CONTROL SYSTEM DESIGN FOR SPACECRAFT FORMATION FLYING: THEORY AND EXPERIMENT

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Andrew D. Robertson
June 2001
Copyright © 2001 by Andrew D. Robertson
All Rights Reserved.

Sun and UltraSPARC are trademarks of Sun Microsystems, Inc.
Matlab is a trademark of The Mathworks, Inc.
ControlShell is a trademark of Real-Time Innovations, Inc.
VxWorks is a trademark of Wind River Systems, Inc.
I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Jonathan P. How
Department of Aeronautics and Astronautics
(Principal Adviser)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Stephen M. Rock
Department of Aeronautics and Astronautics

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Robert H. Cannon, Jr.
Department of Aeronautics and Astronautics

Approved for the University Committee on Graduate Studies:
To Paul Garrues and To Robert Cannon
Abstract

Spacecraft formation flying is an enabling technology for many future space science missions, such as separated spacecraft interferometers (SSI). However, the sensing, control and coordination of such instruments pose many new design challenges. SSI missions will require precise relative sensing and control, fuel-efficient, fuel-balanced operation to maximize mission life and group-level autonomy to reduce operations costs. Enabling these new formation-flying capabilities requires precise relative sensing and estimation, enhanced control capabilities such as cooperative control (multiple independent spacecraft acting together), group-level formation management and informed design of a system architecture to manage distributed sensing and control-system resources.

This research defines an end-to-end control system, including the key elements unique to the formation flying problem: cooperative control, relative sensing, coordination, and the control-system architecture. A new control-system design optimizes performance under typical spacecraft constraints (e.g., on-off actuators, finite fuel, limited computation power, limited contact with ground control, etc.) Standard control techniques have been extended, and new ones synthesized to meet these goals.

In designing this control system, several contributions have been made to the field of spacecraft formation flying control including: an analytic two-vehicle fuel-time-optimal cooperative control algorithm, a fast numeric multi-vehicle, optimal cooperative control algorithm that can be used as a feedforward or a feedback controller, a fleet-level coordinator for autonomous fuel balancing, validation of GPS-based relative sensing for formation flying, and trade studies of the relative-control and relative-estimation-architecture design problems. These research contributions are mapped to possible applications for three spacecraft formation flying missions currently in development. The lessons learned from this research have been validated in simulation and experiment.
Acknowledgments

I'd like to thank the NASA Graduate Student Research Program, Hughes Space and Communications Company (now Boeing Satellite Systems), and the Stanford University Aerospace Robotics Laboratory for supporting my research.

I’d like to thank the technical experts who generously took time from their busy schedules to support my research: Robert Laskin of the Jet Propulsion Laboratory, Eric Gustafson of the Stanford University Applied Physics Laboratory, Doug Zimpfer of Draper Laboratory, Kurt Zimmerman of Integrinatics, Arthur Bryson of the Stanford University Department of Aeronautics and Astronautics, and Walter Murray of the Stanford University Department of Management Science and Engineering. In addition, the Numerical Algorithms Group and Real-Time Innovations provided software used in this work.

I’d like to thank the Aerospace Robotics Laboratory. Jane Lintott, Godwin Zhang and Gad Shelef all provided invaluable support during my many years here. My gratitude also goes to the ARL student computer system administrators (Dave, Andreas, Bob, Jorge, Chad, and several others): don’t think I didn’t appreciate all of your help (well, most of it anyway).

I am very grateful for all of the help I received from Sally Gressens and from Yasmeen Husain.

My deepest thanks go to my peers in the basement: Tobé, Bob, Gokhan, Gordon, Kortney, Bijan, Chris, Tim, Claudia, Jorge, Ed, Hank, Eric, the other Eric, and even the old Eric, Dave, Kurt, both Steves, Jason, Andrew (the other one), Mel, Heidi, Stef, H.D., and, of course, Bruce.

Finally, I thank my friends and my family (especially Katie) for the support that carried me through.
Contents

Abstract v

Acknowledgments vi

List of Tables xi

List of Figures xiii

1 Introduction 1

1.1 Motivation ....................................................... 2

1.1.1 Interferometric Astronomy .................................. 3

1.1.2 Formation Flying Applications ............................. 10

1.1.3 Motivation Summary ......................................... 14

1.2 Problem Statement ............................................... 14

1.2.1 The Virtual Spacecraft Bus ................................. 15

1.2.2 SSI-Observing Flow ........................................ 16

1.2.3 Formation Flying Mission Requirements ..................... 18

1.2.4 Formation Flying Control System .......................... 20

1.2.5 The Science Instrument .................................... 23

1.2.6 Problem Statement Summary ................................ 24

1.3 Related Research .................................................. 24

1.3.1 Spacecraft Formation Flying ................................. 24

1.3.2 Distributed Control .......................................... 25

1.3.3 GPS-Based Relative Sensing ................................. 26

1.4 Contributions of This Research ................................. 26

1.5 Overview of this Thesis .......................................... 28
2 The FormationFlying Testbed

2.1 Free-Flying Space Robots ........................................... 29
  2.1.1 Robot Structure ............................................. 31
  2.1.2 Computer System .............................................. 31
  2.1.3 Real-Time Control Software ................................... 31
  2.1.4 Communications ............................................... 31
  2.1.5 Electrical Power System ...................................... 32
  2.1.6 Reaction Wheel ............................................... 32
  2.1.7 Drag-Free Motion .............................................. 32
  2.1.8 Propulsion System ............................................. 33
  2.1.9 Pneumatic Gripper System .................................... 34
  2.1.10 Overhead Vision System ..................................... 34
  2.1.11 Indoor GPS System ........................................... 34

