression a special symbol, C_m. The vector form of the far-field equations are useful and very concise way to describe the full 3-dimensionally relationship but they are not terribly informative or insightful analytically. Therefore we can also confine the dynamics to two dimensions (as is often the case) and show a much simpler form of Equation (2.4) on the preceding page.

$$\boldsymbol{F}_{ij} = 3C_m \frac{\mu_i \mu_j}{r_{ij}^4} \begin{bmatrix} 2\cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta) \\ -\cos(\alpha)\sin(\beta) - \sin(\alpha)\cos(\beta) \end{bmatrix} \tag{2.5a}$$

$$\boldsymbol{\tau}_{ij} = C_m \frac{\mu_i \mu_j}{r_{ij}^3} \left(2\sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta)\right) \tag{2.5b}$$

Note the subscript order for relative vectors is defined as $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$.

The angles α and β are the dipole angles of satellites i and j respectively. Note: the angles α and β are measured relative to the relative position vector $\boldsymbol{r}_{ij}$, not the inertial axes. Therefore there is a vector rotation required to express these forces in terms of an inertial X-Y plane.

Chapter 3

Verification & Validation

Essentially, all models are wrong, but some are useful
– George E. P. Box, 1987

The goal of verification and validation is to demonstrate two important facts:

1. The model was developed *correctly*
2. The model developed is *correct*

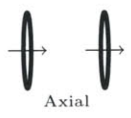
Once both have been established, the model becomes useful. Without validation the model may produced the results we intended (per the model algorithm) but have no relation to the reality it claims to model. Without verification a model may produce results consistent with reality but no conclusions based off its results can be drawn as the implemented model is not necessarily the intended theoretical model.

3.1 Verification

Verification is the process of assessing whether something was correctly built or developed. In the case of a model of a physical system a good way to do this is to be able to predict the behavior of the real system under known conditions.

To verify the near-field algorithm properly computes forces and torques we compare against a known and proven algorithm, the far-field model. We evaluate both our near-field algorithm and the far field model from literature over several ranges and orientations. For brevity we only present the axial force and shear torque cases as the most illustrative. The vector norm of the resultant force or torque vector is graphed for the given geometries.

Note, the value plotted in the Y-axis is Force or Torque per Amp^2. They are presented this way because in both the force models there is an I^2 multiplier. The force scales linearly with the product of current in both coils. In the far-field (Equation (2.4) on page 31) it is contained within $\mu_i\mu_j = (NIA)_i(NIA)_j$. In the near-field (Equation (2.1) on page 26) it is expressly in the integral cross product, $I_i \boldsymbol{d\ell}_i \times (I_j \boldsymbol{d\ell}_j \times \hat{\boldsymbol{r}}_{ji})$. In practice what this means is for two coils of wire the distribution of electrical current is unimportant, all that matters is the product I_iI_j.



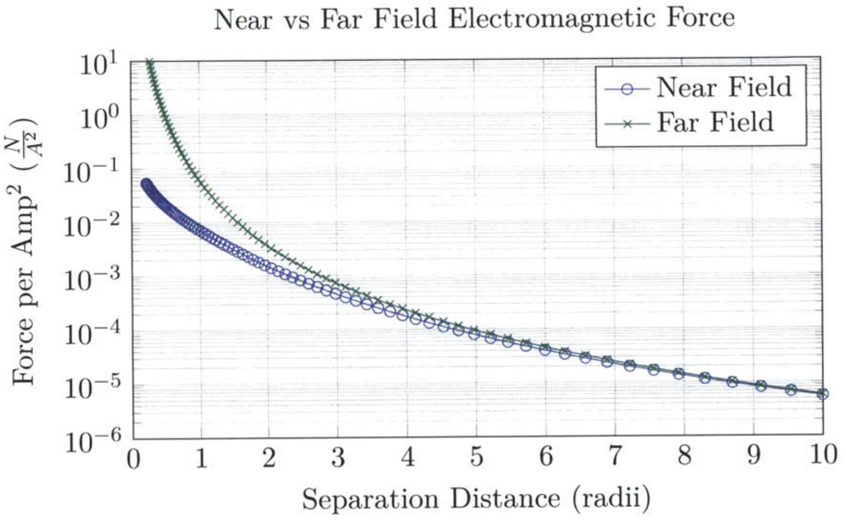
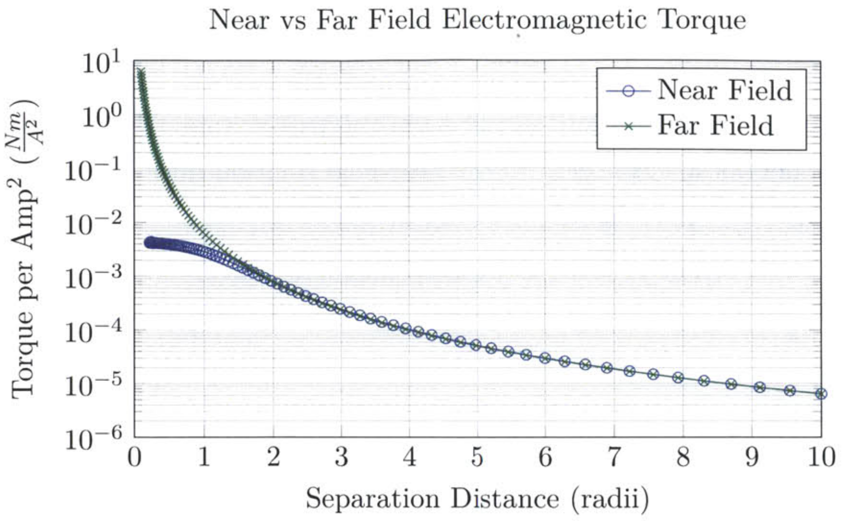


Figure 3-1: Near and far field forces for the axial dipole case

The near and far field models converge at around 6.67 radii, which is the expected range at which the two models converge. This visual comparison of the two models is useful, but the metric we are concerned about when deciding to use near-field or far-field is the relative error. This should converge to zero and be within roughly 10% at 7 radii. The dip in the relative torque error near 2 radii is because the near-field model predicts torques slightly larger than the far-field model at large separation distances. At precisely 2 radii (1 diameter) the near-field and far-field models are equal. Inside of 2 radii the near-field torques are lower than the far-field, which asymptotically goes to infinty due to the



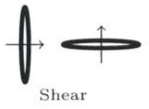
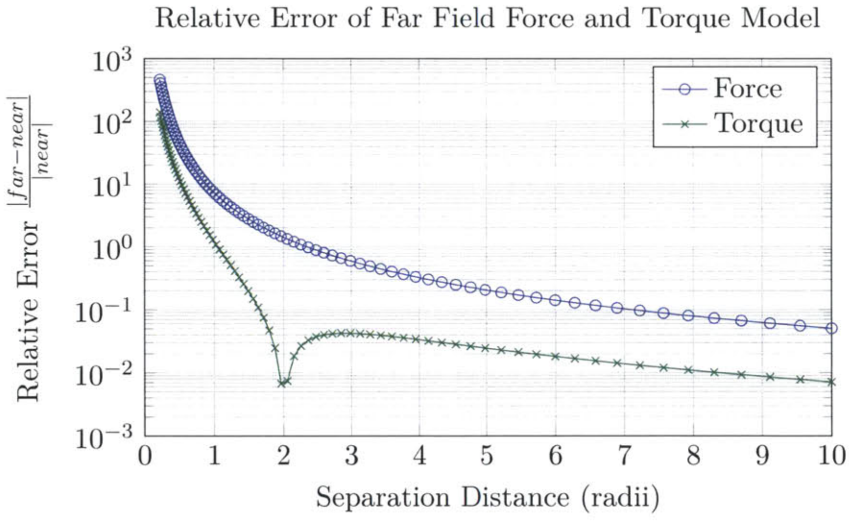


Figure 3-2: Near and far field torques for the shear dipole case



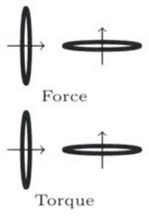
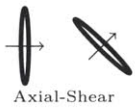


Figure 3-3: Relative error of the far field model in axial force and shear torque

$\frac{1}{r^3}$ term.

The relative force error gradually diverges as the range decreases. Just like as it did for torque, the far-field force model diverges to infinity at zero range due to the $\frac{1}{r^4}$ term.

It is worth presenting one more chart. From the view of Figure 3-2 it appears that the far-field torque model is accurate to within about 1.5 radii. This is simply coincidental. Due to the symmetry involved in the pure shear geometry the far-field model does a good job approximating this particular case. However, when we rotate one coil 45° rather than 90° this symmetry is broken and it is very apparent that the far-field torque model doesn't converge to the near-field solution until at least 7 radii, roughly the same length scale required for force.



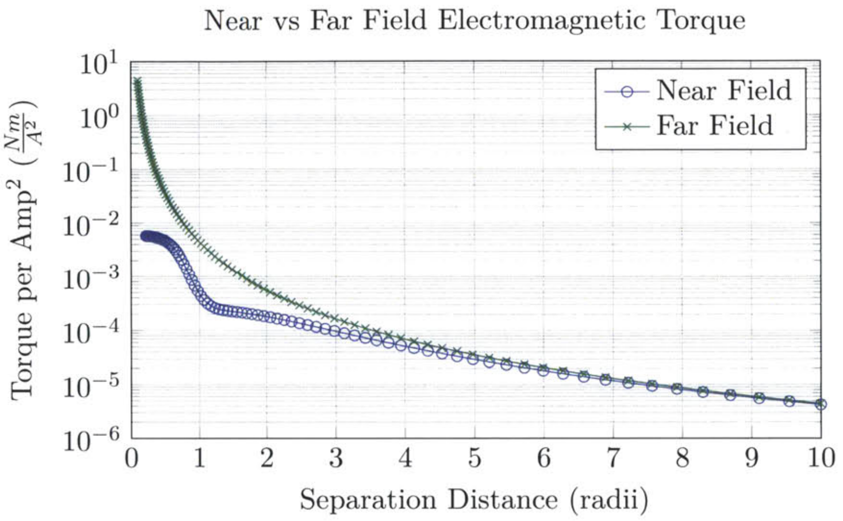


Figure 3-4: Near and far field torques for the 45° dipole case

3.2 Validation

To validate the model to the RINGS system hardware, we compare the force/torque model output to experimental data collected by operating the RINGS in a reduced gravity environment. The RINGS system will eventually operate in the permanent microgravity environment of the International Space Station (ISS)

For operating on the ground, the RINGS are placed into an air carraige which allows for 3 degrees of freedom (DOF), two translational and one rotational. This reduction in DOF's eliminates much of complexity in the dynamics, additionally the air-carriage introduces additional mass and surface friction that is not present in the full 6-DOF system (i.e. in microgravity). Short of actually sending the payload into orbit, one way to achieve an environment momentarily similar to microgravity is the Reduced Gravity Aircraft (RGA). This was the method used to achieve the short-duration microgravity environment needed for model V&V.

3.2.1 Data Collection: Zero G

The RGA is a Boeing 727-200 Series aircraft. The RGA is operated by the ZeroG corporation and flies out of Ellington Field, TX. The flight is provided by the NASA Flight Opportunities Program: Reduced Gravity Office (RGO).

The aircraft flies a series of parabolic arcs as shown in Figure 3-5. After each reduced gravity parabola (durations vary with gravity reduction), there is a 1.8g pull up maneuver that lasts approximately 40 seconds. The transitions between reduced gravity and 1.8g are approximately 2-4 seconds long.[14]. Each flight lasts 2 hours, during which the RGA flies 40 parabolas. Each flight consists of 4 sets of 10 parabolas with short 5 minute breaks between each set.

[14] *Interface Control Document: Boeing 727-200*, Lichtenberg 2009

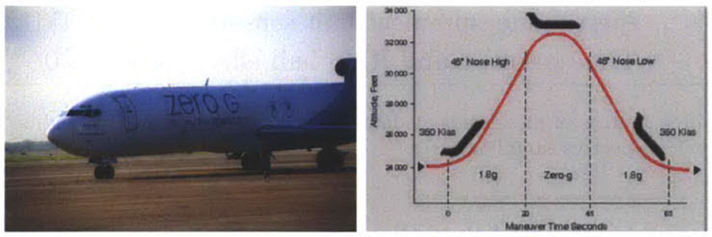


Figure 3-5: Zero G aircraft and parabola

The RGA can simulate Martian, Lunar and micro-gravity levels. The number of each type of reduced gravity parabolas on a flight is dependent on the manifest and payload requirements which varies week to week. Data collection for the EMFF model validation requires a microgravity (0g) environment. Each flight during the week of data collection flew 30 zero gravity parabolas and 10 lunar gravity parabolas.

The work area onboard the aircraft is similar in volume to that of the ISS working volume, as seen in Figure 3-6 on the next page. Using the supplied camera posts the SPHERES beacons were set-up in a known configuration. During each parabola up to three types of data are collected as seen in Table 3.1 on the following page.

Data is collected for an 8 second period during the 17 second microgravity parabola. This provides time before and after for deployment

Figure 3-6: Inside of the Zero G aircraft with the RINGS hardware operating

Abbrev.	Name	Rate
US	Ultrasound ranging measurements	5Hz
IMU	Inertial measurement unit sensors	1kHz
HE	Time resolved and RMS hall effect values	$30\ \frac{\text{samples}}{\text{cycle}}$

Table 3.1: Three sources of measurement data were collected in the RGA flights. They are listed with their respective sampling rates.

and stowage prior to the 1.8g pull up maneuver.

3.2.2 Data Processing

The method for validating the electromagnetic force/torque model involves measuring the trajectory of both spacecraft, recording linear and angular accelerations (IMU) and control inputs (HE) along the trajectory and finally computing the model predicted forces and torques based on the recorded relative trajectory and control inputs.

Model Predictions

"Dynamic Programming for Electromagnetic Spacecraft Actuation", Eslinger 2013

Eslinger has shown how to reconstruct the trajectories of both RINGS satellites based on ultrasound ranging data[15]. The important piece of information as far as the dynamics are concerned is the relative position and the inertial attitudes. Each trajectory is expressed inertially. To determine the relative position simply difference the two positions.

There were complications with the data collection during the tests such that the HE data from one or both satellites was not recorded. Each conductive coil was being driven at the same intensity during the course of a single test. Each test was driving at the same intensity as well. If electrical current readings from one coil were not recorded, for purposes of data analysis, it is assumed that both coils had the same current. If readings from neither coil were recorded, it is assumed that the coils had the overall average value of current, or 14 Amps.

Given a relative position, independent attitudes, and electrical current input, the model can be evaluated to determine the forces and torques, and by application of mass properties, the linear and angular accelerations expected.

Recorded Accelerations

During each test the IMU recorded the body-frame linear accelerations and angular rotation rate. The linear acceleration data is noisy, with noise amplitudes several times the magnitude of the constant term induced by accelerated motion. The source of this noise is the largely the vibration due to the EM interaction. Given that the current signal in each coil is roughly $I_p \sin(2\pi f)$ with $f = 83\text{Hz}$ and the electromagnetic force is $\propto I^2(t)$ then the force is $\propto \sin^2(2\pi f)$. Applying trigonometric identities and it is seen that the force is $\propto (\frac{1}{2} - \frac{1}{2}\cos(4\pi f)$. For a base frequency f of 83 Hz we expect that the vibration due to the force interaction should be at double that, or roughly 166 Hz. This is exactly what the accelerometer data shows. Figure 3-7 on the next page is the peak power spectral density of all accelerometers during the RGA testing. There is also a smaller peak at 130Hz which may be a structural frequency as it does not correpsond to the electromagnetic forces.

Knowing the spectral content of the IMU signal should allow for more intelligent filtering, since the problem frequencies are now known. Prior to the addition of RINGS there was a known 333Hz signal in the accelerometers, however it appears that operating the RINGS introduces energy at the 130 and 166Hz frequencies. Filtering is essential because the unfiltered IMU data (Figure 3-8 on the following page) is essentially unusable. Using a butterworth pattern lowpass filter design with a pass

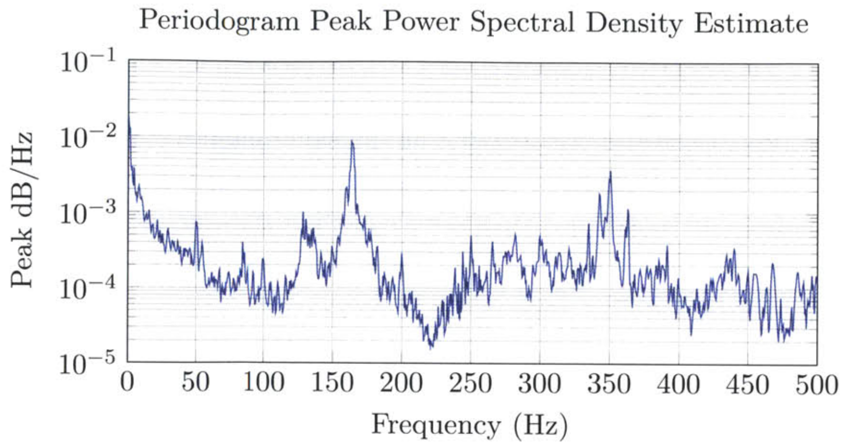


Figure 3-7: This is an aggregate of all IMU data collected during the RGA. The peak power spectral density at every frequency over all runs is recorded. The vibration due to the alternating magnetic field force interaction is at twice the base frequency, there is a very large spike in the accelerometer data at this frequency.

frequency of 1Hz, a stop frequency of 4 Hz (there is a 5Hz vibration due to IR sync), a pass band of 0.1dB, and a stop attenuation of 60dB the result is the steady line in Figure 3-8, which is zoomed-in in Figure 3-9 on the facing page. This filter was chosen as the motion itself is slow (<1Hz) and there is expected to be a 5Hz vibration source due to the Infrared synchronization scheme which causes the 83Hz waveform to "reset" every 5Hz. The pulse is because the 83Hz and 5Hz do not line up. The electrical current is 66.7% of the way through its 17th cycle when the IR flash occurs causing the drive electronics to jump back to 0% through a cycle. This discontinuity results in a descrease in current and a corresponding pulse in IR.

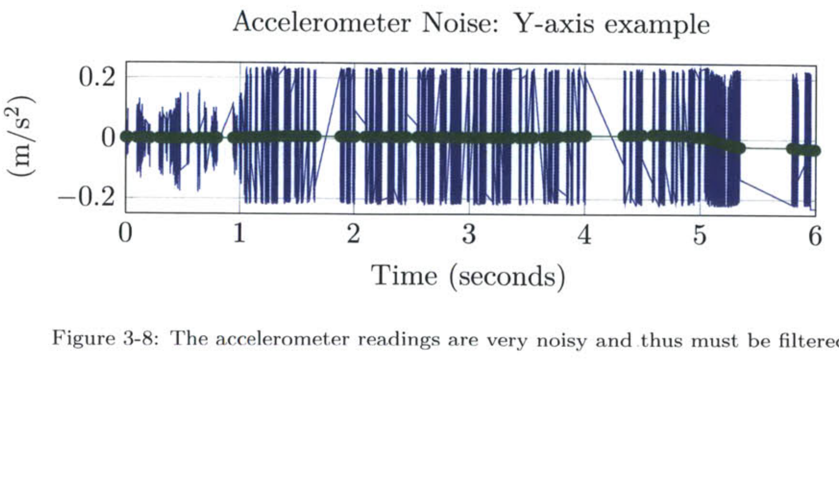


Figure 3-8: The accelerometer readings are very noisy and thus must be filtered.

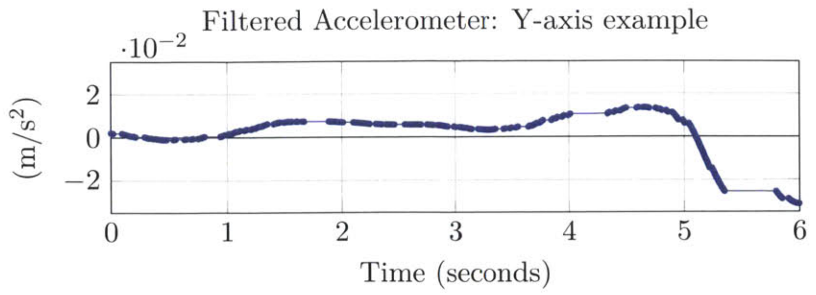


Figure 3-9: The accelerometer readings are still unstead even at low frequencies.

The problem with this result is that the filtered acceleration (with the end clipped off) has a mean of $E = 0.0045 m/s^2$ and a standard deviation $\sigma = .0032 m/s^2$. That relative level of variance makes it very difficult to discern any sort of pattern in the data.

Recorded Angular Rates

The angular rotation rate is fairly clean in comparison, except there is often significant data loss in the rate-gyro data. Also the requisite numerical differentiation (on the already sparse data) increases the noise level as well. Savitsky-Golay methods for smooth numerical differentiation were applied[16,17] to improve the differentiated result.

"Smoothing and Differentiation of Data by Simplified Least Squares Procedures.", Savitzky and Golay 1964

"Properties of Savitzky–Golay digital differentiators", Luo et al. 2005

SG smoothing and differentiation is a type of least-squares polynomial fitting smoothing function. At point x^* the signal $x(\cdot)$ is approximated locally as a polynomial of order m using n points before and after. The span of data points included in each approximation is the frame. SG filters can use arbitrary lead-lag frame sizes however using an asymmetric frame (unequal sample points before and after x^*) induces a phase change into the data (either lead or lag, depending on direction of imbalance). As such, using a symmetric frame makes SG a zero-phase filter. The frame size must be chosen sufficiently large (in time, not index) such that the period of the frame is larger than the period of the noise frequencies being filtered out.

SG methods can be used for arbitrary order differentiation, including 0^{th} order (i.e. data smoothing, no differentiation). SG 0-order smoothing was used to smooth the linear accelerations and 1^{st}-order smooth differentiation was used to determine the angular accelerations. Using

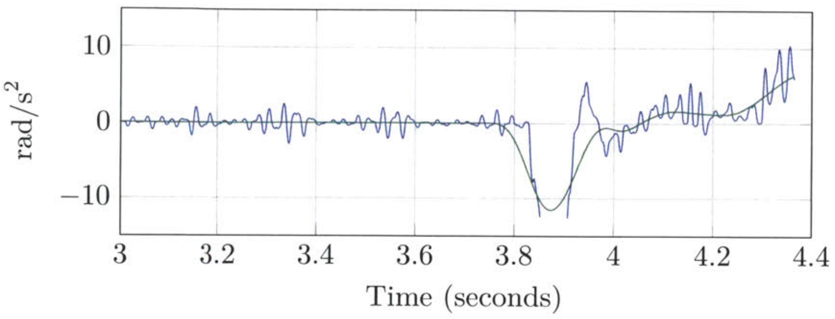


Figure 3-10: Savitzky-Golay smooth differencing produces a much smoother result than that achieved by central differencing.

smooth differentiation is essential, as seen in Figure 3-10.

Unfortunately while the rate gyro data is cleaner it also did not record properly and so much of it is missing as in Figure 3-11. Over all the tests where data was received from both satellites essentially none of them have good rate gyro data. Furthermore, even when filtered the accelerometer data (Figure 3-9 on the previous page) does not provide a steady reading of the acceleration experienced on the satellite.

Going Forward

The trajectory reconstruction does not dynamically constrain the state meaning adjacent time steps are unrelated numerically which is important for model validation. If the state was dynamically constrained then it must have been constrained using a model of the dynamics. The model

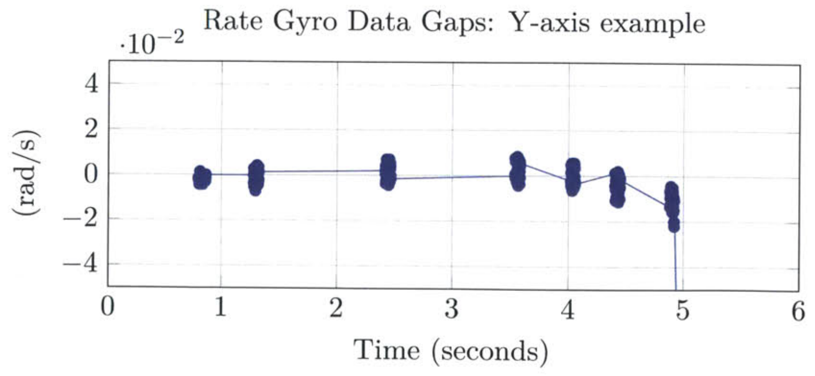


Figure 3-11: There are significant gaps in the rate gyro data, rendering it unusable

can't be used in the process of validating the same model, to do so would be circular logic with no clear proof the model is actually correct.