2.2 System Identification ............................................. 35
  2.2.1 Analytical Thruster Model .................................... 35
  2.2.2 System Identification Experiments ........................... 36
  2.2.3 Experimental Validation of the Revised Model ............... 41
  2.2.4 System Identification Summary ................................ 44

2.3 Testbed in Simulation ............................................. 45
  2.3.1 High-Fidelity FFTB Simulation ................................ 45
  2.3.2 Six-Vehicle Simulation ....................................... 46

2.4 Summary .......................................................... 47

3 GPS Sensing for Spacecraft Formation Flying ......................... 48

3.1 The Relative-Sensing Application of GPS .......................... 49

3.2 Estimation Problem ............................................... 50

3.3 GPS Issues ........................................................ 53
  3.3.1 Multipath .................................................... 53
  3.3.2 Self-Constellation ........................................... 54
  3.3.3 Integer Initialization ....................................... 54

3.4 GPS-Based Estimator Experiments ................................ 57
  3.4.1 Estimation Algorithms ....................................... 58
  3.4.2 Estimator Accuracy Performance Metric ....................... 60
3.4.3 Two-Vehicle Experiments ........................................... 61
3.4.4 Three Vehicles: Simulated Performance .......................... 64
3.4.5 Summary .......................................................... 67
3.5 Estimation Architectures ............................................. 68
  3.5.1 Example: Six-Vehicle Estimation Architectures ................. 68
  3.5.2 Estimation Architecture Summary ................................ 70
3.6 Conclusions ........................................................ 72

4 Formation Flying Control .............................................. 74
  4.1 Introduction ....................................................... 74
  4.2 Bang-Off-Bang Control ............................................ 75
    4.2.1 Derivation/Review ........................................ 76
    4.2.2 Deadband Design ........................................... 81
    4.2.3 Controller Experimental Performance ....................... 82
  4.3 Control for Two Vehicles ......................................... 84
    4.3.1 Two-Vehicle Optimal Control Law Derivation .................. 85
    4.3.2 Simulation Results .......................................... 95
    4.3.3 Experimental Results ....................................... 96
  4.4 Control for Three or More Vehicles ............................. 99
    4.4.1 Inverse Dynamic Optimization ................................ 100
    4.4.2 Heuristic (Sub-Optimal) Control Method ..................... 100
    4.4.3 Numerically Optimized Control .............................. 102
    4.4.4 Simulation Results .......................................... 108
  4.5 Control Architectures ............................................ 110
    4.5.1 Candidate Formation Flying Control Architectures .......... 111
    4.5.2 Control Architecture Simulation Results .................... 112
  4.6 Conclusions ...................................................... 115

5 Coordination ........................................................ 116
  5.1 Formation Coordinator ............................................ 117
  5.2 Fuel Balancing ................................................... 118
  5.3 Fuel-Balancing Coordinator Algorithm ............................ 119
    5.3.1 Algorithm Description ...................................... 120
    5.3.2 Algorithm Performance ..................................... 121
List of Tables

2.1 Estimated acceleration (m/s²) per thruster, estimated disturbance acceleration $d$, and resulting accumulated errors (m and m/s) at four air-pressure levels. ........................................... 37

2.2 Estimated acceleration (m/s²) per thruster, estimated disturbance acceleration $d$, and resulting accumulated magnitude of the errors (m and m/s) for each FFTB vehicle. ........................................... 43

2.3 Thruster model validation experiment results. ........................................... 43

2.4 Noise variances for three-vehicle simulation code. ................................. 46

2.5 Noise variances for six-vehicle simulation code. ................................. 46

3.1 Relative-position sensing for two vehicles. Computational cost per step and estimation performance is listed for several FFTB cases, including: simulation, post-processed experimental data, and real-time experimental estimation. EKF is extended Kalman filter, IWLS is iterated weighted least squares, and IEKF is iterated extended Kalman filter. ........................................... 62

3.2 Simulated relative-position sensing performance for three vehicles. Computational cost per measurement epoch and estimation accuracy are listed for sensing performed as three pairs and as one trio. ................................. 65

4.1 Two-vehicle relative-position offset response simulation (fuel numbers are normalized). Cost is found from Equation 4.44 with $\lambda_1 = 1.86, \lambda_2 = 1.99$. ................................. 96

4.2 Two-vehicle relative-position offset response experiment (normalized fuel). Cost is found from Equation 4.44 with $\lambda_1 = 1.86, \lambda_2 = 1.99$. ................................. 99

4.3 Simulation results for three three-vehicle relative-position controllers in simulation. Maneuver times and fuel used by each vehicle (in seconds of thrust) are listed for each controller. Costs are computed according to Equation 4.122 with $\lambda_1 = 0.5, \lambda_2 = 1.0, \lambda_3 = 2.0$. ................................. 109
5.1 Fuel used by three simulated formation flying spacecraft with a fuel-balancing coordinator for three $L_1$ values. Maneuver lasts 20 minutes in each case. . . 124

5.2 Maximum spacecraft-pair fuel-difference percentage at conclusion of a six-spacecraft aperture filling maneuver simulation; results shown without and with a fuel-balancing coordinator. . . . . . . . . . . . . . . 127

5.3 Fuel used by the vehicle using the most fuel, at the conclusion of the six-vehicle formation observing-sequence simulation without and with a fuel-balancing coordinator. . . . . . . . . . . . . . . . . . 128

6.1 Virtual spacecraft bus maneuver experiment results. Standard deviations of relative-position and relative-angle errors during the maneuver are listed for distributed and centralized control architectures. Total fuel use (in seconds of thrust) is also listed for each case. . . . . . . . . . . . . . . . . . . . . 136
List of Figures

1.1 An early concept for the NASA New Millennium Interferometer (Space Technology 3) mission. ......................................................... 4
1.2 The Very Large Array in New Mexico constructs images by interfering radio signals from an array of 27 radiotelescopes. The length of each leg is 13 miles. 6
1.3 The Keck Interferometer in Hawaii will combine the light from two 10 meter telescopes to increase image resolution. Light from each telescope is directed into a tunnel running beneath the two telescopes where the two beams are interfered. .......................................................... 7
1.4 A recent concept for the SIM mission. Notice the apertures on each end of the central boom: the observing baseline is fixed, thus the maximum image resolution is fixed. ......................................................... 8
1.5 The Terrestrial Planet Finder mission may use separated-spacecraft interferometry to synthesize an aperture of potentially unlimited size to form an observatory in space capable of detecting and characterizing planets in orbit about distant stars. ......................................................... 9
1.6 Multi-level control concept for SSI alignment. ............................... 16
1.7 SSI observing procedure flow diagram. ........................................ 17
2.1 Stanford’s formation flying testbed (FFTB) consists of three free-flying space robots floating on air bearings over a 9 ft by 12 ft granite surface plate. . 30
2.2 This schematic shows the free-flying space robot’s compressed-air system including: supply valve (1), spherical tanks (2), three pressure regulators (3,4,5), eight thrusters (6), pneumatic actuators (7), and floatation cushion (8). ................................................................. 32
2.3 Each vehicle has eight compressed-air thrusters (two each in +x, -x, +y, and -y) which can provide translational thrust as well as torque. The diameter of one thruster (shown) is 1 cm.

2.4 Velocity versus time at four air-pressure levels. Measured and PEM-predicted values are shown (solid) with the analytic model-predicted values (dash-dot) for comparison.

2.5 Acceleration per thruster (a) versus off-time for 2 thrusters (*) and for 4 thrusters (*).

2.6 Acceleration per thruster (a) versus on-time for 2 thrusters (*) and for 4 thrusters (*). The marked point (O) represents an aberration in the data and was ignored.

2.7 Identification experiment data for vehicle 1 (Huey) firing 4 thrusters continuously for 10 seconds. Measured and model-predicted values are shown with resulting errors (position and velocity) below.

2.8 Position (x and y) versus time for three vehicles following a trajectory with feedforward (open loop) control only: desired path (dotted), vehicle data without (solid) and with (dashed) thrust correction maps. Note that dotted and dashed lines coincide on the Louie-y plot.

3.1 GPS measurement definition for the FFTB. The range from pseudolite k to antenna j on vehicle i is defined as $D_{ijk}$.

3.2 Definition of relative-position estimate-error vector $\vec{E}$. $\vec{R}_{\text{vis}}$ and $\vec{R}_{\text{GPS}}$ are the vision and GPS-based relative-position estimates. Performance metric $J$ is the magnitude of $\vec{E}$.

3.3 Computational cost (per epoch) of centralized estimation versus number of spacecraft. Theoretical upper limit (solid line) is based on analysis of IEKF algorithm implemented for FFTB vehicles. Dotted line is expected cost, extrapolated from known cost of 2- and 3-vehicle cases.

3.4 Three estimation architectures for a six-vehicle satellite cluster. 1) Distributed (pair-based), 2) Distributed (trio-based), and 3) Centralized.

3.5 Estimation accuracy versus computational cost for three architectures: 1) distributed (pair-based), 2) distributed (trio-based), and 3) centralized. Solid circles are based on results in Section 3.4.4; the centralized accuracy and cost (broken circle) is extrapolated from those results.
4.1 Phase-plane portrait of single-vehicle bang-off-bang controller. Two switching lines and the optimal trajectory from an initial position of 0.2 and initial velocity -0.15 are plotted (the starting point is marked by a 'o' on the plot).

4.2 Phase-plane portrait of relative controller experimental performance with GPS-based sensing only. This plot shows GPS-based estimated-state history. The estimate includes some time-lag plus multipath noise, distorting the expected parabolic trajectory shape (shown in Figure 4.1).

4.3 Phase-plane portrait of two-vehicle cooperative controller. Shows four switching lines labelled \( g_1 - g_4 \) and the optimal trajectory from \( x(t_0) = 1 \) m and \( v(t_0) = -0.4 \) m/s (the starting point is marked by a 'o' on the plot). The respective thruster settings are noted on the figure, for example at \( x < 0, v = 0 \), the thruster settings are \(+/0\), i.e. \( u_1 = +1, u_2 = 0 \).

4.4 Phase-plane portraits of two-vehicle cooperative controller for \( K_1 = K_2 = 0.013 \), \( \lambda_1 = 1.0 \) and \( \lambda_2 = 1.5, 2.5, 5 \). Note how the \( g_4 \) switching line (solid line) changes location as \( \lambda_2 \) increases.