This means the measurement system must be very good because they are the only data available. State estimation can be robust because it has both a measurement model and a process model that it can integrate. In this case there is only a measurement model. Furthermore as discussed by Simon Nolet[18] using the ultrasound beacons for measurements is a fairly difficult process, even more so when the ultrasound sources are attached to 2m tall x 2 inch diameter posts only fixed at one end. As stated earlier Eslinger has done this though further improvements in the estimation algorithm will translate directly into improvements in this model validation.

"Development of a Guidance, Navigation and Control Architecture and Validation Process Enabling Autonomous Docking to a Tumbling Satellite", Nolet 2007

Chapter 4

Underactuated Closed Loop Position Control

For the objective of position control of a two satellite system a 2-dimensional underactuated system is considered. The reduction to two dimensions can be done because any initial and final positions for a two satellite define a plane, therefore the trajectory between an initial and final position lies within that plane. For position $\boldsymbol{r} = (x, y, z)$ and final position $\boldsymbol{r}_{tgt}$ the plane normal vector is $\hat{\boldsymbol{n}} = \hat{\boldsymbol{r}} \times \hat{\boldsymbol{r}}_{tgt}$ as shown in Figure 4-1 on the following page. The coordinate system $(\hat{\boldsymbol{\imath}}, \hat{\boldsymbol{\jmath}}, \hat{\boldsymbol{k}})$ of the planar motion is defined by the normal vector $\hat{\boldsymbol{z}} = \hat{\boldsymbol{n}}$ and the target vector $\hat{\boldsymbol{\imath}} = \hat{\boldsymbol{r}}_{tgt}$. Their cross product completes the right-hand set, $\hat{\boldsymbol{\jmath}} = \hat{\boldsymbol{n}} \times \hat{\boldsymbol{r}}_{tgt}$. This coordinate system is ill defined as $\hat{\boldsymbol{r}}$ approaches $\hat{\boldsymbol{r}}_{tgt}$ which is acceptable because it is only undefined when $\hat{\boldsymbol{r}} = \hat{\boldsymbol{r}}_{tgt}$, that is, when the satellite is already where it is trying to go.

One possible way to avoid this ill defined system is to define the normal vector $\hat{\boldsymbol{n}}$ from the initial position and the target position, not the current position. For any non-trivial problem $\hat{\boldsymbol{r}}_i \neq \hat{\boldsymbol{r}}_{tgt}$ therefore $\hat{\boldsymbol{n}}$ will remain well defined. This method is susceptible to out of plane drift, that is, any drift induced in the $\hat{\boldsymbol{n}}$ direction will cause the system to depart the plane defined by $\hat{\boldsymbol{n}}$.

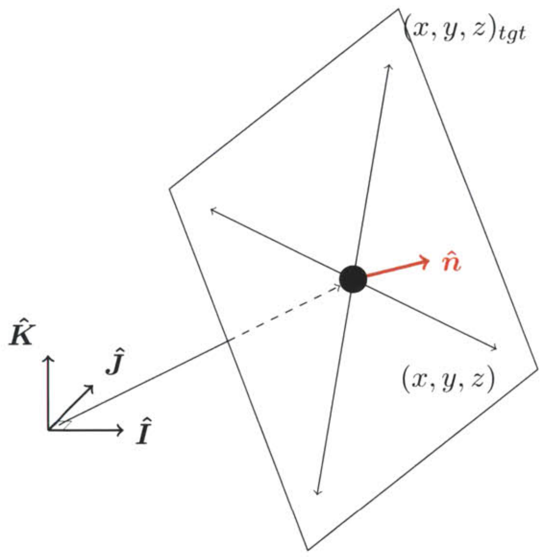


Figure 4-1: For any 2 satellite array, the current and final positions define a plane, so all motion can be described in 2 dimensions.

4.1 State Representation

Given that the system can be reduced to a two dimensional problem the position vector can be expressed in just its x and y components in the plane as in Figure 4-2 on the next page which reduces the state representation to $\mathbf{x} = [x, y, \alpha_i, \alpha_j]^\intercal$.

The kinematics are a second order integrator, that is, the kinematics of a rigid body in free space.

$$\begin{aligned} m_i\ddot{\boldsymbol{x}}_i - \boldsymbol{F}(\boldsymbol{x}_i, \boldsymbol{x}_j, ..., \boldsymbol{\mu}_i, \boldsymbol{\mu}_j, ...) = 0 \\ m_j\ddot{\boldsymbol{x}}_j - \boldsymbol{F}(\boldsymbol{x}_i, \boldsymbol{x}_j, ..., \boldsymbol{\mu}_i, \boldsymbol{\mu}_j, ...) = 0 \end{aligned} \tag{4.1}$$

The dynamics are thus goverened by Equation (4.1) where $\boldsymbol{F}$ is the electromagnetic force, described in the near-field by Equation (2.1) on page 26 and in the far-field by Equation (2.4) on page 31. In a fully actuated system both dipoles $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ are fully controllable thus the direction and magnitude of $\boldsymbol{F}$ is fully controllable by setting the dipoles appropriately.

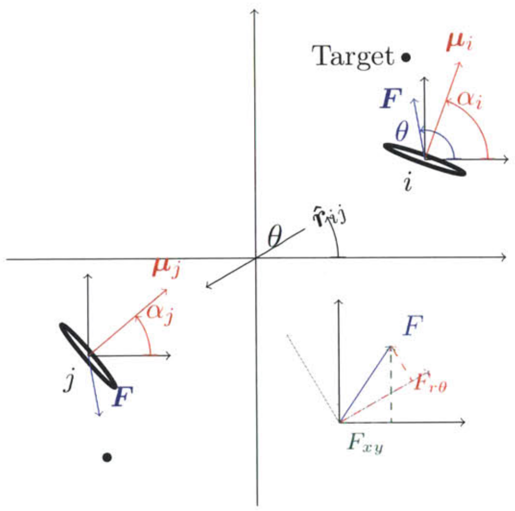


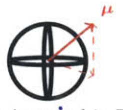
Figure 4-2: Visual definition of the relevant angles and unit vectors for the 2-dimensional system. The whole system is symmetric around the origin due to conservation of momentum. The direction of force θ is defined by α_i, α_j and $\hat{\boldsymbol{r}}_{ij}$.

4.2 Fully Actuated Control

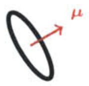
The direction of force is uniquely defined by the dipole vectors of each spacecraft in the system. However, the converse is not true. The dipole vectors of every satellite in a system are not uniquely defined by a given or desired force direction. This is because there is a linear dependence in Equation (4.1) on the preceding page caused by the constraint $\boldsymbol{F}_i + \boldsymbol{F}_j = 0$. This provides two unconstrained degrees of freedom (one for each dimension). These solutions are referred to as the *dipole solutions* as they are the set of magnetic dipole vectors that fully define the formation. The dipole solutions arise by solving the force model equations for dipole vectors given a desired force vector[10].

"Electromagnetic Formation Flight Dipole Solution Planning", Schweighart 2005

If the system is fully actuated, that is, if there are as many electromagnetic coils as there are translational dimensions of interest, there is at least one pair of dipole orienations that produce the desired forces at any moment in time. As such a system is fully actuated there always exists a set of two control inputs $[\mu_x, \mu_y]$ per spacecraft that can produce the desired dipole vector, $\boldsymbol{\mu} = \mu_x\hat{\boldsymbol{\imath}} + \mu_y\hat{\boldsymbol{\jmath}}$. Assuming the dynamics of the electrical current within a coil is *fast* relative to the rigid body motion and attitude dynamics, the dipole vector itself is taken as the control



Fully Actuated in $\mathbb{R}^3$



Underactuated above $\mathbb{R}^1$

input. Fast means sufficiently fast that the settling time of the system is negligible on the timescales of interest.

A linear control law can regulate position for this system. This can be seen easier by ignoring the real force model and just thinking of the system in terms of two inputs, F and θ.

$$m\ddot{\boldsymbol{x}} - F \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} = 0 \tag{4.2}$$

Viewed this way the system is very easy to control. There are two unknowns (F and θ) and two equations (x and y). If the controlled state response is designed such that

$$\ddot{\boldsymbol{x}} = -\lambda_d \dot{\tilde{\boldsymbol{x}}} - \lambda_p \tilde{\boldsymbol{x}} \tag{4.3}$$

the system will have an equilibrium point at $\tilde{\boldsymbol{x}} = 0$. This is simply a system of two equations with two unknowns (F and θ). Solving for the inputs gives the control law.

The forced response is defined as the following: $\boldsymbol{x}_f = \lambda_d \dot{\tilde{\boldsymbol{x}}} + \lambda_p \tilde{\boldsymbol{x}}$ This is the desired response of the system under control actuation

$$F = \sqrt{(\lambda_d \dot{\tilde{\boldsymbol{x}}} + \lambda_p \tilde{\boldsymbol{x}})^{\mathsf{T}} (\lambda_d \dot{\tilde{\boldsymbol{x}}} + \lambda_p \tilde{\boldsymbol{x}})} = \sqrt{\boldsymbol{x}_f^{\mathsf{T}} \boldsymbol{x}_f} \tag{4.4a}$$

$$\theta = \tan^{-1}\left(\frac{\lambda_d \dot{\tilde{y}} + \lambda_p \tilde{y}}{\lambda_d \dot{\tilde{x}} + \lambda_p \tilde{x}}\right) \tag{4.4b}$$

In this fully actuated system the position dynamics and control system do not depend on the attitudes α and β. Thus to achieve total state control a second linear controller could apply PD feedback control to the angles. Unlike position the angle states are decoupled therefore each angle would have its own scalar feedback control law.

4.3 Underactuated System Controllability

Unfortunately the single coil EMFF system does not satisfy linear controllability. The question of linear controllability, tested by evaluating the rank of the controllability matrix

$$\mathcal{M}_c = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \ldots \end{bmatrix} \tag{4.5}$$

is one that asks, in the linearized system, is there any combination of control inputs that can take me to any arbitrary point within the state vector? Upon thinking this is clearly not the case, the matrix math will soon follow to show this.

The force at any moment in time is uniquely defined by the attitudes of each spacecraft, e.g. when aligned axially it is impossible to directly move in the cross-track direction. Thus the linearization about the state at any moment in time results in only one direction of force application. Thus any linearized system should end up with at least one uncontrolled mode (velocity in the cross track direction) and if the linearization was about a stationary point it should have two uncontrolled modes (position and velocity in the cross track direction).

Now the math to support this. The state vector $\mathbf{x}$ is

$$\mathbf{x} = \left[x, y, \alpha, \beta, \dot{x}, \dot{y}, \dot{\alpha}, \dot{\beta}\right] \tag{4.6}$$

with control inputs

$$\mathbf{u} = [\mu_i\mu_j, T_\alpha, T_\beta] \tag{4.7}$$

The three inputs are composite dipole intensity, $\mu_i\mu_j$, the thruster torque on satellite α and the thruster torque on satellite β

for which a linearized system of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{4.8}$$

is found by linearizing the nonlinear model dynamics

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{4.9}$$

such that

$$\dot{\mathbf{x}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\boldsymbol{\delta}\mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\boldsymbol{\delta}\mathbf{u} \tag{4.10}$$

If we now evaluate each of these terms around the linearized conditions (axial alignment, arbitrary separation distance, stationary):

$$\begin{aligned} \mathbf{x}^* &= [1, 0, 0, 0, 0, 0, 0, 0] \\ \mathbf{u}^* &= [0, 0, 0] \end{aligned} \tag{4.11}$$

The 1's in the upper corner of $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ are simply an artifact of transforming the second order system of N states into a first order system of $2N$ states. Otherwise there are no unforced dynamics in this system. The only motion comes from control input, as seen in $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$

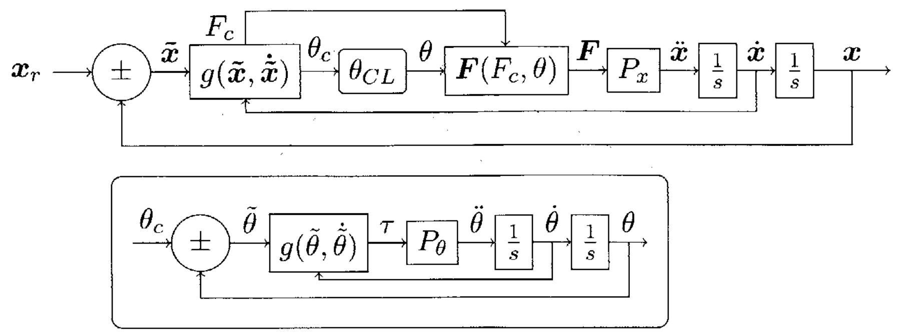
This results in the following

$$\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.12a}$$

$$\mathbf{B} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{6C_m}{(x^2+y^2)^2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.12b}$$

It is now obvious there is a problem. There is no way for the control inputs to effect the $\dot{y}$ state. This is not just a side effect of the state we chose to linearize around, there is a coupling between x and y such that it is not possible to have arbitrary x and y control. As the linearization point changes the values of the 5th and 6th row of $\mathbf{B}$ will indeed change, but in a linearly dependent way.

The result is that the controllability matrix loses rank, in this case having rank 6 while the required rank for full controllability is rank 8. This is because the system is underactuated and the linearized $\mathbf{B}$ matrix only contains information about pushing the system in a single direction. Information about the orthogonal direction is lost during linearization. This means to properly control the full two-dimensional position of the system will require that the dynamics are *not* linearized. One way to approach this problem is to use the attitude state as a pseudo-control input to help drive the position error to zero.



$g(\cdot)$ is the generalized controller function. $g(\cdot)$ for different inputs *is* a different function.
P is the generalized plant, $P_{(\cdot)}$ is the plant for the variable described.

Figure 4-3: Representative diagram of heirarchical close loop position control. Steer θ to particular value that moves $\boldsymbol{x}$ in the desired direction. Note that many of these blocks are in fact non-linear

4.4 Heirarchical System Convergence

When the dipole angle is not an input but rather is itself a state being controlled there is a heirarchy of control. Looking at the block diagram of this system in Figure 4-3 shows how the position controller is wrapped around the angle controller. Note, θ isn't a real angle, the real angles are $\boldsymbol{\alpha} = [\alpha_i, \alpha_j]^\mathsf{T}$ so θ needs to be converted to $\boldsymbol{\alpha}$ by solving for the dipole solutions. Additionally, the commanded force F_c is not real and needs first to be translated to commanded currents I_c, which can be considered the *real* input to the EMFF plant.

In the construct of Figure 4-4 on the next page the desired rates $\dot{\boldsymbol{x}}$ and $\dot{\boldsymbol{\alpha}}$ are treated as zero. It not true that these rates are zero however they were not explicitly solved here so they are assumed zero. The desired translational rates can be easily found given a smooth (C^1) trajectory $\boldsymbol{x}(t)$. To properly reflect this in the block diagram the rate feedback would be compared against the reference rate and the error fed to the controller.

The desired angles are a function of $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ indirectly through θ. Therefore the desired angular rates are dependent on the position and velocity, though deriving what the angular rates should be is much less direct. This remains a task to be completed. Tracking $\boldsymbol{\alpha}$ while providing the correct $\dot{\boldsymbol{\alpha}}$ will improve angle tracking performance beyond the results presented here.

The result is a heirarchical relationship between the angle control

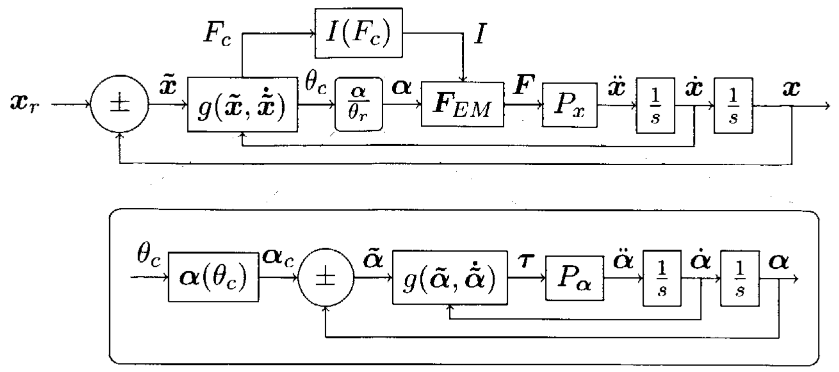


Figure 4-4: Representative diagram of heirarchical close loop position control for the EMFF system. Steer $\boldsymbol{\alpha} = \begin{bmatrix}\alpha_i & \alpha_j\end{bmatrix}^T$ such that the force direction θ is such that $\boldsymbol{x}$ moves in the desired direction. Note that many of these blocks are in fact non-linear

and the position control. Contraction analysis will allow us to state some properties of this heirarchical system.

4.4.1 Contraction Analysis

Contraction analysis states that for a system with generalized Jacobian

$$\mathbf{F} = \left(\dot{\boldsymbol{\Theta}} + \boldsymbol{\Theta}\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)\boldsymbol{\Theta}^{-1} \tag{4.13}$$

"On Contraction Analysis for Non-linear Systems", Lohmiller and Slotine 1998

where $\boldsymbol{\Theta}$ is a square uniformly *p.d.* matrix the system is said to be *Contracting* if $\mathbf{F} < 0$ or $\mathbf{F}$ is negative definite uniformly[19]. *Contracting* means that trajectories "forget" their past. Properly it means that all trajectories tend towards an equilibrium by means of all trajectories tending towards each other. If the virtual separation between trajectories vanishes in effect this means they tend towards a common state (an equilibrium, a limit cycle, etc). In the case of the underactuated EMFF system the identity matrix is a transformation matrix that allows for a *n.d.* generalized Jacobian. Different transforms ($\boldsymbol{\Theta}$) may be useful for future analysis as the theory can guarantee convergence to within a ball of some radius around the equilibrium where the size of the ball is determined by the transformation matrix Θ and the magnitude of the disturbance signals.

For the entire position-attitude control system expressed as a first order system the state vector is

$$\mathbf{x} = \left[\tilde{\alpha}, \dot{\tilde{\alpha}}, \tilde{\beta}, \dot{\tilde{\beta}}, x, \dot{x}, y, \dot{y}\right]^{\mathsf{T}} \tag{4.14}$$

which can be broken into the component pieces $\mathbf{x}_1$ and $\mathbf{x}_2$ defined as

$$\begin{aligned} \mathbf{x} &= \left[\mathbf{x}_1, \mathbf{x}_2\right]^{\mathsf{T}} \\ \mathbf{x}_1 &= \left[\tilde{\alpha}, \dot{\tilde{\alpha}}, \tilde{\beta}, \dot{\tilde{\beta}}\right]^{\mathsf{T}} \\ \mathbf{x}_2 &= \left[x, \dot{x}, y, \dot{y}\right]^{\mathsf{T}} \end{aligned} \tag{4.15}$$

with the derivative functions $\mathbf{f} = \dot{\mathbf{x}}$ split in the same manner. The Jacobian of this partitioned system is structured as follows.

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \left[\begin{array}{c|c} \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_2} \\ \hline \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_2} \end{array}\right] = \left[\begin{array}{c|c} \mathbf{F}_1 & \mathbf{F}_{12} \\ \hline \mathbf{F}_{21} & \mathbf{F}_2 \end{array}\right] \tag{4.16}$$

If it can be shown that $\mathbf{F}_{12}$ is zero and that $\mathbf{F}_{21}$ is bounded then the entire system is contracting for $\mathbf{F}_1$ and $\mathbf{F}_2$ contracting under the heriarchical principle[19] of contraction analysis.

"On Contraction Analysis for Non-linear Systems", Lohmiller and Slotine 1998

Angle Subsystem

First we note that the angle subsystem is two systems in parallel, α and β and therefore it too can be split into $\mathbf{x}_1 = [\mathbf{x}_\alpha, \mathbf{x}_\beta]^{\mathsf{T}}$ whose Jacobian is

$$\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1} = \left[\begin{array}{c|c} \mathbf{F}_\alpha & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{F}_\beta \end{array}\right] \tag{4.17}$$

Which has eigenvalues $\boldsymbol{\lambda}_\alpha$ and $\boldsymbol{\lambda}_\beta$, therefore it is contracting for $\mathbf{F}_\alpha$ and $\mathbf{F}_\beta$ contracting. Therefore it is safe to consider the single case of $\mathbf{F}_\alpha$ and know that $\mathbf{F}_\beta$ is contracting under the same principles.

If we assume that the torque available, T, is always greater than the electromagnetically induced torque τ by some margin η

$$\text{Assume: } |T - \tau| > \eta \tag{4.18}$$

and further that controller gains λ_d and λ_p are chosen such that

$$\left|-\lambda_d\tilde{\alpha} - \lambda_p\dot{\tilde{\alpha}}\right| <= \eta \tag{4.19}$$

it can then be known that the angle controller will always have excess torque $T' = T - \tau$ sufficient to satisfy the control law equation (4.20). A proposed angle controller is a simple PD control law on the angle error signal.

$$T' = -\lambda_d\tilde{\alpha} - \lambda_p\dot{\tilde{\alpha}} \tag{4.20}$$

The real applied torque T would then be $T = T' - \tau$ such that the resultant torque on the system when the electromagnetic torque is added is just T'. The angle dynamics are then

$$\ddot{\tilde{\alpha}}I_{zz} = -\lambda_p\tilde{\alpha} - \lambda_d\dot{\tilde{\alpha}} \tag{4.21}$$

Under these dynamics the angle error $\tilde{\alpha}$ has an equilibrium point at $[\dot{\alpha}, \alpha] = [0, 0]$. The Jacobian $\frac{\partial \mathbf{f}_\alpha}{\partial \mathbf{x}_\alpha}$ is then

$$\mathbf{F}_\alpha = \frac{\partial \mathbf{f}_\alpha}{\partial \mathbf{x}_\alpha} = \begin{bmatrix} 0 & 1 \\ -\lambda_p & -\lambda_d \end{bmatrix} \tag{4.22}$$

which has eigenvalues

$$\boldsymbol{\lambda}_\alpha = -\frac{\lambda_d}{2} \pm \frac{\sqrt{\lambda_d^2 - 4\lambda_p}}{2} \tag{4.23}$$

For $\lambda_p < 0$ the eigenvalues $\boldsymbol{\lambda}_\alpha$ are always negative. Therefore for $\lambda_p < 0$ the angle system, $\mathbf{F}_1$ is *contracting.*

The assumption $T - \tau = T' > \eta$ allows us to effectively decouple the attitude system from the position system. The statement means that the attitude control system *always* has sufficient control authority to go where it needs to regardless of what the position controller is trying. This is a conservative assumption and systems that violate it may still be stable

Equation (4.21) also leads us to show the off diagonal block term $\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_2} = 0$ as there is no dependence on position. The updated overall Jacobian is then

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \left[\begin{array}{cc|c} \mathbf{F}_\alpha & & \mathbf{0} \\ & \mathbf{F}_\beta & \\ \hline \multicolumn{2}{c|}{\mathbf{F}_{21}} & \mathbf{F}_2 \end{array}\right] \tag{4.24}$$

The lower two blocks (the position subsystem and its coupling with angle error) now need to be analyzed.

Position Subsystem

The position subsystem is not as separable as the angle subsystem is due to the dependence of force on range, $r^2 = x^2 + y^2$. This couples the x and y dynamics. The position subsystem $\mathbf{x}_2$ thus has dynamics

$$\mathbf{f}_2(\mathbf{x}_2) = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \frac{F_r x}{(x^2+y^2)^{1/2}} - \frac{F_\theta y}{(x^2+y^2)^{1/2}} \\ \dot{y} \\ \frac{F_r y}{(x^2+y^2)^{1/2}} + \frac{F_\theta x}{(x^2+y^2)^{1/2}} \end{bmatrix} \tag{4.25}$$

The forces F_r and F_θ are the forces described in the rotating coordinate frame fixed aligned with the inertial orientation of the satellite pair. F_r could also be called F_{axial} and F_θ could be called $F_{transverse}$. The sines and cosines rotate these forces into the inertial X-Y coordinate frame, see Figure 4-2 on page 47.