4.5 Position-versus-time experimental data for three different controllers of the relative spacing of a pair of vehicles. The initial relative separation is 2 m. The desired relative separation is 1 m, which is achieved in each case. Experimental data is shown for the PD controller (\( \triangle \)), two independent fuel-time-optimal controllers (\( o \)), and the fuel-time-optimal cooperative controller (x). The latter is better: see Table 4.2.

4.6 Relative-position error for relative-controller experiment. Experimental data is shown for the PD controller (\( \triangle \)), two independent fuel-time-optimal controllers (\( o \)), and the fuel-time-optimal cooperative controller (x).

4.7 Optimal actuator histories for a three-vehicle relative-position offset response example, computed via the inverse dynamic optimization technique. The resulting motion is shown in Figure 4.8.

4.8 Relative-state histories (components in one inertial direction) for a three-vehicle relative-position offset response example, computed via the inverse dynamic optimization technique. Relative-position and relative-velocity errors are plotted for vehicles 1 and 2 (x) and 2 and 3 (o).
4.9 Optimal actuator histories for a three-vehicle relative-position offset response example, computed via switch-time optimization. The resulting motion is shown in Figure 4.10.

4.10 Relative-state histories (components in one inertial direction) for a three-vehicle relative-position offset response example, computed via switch-time optimization. Relative-position and relative-velocity errors are plotted for vehicles 1 and 2 (dash-dot) and 2 and 3 (solid). Compare with Figure 4.8.

4.11 Switch-time optimizer configured for feedback control.

4.12 Simulated three-vehicle maneuver with the switch-time optimizer operating as a receding-horizon controller. The three absolute vehicle positions (along one axis) are plotted versus time on the upper plot (the desired separation is 1 m between vehicles one and two and vehicles two and three). The absolute velocities are shown on the lower plot with re-planning events indicated by vertical lines.

4.13 Simulated absolute position versus time for three vehicles under three relative-control laws: distributed (*), heuristic (o) and switch-time optimizer (Δ).

4.14 Three six-vehicle control architectures. Each vehicle with a “*” regulates its state relative to other spacecraft as indicated by the arrows.

4.15 Simulation-based fuel/time trade curves for three control architectures for six vehicles performing 15-point stop-and-go observing slews: distributed (o), hierarchy (*), and centralized (Δ).

5.1 The coordinator is the higher level in a two-level decentralized control formulation for large-scale problems. The lower level consists of many subsystem controllers communicating with the coordinator via information variables $I_1, \ldots, I_N$.

5.2 Fuel-balancing coordinator functional flow diagram. The coordinator tunes the fuel-balance using the information variables $\lambda$ and $f$.

5.3 Total fuel used versus total time for a six-vehicle formation to complete a 15-point stop-and-go observing slew. Simulation with centralized relative control and a fuel-balancing coordinator for three values of $\Lambda$: 1.8 (o), 6 (o), and 18 (Δ).
5.4 Fuel used by three simulated formation flying spacecraft (numbered one (\(\triangle\)),
two (\(\circ\)) and three (\(\times\))) with a fuel-balancing coordinator operating once
per minute for three values of \(L_1\): 0, 0.5, and 1. \(\Lambda = 3\) throughout, total
maneuver time was 20 minutes in each case. See Table 5.1. ............... 123

5.5 Time history of \(\lambda\) for three-spacecraft aperture-filling maneuver simulation
with a coordinator, with \(L_1 = 0.5\), and spacecraft one starting with less fuel
than two and three. Spacecraft are numbered one (\(\triangle\)), two (\(\circ\)) and three (\(\times\)). 125

5.6 Fuel-use history for a three-spacecraft aperture-filling maneuver simulation
with coordinators running at three different update rates and \(\Lambda = 3, L_1 = 0.5\). These plots show that 1/120 Hz is an acceptable update rate, with little
improvement from higher rates. ........................................ 126

6.1 Phase-plane portrait of relative state error in one axis with GPS-based sensing only (experimental data). Deadband region and control switching lines
are also shown. ........................................................... 131

6.2 Laser tracking performance with coarse control (GPS plus thrusters) only.
(See footnote on following page.) ..................................... 132

6.3 Laser tracking performance with both control layers active. ............... 134

6.4 Virtual-spacecraft-bus maneuver experiment with distributed relative control.
The starting and ending positions are shown connected to illustrate the
equilateral-triangle formation (dash-dot lines). Two intermediate points are
marked (\(\circ\)'s and \(\triangle\)'s). The starting triangle is on the right. At the start,
Huey is at the bottom of the triangle and Heavenly and Louie are at the
upper left and upper right, respectively. Position histories in the x-y space
of the FTFB reference frame are plotted for all three vehicles (solid lines).
The desired absolute positions are also plotted (dotted lines). ............... 137

6.5 Virtual-spacecraft-bus maneuver experiment with centralized relative control. 138

A.1 Interference of light from source (A) arriving from two different slits in plane
(B) at viewing plane (C); constructive interference occurs at point (D). .... 153

A.2 An interferometer consisting of two apertures in plane B separated by baseline
\(b\) and four detectors in plane C. The detectors are spaced along plane C a
distance \(x\) from the boresight. Observing target A is located a small angle \(\theta\)
off the instrument boresight. ............................................. 156
Chapter 1

Introduction

This dissertation addresses control-system design for the spacecraft formation flying problem. This problem consists of multiple spacecraft performing a unified task by coordinating their actions. For an intriguing example of the benefits of this approach, consider a multi-element interferometer. On a single-spacecraft bus, the maximum observing baseline is limited by the dimensions of the spacecraft. However, placing interferometer elements on separate spacecraft removes this constraint: the instrument baselines are limited only by the maximum separation the vehicles can achieve while still flying in precise formation. The potential benefits of such formation flying missions are driving many current research efforts aimed at controlling a formation of spacecraft.

This new class of space mission requires new formation flying control designs that coordinate and control independent spacecraft in the formation as a unified system. The primary objective (e.g., collecting science data) must be met while balancing many secondary needs, including: relative-state sensing, multi-vehicle control, resource allocation, inter-vehicle communications, autonomy, and trajectory planning. The capability and complexity of formation flying missions increase with the number of vehicles. Therefore, new control-system tools and insights are required to enable upcoming generations of increasingly capable spacecraft formation flying missions.

This thesis documents an efficient formation flying control system capable of cm-level relative alignment. It derives new multi-vehicle relative sensing and control tools. It addresses design challenges and trade-offs at appropriate levels within the control-system framework.
This work examines the design at each level via simulation and experiment using a three-vehicle formation flying testbed which simulates physically the drag-free dynamics of a spacecraft formation in a plane.

Section 1.1 lists numerous space and terrestrial formation flying applications, including separated-spacecraft interferometry which will revolutionize space-based astronomy. Separated-spacecraft interferometry requires a formation flying control system, as detailed in Section 1.2. Section 1.3 lists related research efforts in spacecraft formation flying, distributed control, and GPS-based relative sensing.

1.1 Motivation

Formation flying of multiple spacecraft is an enabling technology for many future space-science missions including enhanced stellar optical interferometers and virtual platforms for Earth observing and space science. The goal of these missions is to accomplish science tasks using a distributed array of relatively simple, but highly coordinated vehicles. These vehicles will be controlled so that the physical spacecraft buses form a virtual spacecraft bus whose size (for the purposes of observing baselines and instrument apertures) is limited not by the physical dimensions of the vehicles, but rather by the active workspace of the formation control system (e.g., possibly limited by the range of the relative sensing system, or by the precision of the attitude control systems or by the sensitivity of the science instrument). Ultimately, a space-based interferometer that takes advantage of these extended baselines may yield the first pictures of planets orbiting other stars.

Virtual-spacecraft-bus missions benefit compared to more conventional, monolithic-spacecraft missions in several important ways:

1. *Instrument Performance*

   Increased instrument baselines translate into improved resolution. For example, the angular resolution of an astronomical interferometer with baseline $B$ at observing wavelength $\lambda$ is proportional to $\lambda/B$.

2. *Flexibility and Robustness*

   Replacing one failed part of a distributed science instrument is easier than replacing the entire instrument. Similarly, upgrades or enhancements can be made to the instrument by “swapping in” spacecraft with upgraded parts when they are ready.
3. **Cost**

Splitting the complex science payload into several pieces simplifies the design for the supporting spacecraft bus: several small spacecraft are easier to design, test and launch than a single large spacecraft. This may lead to significant cost savings.

Systems that are not space-based also benefit from formation flying technologies. In particular there has been great interest in autonomous formation control for aircraft, ground vehicles, and mobile robots. The benefits listed above carry over to some extent into these arenas; and additional advantages appear. For example, aircraft flying in a tight “V” formation benefit from reduced total aerodynamic drag (migrating flocks of birds are believed to take advantage of this effect for long-distance flights). Some space and non-space applications of formation flying technology are outlined in Section 1.1.2.

This work was initially motivated by the NASA New Millennium Interferometer (NMI) mission (see Figure 1.1) which was later renamed Space Technology 3 (ST3). As originally conceived, this mission was to consist of a three-vehicle formation carrying an optical-interferometer payload.\(^1\) ST3 will be the first demonstration of separated-spacecraft interferometry (SSI). While ST3 will not have the necessary resolution for observing extra-solar planets, follow-on SSI missions will leverage the technology developed for ST3 — including formation flying — to make interferometric observations at increasing resolution. Such missions will detect, characterize and ultimately image planets orbiting stars other than our sun.

### 1.1.1 Interferometric Astronomy

At present, astronomical targets such as stars are imaged at many frequencies. However, strong interest lies in observation at visible and near-visible light frequencies. The goal that has driven the development of astronomical tools beginning with Galileo’s first telescopes in 1610 is to improve resolution in order to allow scientists to learn continually more about the universe.

Today, visual astronomy has two fundamental limits: ground-based observatories are limited by atmospheric disturbance effects, and space-based observatories are limited by the maximum diameter of mirrors that can be launched into space. Adaptive optics are

\(^1\)It was subsequently redesigned as a two-vehicle mission with the same science goal.
improving the performance of ground-based systems by compensating for atmospheric effects in real time (see Reference [1]). However, by moving to space the atmosphere problem is removed entirely. Also a space-based telescope has more pointing freedom, can operate without interruption due to daylight, does not suffer from light pollution, is isolated from vibration sources, and is not constrained in size by the need to compensate for gravity (i.e., with a large support structure). Telescopes operating in space have many advantages over ground-based telescopes.