Chapter 2 discussed how the forces $\mathbf{F}_r$ and $\mathbf{F}_\theta$ are aligned in the rotated reference $R\Theta$. The two frames are rotated by angle θ relative to each other

Referencing the far-field model in Equation (2.5) on page 32 the magnitude of the force vector $[F_r, F_\theta]$ is

Note that C_m is the constant variabel that encapsulates several other commonly occuring constants in electromagnetic force and torque expressions. It is defined in Chapter 2.

$$\left\| \begin{bmatrix} F_r \\ F_\theta \end{bmatrix} \right\| = \frac{3C_m\mu_i\mu_j}{16(x^2+y^2)^2}\sqrt{\gamma_r^2 + \gamma_\theta^2} \tag{4.26}$$

where γ_r and γ_θ are sine and cosine parts of the far field model Equation (2.5) on page 32 as restated in Equation (4.27) with some modification. In the mode as originally stated the angles were with reference to the rotating reference frame. The angles α and β here are inertially defined, so there is an offset of θ required.

$$\gamma_r = \frac{1}{2}(3\cos(\alpha+\beta-2\theta) + \cos(\alpha-\beta)) \tag{4.27}$$

$$\gamma_\theta = \sin(\alpha+\beta-2\theta) \tag{4.28}$$

Therefore the force model restated is

$$\mathbf{F} = \begin{bmatrix} F_r \\ F_\theta \end{bmatrix} = \frac{3C_m\mu_i\mu_j}{16(x^2+y^2)^2}\begin{bmatrix} \gamma_r \\ \gamma_\theta \end{bmatrix} \tag{4.29}$$

Inserting Equation (4.29) into Equation (4.25) yields

$$\mathbf{f}_2(\mathbf{x}_2) = \begin{bmatrix} \dot{x} \\ \frac{3C_m\mu_i\mu_j}{16(x^2+y^2)^{5/2}}\left(x\gamma_r + y\gamma_\theta\right) \\ \dot{y} \\ \frac{3C_m\mu_i\mu_j}{16(x^2+y^2)^{5/2}}\left(y\gamma_r - x\gamma_\theta\right) \end{bmatrix} \tag{4.30}$$

Now it is time to find a dipole solution magnitude $\mu_i\mu_j$ and directions α and β. Prior work has described in length how to do this for N satellite formations or to do it for arbitrary extra constraints. As this work is limited to two satellite systems, and because it is interesting that this solution exists, the dipole solution used will be based on assuming $\beta = \theta$ (the second satellite is controlled to point directly at the first, the angle in the rotating reference frame $\beta_{r\theta} = 0$) from which there is an analytic solution for α_c, the commanded angle. See Appendix A on page 95 for more information. With this assumption we have

$$\gamma_r = 2\cos(\alpha_c) \tag{4.31}$$

$$\gamma_\theta = \sin(\alpha_c) \tag{4.32}$$

Updating the dynamics $\mathbf{f}_2$ yields

$$\mathbf{f}_2(\mathbf{x}_2) = \begin{bmatrix} \dot{x} \\ \frac{3C_m\mu_i\mu_j}{16(x^2+y^2)^{5/2}}\left(2x\cos(\alpha_c) + y\sin(\alpha_c)\right) \\ \dot{y} \\ \frac{3C_m\mu_i\mu_j}{16(x^2+y^2)^{5/2}}\left(2y\cos(\alpha_c) - x\sin(\alpha_c)\right) \end{bmatrix} \tag{4.33}$$

Now to find the value of $\mu_i\mu_j$ combine Equation (4.26) on the previous page and Equation (4.4) on page 48

$$\mu_i\mu_j = \sqrt{\mathbf{x}_f^\mathsf{T}\mathbf{x}_f}\,\frac{16(x^2+y^2)^2}{3C_m\sqrt{\gamma_r^2+\gamma_\theta^2}} \tag{4.34}$$

This then recreates the linear control law described in Equation (4.4) on page 48 which leads to a Jacobian

$$\mathbf{F}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\lambda_p & -\lambda_d & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\lambda_p & -\lambda_d \end{bmatrix} \tag{4.35}$$

that has eigenvalues

$$\boldsymbol{\lambda}_{\mathbf{f}_2} = -\frac{\lambda_d}{2} \pm \frac{\sqrt{\lambda_d^2 - 4\lambda_p}}{2} \tag{4.36}$$

which are always less than 0 for $\lambda_p < 0$. Therefore for $\lambda_p < 0$ the position system, $\mathbf{F}_2$ is *contracting.*

Note that the angles α and β are not precisely α_c and β_c (the commanded angles) therefore the error signals $\tilde{\alpha}$ and $\tilde{\beta}$ are nonzero. This adds in bounded disturbances that show up in $\mathbf{F}_{21} = \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_1}$ (i.e. the lower left hand quadrant in the overall Jacobian). The precise analytical expression of $\mathbf{F}_{21}$ has not yet been fully described, however knowing that Jacobian precisely is not important in the context of contraction analysis. Contraction theory states that for a Jacobian such as Equation (4.37) so long as $\mathbf{F}_{21}$ is bounded the overall system is contracting.

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \left[\begin{array}{c|c} \mathbf{F}_1 & \mathbf{0} \\ \hline \mathbf{G} & \mathbf{F}_2 \end{array}\right] \tag{4.37}$$

We can say that $\mathbf{F}_{21}$ is bounded by noting that it consists of terms from Equation (4.33) on the preceding page multiplied by sines and cosines of small angles $\tilde{\alpha}$ and $\tilde{\beta}$. Further evidence to support the claim that $\mathbf{F}_{21}$ is bounded is that numerical simulation of this control law demonstrates its stability.

The way to interpret Equation (4.37) is that system 1 is exponentially convergent to its equilibrium and system 2 is contracting subject to a decaying disturbance signal thus is also convergent to its equilibrium.

4.5 Step Response

The control prinicples just described were implemented in simulation and the following results are based off of this simulation. The scenarios being tested are for a step input trajectory which is actually a worst case for a controller of this form as the desired trajectory has a discontinuity. One feature not described above that was added to the controller implementation was a so-called "Force Roll-off" whereby the commanded current rolls off with the cosine of angle error. As described at the end of Section 4.4 on page 51 the bottom left quadrant of the Jacobian $\mathbf{G}$ is composed of the residuals from angle errors. The effect of $\mathbf{G}$ is that it acts as a disturbance on $\mathbf{F}_2$. The smaller $\mathbf{G}$ the smaller the disturbance and thus the faster $\mathbf{F}_2$ converges. By having the command roll-off with the angle error it reduces the effect of angle errors. The formal statement is

$$I_{actual} = \cos\tilde{\alpha}\cos\tilde{\beta} I_{des} \tag{4.38}$$

The mass properties were chosen similar to that of the RINGS hardware units, though this choice was arbitrary.

For large angle controller gains the trajectory is essentially a straight line from the initial to final point. This is expected because when at rest the desired force is determined purely by the position error (the velocity error is zero when at rest) thus it will point directly at the target. As the system accelerates in that direction the vecocity error increases but still along the same vector so the direction of motion doesn't change. However when the P-D tradeoff occurs, that is, when the control input needs to reverse and begin slowing down the system, due to the 2-dimensional nature of this the dipole vector does not necessarily decrease to zero magnitude but rather "swings" by the origin as in Figure 4-5 on the facing page.

The consequence of this "swing" is that the commanded angle α_c changes very rapidly and the angle controller will try to track it. This angle can swing arbitrarily fast depending how close to the origin the vector passes (this actually means the "better" the flip, that is, the closer it is to actually flipping directly over the origin, the faster the angle swings). This would be a bad thing if α had to track α_c mod 2π.

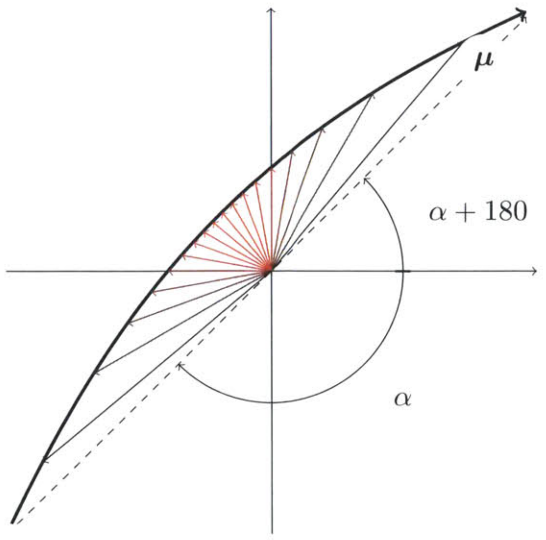


Figure 4-5: Dipole Swing caused by the two dimensional nature of the problem. The dipole vector doesn't pass directly over the origin, instead the dipole angle swings very fast from α to $\alpha + 180$. The rate is fastest when closest to the origin (red arrows).

However, this is not the case for a system that can drive $\mu < 0$. For a direct current system that means flowing current in the opposite direction from the "positive" direction. For an alternating current system that means driving at phase angle $\phi = \pi$ such that

$$\mu_i \mu_j = \cos(2\pi\omega t)\cos(2\pi\omega t + \pi) = -\cos^2(2\pi\omega t) \tag{4.39}$$

By driving at the opposed phase the resultant force is the negative of the in-phase value.

The ability to actuate with "negative current" means the system only needs to track to $\alpha_c \bmod \pi$. In these simulations this is implemented by going to whichever angle, $\alpha_c \bmod \pi$ or $(\alpha_c + \pi) \bmod \pi$ is closer. This could be made more intelligent by including the angle rate in this consideration. The most intelligent solution is to detect when the angle starts to swing around and not track it until it settles on the far side of the origin, at which point if $(\alpha - \alpha_c)_{\text{before}} = (\alpha - \alpha_c)_{\text{after}}$. Essentially if the system can detect an almost 180 degree swing, instead of tracking the full swing around it can short circuit and instead wait until $\alpha_c + pi$ is closer than α_c and track to it instead. That being said, such a heuristic is not included in these results, and it is visible as large torque "spikes" when the system passes by the target point and $\tilde{\mathbf{x}}$ approaches zero.

4.5.1 Unbounded Controls: Fast Angle Control

The first result (Figure 4-6 on page 63) is a step from $\mathbf{x} = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}^{\mathsf{T}}$ to $\mathbf{x} = \begin{bmatrix} 1 & 0 \end{bmatrix}^{\mathsf{T}}$. This is run with no control saturation. To keep position and angle gains distinct position gains will be λ and angle gains will be k. The gains used are $\begin{bmatrix} \lambda_p & \lambda_d & k_p & k_d \end{bmatrix} = \begin{bmatrix} 0.1 & 2 & 25 & 10 \end{bmatrix}$. The values were selected arbitrarily as the contraction analysis has shown that for $\lambda_p > 0$ and $k_p > 0$ the systems are contracting. Therefore so long as the gains remain positive the precise value is merely a matter of the desired performance, which is of little concern for now as this work is intended to demonstrate convergence, not particular performance metrics.

With the given parameters this system has exceptionally good performance, within seconds the angles are tracking the desired values and the angle error is very low. When the control inputs can exactly satisfy the feedback control law the performance is exceptional.

4.5.2 Saturated Controls: Fast Angle Control

The next result (Figure 4-7 on page 64) is the exact same problem run with the control inputs bounded to $-1 < \tau < 1$ and $18 < I < 18$. As expected the overshoot is larger. For large angle gains even with the saturation the angle converges much faster than the position such that the rajectory remains very straight. There is more of a spiral in the phase plane as this system converges onto the target slower. Note the spikes in the angle and angle rates. This is possibly the most prominent impact that saturation has, it prevents the angles from tracking the commanded angles all the way around as in Figure 4-5 on the previous page. By saturating the torque as the commanded angle starts to swing around the dipole follows (the initial spike in angle) but at some point the angle error exceeds $\pi/2$ and α is now closer to the opposite angle $\alpha_c + \pi$ at which point it starts tracking that opposite angle and flips the sign of the control input. Note that the control current in Figure 4-6 on page 63 is always positive, this is because the angle system has sufficient control bandwidth to track α_c. The saturated system does not have sufficient bandwidth and so when it switches from tracking α_c to $\alpha_c + \pi$ there is a corresponding switch in the sign of the electrical current. Additionally, the current always passes smoothly through zero due to the "Force Rolloff" feature described in Equation (4.38) on page 58, otherwise it would have a discontinuity when the angle switches where it is tracking.

4.5.3 Unbounded Controls: Slow Angle Control

The previous two scenarios almost felt like cheating when talking about underactuated control because the angle controller was so fast that there was very little interaction between the angle dynamics and position dynamics. In the extreme, the angle dynamics are so fast that they are assume instantaneous relative to the position dynamics. At that point the system is hardly "underactuated". So in the spirit of not cheating, if the angle controller is slowed down significantly such that the gains are now

$$\begin{bmatrix}\lambda_p & \lambda_d & k_p & k_d\end{bmatrix} = \begin{bmatrix}0.1 & 2 & .04 & .4\end{bmatrix}$$

which should reveal a stronger coupling between the angle and position dynamics and really show off the underactuated controller.

Unsurprisingly Figure 4-8 on page 65 performs wonderfully despite heavily overlapping the time-constants of the angle and position systems. Note the significantly longer timespan for this maneuver and note the attempted rotation (and electrical current zero-crossing) around 20 seconds which it soon gave up on and decided to track the opposite control angle instead.

This is about as "loose" as the system can get in the current configuration. Subsequent slowing down of angle dynamics yields a system that is still contracting but the satellites collide because the contraction is so slow.

4.5.4 Bounded Controls: Slow Angle Control

Finally, for completeness, the bounded controls with slow angle controller scenario is examined (Figure 4-9 on page 66). As it turns out this is an interesting scenario for it violates the constraint that the requisite force be satisfiable. It fails this because of the $\frac{1}{r^4}$ term. The trajectory sends the system outwards and within the time-span simulated it never recovers. It *is* possible to actually "escape" the formation. That is, there is some speed v_{esc} above which it is *impossible* to maintain formation, the system will drift apart to infinity. This happens when the specific mechanical energy of the system is larger than the potential energy well defined by the magnetic field of the other satellite. This is not an aspect considered in this controls analysis, further study could look at including the escape velocity limit in the stability analysis.

4.6 Summary

Contraction analysis provides a very powerful method for demonstrating the convergence of the underactuated EMFF system. This statement holds up under simulation very well providing convergent behavior in position and attitude. This still remains an underactuated system which means there are not sufficient inputs to properly control both position *and* attitude simultaneously therefore the angles themselves are used as inputs to the position system, which then has sufficient control authority to be fully controlled. This leads to a heirarchical control structure where the angles track a commanded angle that guides the position to the reference position with the desired dynamics, in this case the dynamics of a heavily damped oscillator.

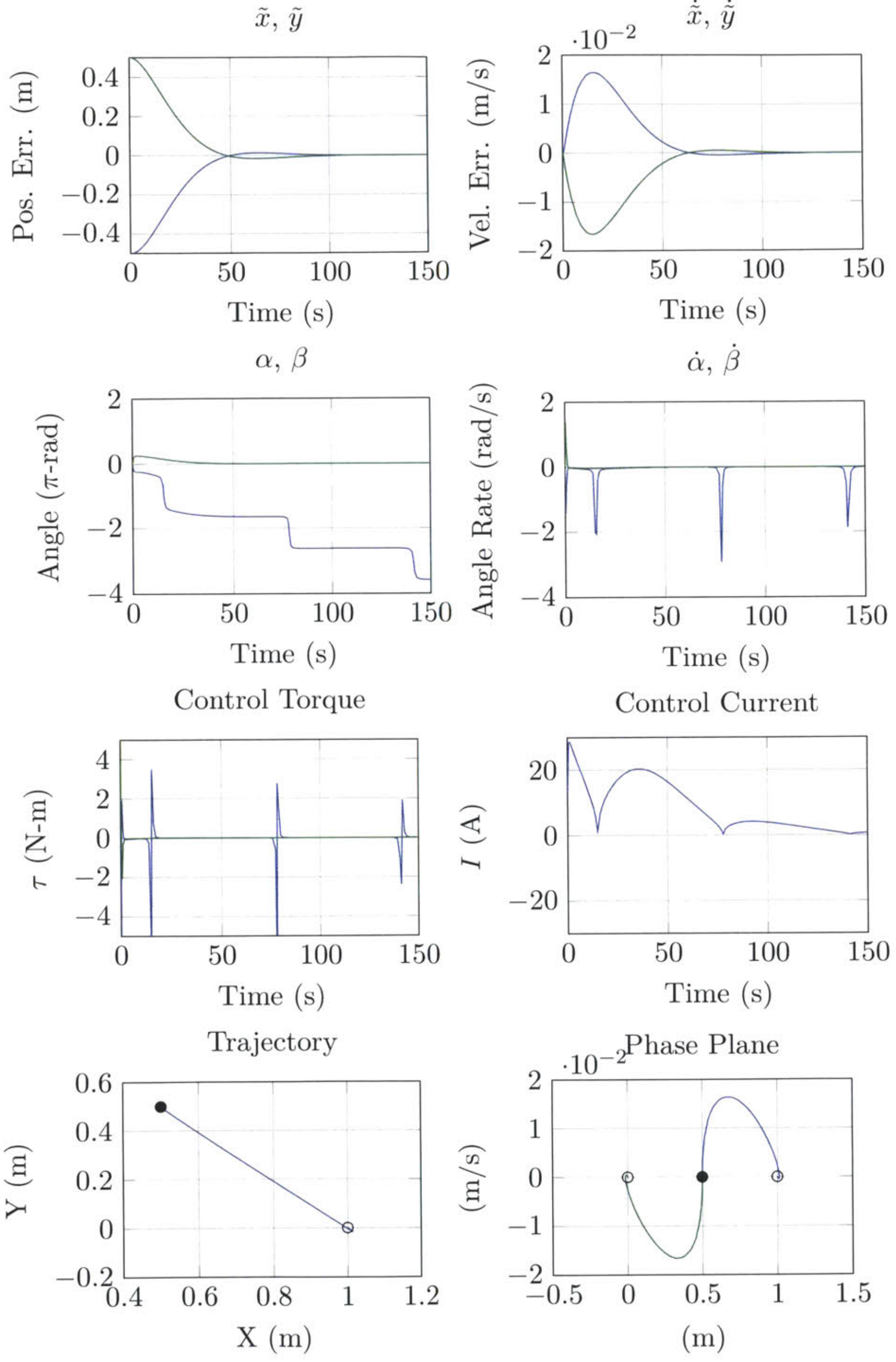


Figure 4-6: Unbounded control inputs. The performance is very good but the control inputs used are quite large at times. Start:● End:○.

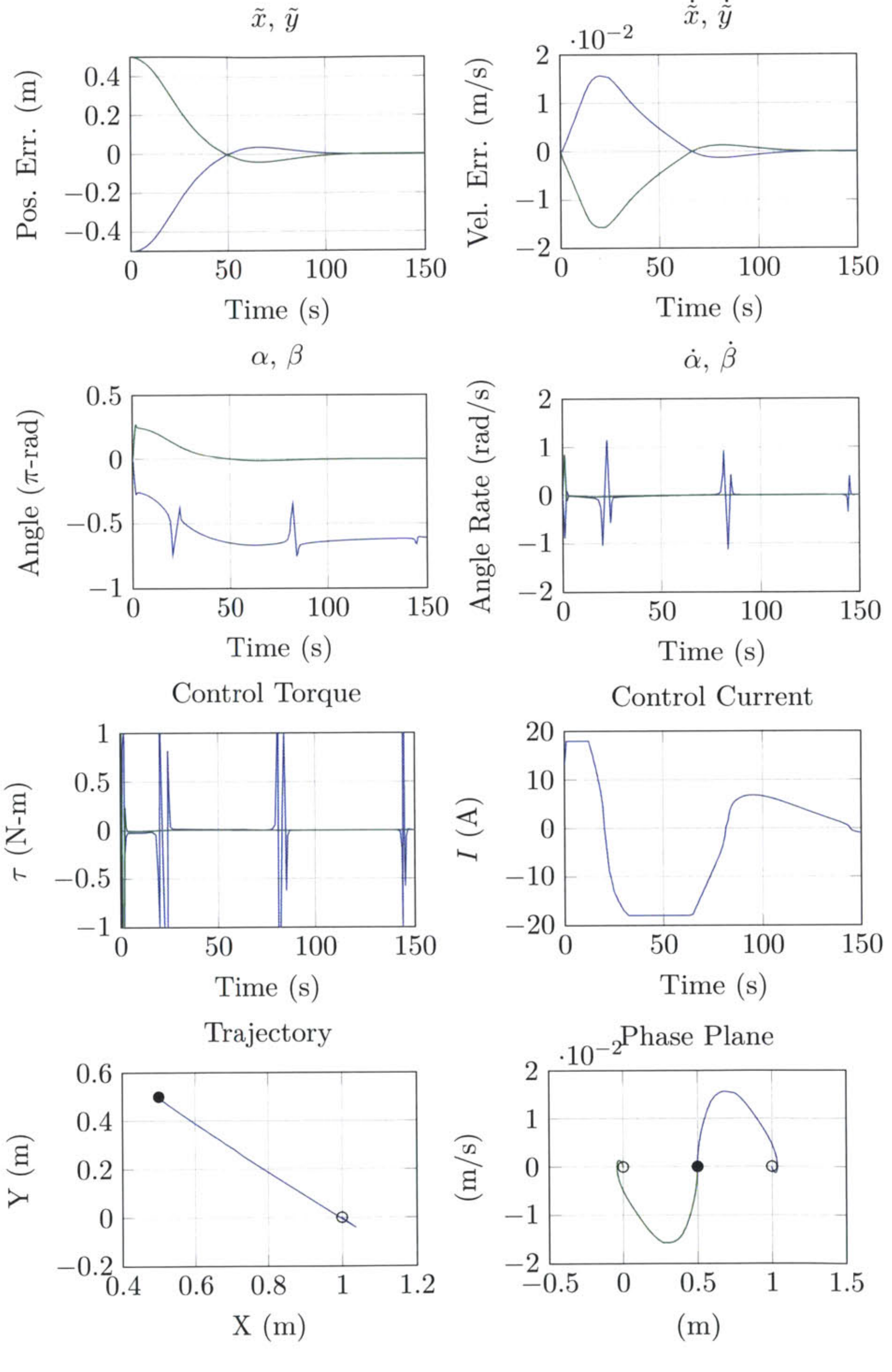


Figure 4-7: Bounded control inputs. The performance is still very good but the overshoot is larger with the saturated controls. Start:● End:○.

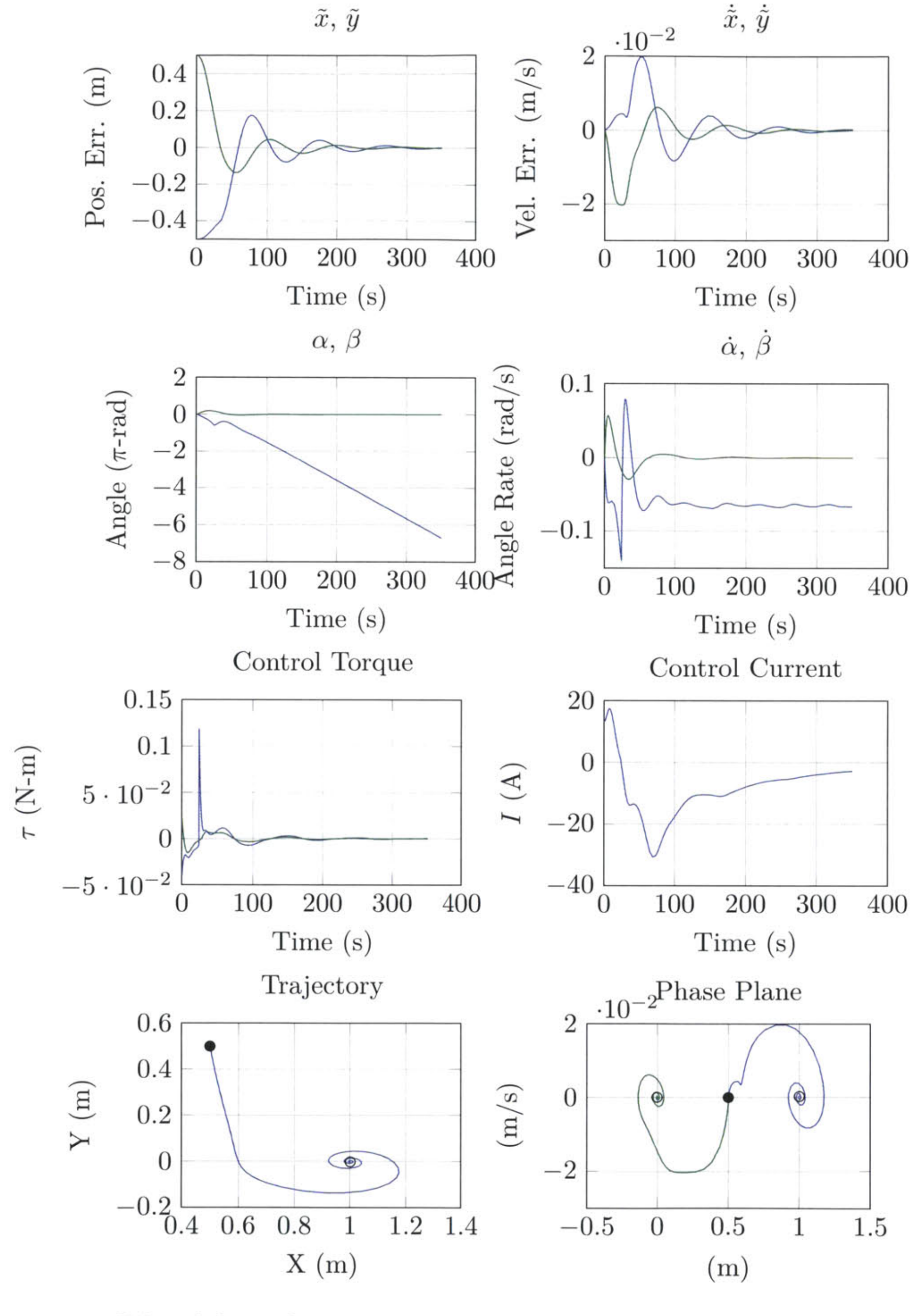


Figure 4-8: Unbounded control inputs with a slow angle controller. The performance performance is now rather poor, however, importantly, the system is still contracting (as expected). Start:● End:○.