Aside from cost, the primary limitation on the performance of a space telescope like the Hubble Space Telescope is that the size of the primary mirror (which determines the instrument's resolution) is limited to what can fit inside a launch vehicle. Even segmented (deployable) mirrors are still fundamentally limited in resolution to the maximum launchable mirror size. Space-based interferometers overcome this limit because they do not use a primary mirror.
The technique of interferometry has been in use for years by ground-based radioastronomy observatories to improve the resolution of radio-wave images. Instead of a primary mirror, an optical interferometer collects light from two distant apertures, and directs it to a common point, where the two light beams are combined, and the resulting interference fringe is measured. A set of fringe measurements from many different observing configurations can be transformed into an image with resolution equal to that which would be achieved by a mirror of diameter equal to the separation of the two apertures (see Appendix A). In effect, a large instrument aperture is formed piece-by-piece from individual interference fringe measurements. This technique is called aperture synthesis. If each one of the apertures is on a separate spacecraft, then there is no fundamental limit to the imaging resolution of such an instrument.

In 1991, the National Research Council recommended that NASA pursue space-based optical and infrared interferometry as a prerequisite for next-generation space astronomy missions (see Reference [1]). As NASA undertook this task, it became apparent that a fundamental design decision must be made. Should space-based interferometers be monolithic (contained on one spacecraft) or separated? Preliminary designs have been made for both kinds of mission (briefly reviewed below). Separated spacecraft interferometers have the advantage of longer baselines, but require the ability to fly in precise formation. Thus formation flying is a necessary technology for SSI missions, which could provide — by orders of magnitude — the best potential image resolution. The focus of this thesis is developing the control-system tools that will enable efficient spacecraft formation flying for applications such as SSI.

Appendix A presents an overview of the principles of interferometric telescopes. The following sections review some current and planned interferometers (both ground-based and space-based).

Earth-Based Interferometers

Interferometers were first developed by A.A. Michelson for a variety of applications\(^2\). One application was to measure the diameter of stellar disks. Michelson calculated the relation

\[^{2}\text{Michelson was primarily interested in detecting the motion of the Earth relative to the} \text{ether of space which would presumably have been fixed in a universal inertial frame. He used an interferometer to look for the minute change in the pathlength of a light beam that would be expected for light traveling in the direction the Earth was moving relative to the ether. His negative results laid the ground work for Einstein's special theory of relativity. Michelson's work is documented in Reference [2].}\]
between the baseline at which the fringe vanishes and the diameter of the disk and then measured the baseline at which the fringe vanished. In this way, the diameter of the star Betelgeuse (α Orionis) was measured at Mount Wilson Observatory using a 20ft interferometer in 1920 by Michelson and F. G. Pease (see Reference [3]). Michelson’s interferometer technique was also applied to measure the angular separation of the double star Capella (see Reference [4]). These astrometric measurements were quite useful and astrometry science continues to be important, but interferometric imaging was not possible with the optical instruments of the time.

The first imaging interferometers thus operated at radio wavelengths. Separate radiotelescopes were combined to form one instrument using aperture synthesis. The radio waves were recorded on tape at each radio telescope. The recordings were then interfered with each other, aided by timestamp signals on the tapes. The first radio interferometer was built in 1946 (see Reference [5]) using two radio antennas. It was used to measure radio wave sources in the Sun. The development of the closure phase technique enabled radio source imaging in 1974 by A. E. E. Rogers et. al. (see Reference [6]). The Very Large Array (VLA) in New Mexico (shown in Figure 1.2) began performing aperture synthesis measurements with its array of 27 radiotelescopes spread over 22 miles in 1980. These telescopes
Figure 1.3: The Keck Interferometer in Hawaii will combine the light from two 10 meter telescopes to increase image resolution. Light from each telescope is directed into a tunnel running beneath the two telescopes where the two beams are interfered.

are mounted on rails, which allows them to be configured for many observing baselines. The use of many telescopes allows a large number of baseline measurements to be formed during one observing period. However, the pointing of the array is limited by the Earth’s motion; stellar targets must be in view in order to be imaged by the VLA.

Spurred by the development of sophisticated optical sensors (including the CCD), interest in ground-based optical interferometers has grown recently. The first two-telescope optical interferometer was constructed in 1975 by A. Labeyrie to observe Vega (see Reference [7]). The first use of an optical-telescope interferometer for aperture synthesis was in 1987 (see Reference [8]). This interferometer employed a single telescope and an aperture mask. In 1996 the double star Capella was imaged by the COAST instrument, an array of four optical telescopes performing aperture synthesis (see Reference [9]). At present, COAST and several other telescope arrays are currently being developed and tested, with
even more ambitious projects planned. Reference [10] outlines some of the current and planned ground-based instruments, including the Keck Interferometer project which will combine light from two very large (10 meter) optical telescopes (see Figure 1.3).

**Space-Based Interferometers**

The 1991 National Research Council report ([1]) recommended that NASA pursue space-based optical and infrared interferometry as a prerequisite for next-generation space astronomy missions. Scientists have identified many potential science targets that could be observed by space-based optical interferometers (see Reference [11]).

To meet these goals, groups in NASA have developed ground-based testbeds to refine interferometer control methods and have outlined possible space-based interferometer missions. Reference [12] outlines some of these efforts, including the Space Interferometry Mission (SIM). A recent SIM concept is shown in Figure 1.4. Other space-based interferometry missions proposed by NASA include the Orbiting Stellar Interferometer (OSI), and the Small OSI for Narrow Angle Astrometry with Two Apertures (SONATA) which is an astrometric mission only (see References [13, 14]).
CHAPTER 1. INTRODUCTION

Figure 1.5: The Terrestrial Planet Finder mission may use separated-spacecraft interferometry to synthesize an aperture of potentially unlimited size to form an observatory in space capable of detecting and characterizing planets in orbit about distant stars.

Such monolithic missions have observing-instrument baselines limited by the size of the spacecraft (usually 10 meters or less). In addition, the structure that supports the instrument must be actively controlled to suppress the vibrations that are common in lightweight space-based structures. One alternative is to employ multiple spacecraft flying in formation ([12]).

SSI - Long-Baseline Space-Based Interferometry

To achieve the resolution necessary to image planets orbiting stars other than our own, an interferometer’s baseline must be quite long (e.g., 10-1000 kilometers). Separated spacecraft interferometry offers the only way to achieve such baselines. However, the baselines between observing telescopes must be controlled with a high level of precision. Thus spacecraft formation flying is an enabling technology for such missions. Several SSI missions have been proposed. Reference [15] is an early SSI proposal (1983). It outlines a three-vehicle mission
in Earth orbit called Spacecraft Array for Michelson Spatial Interferometer (SAMSI). Reference [16] describes the proposed MUSIC mission which would be an eighteen-spacecraft mission with observing baselines from 1 to 100 km.

The NASA New Millennium Program’s Space Technology 3 Mission (shown in Figure 1.1) was originally planned to be a formation flying-technology demonstration mission with three vehicles forming an interferometer. Reference [17] describes the mission design for ST3 as of March 1998. Though ST3 was subsequently redesigned as a two-vehicle mission, the three-vehicle problem is of interest as a step toward formation flying with many vehicles.

SSI with more than two vehicles allows simultaneous measurements at multiple baselines. An SSI mission with three or more spacecraft collects measurements from multiple baselines during a single observing period, and thus the instrument can take a faster “snap-shot” of the science target. Current design options for the Terrestrial Planet Finder (TPF) mission include SSI with five spacecraft working together (shown in Figure 1.5). Reference [18] lists the goals of the TPF mission.

1.1.2 Formation Flying Applications

Current interest in non-SSI applications of spacecraft formation flying has lead to a variety of proposed space missions. Some of these will now be described, followed by some terrestrial formation flying applications.

Spacecraft Applications

Formation flying techniques will be applied to upcoming space missions with a variety of goals. Earth science missions will array virtual science instruments in space to measure the gravity field and to perform coordinated imaging. Colocated satellites (sharing geosynchronous orbit slots) will use a close formation of spacecraft to improve capability and flexibility of communications satellites. Military Earth-observing missions will use formation flying to form virtual radar antennas in orbit. A NASA mission will attempt to detect gravity waves using three spacecraft in precise formation.

The GRACE mission will employ two Earth-orbiting spacecraft flying in precise formation. A microwave interferometer will track the relative range between them. The variations in this range correspond to variations in Earth’s gravity field, which will allow scientists to characterize the behavior of the oceans (see Reference [19]). The deep-space LISA mission
CHAPTER 1. INTRODUCTION

will use three spacecraft flying in precise formation. This mission will try to detect gravity waves from galactic and extra-galactic sources by measuring the variation in ranges between the spacecraft using laser interferometers (see Reference [20]).

Synthetic-aperture radar (SAR) observatories in space formed by fleets of spacecraft in formation will provide the capability to closely monitor activity on Earth. Such missions will have military and defense applications. The Air Force is currently developing the TechSat 21 technology demonstration mission to explore this concept (see Reference [21]). A coordinated Earth-observing program involving Landsat-7 and Earth Observing-1 (EO-1) flying in formation will validate sensing technology by observing the same site near-simultaneously with two different instruments (see Reference [22]). Finally, placing several communications satellites in a coarse formation around an assigned orbital slot greatly enhances capability and flexibility. However, the satellites must all stay near the slot without colliding while preserving fuel. Reference [23] examines orbit designs for such satellite clusters.

Land Vehicles

There has been a high interest level recently in the intelligent vehicle highway system (IVHS). In this concept, passenger automobiles benefit from automated systems that can assist the driver or even take over the driving task. Groups of automated vehicles may travel together in tight packs with large separations between the packs; this concept is called platooning. Platooning is an interesting formation flying application that involves precise control within platoons, coarse control between platoons and potential coordination with the highway infrastructure (e.g., traffic flow monitoring and ramp metering).

Among the goals of this system are increased safety, reduced driver stress, increased highway capacity, increased mobility, and improved vehicle efficiency (which decreases fuel use and pollution). Reference [24] suggests that an automated platooning system with 20 cars per platoon separated by 1-2 meters could improve highway capacity by 400%. A platoon of cars traveling together results in aerodynamic drag reduction for all cars (including the leader) in the platoon, with an average drag reduction up to 50% which translates to 25% fuel savings and up to 50% reduction in pollution ([24]).

The potential advantages of the IVHS are considerable. However, there are many challenges to implementing such a system. One key challenge is to design the control system for many vehicles driving in close proximity. Though this application is quite different from
space interferometry, it is nevertheless a formation flying problem. The designers of the IVHS will face many of the same challenges discussed in this thesis including: sensing, low-level control, and fleet-level coordination. Initial experiments in References [24, 25, 26] have demonstrated two-, three-, and four-car formations operating on closed roads.