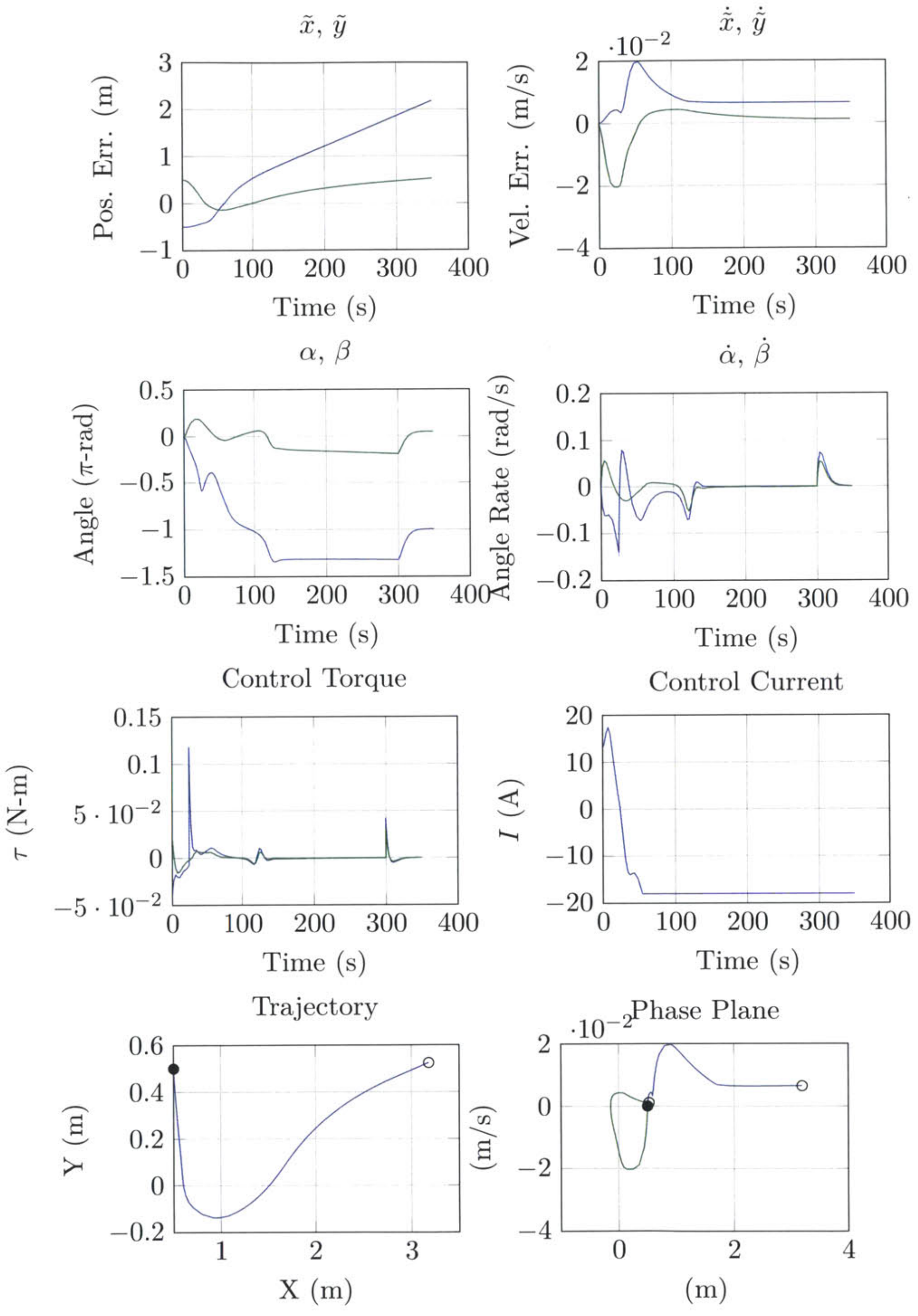


Figure 4-9: Bounded control inputs with a slow angle controller. The performance performance is now rather poor, in fact so poor the system appears to not converge. Start:● End:○.

Chapter 5

Underactuated Path Planning with Optimal Control

Space systems design is often about doing more with less. In such an environment optimization is a wonderful tool to squeeze out every bit of performance with the smallest cost. To that end, optimal control theory is applied to the underactuated electromagnetic formation flight system.

5.1 Optimal Control

The Nonlinear Optimal Control Problem (NLOCP) is a problem that asks to find the state-control trajectory, $[\mathbf{x}(\cdot), \mathbf{u}(\cdot)]$, and optionally the terminal conditions, $[\mathbf{x}_f, t_f]$, that minimize the Bolza cost functional

$$J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t] = E(\mathbf{x}_f, t_f) + \int_{t_0}^{t_f} L(\mathbf{x}, \mathbf{u}, t)\, dt \tag{5.1}$$

subject to dynamic constraints

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \tag{5.2}$$

that steer the system from initial conditions $\mathbf{x}_0$ and t_0 to the terminal manifold

$$\mathbf{e}(\mathbf{x}_f, t_f) = 0 \tag{5.3}$$

The cost function from Equation (5.1) on the preceding page is comprised of a Meyer cost (Endpoint cost) and a Lagrange cost, which is integrated across the timespan of interest[20].

A Primer on Pontryagin's Principle in Optimal Control, Ross 2009

This process involves finding the cost functional inputs, $[\boldsymbol{x}(\cdot), \boldsymbol{u}(\cdot)]$, such that small perturbations in their values, $[\boldsymbol{\delta x}(\cdot), \boldsymbol{\delta u}(\cdot)]$ only ever cause a positive perturbation in the cost functional, $\delta J > 0$. This is described as finding the inputs that drive the first *variation* of the cost to 0, conceptually very similar to finding maxima and minima in standard calculus.

Simply differentiating J is not enough because the minimal value of J does not necessarily satisfy the dynamic or boundary constraints. In order to enforce the constraints they must be "added" in to the cost function[21]. This modification must still preserve the value of the cost function, so the constraints are added in a value neutral way. Express each constraint function as an expression equal to zero i.e. $\dot{\mathbf{x}} - \mathbf{f}(\mathbf{x}, \mathbf{u}, t) = 0$ rather than $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$. Then *augment* the constraints from Equation (5.2) on the previous page and Equation (5.3) to the cost function from Equation (5.1) on the previous page using Lagrange multipliers.

Applied Optimal Control: Optimization, Estimation, and Control, Bryson and Ho 1975

$$J_a[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t] = E(\mathbf{x}_f, t_f) + \int_{t_0}^{t_f} \left[L(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^\intercal (\mathbf{f}(\mathbf{x}, \mathbf{u}, t) - \dot{\mathbf{x}})\right] dt + \boldsymbol{\nu}^\intercal \mathbf{x}_f \tag{5.4}$$

This augmented Lagrangian explicitly enforces the constraints. Part of the integrand shows up together often enough that it has earned itself its own name: the Hamiltonian, defined in Equation (5.5).

$$\mathcal{H}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}) = L(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^\intercal \mathbf{f} \tag{5.5}$$

With the constraints baked into the cost function so to speak, the result from setting $\delta J_a = 0$ will satisfy the constraints. The process of actually taking derivatives is fairly mundane though does require integration by parts. Once finished the result is a set of *necessary conditions* (differential/algebraic equations shown below) and a set of*boundary conditions* (not shown for brevity).

$$\begin{aligned} \dot{\mathbf{x}} &= \mathcal{H}_{\boldsymbol{\lambda}}^{\mathsf{T}} = \mathbf{f} \\ -\dot{\boldsymbol{\lambda}} &= \mathcal{H}_{\mathbf{x}}^{\mathsf{T}} \\ 0 &= \mathcal{H}_{\mathbf{u}} \end{aligned} \tag{5.6}$$

Equation (5.6) forms a set of differential equations with an equal number of boundary conditions whose solutions $\mathbf{x}$, $\mathbf{u}$, and $\boldsymbol{\lambda}$ satisfy the necessary conditions for optimality. The problem is that they are mixed inital and terminal conditions thus the solution cannot simply be integrated forward in time. This is where the NLOCP transforms into a numerical problem solvable in a variety of ways. The method that has received much attention of late is the pseudospectral direct transcription method.

5.1.1 Numerical Solution: Pseudospectral Method

The pseudospectral method of solving the NLOCP involves transcribing the problem into a massive sparse nonlinear program (NLP). Rao and Patterson have created a very powerful and user-friendly program that operates in the MATLAB computing environment called $\mathbb{GPOPS} - \mathbb{II}$ for the exact purpoes of interfacing a NLOC problem statement with a NLP solver such as IPOPT or SNOPT[22,23]. $\mathbb{GPOPS} - \mathbb{II}$ was designed so that it could be used "blindly" however a solid understanding of the principles of optimal control help significantly in properly posing the right problem with the right constraints.

$\mathbb{GPOPS}$: General Pseudospectral Optimal Control Solver

"SNOPT: An SQP algorithm for large-scale constrained optimization", Gill et al. 1997

"Line Search Filter Methods for Nonlinear Programming: Local Convergence", Wächter and Biegler 2005

The user is required to supply: the requisite parameters for the problem being posed, such as initial/final state bounds, path constraints, control bounds; the dynamic constraint function; the cost function; linkage information if the problem spans multiple phases; NLP options configuration options; and various other assorted options. For a full reference on how to set-up and use $\mathbb{GPOPS} - \mathbb{II}$ refer to the user-guide. The source code for one of the the spin-up trajectory is included in Appendix C on page 111.

5.2 Applied to Underactuated EMFF Systems

5.2.1 State Representation

As done in previous sections, the system will be described with 2-dimensional dynamics, given that any initial and starting point define a plane. As opposed to the state representation in Chapter 4 on page 45, it is more convinient to represent position in polar co-ordinates rather than cartesian, in particular when dealing with minimum range constraints as is done in Section 5.3.4 on page 81 the constraint is only on a single state, r rather than a combination of x and y. Restricting the state to two dimensions *does* rule out a class of maneuvers that are inheritly 3-dimensional. The fully 3-dimensional state is 18 elements long, (3 positions, 3 position rates, 6 angles and 6 angle rates). Finding the optimal solutions to an 18 element state-vector is numerically very challenging. Reducing the number of state variables is one way to simplify the problem and make it more numerically tractible. For any planar maneuver (of which there are many) many of these states are irrelevant so there is no loss of generality in reducing to a 2-dimensional representation.

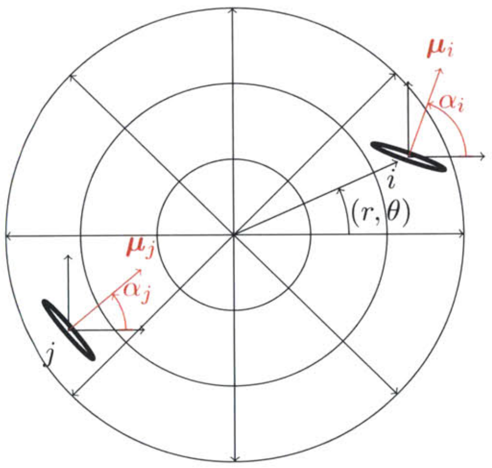


Figure 5-1: Visual definition of the relevant angles and unit vectors for the polar 2-dimensional system. The whole system is symmetric around the origin due to conservation of momentum. The direction of force θ is defined by α_i, α_j and $\hat{\boldsymbol{r}}_{ij}$.

In this form the state vector is $\mathbf{x} = \left[\theta, r, \alpha_i, \alpha_j, \dot{\theta}, \dot{r}, \dot{\alpha}_i, \dot{\alpha}_j\right]^\mathsf{T}$. As the $[r, \theta]$ coordinate frame is a non-inertial frame care must be taken to properly account for the ficticious forces that arise from a non-inertial reference frame. These forces are the coriolis force, the centrifugal force and the euler force.

$$\ddot{\mathbf{r}}_{coriolis} = -2\boldsymbol{\Omega} \times \mathbf{v} \tag{5.7}$$

$$\ddot{\mathbf{r}}_{centrifugal} = -\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{r}) \tag{5.8}$$

$$\ddot{\mathbf{r}}_{euler} = -\frac{d\Omega}{dt} \times \mathbf{r} \tag{5.9}$$

The rotating reference frame is chosen so that state constraints and boundary conditions can be more easily stated. For example if targetting a final state where the satellites are "orbiting" their center of mass, in cartesian coordinates this would be a complicated expression coupling the x and y states together. In polar coordinates it is nicely separated with a fixed target radius r and a fixed target angular rate $\dot{\theta}$.

5.2.2 Cost Metrics

Two different cost functions will be evaluated for each maneuver: minimum-time and minimum-energy-fixed-time. Speaking generally these are the most interesting cost metrics. A possible third interesting metric would be a LQR type cost that is a quadratic state error and quadratic control effort cost function. This would produce faster results than the minimum-energy solution but without being a bang-bang control like the minimum-time solution. This remains a possibility for future work.

Minimum Time: Problem Formulation

The minimum time problem asks to step from an initial state to a final state in the minimum time with bounded states and control inputs. This makes the cost function very simple, simply minimize final time.

$$\left\{ \begin{array}{ll} \text{Minimize:} & J\left[t_f\right] = t_f \\ \text{Subject to:} & \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \\ & t_0 = 0 \\ & \mathbf{x}(t_0) = \mathbf{x}^{\mathbf{0}} \\ & \mathbf{e}(\mathbf{x}(t_f)) = 0 \end{array} \right. \tag{5.10}$$

Minimum Energy Fixed Time: Problem Formulation

The minimum energy problem asks to perform a maneuver, stepping from an initial state to a final state with bounded states and control inputs within a fixed length of time and to use the least amount of energy possible. Energy is the time integral of power ($E = \int P$) and power is $\propto I^2$ so using a quadratic control integral cost is appropriate.

$$\begin{cases} \text{Minimize:} & J\left[t_0, t_f, \mathbf{u}(\cdot)\right] = \int_{t_0}^{t_f} \mathbf{u}^{\mathsf{T}} \mathbf{R}_{\mathbf{uu}} \mathbf{u} dt \\ \text{Subject to:} & \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \\ & (t_0, t_f) = (t^0, t^f) \\ & \mathbf{x}(t_0) = \mathbf{x}^{\mathbf{0}} \\ & \mathbf{e}(t_f, \mathbf{x}(t_f)) = 0 \end{cases} \tag{5.11}$$

Where $\mathbf{R}_{\mathbf{uu}}$ is the control weighting matrix, this allows for trading the relative cost between the various control inputs. In this case that is trading the cost of torquing against the cost of electrical current.

5.3 Path Planning Trajectories

A variety of maneuvers (Table 5.1 on the next page) of increasing difficulty have been evaluated. The first is the simplest, numerically easiest and also should produce predictable results, so it is a good maneuver to start with to ensure the optimization software is working properly. From there the trajectories move into two dimensional step. The primary increase in complexity is the system must now handle angular rotations of each satellite. Next an obstacle is added such that the solution must route around a "keep-out-zone". Finally the spinup maneuver is evaluated. The spinup and orbit are two maneuvers that EMFF excells at it they requires continual force application to maintain a circular trajectory. Such a maneuver performed with convetional thrusters would be very mass costly.

For all maneuvers the control bounds are the same: $||[\tau_i, \tau_j, I^2]|| <$ [0.02mNm, 0.02mNm, 225A]. These limits were chosen due to their similarity to the RINGS hardware testbed. This choice is arbitrary but may

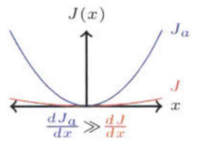
Maneuver	Description
Axial Step	1-dimensional step
Formation Slew	2-dimensional step
Collision Avoidance	2-dimensional step with obstruction
Formation Spin-up	2-dimensional dynamic target

Table 5.1: List of optimal control maneuvers

facilitate eventual hardware testing of this control principle.

5.3.1 Nuances of optimal trajectory solutions

While the cost functions described in Equation (5.10) on page 71 and Equation (5.11) on the preceding page are conceptually the "minimum time" and "minimum energy" cost functions, when implemented numerically they sometimes yield poor results. When there is a dimension of actuation or motion that very weakly affects the cost, sometimes the NLP solver has difficulty finding the minimal trajectory along that dimension. In the results shown here this often resulted in solutions with excessive satellite rotations, sometimes revolving a full 2π radians while translating. Another case where this happens is when solving a minimal time problem, there can be moments where the control input very weakly affects the cost function so the NLP solver will find it difficult to find the time minimal choice and the resulting control output might "float" almost randomly for a short period of time as is seen in Figure 5-2 on the following page between 20 and 40 seconds. What otherwise appears to be a well formed minimum-time bang-bang control solution has a hiccup where the control input does not strongly affect the outcome for a period of time, thus the almost random control input for the duration. It *may* be possible that such a maneuver is in fact time- or energy-optimal however in practice is is undesirable when a similar trajectory with much less rotation is feasible. This gives rise to an augmented cost function by application of heuristics.



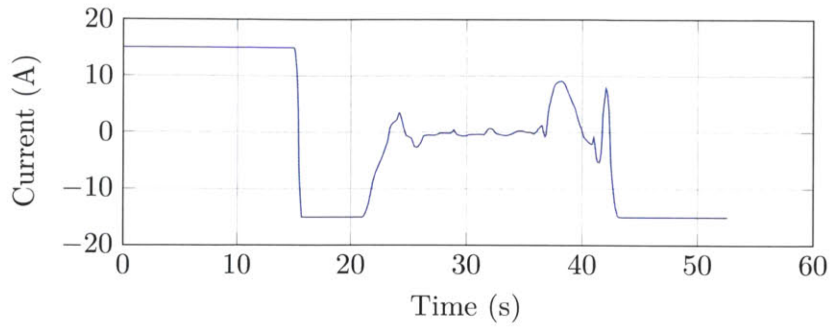


Figure 5-2: Floating control input when weakly connected to cost function

Heuristics

Sometimes the NLP solver needs soem extra help arriving at the desired solution. To that end terms can be augmented to the cost function to help shape the solution space more favorably. Take the simplest time-optimal cost function, $J = t_f$, this cost does not care if the system is rotating at 0.1, 1 or 10 $\frac{rad}{s}$ along the way so long as t_f does not change. However, given the option to choose from those options, it would be preferred (usually) to have the lower rotation rate, therefore an augmented cost term could be

$$J_a = t_f + \int_{t_0}^{t_f} k_\omega \omega^2 dt \tag{5.12}$$

which would serve to steepen the gradient in cost with ω. The tuning parameter k_ω is used to vary how strongly the solution penalizes angular rotations. If time-optimality is still the objective, k_ω should be chosen such that the augmented cost is still small compared to t_f. Experience has shown that at a 10:1 ratio still produces results of approximately the same t_f but with reduced angular rates.

This gives rise to the use of the cost function to "penalize" certain behavior. Suppose there is a certain motion that is undesirable, it isn't necessarily a good idea to completely disallow the behavior but rather penalize it in the cost function. This means its still permissable but only if its really worth it, and the gain multiplier is the knob that determines how worth it it needs to be.

A second augmentation is to slightly penalize control effort even in the time-optimal case (where control effort is not ordinarily part of cost). The purpose of this is the same as penalizing angular rates, if control effort momentarily does not map to cost function (or does so weakly) it is preferred to have no control input.

$$J_a = t_f + \int_{t_0}^{t_f} k_\omega \omega^2 + \mathbf{u}^\mathsf{T} \mathbf{R}^*_{\mathbf{uu}} \mathbf{u} dt \tag{5.13}$$

This addition enforces that even if $\frac{dJ}{d\mathbf{u}} = 0$ that $\frac{dJ_a}{d\mathbf{u}} > 0$ for *p.d.* weighting matrix $\mathbf{R}^*_{\mathbf{uu}} > 0$. The use of (*) in this case is simply to differentiate this matrix from the weighting matrix used in minimum energy maneuvers.

5.3.2 Axial Step

Minimum Time

$$\begin{cases} \text{Minimize:} & J[t_f] = t_f \\ \text{Subject to:} & \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \\ & t_0 = 0 \\ & \mathbf{x}_0^\mathsf{T} = [\frac{\pi}{2}, 0.7, 0, 0, 0, 0, 0, 0]^\mathsf{T} \\ & \mathbf{x}_f^\mathsf{T} = [\frac{\pi}{2}, 0.4, 0, 0, 0, 0, 0, 0]^\mathsf{T} \end{cases} \tag{5.14}$$

The minimum time axial step should produce a bang-bang type control. The actuators accelerate inwards at maximum force until the switching point where they reverse and decelerate at max effort, coming to a stop right at the target position, shown in Figure 5-3 on page 77.

Minimum Energy

$$\begin{cases} \text{Minimize:} & J[t_0, t_f, \mathbf{u}(\cdot)] = \int_{t_0}^{t_f} \mathbf{u}^\mathsf{T} \mathbf{R}_{\mathbf{uu}} \mathbf{u} dt \\ \text{Subject to:} & \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \\ & (t_0, t_f) = (0, 60) \\ & \mathbf{x}_0^\mathsf{T} = [\frac{\pi}{2}, 0.7, 0, 0, 0, 0, 0, 0]^\mathsf{T} \\ & \mathbf{x}_f^\mathsf{T} = [\frac{\pi}{2}, 0.4, 0, 0, 0, 0, 0, 0]^\mathsf{T} \end{cases} \tag{5.15}$$

The minimum energy solution was given 60 seconds to perform the same maneuver that the minimum time solution performed in 20 seconds. Because the cost is quadratic with current ($P \propto I^2$) the cost favors constant low current rather than large short spikes of high current. This is seen in the roughly constant but low input in Figure 5-4 on page 78.

5.3.3 Formation Slew

The formation slew maneuver aims to rotate the entire formation relative to inertial space. The inital and final orientations are unconstrained and the final rates are enforced as zero.

Minimum Time

The minimum time slew maneuver demonstrates the benefit of using optimization when working with non-linear systems. The target state is a pure rotation of the formation with no change in range, however it is more expedient to first decrease the range, whereby there is a larger force available. Thus the resulting trajectory shown in Figure 5-9 on page 84 and in detail in Figure 5-6 on page 80 is an inward curving arc. There are some numerical artifacts in the control input that are likely not part of the true optimal solution. More time spent massaging the NLP solver could help resolve small issues such as that. In absence of an improved result, the control profile could be filtered or smoothed to remove the spurious spikes and still retain a near optimal solution.

$$\begin{cases} \text{Minimize:} & J[t_f] = t_f \\ \text{Subject to:} & \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \\ & t_0 = 0 \\ & \mathbf{x}_0^{\mathsf{T}} = [-\frac{\pi}{10}, 0.5, \cdot, \cdot, 0, 0, 0, 0]^{\mathsf{T}} \\ & \mathbf{x}_f^{\mathsf{T}} = [+\frac{\pi}{10}, 0.5, \cdot, \cdot, 0, 0, 0, 0]^{\mathsf{T}} \\ & [-\pi, -\pi]^{\mathsf{T}} \leq [\alpha_i, \alpha_j]_0^{\mathsf{T}} \leq [\pi, \pi]^{\mathsf{T}} \\ & [-\pi, -\pi]^{\mathsf{T}} \leq [\alpha_i, \alpha_j]_f^{\mathsf{T}} \leq [\pi, \pi]^{\mathsf{T}} \end{cases} \tag{5.16}$$

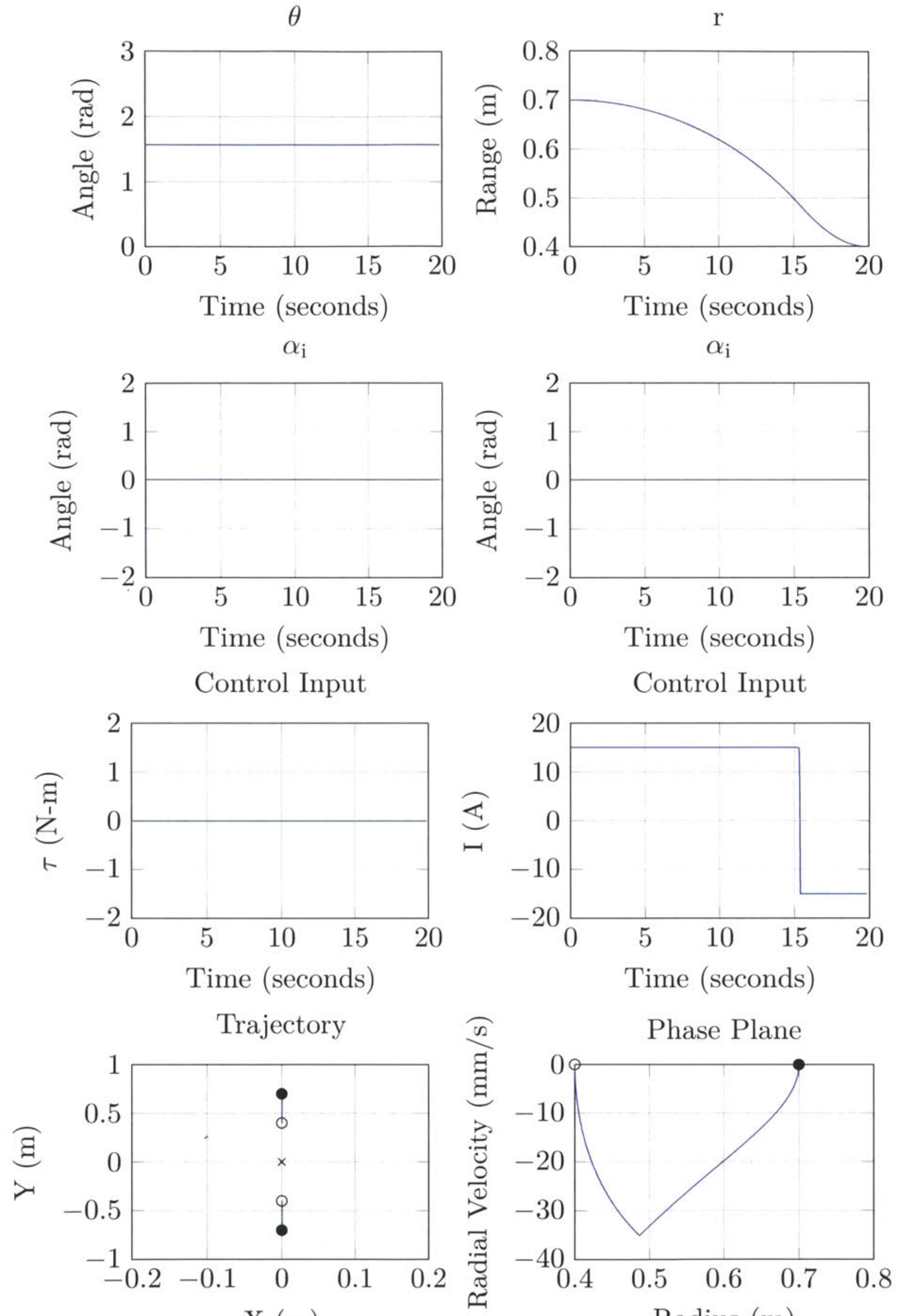


Figure 5-3: Minimum time bang-bang control for axial step maneuver. The nonlinearity of the force model is visible in the shape of the phase-plane. Note: Filled circle: start point. Empty circle: end point.

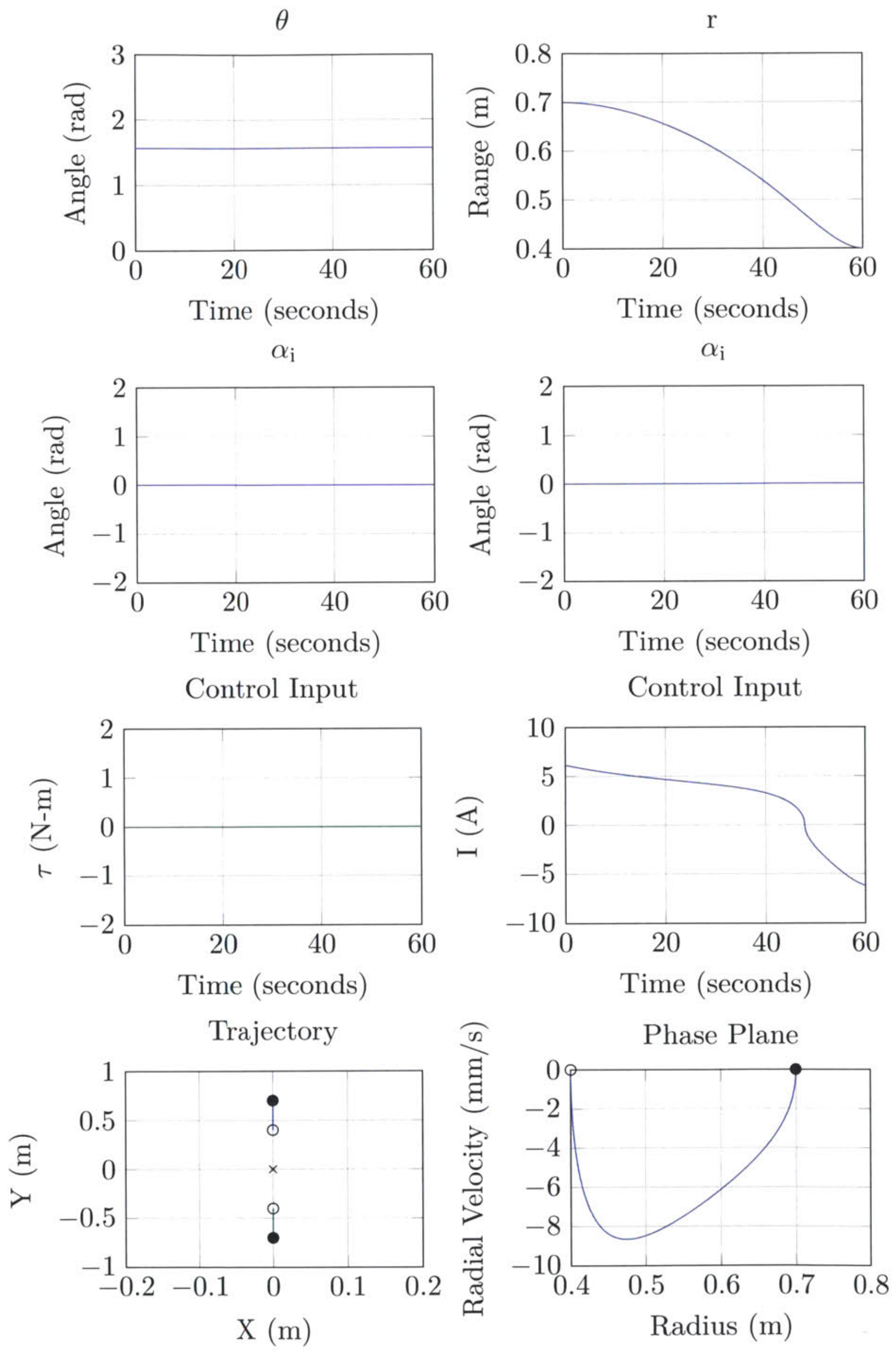


Figure 5-4: Minimum energy control for axial step maneuver. Note: Filled circle: start point. Empty circle: end point.

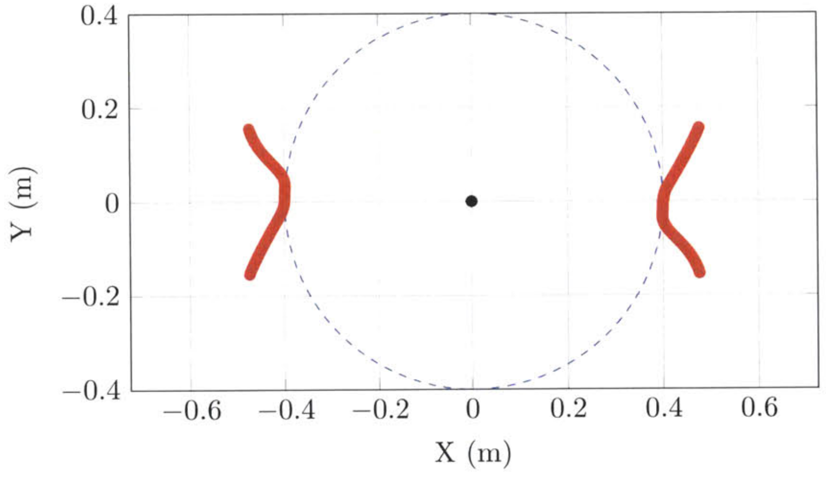


Figure 5-5: Minimum time slew maneuver trajectory. Trajectory started at $\theta = -\frac{\pi}{10}$ radians and slewed counterclockwise to $\theta = \frac{\pi}{10}$. That is a 36 degree slew. The dashed line is the lower bound on the radius r.

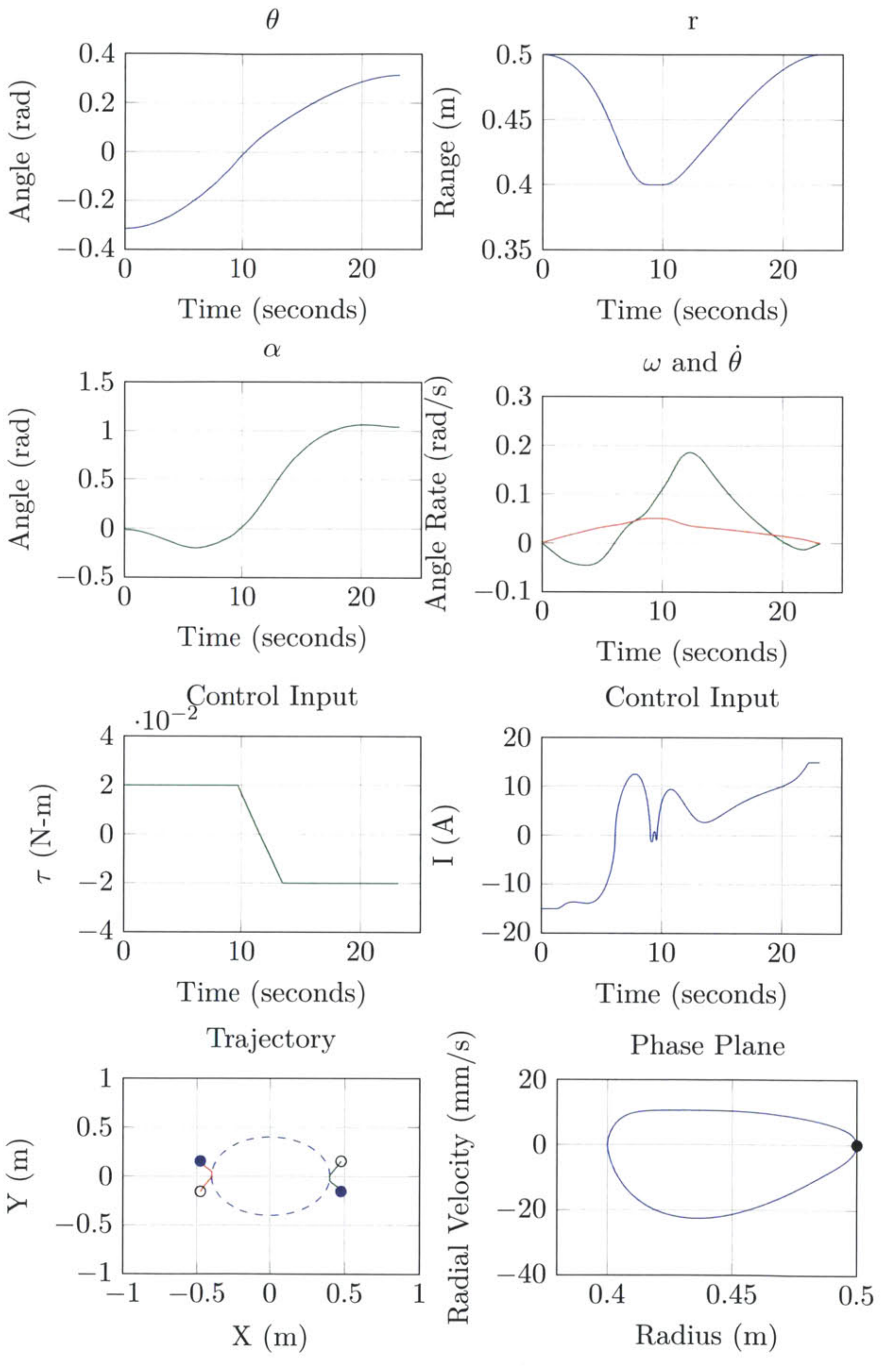


Figure 5-6: Minimum time control for slew maneuver. Note: Filled circle: start point. Empty circle: end point.

Minimum Energy

$$\begin{cases} \text{Minimize:} & J\left[t_0, t_f, \mathbf{u}\right] = \int_{t_0}^{t_f} \mathbf{u}^\mathsf{T}\mathbf{R}_{\mathbf{uu}}\mathbf{u}dt \\ \text{Subject to:} & \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \\ & (t_0, t_f) = (0, 60) \\ & \mathbf{x}_0^\mathsf{T} = [-\frac{\pi}{10}, 0.5, \cdot, \cdot, 0, 0, 0, 0]^\mathsf{T} \\ & \mathbf{x}_f^\mathsf{T} = [+\frac{\pi}{10}, 0.5, \cdot, \cdot, 0, 0, 0, 0]^\mathsf{T} \\ & [-\pi, -\pi]^\mathsf{T} \le [\alpha_i, \alpha_j]_0^\mathsf{T} \le [\pi, \pi]^\mathsf{T} \\ & [-\pi, -\pi]^\mathsf{T} \le [\alpha_i, \alpha_j]_f^\mathsf{T} \le [\pi, \pi]^\mathsf{T} \end{cases} \quad (5.17)$$

The minimum energy slew maneuver (Figure 5-7 on the next page) should produce a result visually similar to the minimum time. There is less of an advantage to decreasing range but it remains because for a given force magnitude the required current I can be lower, thus reducing energy cost, so the trajectory arc inwards, possibly running into the lower range bound.

5.3.4 Avoidance Maneuver

All of the previous maneuvers could have been accomplished (to some level of similar performance) by the linear controller described in Chapter 4 on page 45. The avoidance maneuver however is not directly achievable through the feedback control law. This maneuver is defined by its use of a "keep-out-zone". The keep-out-zone acts as a collision-avoidance-zone for a two-satellite configuration. This allows for safely reconfiguring a formation where the straight-line path to the final state would cause the spacecraft to collide. As needed more complicated state path constraints can be formulated. Due to the state representation the lower

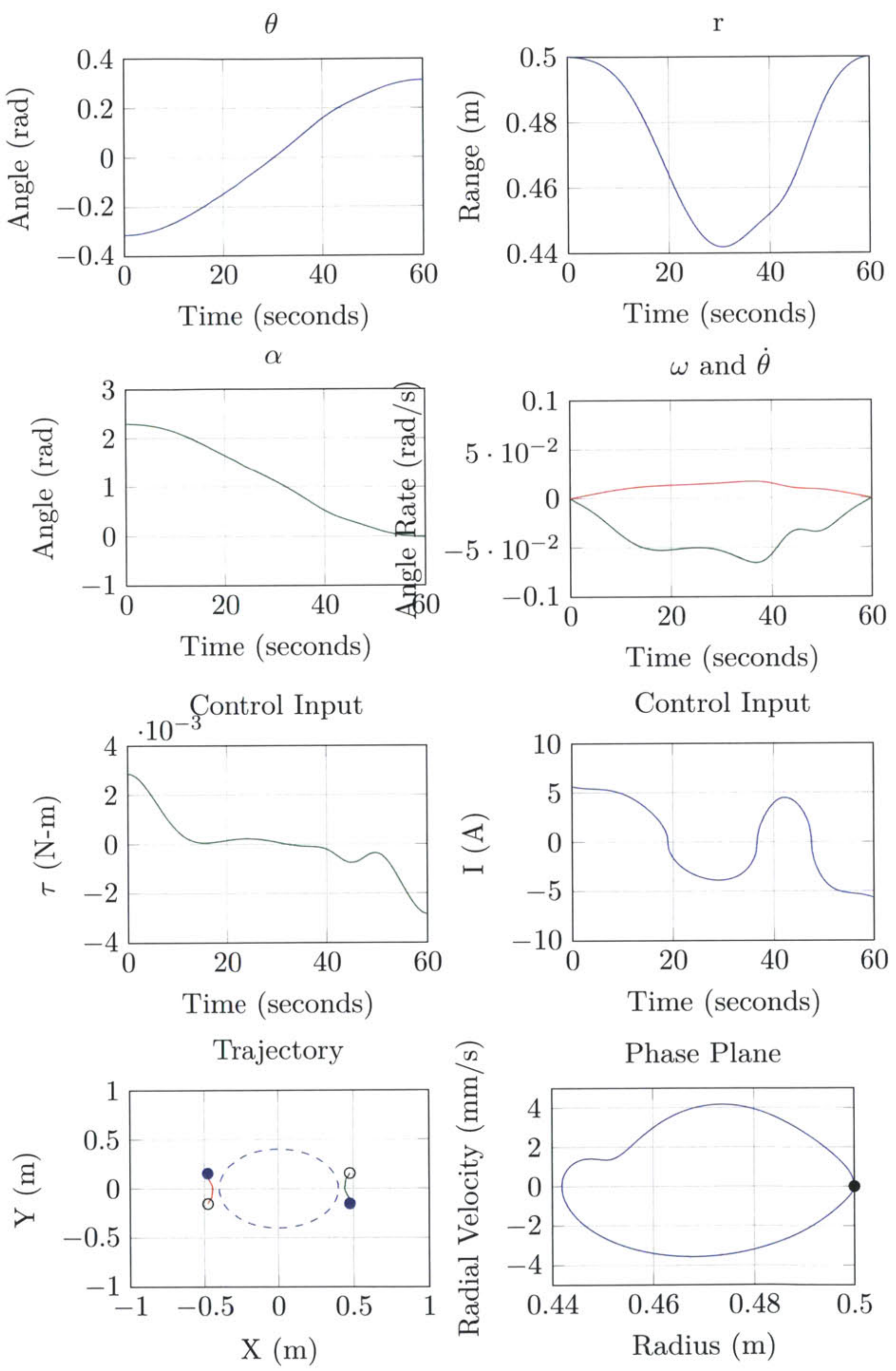


Figure 5-7: Minimum energy control for slew maneuver. Note: Filled circle: start point. Empty circle: end point.

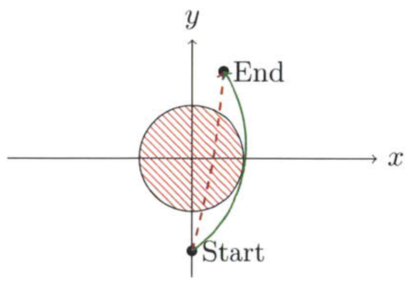


Figure 5-8: Red "keep-out-zone". Set as a 40cm radius ball about the origin within which the satellites cannot go. The direct path shown above is not allowed This means the closest approach is 80cm or just under 3 coil radii.

bounded range constraint ends up being very simple to enforce.

$$\begin{cases} \text{Minimize:} & J[t_f] = t_f \\ \text{Subject to:} & \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \\ & t_0 = 0 \\ & \mathbf{x}_0^\mathsf{T} = [-\frac{\pi}{10}, 0.5, \cdot, \cdot, 0, 0, 0, 0]^\mathsf{T} \\ & \mathbf{x}_f^\mathsf{T} = [+\frac{\pi}{10}, 0.5, \cdot, \cdot, 0, 0, 0, 0]^\mathsf{T} \\ & [-\pi, -\pi]^\mathsf{T} \leq [\alpha_i, \alpha_j]_0^\mathsf{T} \leq [\pi, \pi]^\mathsf{T} \\ & [-\pi, -\pi]^\mathsf{T} \leq [\alpha_i, \alpha_j]_f^\mathsf{T} \leq [\pi, \pi]^\mathsf{T} \\ & r > 0.4 \end{cases} \tag{5.18}$$

The minimum energy problem statment is excluded for brevity. It follows the same pattern as the axial and slew minimum energy maneuvers but with the modifications specific to the collision avoidance case.

Minimum Time

The minimum time trajectory (Figure 5-9 on the next page) should get as close as possible to have access to the largest forces then orbit around at the minimum distance until close to the target where it will detatch from the range lower bound and proceed to the target. The full detail is shown in Figure 5-10 on page 85. This maneuver was found by using the penalty augmented cost function in Equation (5.13) on page 75.

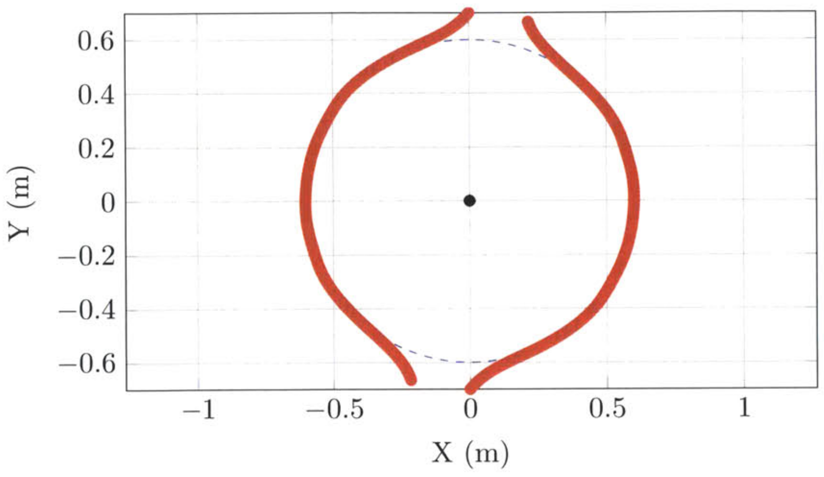


Figure 5-9: Minimum collision avoidance maneuver trajectory.

Minimum Energy

The minimum energy trajectory is expected to behave similarly to the slew maneuver with respect its minimum time trajectory. The path will arc around the minimum bound. The trajectory looks very similar to the minimum time path except the duration is much longer, indicative of the slower pace for the minimum energy solution. Other than that the two are in fact very similar. If the lower range bound was very low it is possible for the minimum energy trajectory to only run against the bound for a short time if at all, however for such a large minimum bound the energy minimal trajectory rides the minimal range bound.

5.3.5 Spin-up Maneuver

This is perhaps the most interesting maneuver as it is directly applicable to distributed telescope systems and other interferometry missions. It is also one of the more numerically challenging. The spin-up maneuver is where the system begins at rest and ends orbiting around the center of mass. Therefore the final state is not a stationary state (constant position/angle), but rather has constant rates. Note the trajectory presented here is only the *spinup* not the orbit. This trajectory ends once the correct terminal conditions are satisfied such that orbiting could proceed.

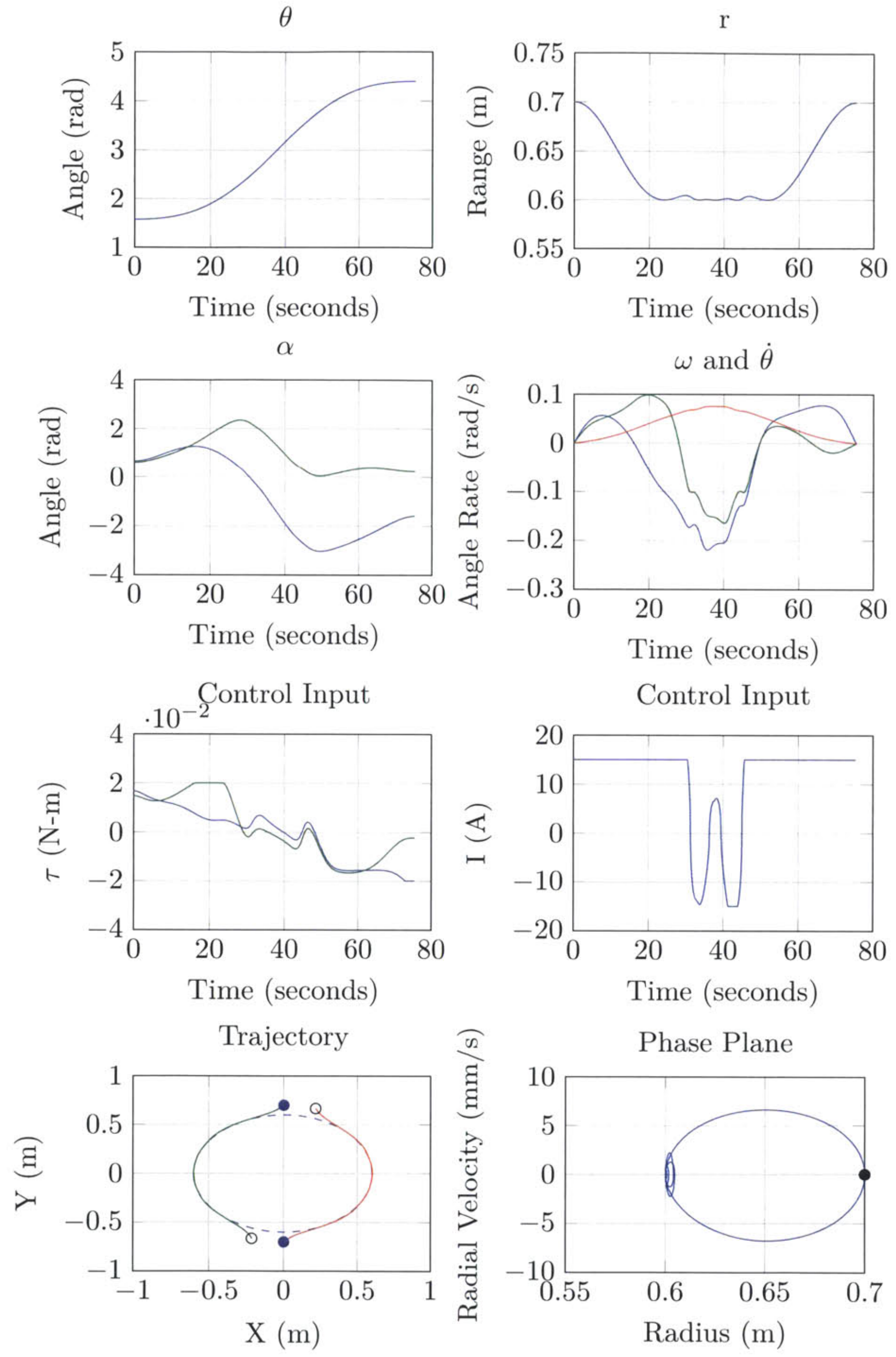


Figure 5-10: Minimum time control for the collision avoidance maneuver. Note: Filled circle: start point. Empty circle: end point.

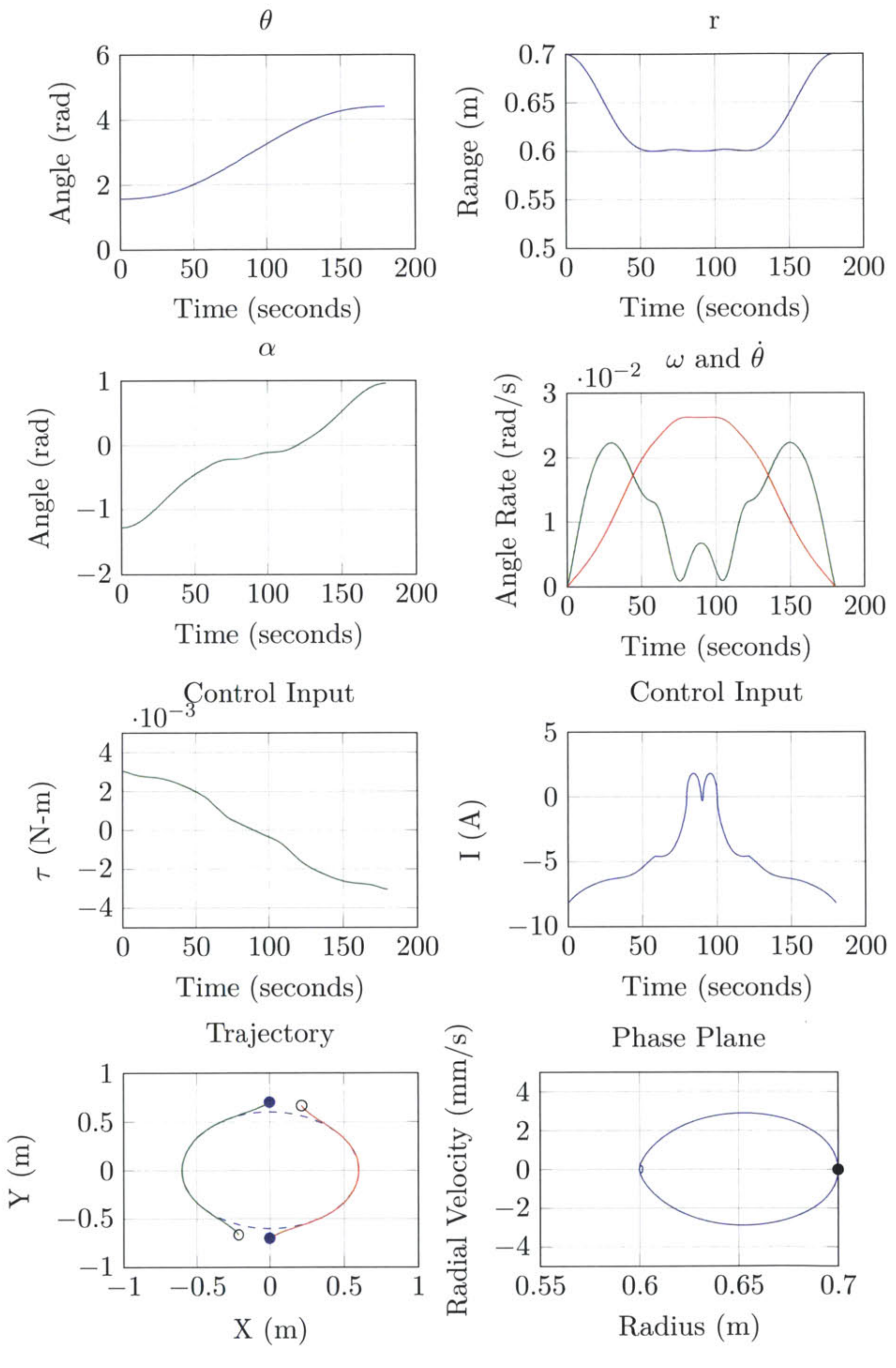


Figure 5-11: Minimum energy control for the collision avoidance maneuver. Note: Filled circle: start point. Empty circle: end point.

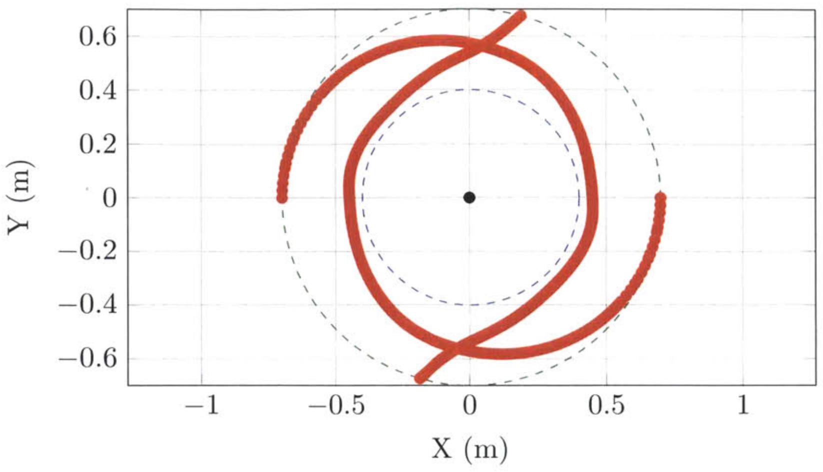


Figure 5-12: Minimum time spinup maneuver, starting from rest along the circle perimiter, begin orbiting in minimum time. The final state is fixed at $\theta = 0$ and the inital state is free. The inner dashed circle is the lower state bound on radius and the outer circle is the targetted radius for the spinning formation.

Minimum Time

As is typically the case for minimum time maneuvers, the control inputs (at least in electrical current) are fully saturated, shown in Figure 5-15 on page 90. Furthermore the system steps inwards to decrease the range temporarily and increase the available force so it can move faster (Figure 5-14 on page 89). What is most interesting is that the system first decreases range to have access to higher forces then it increases its angular rate very quickly at which point it lets centrifugal acceleration pull the satellite out to the desired radius while letting the angular rate slow down as the radius increases.

Minimum Energy

Similar to the collision avoidance minimal energy trajectory the minimum energy solution (Figure 5-14 on page 89) is similar in appearance to the time minimal solution. The primary difference is the whole process is slowed down in the energy minimal path, thus as the rotational dynamics take hold there is a slightly different trajectory shape than the time minimal case.

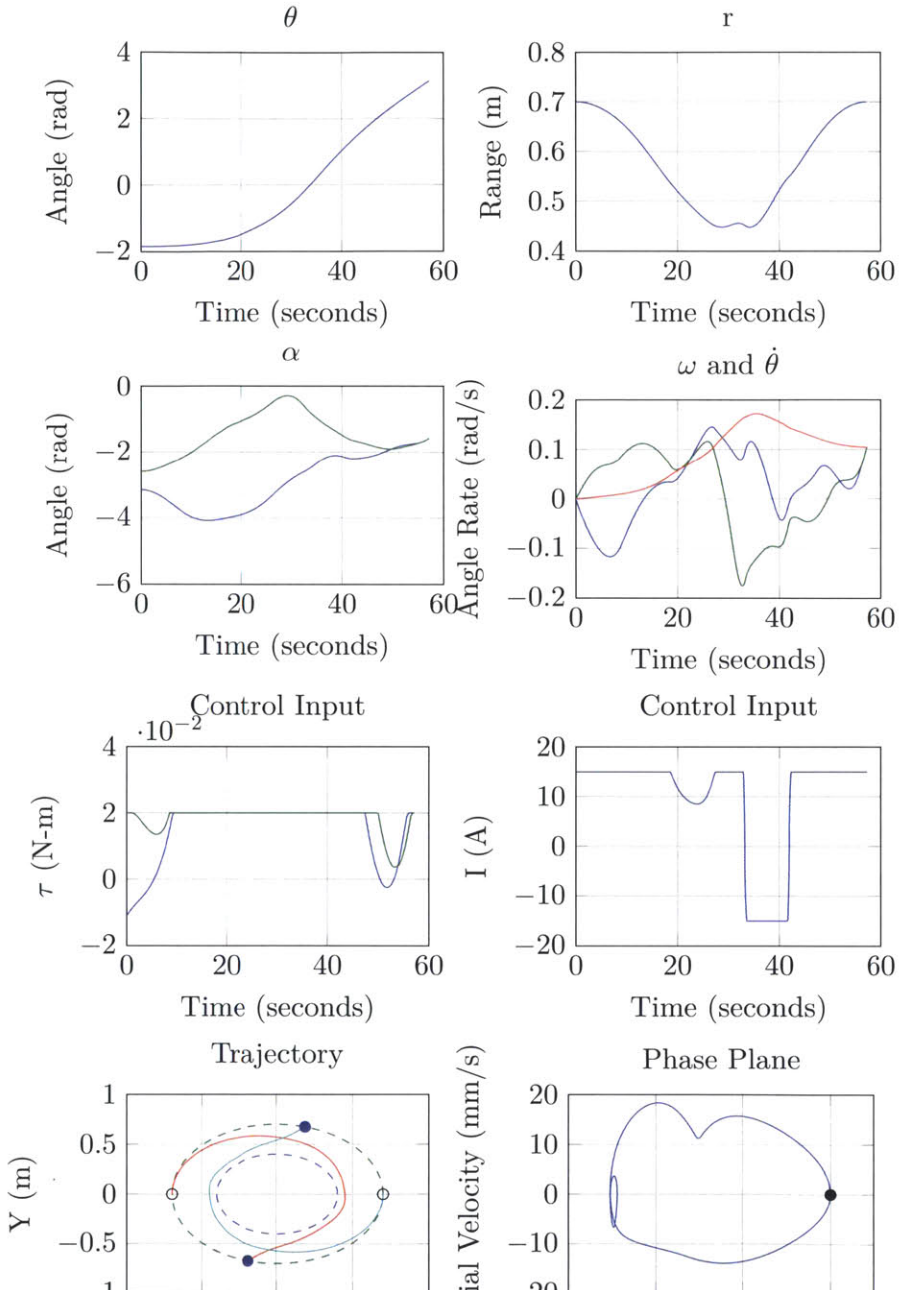


Figure 5-13: Minimum time control for spinup maneuver. Note: Filled circle: start point. Empty circle: end point.

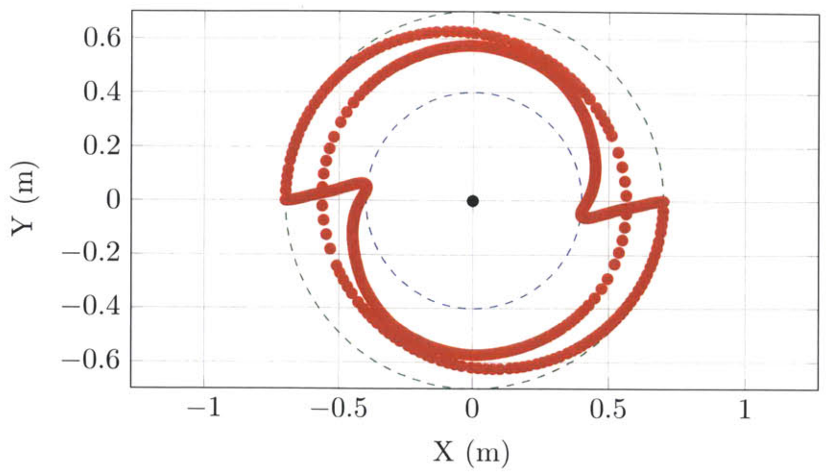


Figure 5-14: Minimum time spinup maneuver, starting from rest along the circle perimiter, begin orbiting in minimum time. The final state is fixed at $\theta = 0$ and the inital state is free.

5.4 Summary

Despite being underactuated this class of system is able to satisfy time and energy minimal trajectories of interest. Solutions were presented for this set of interesting trajectories where the physical parameters of the simulated object are similar to those of the RINGS hardware unit. These solutions demonstrate the ability to generate both feasible and optimal trajectories with the underactuated system for a variety maneuvers. The solutions, as contrasted with the resulting trajectories in Chapter 4, take advantage of the nonlinearities in the dynamics. These trajectories are targetted at both static and dynamic targets leaving the door open for more sophisticated planar maneuvers with interesting and time varying constraints. Extending this result into 3-dimensional maneuvers (such as sweeping out a cone with the position vector) where there either isn't a well defined plane using the method described above or the plane is itself changing and rotating in space can go forward in one of two ways. It can focus on mapping the effect of rotating the plane onto disturbances within the plane such that any trajectory can be considered "planar" albeit with non-inertial dynamics, or it can focus on streamlining the NLP solution for the expanded 18 vector 3-dimensional vector or reducing the degrees of freedom of the expanded state to aid

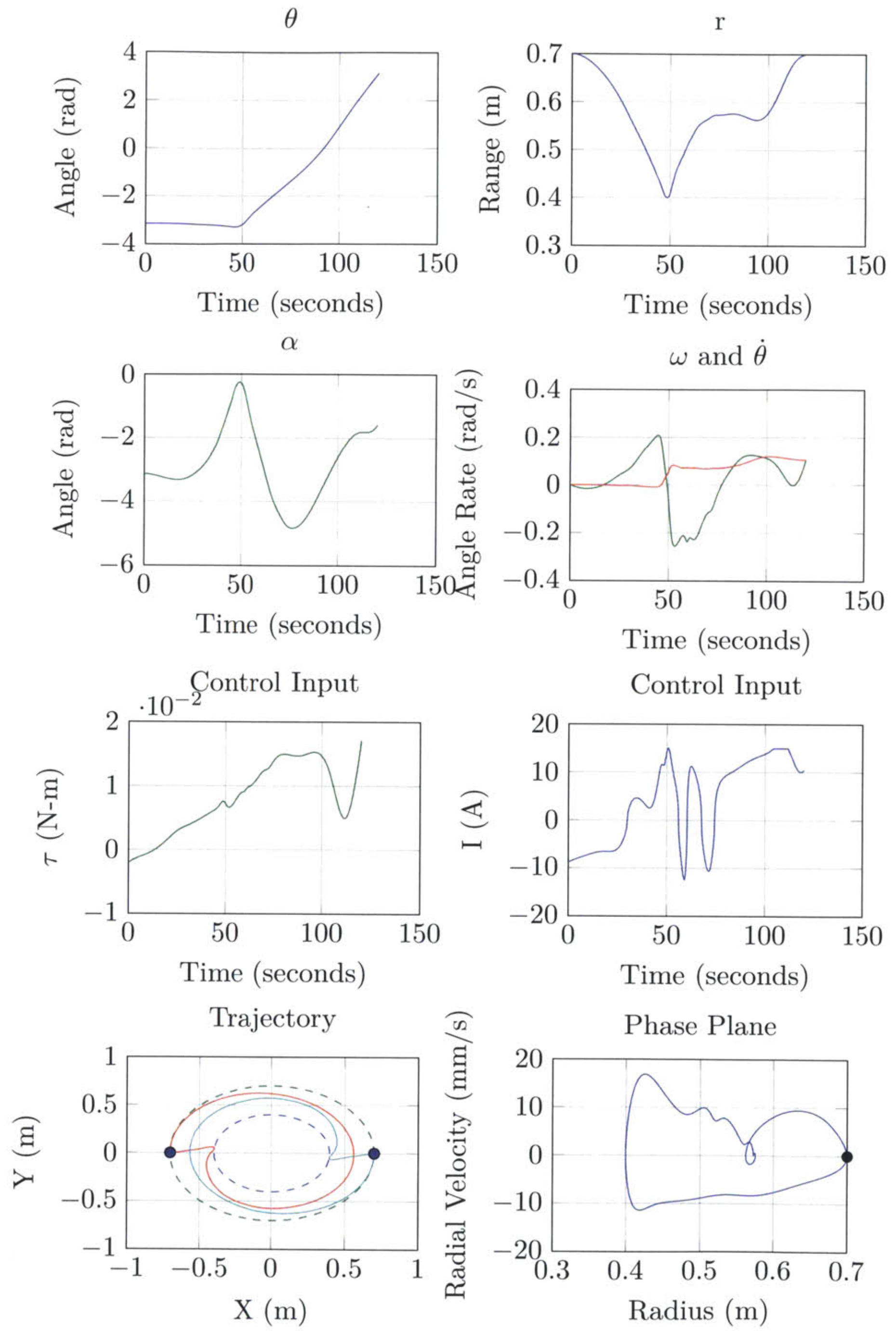


Figure 5-15: Minimum time control for spinup maneuver. Note: Filled circle: start point. Empty circle: end point.

Chapter 6

Conclusions

Electromagnetic formation flight is becoming reality with the RINGS hardware testbed launching to the International Space Station in late summer 2013. In any real EMFF system it is ideal to have three fully actuated magnetic actuators, however mass, power, volume or cost constraints might lead to a loss of one or more actuators. This thesis has shown that depending on the mission attitude control requirements an underactuated system may not be problematic.

6.1 Models & Validation

The thesis began by discussing the prior electromagnetic force models developed and reproduces the results in the near-field. Importantly for when using a near-field model, a limit is found to how discritized the model needs to be before losing numerical precision as a function of separation distance. Further, the same process as was used to find the machine precision limit can be used to find the discretization limit for any arbitrary required numerical precision up to machine precision. This allows for trading computational run-time performance for numerical precision depending on what the requirement is.

Chapter 3 attempts to verify and validate the near-field model. The near-field model developed for this thesis agrees with the far-field model outside of 6.67 radii and diverges in the expected manner in the near-field. Validation is conducted experimentally with data collected from step responses taken in a very short duration 6DOF environment, the reduced gravity aircraft. The process uses the reconstructed trajectory based on ultrasound beacon ranging measurments as the state input for the electromagnetic force model. The modelled force is then compared to the measured linear and angular accelerations. The rate-gyro angular rate sensors are far superior to the on-board accelerometers in noise characteristics, however for many of the zero-g tests the rate-gyro data is very sparesely populated rendering it largely unusable quantitatively. The reconstructed trajectory did not readily generate results in alignment with the measured linear acceleration data and no reasonable explanation for this is presented. The process as described should be sufficient with a more complete data set.

6.2 Underactuated Closed Loop Position Control

Chapter 4 first looks at what it would take to control a fully actuated system. Then it describes how the system in question is not fully actuated nor does it satisfy linear controllability. Next it applies contraction analysis theory to produce a dipole-steering feedback control law that stabilizes the position exponentially. This control law is then simulated and evaluated within the near-field region, the contol method uses a dynamic inversion based on the far-field model and the system remained stable. This proves the ability of a single actuator system to achieve position control. The underactuated system is not capable of simultaneous arbitrary attitude and position control due to the nature of the position control. This is proven by noting the force direction is uniquely defined by the spacecraft attitudes, therefore at any moment in time only disturbances along a single axis can be rejected without rotating the spacecraft.

6.3 Path Planning with Optimal Control

Chapter 5 uses the same underactuated system from Chapter 4 and applies optimal control theory with a numerical optimzation tool, $\mathbb{GPOPS} - \mathbb{II}$, and found time and energy optimal trajectories for a series of maneuvers. Modifications to the pure minimum time and minimum energy cost functions are propsed and used to improve numerical convergence to a well behaved optimal trajectory. The maneuvers tested were: axial step, lateral step (slew), a collision-avoidance maneuver, and a formation spin-up maneuver. These maneuvers were selected for their gradual increase in complexity and for their application to real scenarios. The axial and lateral step test basic position control, the collision-avoidance test demonstrates a crucial ability when operating a multi-satellite formation in close proximity and the formation spin-up test is an essential maneuver for a system such as a sparese telescope array where the satellites orbit the overall center of mass.

When it comes to performance, while the maneuvers themselves were not exactly the same across Chapter 4 and Chapter 5, a quick comparison of maneuver times reveals how much faster the optimal trajectories can be than those guided by the feedback control law. For instance the collision avoidance maneuver has the system moving roughly 1.4 meters in about 75 seconds. Even for the fast angle controller the step maneuver was 0.7 meters and it had a rise time of around 40 seconds and a settling time of 70 seconds. Furthermore the optimal avoidance maneuver could not take a straight-line path to the final state. All told the optimal trajectory outperforms the feedback controller by a factor of 2.

6.4 Future Work

Overall the thesis shows that the underactuated system is controllable however the attitude is not independently controllable as the dipole vector is driven by the attitude. In this sense the attitude of the electromagentic actuator is one of the control inputs to the position control system. With this limitation the system proved controllable. There yet remains useful work to be done pertaining to underactuated EMFF systems such as solving for the desired angular rates derived from the desired angle control based on position error. Including desired angular rates in the feedback control law will improve angle tracking performance which translates into faster position convergence.

Other topics of future work include:

- Analyzing the effect of relative rotation angle on the discritization limit.
- Collecting further (and cleaner) data for model validation.
- Deriving the desired angle *rates* $\dot{\boldsymbol{\alpha}}$ for the dipole steering position feedback control.
- Consider the effects of escape velocity on nonlinear system stability.
- Multi-satellite (N>2) underactuated planar path planning.
- Apply a LQR type cost function for numerical optimzation, i.e. $J = \int_{t_0}^{t_f} \mathbf{x}^\intercal \mathbf{R_{xx}} \mathbf{x} + \mathbf{u}^\intercal \mathbf{R_{uu}} \mathbf{u} dt$. This *should* produce results somewhere inbetween min-time and min-energy.
- Conceiving hardware solutions that decouple the payload attitude from the actuator attitude.

Final Remarks

This thesis sought to model and control a pair of underactuated electromagnetically driven satellites. The system has been modelled and is awaiting improved validation, the system has been stabilized under modest assumptions, and the system has had trajectories for complex and intesting maneuvers planned all while operating in the near-field domain. This research provides strong evidence that from the dynamics and control standpoint the underactuated electromagnetic formation flight system is a viable space system design.

Appendix A

Dynamic Inversion

Note: $C_m = \frac{3\mu_0}{4\pi}$ and $\boldsymbol{r}_{ij} = [-2x, -2y]^\intercal$ per the definition of x, y and r, see Figure 4-2 on page 47.

Starting with the far-field equation

$$\boldsymbol{F} = C_m \frac{\mu_i \mu_j}{r_{ij}^4} \Big(5\,(\hat{\boldsymbol{\mu}}_i \cdot \hat{\boldsymbol{r}}_{ij})\,(\hat{\boldsymbol{\mu}}_j \cdot \hat{\boldsymbol{r}}_{ij})\,\hat{\boldsymbol{r}}_{ij} - (\hat{\boldsymbol{\mu}}_i \cdot \hat{\boldsymbol{\mu}}_j)\,\hat{\boldsymbol{r}}_{ij} - (\hat{\boldsymbol{\mu}}_i \cdot \hat{\boldsymbol{r}}_{ij})\,\hat{\boldsymbol{\mu}}_j - (\hat{\boldsymbol{\mu}}_j \cdot \hat{\boldsymbol{r}}_{ij})\,\hat{\boldsymbol{\mu}}_i \Big) \tag{A.1}$$

Setting the dipole $\hat{\boldsymbol{\mu}}_j$ equal to $-\hat{\boldsymbol{r}}_{ij}$.

$$\frac{\boldsymbol{F} r_{ij}^4}{C_m} = \mu_i \mu_j \Big(-5\,(\hat{\boldsymbol{\mu}}_i \cdot \hat{\boldsymbol{r}}_{ij})\,(\hat{\boldsymbol{r}}_{ij} \cdot \hat{\boldsymbol{r}}_{ij})\,\hat{\boldsymbol{r}}_{ij} + (\hat{\boldsymbol{\mu}}_i \cdot \hat{\boldsymbol{r}}_{ij})\,\hat{\boldsymbol{r}}_{ij} + (\hat{\boldsymbol{\mu}}_i \cdot \hat{\boldsymbol{r}}_{ij})\,\hat{\boldsymbol{r}}_{ij} + (\hat{\boldsymbol{r}}_{ij} \cdot \hat{\boldsymbol{r}}_{ij})\,\hat{\boldsymbol{\mu}}_i \Big) \tag{A.2}$$

noting that $\hat{\boldsymbol{r}}_{ij} \cdot \hat{\boldsymbol{r}}_{ij} = 1$ and combining like terms...

$$\frac{\boldsymbol{F} r_{ij}^4}{C_m} = \mu_i \mu_j \Big(\hat{\boldsymbol{\mu}}_i - 3\,(\hat{\boldsymbol{\mu}}_i \cdot \hat{\boldsymbol{r}}_{ij})\,\hat{\boldsymbol{r}}_{ij} \Big) \tag{A.3}$$

Now break the equation out into its components and apply the definition of the dot product, $\hat{\boldsymbol{\mu}}_i \cdot \hat{\boldsymbol{r}}_{ij} = \mu_{ix} r_x + \mu_{iy} r_y + \mu_{iz} r_z$.

$$\begin{aligned}
\frac{\boldsymbol{F}_x r_{ij}^4}{C_m} &= \mu_i \mu_j \Big(\mu_x - 3\,(\mu_{ix} r_x + \mu_{iy} r_y + \mu_{iz} r_z)\, r_x \Big) \\
\frac{\boldsymbol{F}_y r_{ij}^4}{C_m} &= \mu_i \mu_j \Big(\mu_y - 3\,(\mu_{ix} r_x + \mu_{iy} r_y + \mu_{iz} r_z)\, r_y \Big) \\
\frac{\boldsymbol{F}_z r_{ij}^4}{C_m} &= \mu_i \mu_j \Big(\mu_z - 3\,(\mu_{ix} r_x + \mu_{iy} r_y + \mu_{iz} r_z)\, r_z \Big)
\end{aligned} \tag{A.4}$$

Group the terms by $\boldsymbol{\mu}$ components

$$\begin{aligned}
\frac{\boldsymbol{F}_x r_{ij}^4}{C_m} &= \mu_i\mu_j\Big(\mu_{ix}\left(1-3r_x^2\right) - 3\mu_{iy}r_xr_y - 3\mu_{iz}r_xr_z\Big) \\
\frac{\boldsymbol{F}_y r_{ij}^4}{C_m} &= \mu_i\mu_j\Big(-3\mu_{ix}r_xr_y + \mu_{iy}\left(1-3r_y^2\right) - 3\mu_{iz}r_yr_z\Big) \\
\frac{\boldsymbol{F}_z r_{ij}^4}{C_m} &= \mu_i\mu_j\Big(-3\mu_{ix}r_xr_z - 3\mu_{iy}r_yr_z + \mu_{iz}\left(1-3r_z^2\right)\Big)
\end{aligned} \tag{A.5}$$

Factor the vector $\boldsymbol{\mu}_i$ out of the set of equations

$$\frac{\boldsymbol{F} r_{ij}^4}{C_m} = \mu_i\mu_j \begin{bmatrix} 1-3r_x^2 & -r_xr_y & -r_xr_z \\ -r_xr_y & 1-3r_y^2 & -r_yr_z \\ -r_xr_z & -r_yr_z & 1-3r_z^2 \end{bmatrix} \begin{bmatrix} \mu_{ix} \\ \mu_{iy} \\ \mu_{iz} \end{bmatrix} \tag{A.6}$$

Whic reduces to

$$\frac{\boldsymbol{F} r_{ij}^4}{C_m} = \left[I_3 - 3\hat{\boldsymbol{r}}_{ij}\hat{\boldsymbol{r}}_{ij}^{\mathsf{T}}\right]\hat{\boldsymbol{\mu}}_i\mu_i\mu_j \tag{A.7}$$

And if it is assumed (reasonably) that the dipole magnitudes are equal, i.e. $\mu_i = \mu_j = \mu$ then

$$\frac{\boldsymbol{F} r_{ij}^4}{C_m} = \left[I_3 - 3\hat{\boldsymbol{r}}_{ij}\hat{\boldsymbol{r}}_{ij}^{\mathsf{T}}\right]\hat{\boldsymbol{\mu}}_i\mu^2 = \mathbf{M}\hat{\boldsymbol{\mu}}_i\mu^2 \tag{A.8}$$

$$\mathbf{M}^{-1}\frac{\boldsymbol{F} r_{ij}^4}{C_m} = \hat{\boldsymbol{\mu}}_i\mu^2 \tag{A.9}$$

Which is valid so long as $\mathbf{M}$ is invertible.

$$\begin{aligned}
\det(\mathbf{M}) &= 1 - 3r_x^2 - 3r_y^2 - 3r_z^2 \\
\det(\mathbf{M}) &= 1 - 3\|\hat{\boldsymbol{r}}\| \\
\det(\mathbf{M}) &= -2
\end{aligned} \tag{A.10}$$

As shown, the determinant of $\mathbf{M}$ is always non-zero thus the matrix is always invertible and there is always a vector $\boldsymbol{\mu}_i$ that produces force $\boldsymbol{F}$.

Further, applying the definition of $r_{ij} = [-2x, -2y]$ shows a different look at this equation. Starting with $\mathbf{M}$

$$\mathbf{M} = \begin{bmatrix} 1 - 12x^2 & -12xy \\ -12xy & 1 - 12y^2 \end{bmatrix} \tag{A.11}$$

Now note that the direction of force is $\tan^{-1}(\hat{\boldsymbol{\mu}})$ and $\hat{\boldsymbol{\mu}}$ expressed in this form is independent of the scalar magnitude, therefore for this purpose the r_{ij}^4 and C_m terms in Equation (A.9) on the facing page can be ignored (they are all wrapped into scalar value K).

$$\mathbf{M}^{-1}\hat{\boldsymbol{F}} = K\hat{\boldsymbol{\mu}} \tag{A.12}$$

where $\hat{\boldsymbol{F}} = [\cos(\theta_c), \sin(\theta_c)]^\intercal$ and θ_c is the desired force angle determined by

$$\theta_c = tan^{-1}(\frac{\lambda_d\dot{x} + \lambda_p x}{\lambda_d\dot{y} + \lambda_p y}) \tag{A.13}$$

Performing the inverse, multiplication and taking the inverse tangent from Equation (A.12) to get the angle

$$\alpha = \tan^{-1}\left(\frac{-12\lambda_p x^3 - 12\lambda_d\dot{x}x^2 + 12\lambda_p xy^2 + 12\lambda_d\dot{y}xy + \lambda_p x + \lambda_d\dot{x}}{12\lambda_p x^2 y + 12\lambda_d\dot{x}xy - 12\lambda_p y^3 - 12\lambda_d\dot{y}y^2 + \lambda_p y + \lambda_d\dot{y}}\right) \tag{A.14}$$

This is an algebraic expression for α as measured in the inertial reference frame.

Appendix B

Near-field Model Matlab Code

The following source files are also located on the SPHERES SVN at
http://ssl.mit.edu/svn/spheres/trunk/TestProjects/RINGS/project_Utilities

daeEM.m

```
function [FT1,FT2] = daeEM(X1,X2,II,radius,Nt,inertialOutput)%#codegen
assert(isa(X1,'double'));
assert(all(size(X1,2)==13));
assert(isa(X2,'double'));
assert(all(size(X2,2)==13));
assert(isa(II,'double'));
assert(all(size(II,2)==1));
assert(isa(radius,'double'));
assert(all(size(radius)==[1 1]));
assert(isa(Nt,'double'));
assert(all(size(Nt)==[1 1]));
assert(isa(inertialOutput,'double'));
assert(all(size(inertialOutput)==[1 1]));
% X1 - State Vector 1
% X2 - State Vector 2
% II - I1 * I2 (electrical currents)
% radius - radius of coils
% Nt - Number of timesteps given
% inertialOutput - Flag to select body or inertial reference outputs

% Initialize Outputs
FT1 = zeros(Nt,6);
Fx1 = zeros(Nt,1); Fy1 = zeros(Nt,1); Fz1 = zeros(Nt,1);
Tx1 = zeros(Nt,1); Ty1 = zeros(Nt,1); Tz1 = zeros(Nt,1);
FT2 = zeros(Nt,6);
Fx2 = zeros(Nt,1); Fy2 = zeros(Nt,1); Fz2 = zeros(Nt,1);
Tx2 = zeros(Nt,1); Ty2 = zeros(Nt,1); Tz2 = zeros(Nt,1);
x1 = X1(:,1); x2 = X2(:,1);
y1 = X1(:,2); y2 = X2(:,2);
z1 = X1(:,3); z2 = X2(:,3);
vx1 = X1(:,4); vx2 = X2(:,4);
```

```matlab
vy1 = X1(:,5); vy2 = X2(:,5);
vz1 = X1(:,6); vz2 = X2(:,6);
q11 = X1(:,7); q12 = X2(:,7);
q21 = X1(:,8); q22 = X2(:,8);
q31 = X1(:,9); q32 = X2(:,9);
q41 = X1(:,10); q42 = X2(:,10);
wx1 = X1(:,11); wx2 = X2(:,11);
wy1 = X1(:,12); wy2 = X2(:,12);
wz1 = X1(:,13); wz2 = X2(:,13);

for i=1:Nt
    Isq = II(i);
    % Extract States
    xA = [x1(i);y1(i);z1(i);vx1(i);vy1(i);vz1(i);...
            q11(i);q21(i);q31(i);q41(i);wx1(i);wy1(i);wz1(i)];
    xB = [x2(i);y2(i);z2(i);vx2(i);vy2(i);vz2(i);...
            q12(i);q22(i);q32(i);q42(i);wx2(i);wy2(i);wz2(i)];
    R1 = xA(1:3);
    R2 = xB(1:3);
    q = xA(7:10);
    p = xB(7:10);
    R  = R2-R1;
    nR = sqrt(sum(R.*R));
    nRrad = nR/radius;
    % See Alex Buck S.M Thesis for description of N-Segment ...
        selection. This curve describes
    % the polynomial 'w' such that the limiting curve, 'y' is y=1/w. ...
        N should lie on this
    % curve, but be no larger in order to ensure machine precision ...
        without any excess
    % computation.
    LimPolyInv = [    -0.000871941889380    0.015960066294433   ...
        -0.003414459128870    ];
    N = uint32(1/polyval(LimPolyInv,nRrad));
    F1 = zeros(1,3); F2 = F1;
    T1 = zeros(1,3); T2 = T1;
    if ( Isq~=0 ) % If either coil isn't driving, just skip ...
        computation, output will be 0.
        coder.ceval('EMWrapper',coder.ref(F1),coder.ref(T1),...
                                coder.rref(xA),coder.rref(xB),...
                                Isq,radius,N);
        for ii=1:3
            if isnan(F1(ii))
                F1(ii)=0;
            end
            if isnan(T1(ii))
                T1(ii)=0;
            end
        end
        F2 = -F1;
        T2 = -T1 - cross(R1(:),F1) - cross(R2(:),-F1);
        if ~inertialOutput
            % Rotate to body frame coordinates
            [F1,T1] = rotateToBody(F1,T1,q);
            [F2,T2] = rotateToBody(F2,T2,p);
        end
    end
    Fx1(i) = F1(1); Fy1(i) = F1(2); Fz1(i) = F1(3);
    Tx1(i) = T1(1); Ty1(i) = T1(2); Tz1(i) = T1(3);
    Fx2(i) = F2(1); Fy2(i) = F2(2); Fz2(i) = F2(3);
    Tx2(i) = T2(1); Ty2(i) = T2(2); Tz2(i) = T2(3);
end
FT1 = [Fx1(:) Fy1(:) Fz1(:) Tx1(:) Ty1(:) Tz1(:)];
FT2 = [Fx2(:) Fy2(:) Fz2(:) Tx2(:) Ty2(:) Tz2(:)];

function [F,T] = rotateToBody(F,T,q)
q1 = q(1);  q2 = q(2);  q3 = q(3);  q4 = q(4);
RB2I = [q4*q4+q1*q1-q2*q2-q3*q3, 2*(q1*q2-q3*q4), 2*(q1*q3+q2*q4);
```

```
    2*(q1*q2+q3*q4), q4*q4-q1*q1+q2*q2-q3*q3, 2*(q2*q3-q1*q4);
    2*(q1*q3-q2*q4), 2*(q2*q3+q1*q4), q4*q4-q1*q1-q2*q2+q3*q3];
RI2B = RB2I';
F = (RI2B * F(:))';
T = (RI2B * T(:))';
```

daeDyn.m

```
function [dae] = daeDyn(X1,X2,II,Tt1,Tt2,radius,Nt,mass,I,InvI)
% v,a,qd,wd,pd,ud
%#codegen
assert(isa(I,'double'));
assert(isa(InvI,'double'));
assert(all(size(I)==[3 3]));
assert(all(size(InvI)==[3 3]));
assert(isa(Nt,'double'));
assert(all(size(Nt)==[1 1]));
assert(isa(II,'double'));
assert(all(size(II,2)==1));
assert(all(size(X1,2)==13));
assert(all(size(X2,2)==13));
assert(all(size(Tt1,2)==3));
assert(all(size(Tt2,2)==3));
assert(isa(mass,'double'));
assert(isa(X1,'double'));
assert(isa(X2,'double'));
assert(isa(Tt1,'double'));
assert(isa(Tt2,'double'));
assert(isa(radius,'double'));

m = mass; % kg
% Initialize Outputs
dx=zeros(Nt,1);dy=zeros(Nt,1);dz=zeros(Nt,1);
dvx=zeros(Nt,1);dvy=zeros(Nt,1);dvz=zeros(Nt,1);
dq1=zeros(Nt,1);dq2=zeros(Nt,1);dq3=zeros(Nt,1);
dw1=zeros(Nt,1);dw2=zeros(Nt,1);dw3=zeros(Nt,1);
dp1=zeros(Nt,1);dp2=zeros(Nt,1);dp3=zeros(Nt,1);
du1=zeros(Nt,1);du2=zeros(Nt,1);du3=zeros(Nt,1);
x1 = X1(:,1); x2 = X2(:,1);
y1 = X1(:,2); y2 = X2(:,2);
z1 = X1(:,3); z2 = X2(:,3);
vx1 = X1(:,4); vx2 = X2(:,4);
vy1 = X1(:,5); vy2 = X2(:,5);
vz1 = X1(:,6); vz2 = X2(:,6);
q11 = X1(:,7); q12 = X2(:,7);
q21 = X1(:,8); q22 = X2(:,8);
q31 = X1(:,9); q32 = X2(:,9);
q41 = X1(:,10); q42 = X2(:,10);
wx1 = X1(:,11); wx2 = X2(:,11);
wy1 = X1(:,12); wy2 = X2(:,12);
wz1 = X1(:,13); wz2 = X2(:,13);


for i=1:Nt
    Isq = II(i);
    % Extract States
    xA = [x1(i);y1(i);z1(i);vx1(i);vy1(i);vz1(i);...
            q11(i);q21(i);q31(i);q41(i);wx1(i);wy1(i);wz1(i)];
    xB = [x2(i);y2(i);z2(i);vx2(i);vy2(i);vz2(i);...
            q12(i);q22(i);q32(i);q42(i);wx2(i);wy2(i);wz2(i)];
    R1 = xA(1:3);
    R2 = xB(1:3);
    V1 = xA(4:6);
    q = xA(7:10);
    p = xB(7:10);
    w = xA(11:13);
    u = xB(11:13);
    R  = R2-R1;
    nR = sqrt(sum(R.*R));
    nRrad = nR/radius;
    % See Alex Buck S.M Thesis for description of N-Segment ...
        selection. This curve describes
    % the polynomial 'w' such that the limiting curve, 'y' is y=1/w. ...
        N should lie on this
    % curve, but be no larger in order to ensure machine precision ...
        without any excess
    % computation.
    LimPolyInv = [   -0.000871941889380    0.015960066294433   ...
        -0.003414459128870    ];
    N = uint32(1/polyval(LimPolyInv,nRrad));
    F1 = zeros(1,3); F2 = F1;
    T1 = zeros(1,3); T2 = T1;
    if ( Isq~=0 ) % If either coil isn't driving, just skip ...
        computation, output will be 0.
        coder.ceval('EMWrapper',coder.ref(F1),coder.ref(T1),...
                                coder.rref(xA),coder.rref(xB),...
                                Isq,radius,N);
        for ii=1:3
            if isnan(F1(ii))
                F1(ii)=0;
            end
            if isnan(T1(ii))
                T1(ii)=0;
            end
        end
        F2 = -F1;
        T2 = -T1 - cross(R1(:),F1) - cross(R2(:),-F1);
        [F1,T1] = qrotate(F1,T1,q);
        [~,T2] = qrotate(F2,T2,p);
    end
    TT1 = T1+Tt1(i,1:3);
    TT2 = T2+Tt2(i,1:3);

    dq = quatDeriv(q,w);
    dp = quatDeriv(p,u);
    dw = (InvI * (TT1' - cross(w,I*w)))';
    du = (InvI * (TT2' - cross(u,I*u)))';

    dx(i) = V1(1);   dy(i) = V1(2);   dz(i) = V1(3);
    dvx(i)= F1(1)/m; dvy(i)= F1(2)/m; dvz(i)= F1(3)/m;
    dq1(i)= dq(1);   dq2(i)= dq(2);   dq3(i)= dq(3);
    dw1(i)= dw(1);   dw2(i)= dw(2);   dw3(i)= dw(3);
    dp1(i)= dp(1);   dp2(i)= dp(2);   dp3(i)= dp(3);
    du1(i)= du(1);   du2(i)= du(2);   du3(i)= du(3);
end
dae = [dx(:)  dy(:)  dz(:)  dvx(:) dvy(:) dvz(:)...
       dq1(:) dq2(:) dq3(:) dw1(:) dw2(:) dw3(:)...
       dp1(:) dp2(:) dp3(:) du1(:) du2(:) du3(:)];


function qd = quatDeriv(q,w)
qd = zeros(1,3);
qd(1) = 0.5*( q(2)*w(3) - q(3)*w(2) + q(4)*w(1));
qd(2) = 0.5*( q(3)*w(1) - q(1)*w(3) + q(4)*w(2));
qd(3) = 0.5*( q(1)*w(2) - q(2)*w(1) + q(4)*w(3));

function [F,T] = qrotate(F,T,q)
q1 = q(1);  q2 = q(2);  q3 = q(3);  q4 = q(4);
RB2I = [q4*q4+q1*q1-q2*q2-q3*q3, 2*(q1*q2-q3*q4), 2*(q1*q3+q2*q4);
     2*(q1*q2+q3*q4), q4*q4-q1*q1+q2*q2-q3*q3, 2*(q2*q3-q1*q4);
     2*(q1*q3-q2*q4), 2*(q2*q3+q1*q4), q4*q4-q1*q1-q2*q2+q3*q3];
RI2B = RB2I';
F = (RI2B * F(:))';
```

```
T = (RI2B * T(:))';
```

EMWrapper.c

```
//
//  EMWrapper.c
//
//
//  Created by Alexander Buck on 11/19/12.
//

#ifdef __cplusplus
extern "C" {
#endif

#include <math.h>
#include <stdio.h>
#include <string.h>
#include "globals.h"
#include "EMWrapper.h"
#include "utilities.h"
#include "EM_ForceTorque.h"

void EMWrapper(double* Fm,double *Tm, double *x1, double *x2,
               double I2,double radius,unsigned int Nseg)
{
    double coil_1[MAX_Segments*3] = {0};
    double coil_2[MAX_Segments*3] = {0};
    double dL_1[MAX_Segments*3] = {0};
    double dL_2[MAX_Segments*3] = {0};
    double coil_std[MAX_Segments*3] = {0};
    double   dL_std[MAX_Segments*3] = {0};
    static double Z[3] = {0,0,0};
    double dl;
    unsigned int idx,N;
    memset(coil_1,0,MAX_Segments*3*sizeof(double));
    memset(coil_2,0,MAX_Segments*3*sizeof(double));
    memset(dL_1,0,MAX_Segments*3*sizeof(double));
    memset(dL_2,0,MAX_Segments*3*sizeof(double));

    N  = (Nseg>MAX_Segments)?MAX_Segments:Nseg;
    dl = 2*pi*radius/N;
    for(idx=0;idx<N;idx++){
        coil_std[3*idx+0] = cos(2*pi*idx/N);
        coil_std[3*idx+1] = 0;
        coil_std[3*idx+2] = sin(2*pi*idx/N);
          dL_std[3*idx+0] = -coil_std[3*idx+2];
          dL_std[3*idx+1] = 0;
          dL_std[3*idx+2] = coil_std[3*idx+0];
    }

    /* Rotate the standard coil and standard dL vectors */
    mat_rotateScaleTranslate(coil_std, &x1[QUAT], radius, &x1[POS], ...
        coil_1, N, 3);
    mat_rotateScaleTranslate(coil_std, &x2[QUAT], radius, &x2[POS], ...
        coil_2, N, 3);
    mat_rotateScaleTranslate(  dL_std, &x1[QUAT],     dl,        Z,   ...
        dL_1, N, 3);
    mat_rotateScaleTranslate(  dL_std, &x2[QUAT],     dl,        Z,   ...
        dL_2, N, 3);
    /* Call the EMFF_3D subroutine. */
    EM_ForceTorque(&x1[POS], (double*)coil_1, (double*)dL_1, N,
                   &x2[POS], (double*)coil_2, (double*)dL_2, N,
                   I2,       N_turns,         N_turns,
                   Fm,       Tm);
}

#ifdef __cplusplus
}
#endif
```

EM_ForceTorque.c

```
//
//  EM_ForceTorque.c
//
//
//  Created by Alexander Buck on 11/19/12.
//

#ifdef __cplusplus
extern "C" {
#endif

#include <math.h>
#include "globals.h"
#include "utilities.h"
#include "EM_ForceTorque.h"

/* Mex Function */
void EM_ForceTorque( double* R_1, double* coil_1, double* dL_1, ...
     double Nseg_1,
                     double* R_2, double* coil_2, double* dL_2, ...
                          double Nseg_2,
                     double I2,  double Nturns_1, double Nturns_2,
                     double* F1, double* T1){
    // coil_i and dL_i are (Ncoil_i)x3 arrays where
    // Ncoil_i is the number of segments in coil_i
    int i,j;
    double temp;
    double Atemp[3];
    double Btemp[3];
    double dr[3];
    double Tseg1[3]={0};
    double Fseg1[3]={0};

    // Loop over each element on Coil 1
    for(i=0;i<Nseg_1;i++) {
        // Loop over each element on Coil 2
        for(j=0;j<Nseg_2;j++) {
            /*
             * NOTE: Do not multiply in constants yet, they are extra
             * computations that are not required until all the segments
             * are added up.
             */
            //  Fseg  =  1/||dr||^3  *  dL_1 x (dL_2 x dr)
            // r_i - r_j
            // dL_2 x ( ^ )
            // dL_1 x ( ^ )
            // ( ^ ) / ||dr||^2 = Ftemp
            vec_sub(    &coil_1[i*3],  &coil_2[j*3],   dr,     3);
            vec_cross(  &dL_2[j*3],    dr,                     Atemp);
            vec_cross(  &dL_1[i*3],    Atemp,                  Btemp);
            vec_div(    Btemp,         pow(vec_norm(dr,3),3), ...
                Fseg1,  3);
            // Tseg = r x F
            // r_i-R_1
            // ( ^ ) x Ftemp    = Ttemp
            vec_sub(    &coil_1[i*3],  R_1,                    ...
                Atemp,  3);
            vec_cross( Atemp,          Fseg1,                    Tseg1);
            // Accumulate the force contributions from each element
            // Accumulate the torque contribution from each element
            vec_add(F1, Fseg1, F1, 3);
            vec_add(T1, Tseg1, T1, 3);
        }
    }
    // Multiply constants after double sum
    temp = km * I2 * Nturns_1 * Nturns_2;
    // Multiply constants into Force and Torque (T = dr x F)
    vec_scale(F1,temp,F1,3);
    vec_scale(T1,temp,T1,3);
}

#ifdef __cplusplus
}
#endif
```

utilities.c

```
//
//  utilities.c
//
//
//  Created by Alexander Buck on 11/19/12.
//

#ifdef __cplusplus
extern "C" {
#endif

#include <math.h>
#include <stdio.h>
#include "utilities.h"
#ifndef DEBUG
    #define printf(...) do {} while(0)
#endif

/* Utility Functions */
/* c cannot be either a or b */
void vec_cross(double* a,double* b,double* c)
{
    c[0] = a[1] * b[2] - a[2] * b[1];
    c[1] = a[2] * b[0] - a[0] * b[2];
    c[2] = a[0] * b[1] - a[1] * b[0];
}

// a and c can be the same
void vec_add(double* a,double* b,double* c,int len)
{
    int i;
    for(i=0;i<len;i++) {
        c[i] = a[i]+b[i];
    }
}

// a and c can be the same
void vec_sub(double* a,double* b,double* c,int len)
{
    int i;
    for(i=0;i<len;i++) {
        c[i] = a[i]-b[i];
    }
}

```

```
// a and c can be the same
void vec_div(double* a,double b,double* c,int len)
{
    int i;
    for(i=0;i<len;i++) {
        c[i] = a[i]/ b;
    }
}
void vec_scale(double* a,double b,double* c,int len)
{
    int i;
    for(i=0;i<len;i++) {
        c[i] = a[i]*b;
    }
}

double vec_norm(double* a,int len)
{
    int i;
    double tmp=0;
    for(i=0;i<len;i++) {
        tmp += a[i]*a[i];
    }
    return sqrt(tmp);
}

void vec_copy(double* a,double*b,int len)
{
    vec_scale(a,1,b,len);
}

// a and c can be the same
void mat_add(double* a,double* b,double* c,int m,int n)
{
    int i,j;
    for(i=0;i<m;i++) {
        for(j=0;j<n;j++) {
            c[i*n+j] = a[i*n+j] + b[i*n+j];
        }
    }
}

// a and c can be the same
void mat_sub(double* a,double* b,double* c,int m,int n)
{
    int i,j;
    for(i=0;i<m;i++) {
        for(j=0;j<n;j++) {
            c[i*n+j] = a[i*n+j] - b[i*n+j];
        }
    }
}

// a and c can be the same
void mat_div(double* a,double b,double* c,int m,int n)
{
    int i,j;
    for(i=0;i<m;i++) {
        for(j=0;j<n;j++) {
            c[i*n+j] = a[i*n+j]/b;
        }
    }
}

// a and c can be the same
void mat_scale(double* a,double b,double* c,int m,int n)
{
    int i,j;
    for(i=0;i<m;i++) {
        for(j=0;j<n;j++) {
            c[i*n+j] = a[i*n+j]*b;
        }
    }
}

// a and c CANNOT be the same
void mat_mult(const double* a,double* b, double* c,int nra,int ...
    nca,int ncb)
{
    int i,j,k;
    for(i=0;i<nra;i++) {
        for(j=0;j<ncb;j++) {
            c[i*ncb+j]=0.0f;
            for(k=0;k<nca;k++) {
        //  The i'th row and j'th column of c is:
        //  k'th entry in the i'th row of a (TIMES) k'th entry in the ...
            j'th column of b
                c[i*ncb+j] += a[i*nca+k]  *  b[k*ncb+j];
                printf("%f * %f = ...
                    %f\n",a[i*nca+k],b[k*ncb+j],c[i*ncb+j]);
            }
            printf("\n");
        }
        printf("\n");
    }
}

// Adds a vector to the rows of a matrix
// a and c can be the same
void mat_vec_add(double* a, double* b, double* c, int m, int n)
{
    int i;
    for(i=0;i<m;i++) {
        vec_add(&a[i*n],&b[0],&c[i*n],n);
    }
}

// Rotate a matrix of M Nx1 vectors
/*
 * a = [               |               |              ]
 *  j =   0 1 2 3 4       0 1 2 3 4      0 1 2 3 4           <---- N
 *  i =       0               1              2               <---- M
 */
void mat_rotate(double* a, double* b, double* c, int M, int N)
{
    int i,j,k;
    for(i=0;i<M;i++) {
        for(j=0;j<N;j++) {
            c[i*N+j] = 0;
            for(k=0;k<N;k++) {
                c[i*N+j] += b[j*N+k]*a[i*N+k];
                printf("%f * %f = %f\n",a[i*N+k],b[j*N+k],c[i*N+j]);
            }
        }
    }
}

/*
 * Scale Factor -----------------------------------------------\
 * Rotation (quaternion) --------------------------\           |
 * Input Matrix --------------------\              |           |
 *                                  V              V           V ...
            */
void mat_rotateScaleTranslate( double* in, double* quat, double scale,
                               double* translate, double* out, int ...
                                   M,int N)
```

```
/* Offset Vector ------------------------/             |           | ...
        |
 * Output Matrix --------------------------------------/           | ...
         |
 * Input Matrix Rows --------------------------------------------/ ...
         |
 * Input Matrix Cols ---------------------------------------------------/
 */
{
    double rotMat[3*3];

    quat_rotation(quat,rotMat);

    mat_rotate(in,rotMat,out,M,N);

    mat_scale(out,scale,out,M,N);

    mat_vec_add(out,translate,out,M,N);
}



double mat_inv33(double* in, double* out)
{
    double m11, m12, m13, m21, m22, m23, m31, m32, m33;
    double temp;

    m11 = in[0];
    m12 = in[1];
    m13 = in[2];
    m21 = in[3];
    m22 = in[4];
    m23 = in[5];
    m31 = in[6];
    m32 = in[7];
    m33 = in[8];

    temp = m11*m22*m33 - m11*m23*m32 + m12*m23*m31
         - m12*m21*m33 + m13*m21*m32 - m13*m22*m31;

    // must have non-zero determinant
    if (temp == 0.0f)
        return 1;

    // make it multiplier to speed things up
    temp = 1/temp;

    out[0] = temp*(m22*m33-m23*m32);
    out[1] = temp*(m13*m32-m12*m33);
    out[2] = temp*(m12*m23-m13*m22);
    out[3] = temp*(m23*m31-m21*m33);
    out[4] = temp*(m11*m33-m13*m31);
    out[5] = temp*(m13*m21-m11*m23);
    out[6] = temp*(m21*m32-m22*m31);
    out[7] = temp*(m12*m31-m11*m32);
    out[8] = temp*(m11*m22-m12*m21);

    return 0;
}

void quat_rotation(double* quat, double* mat)
{
    double q1 = quat[0], q2 = quat[1];
    double q3 = quat[2], q4 = quat[3];
    double qn = sqrt(q1*q1 + q2*q2 + q3*q3 + q4*q4);
    printf("Quat norm: %f\n",qn);
    q1 = q1/qn;
    q2 = q2/qn;
    q3 = q3/qn;
    q4 = q4/qn;
    // make the rotation matrix from the quaternion
    mat[0] = q4*q4+q1*q1-q2*q2-q3*q3;
    mat[1] = 2*(q1*q2-q3*q4);
    mat[2] = 2*(q1*q3+q2*q4);
    mat[3] = 2*(q1*q2+q3*q4);
    mat[4] = q4*q4-q1*q1+q2*q2-q3*q3;
    mat[5] = 2*(q2*q3-q1*q4);
    mat[6] = 2*(q1*q3-q2*q4);
    mat[7] = 2*(q2*q3+q1*q4);
    mat[8] = q4*q4-q1*q1-q2*q2+q3*q3;
}

#ifdef __cplusplus
}
#endif
```

Appendix C

GPOPS – II: Spin-up Trajectory Matlab Code

Main Function

```
function planarSpinupMain(nlpsolver,inputSol,minT,tol)
%-------------------------------------------------%
%----------- planarSpinup Problem ----------------%
%-------------------------------------------------%
clc
if nargin<1
    nlpsolver = 'ipopt';
end
if nargin<2
    inputSol = [];
end
if nargin<3
    minT = 1;
end
if nargin<4
    tol = [1e-6;1e-6];
end
%-------------------------------------------------%
%------- Set Up Auxiliary Data for Problem ---------%
%-------------------------------------------------%
auxdata.radius  = 0.31; %m
auxdata.mu0     = pi*4e-6;
auxdata.m       = 17.2; % kg
auxdata.Inertia = diag([0.3742 1 0.5577]);
auxdata.IInertia= inv(auxdata.Inertia);
if    minT
    auxdata.ki      = 1;
    auxdata.kt      = 1;
    auxdata.ku3     = 1e-6;
    auxdata.ku12    = 100;
    auxdata.kw      = 1;
else %minE
    auxdata.ki      = 1;
    auxdata.kt      = 0;
    auxdata.ku3     = 1e-4;
    auxdata.ku12    = 10000;
    auxdata.kw      = .5;
end
```

```
auxdata.minr = [0.4;0.7];
%-------------------------------------------------%
%--- Set Up Bounds on State, Control, and Time -----%
%-------------------------------------------------%
t0          = 0;
tf          = 120;
minr        = 0.4;
r0u         = [ pi        .7];   %r0 = [ th , r  ]
r0l         = [-pi        .7];   %r0 = [ th , r  ]
a0l         =-[pi         pi];   %a0 = [ a1 , a2 ]
a0u         = [pi         pi];   %a0 = [ a1 , a2 ]
v0          = [0           0];   %v0 = [vth , vr ]
w0          = [0           0];   %w0 = [ w1 , w2 ]
rmax        = [3*pi        2];   %r  = [ th , r  ]
amax        = [3           3]*pi;%a  = [ a1 , a2 ]
vmax        = [1           1]*.2;%v  = [vth , vr ]
wmax        = [1           1];   %w  = [ w1 , w2 ]
rmin        = [-3*pi   minr];    %r  = [ th , r  ]
amin        =   -amax;           %a  = [ a1 , a2 ]
vmin        =   -vmax;           %v  = [vth , vr ]
wmin        =   -wmax;           %w  = [ w1 , w2 ]
          T = 60;              % 2 minute period
          w = 2*pi / T;        % Required angular rate
rfl         = [pi       .7];     %r  = [ th , r  ]
rfu         = [pi       .7];     %r  = [ th , r  ]
afl         =-[pi       pi]/2;   %a  = [ a1 , a2 ]
afu         =-[pi       pi]/2;   %a  = [ a1 , a2 ]
vfl         = [w         0];     %v  = [dth , vr ]
vfu         = [w         0];     %v  = [dth , vr ]
wfl         = [w         w];     %w  = [ w1 , w2 ]
wfu         = [w         w];     %w  = [ w1 , w2 ]
thrust = 0.1; %N
leverR = 0.1; %m
tqlim       = (2*thrust)*leverR;
u1max       = +tqlim;
u1min       = -tqlim;
u2max       = +tqlim;
u2min       = -tqlim;
u3max       = +225; % 15 Amps
u3min       = -225; % 15 Amps

bounds.phase.initialtime.lower  = t0;
bounds.phase.initialtime.upper  = t0;
if   minT
    bounds.phase.finaltime.lower    = t0;
    bounds.phase.finaltime.upper    = tf;
else%minE
    bounds.phase.finaltime.lower    = tf;
    bounds.phase.finaltime.upper    = tf;
end
bounds.phase.initialstate.lower = [r0l, a0l, v0, w0];
bounds.phase.initialstate.upper = [r0u, a0u, v0, w0];
bounds.phase.state.lower        = [rmin, amin, vmin, wmin];
bounds.phase.state.upper        = [rmax, amax, vmax, wmax];
bounds.phase.finalstate.lower   = [rfl, afl, vfl, wfl];
bounds.phase.finalstate.upper   = [rfu, afu, vfu, wfu];

bounds.phase.control.lower      = [u1min, u2min, u3min];
bounds.phase.control.upper      = [u1max, u2max, u3max];

bounds.phase.integral.lower     = 0;
bounds.phase.integral.upper     = 5000;
%-------------------------------------------------%
%-------------- Set Up Initial Guess --------------%
%-------------------------------------------------%
tGuess      = [t0; tf];
rGuess      = [(r0l+r0u)/2; (rfl+rfu)/2];
aGuess      = [(a0l+a0u)/2; (afl+afu)/2];
```

```
vGuess       = [v0; (vfl+vfu)/2];
wGuess       = [w0; (wfl+wfu)/2];
u1Guess      = [.2; 0];
u2Guess      = [.2; 0];
u3Guess      = [225; 225];
if isempty(inputSol)
    guess.phase.time    = [tGuess];
    guess.phase.state   = [rGuess, aGuess, vGuess, wGuess];
    guess.phase.control = [u1Guess, u2Guess, u3Guess];
    guess.phase.integral= [ 50 ];
else
    guess.phase.time    = inputSol.time;
    guess.phase.state   = inputSol.state;
    guess.phase.control = inputSol.control;
    guess.phase.integral= inputSol.integral;
end%-------------------------------------------------%
%------------- Set Up Initial Mesh -----------------%
%---------------------------------------------------%
N = 10;
meshphase.colpoints = 4*ones(1,N);
meshphase.fraction   = ones(1,N)/N;
%---------------------------------------------------%
%--------------- Set Up for Solver -----------------%
%---------------------------------------------------%
setup.name = 'Planar Spinup';
setup.functions.continuous = @commonContinuous;
setup.functions.endpoint = @commonEndpoint;
setup.method = 'RPMintegration';
setup.nlp.solver = nlpsolver;
setup.auxdata = auxdata;
setup.bounds = bounds;
setup.mesh.phase = meshphase;
setup.guess = guess;
setup.derivatives.supplier = 'sparseCD';
setup.derivatives.derivativelevel = 'second';
setup.derivatives.dependencies = 'full';
setup.scales = 'automatic-bounds';
setup.mesh.method = 'hp1';

setup.mesh.tolerance = tol(1);
setup.nlp.options.tolerance = tol(2);

%---------------------------------------------------%
%------ Solve Problem and Extract Solution ---------%
%---------------------------------------------------%
output = gpops2(setup);
assignin('base','output',output);
```

Dynamics Function

```
function [phaseout,dvxy] = planarSpinupContinuous(input)

% input
% input.phase(phasenumber).state
% input.phase(phasenumber).control
% input.phase(phasenumber).time
% input.phase(phasenumber).parameter
%
% input.auxdata = auxiliary information
%
% output
% phaseout(phasenumber).dynamics
% phaseout(phasenumber).path
% phaseout(phasenumber).integrand

```

```
%Extract Auxdata
    auxdata = input.auxdata;
    radius  = auxdata.radius;
    m       = auxdata.m;
    Inertia = auxdata.Inertia;
    ku3     = auxdata.ku3;
    ku12    = auxdata.ku12;
    kw      = auxdata.kw;

%Extract States
    t       = input.phase.time;
    th      = input.phase.state(:,1);
    r       = input.phase.state(:,2);
    a       = input.phase.state(:,[3 4]);
    v       = input.phase.state(:,[5 6]);
    w       = input.phase.state(:,[7 8]);
    u1      = input.phase.control(:,1);
    u2      = input.phase.control(:,2);
    u3      = input.phase.control(:,3);
    z       = zeros(length(t),1);
    o       = ones(length(t),1);

%Rotate [th, r] to [x y]
    [x,y]   = pol2cart(th,r);
    vr      = v(:,2);          % vr
    vth     = v(:,1).*r; % dth * r = vth

%Rotate [vr vth] to [vx vy]
    vxy = zeros(length(t),2);
    for i=1:length(t)
        vxy(i,:) = [cos(th(i)) -sin(th(i));sin(th(i)) cos(th(i))] * ...
            [vr(i);vth(i)];
    end

%Create state vectors for daeEM mex function interface
    x1 = [ [x y z]...
           [vxy z] ...
           [ [z z o].*(sin(a(:,1)/2)*ones(1,3)) cos(a(:,1)/2) ]...
           [z z w(:,1)]];
    x2 = [-[x y z]...
          -[vxy z]...
           [ [z z o].*(sin(a(:,2)/2)*ones(1,3)) cos(a(:,2)/2) ]...
           [z z w(:,2)]];
    [FT1,FT2] = daeEM(x1,x2,u3,radius,length(t),1);
    f1  = FT1(:,[1 2]);
    f2  = FT2(:,[1 2]);
    tq1 = FT1(:,6)  +  u1;
    tq2 = FT2(:,6)  +  u2;

%F = m * a
    dvxy = f1 ./ m;
    dvrth = zeros(length(t),2);
%Rotate [vx vy] back to [vr vth]
    for i=1:length(t) %                        Note the Transpose ----V
        dvrth(i,:) = [cos(th(i)) -sin(th(i));sin(th(i)) cos(th(i))]' *...
            [dvxy(i,1);dvxy(i,2)];
    end
%Transform dvth to dth_dot by dividing by the radius
    dv = [dvrth(:,2)./r dvrth(:,1)]; % [dth_dot dr_dot]
%T = I * a
    dw = [tq1/Inertia(3,3)  tq2/Inertia(3,3)];
%Final Output!
    dae = [   v   w   dv   dw   ];
    phaseout.dynamics  = dae;

    phaseout.path = [];

%Cost Function
```

```
    phaseout.integrand = ku12*(u1.^2 + u2.^2) +...
                         ku3*(u3.^2) +...
                         kw*(w(:,1).^2 + w(:,2).^2);
```

Endpoint Function

```
function output = planarSpinupEndpoint(input)

% Inputs
% input.phase(phasenumber).initialstate -- row
% input.phase(phasenumber).finalstate -- row
% input.phase(phasenumber).initialtime -- scalar
% input.phase(phasenumber).finaltime -- scalar
% input.phase(phasenumber).integral -- row
% input.parameter -- row
% input.auxdata = auxiliary information

% Output
% output.objective -- scalar
% output.eventgroup(eventnumber).event -- row
kt = input.auxdata.kt;
ki = input.auxdata.ki;
% cost
% output.objective = input.phase.integral;
output.objective = kt*input.phase.finaltime + ki*input.phase.integral;
```

Bibliography

[1] O. Brown and P. Eremenko. *The Value Proposition for Fractionated Space Architectures.* AIAA Paper 2006-7506. 2006.

[2] C. A. Beichman, N. J. Woolf, and C. A. Lindensmith. *The Terrestrial Planet Finder (TPF) : a NASA Origins Program to search for habitable planets.* 1999.

[3] David Barnhart. *Phoenix: Initial Technical Elements and Interfaces.* Tech. rep. DARPA, 2011.

[4] Jesse Leitner. *Formation Flying: The Future of Remote Sensing from Space.* Tech. rep. NASA, Jan. 2004.

[5] I.D. Boyd and A. Ketsdever. "Interactions Between Spacecraft and Thruster Plumes". In: *Journal of Spacecraft and Rockets* 38.3 (2001), pp. 380–380.

[6] AndrÃľ Kurs et al. "Wireless Power Transfer via Strongly Coupled Magnetic Resonances". In: *Science* 317.5834 (2007), pp. 83–86. DOI: `10.1126/science.1143254`.

[7] Laila Mireille Elias. "Dynamics of Multi-Body Space Interferometers Including Reaction Wheel Gyroscopic Stiffening Effects: Structurally Connected and Electromagnetic Formation Flying Architectures". PhD thesis. 77 Massachusetts Avenunue, Cambridge MA, 02139: Massachusetts Institute of Technology, Mar. 2004.

[8] Matthew Neave and Raymond J. Sedwick. "Dynamics and Thermal Control of an Electromagnetic Formation Flight Testbed". MA thesis. 77 Massachusetts Avenunue, Cambridge MA, 02139: Massachusetts Institute of Technology, June 2005.

[9] Daniel W. Kwon and David W. Miller. "Electromagnetic Formation Flight of Satellite Arrays". MA thesis. 77 Massachusetts Avenunue, Cambridge MA, 02139: Massachusetts Institute of Technology, Feb. 2005.

[10] Samuel Schweighart. “Electromagnetic Formation Flight Dipole Solution Planning”. PhD thesis. 77 Massachusetts Avenunue, Cambridge MA, 02139: Massachusetts Institute of Technology, June 2005.

[11] Umair Ahsun and David W. Miller. “Dynamics and Control of Electromagnetic Satellite Formations”. PhD thesis. 77 Massachusetts Avenunue, Cambridge MA, 02139: Massachusetts Institute of Technology, June 2007.

[12] Roman Wawrzazek and Marek Banaszkiewicz. “Control and reconfiguration of satellite formations by electromagnetic forces”. In: *Journal of Telecommunications and Information Technology* 1 (2007), p. 5.

[13] H. W. Kuhn and A. W. Tucker. “Nonlinear Programming”. In: *Proceedings of the Berkeley Symposium on Mathematical Statisitcs and Probability.* 1951, 481âĂŞ492.

[14] Byron K. Lichtenberg. *Interface Control Document: Boeing 727-200.* Tech. rep. Zero Gravity Corporation, 2009.

[15] Greg Eslinger. “Dynamic Programming for Electromagnetic Spacecraft Actuation”. MA thesis. 77 Massachusetts Avenunue, Cambridge MA, 02139: Massachusetts Institute of Technology, June 2013.

[16] Abraham. Savitzky and M. J. E. Golay. “Smoothing and Differentiation of Data by Simplified Least Squares Procedures.” In: *Analytical Chemistry* 36.8 (1964), pp. 1627–1639. DOI: 10.1021/ac60214a047.

[17] Jianwen Luo et al. “Properties of Savitzky–Golay digital differentiators”. In: *Digit. Signal Process.* 15.2 (Mar. 2005), pp. 122–136. ISSN: 1051-2004. DOI: 10.1016/j.dsp.2004.09.008.

[18] Simon Nolet. “Development of a Guidance, Navigation and Control Architecture and Validation Process Enabling Autonomous Docking to a Tumbling Satellite”. PhD thesis. 77 Massachusetts Avenunue, Cambridge MA, 02139: Massachusetts Institute of Technology, June 2007.

[19] Winfried Lohmiller and Jean-Jacques E. Slotine. “On Contraction Analysis for Non-linear Systems”. In: *Automatica* 34.6 (1998), pp. 683 –696. ISSN: 0005-1098. DOI: 10.1016/S0005-1098(98)00019-3. URL: http://www.sciencedirect.com/science/article/pii/S0005109898000193.

[20] I.M. Ross. *A Primer on Pontryagin's Principle in Optimal Control.* Collegiate Publishers, 2009. ISBN: 9780984357109.

[21] A.E. Bryson and Y.C. Ho. *Applied Optimal Control: Optimization, Estimation, and Control.* Halsted Press book'. Hemisphere Publishing Company, 1975. ISBN: 9780891162285. URL: `http://books.google.com/books?id=P4TKxn7qW5kC`.

[22] Philip E. Gill et al. "SNOPT: An SQP algorithm for large-scale constrained optimization". In: *SIAM Journal on Optimization* 12 (1997), pp. 979–1006. DOI: `10.1.1.54.7414`.

[23] A. Wächter and L. Biegler. "Line Search Filter Methods for Nonlinear Programming: Local Convergence". In: *SIAM Journal on Optimization* 16.1 (2005), pp. 32–48. DOI: `10.1137/S1052623403426544`.