Another ground-based application of formation flying is military vehicle formations. A formation of scouts using distributed sensing achieves full coverage of the environment. The advantages of automating some or all of the scout vehicles include removing soldiers from hostile environments and enabling systematic and possibly continuous coverage of patrol areas. Experiments have already demonstrated formation behaviors on a two-vehicle testbed consisting of two High Mobility Multipurpose Wheeled Vehicles (HMMWVs). That work aims to field a robotic scout platoon for the US Army (see Reference [27]).

Agricultural applications could also benefit from formation flying. Autonomous farm vehicles traveling in formation could potentially increase efficiency and safety, reduce the dependence on highly skilled human operators and allow continuous operation (i.e., during night and bad weather). Related research has already shown the ability to automate a tractor pulling an agricultural implement using carrier-phase differential GPS as a sensor (see Reference [28]).

Mobile Robots

Teams of robots acting together can be more capable and more flexible than individual robots acting separately. For example, a team of small robots operating in a factory environment may provide the capability to handle payloads too large for one robot while offering more flexibility and robustness. However, the team of robots must work together as a system. The control of mobile robot teams is another application for formation flying.

Terrestrial applications for mobile-robot teams include: construction, exploration, surveillance, and cooperative payload transportation. Teams of mobile robots are well suited to simple tasks in a factory environment such as material and inventory transport and assembly. Robot teams will be useful in environments hostile to humans. Tasks such as lunar base construction, volcanic exploration, and nuclear waste site clean-up are obvious applications for mobile robot teams.

Reference [29] presents some experimental results for a two-robot team cooperating to carry an object while following a trajectory. Reference [30] presents experimental results for many robots performing a cooperative box-pushing task. References [31, 32] present
simulation results for controlling larger groups of mobile robots (from three to thirty) to initialize and move as formations. Reference [33] presents simulation results for mobile robots performing a security task: hunting and capturing an intruder.

**Aircraft**

As mentioned, one benefit of formation flying for fleets of aircraft is the reduced total aerodynamic drag for the formation when flying in a “V” pattern. Another benefit is reduced pilot workload and improved performance for formation flying by squadrons of military aircraft.

Reference [34] discusses the design of an autonomous formation of aerial vehicles capable of very long flight duration. Such a system relies on solar power and efficient energy storage, but the enabling technology is the reduction in aerodynamic drag due to precise formation flying. A formation of fifteen solar-powered aircraft is proposed. Each aircraft has an efficient but modest structural aspect ratio of twelve. However, the formation as a whole achieves a much higher effective aspect ratio without the structural problems inherent in very long, slender wings. In addition, the formation is extremely robust due to vehicle redundancy. Overall, in formation flying mode the system is expected to achieve a reduction of fifty per cent in the power necessary to sustain flight.

The military currently takes advantage of aircraft formation flying (under human control) to keep groups of planes together so they can operate more efficiently and effectively. The development of autonomous formation flight control offers additional benefits and can be applied to current missions where manual formation flying is already employed. Reference [35] develops formation flight controllers that could work in conjunction with current autopilots that are standard on modern military planes. Autonomous formation flight control is expected to achieve robust formation maintenance using less energy. In addition, the pilot workload can be reduced during autonomous operation so that the pilot can be more effective at the strategic level.

**Future Applications**

Additional formation flying applications will surely appear in the future. Any task that may be performed more cheaply, safely, or effectively by an automated team of agents (spacecraft, aircraft, land vehicles or mobile robots) will benefit from formation flying techniques. Potential applications in the future could include the systematic remote exploration
of Mars by ground vehicles or aircraft, the autonomous construction of a base on the moon (or in space) by a team of mobile robots, or the autonomous inspection and maintenance of deep-sea oil-drilling platforms by fleets of autonomous underwater robots.

1.1.3 Motivation Summary

Formation flying has many potential applications. One exciting application is SSI which will enable long-baseline interferometric measurements in space. SSI enables the high resolution necessary to image extra-solar planets. However, coordinating many vehicles to act as a single system is a challenging problem. For the SSI application, small spacecraft with limited resources must be coordinated to maintain precise relative positions. The formation flying controller must conserve fuel in all vehicles as long as possible to maximize the science capability. A global relative-sensing system is needed to track the formation states. The formation needs a high level of autonomy because directly controlling the formation from the ground will be inefficient and possibly even impractical (e.g., for a large constellation of vehicles).

The following section presents a framework for the spacecraft formation flying control system and defines the scope and role of formation flying in SSI missions.

1.2 Problem Statement

The formation flying control system transforms separate spacecraft into a unified virtual spacecraft bus (VSB), as illustrated in Figure 1.1, which is equivalent to a highly capable and flexible monolithic spacecraft bus from the science objective’s perspective. Precise relative sensing and cooperative control are required to achieve this challenging goal. Fuel must be conserved, and fuel-use must be balanced between spacecraft to prevent one vehicle from prematurely exhausting its supply. Relative-state estimation and relative control will be distributed among the vehicles, which necessitates the investigation of efficient and effective control and estimation architectures. These new control-system tools will enable VSB separated-spacecraft interferometry missions.

Sections 1.2.1–1.2.3 describe the virtual spacecraft bus concept, typical mission flow, and requirements for the formation flying control system. Section 1.2.4 organizes the formation flying control system into three functions: sensing, control and coordination. Finally, Section 1.2.5 briefly outlines how the finer levels of alignment will be achieved on SSI missions.
1.2.1 The Virtual Spacecraft Bus

The goal of a separated-spacecraft interferometer (as discussed in Section 1.1.1) is to form an image by measuring interference fringes at many baselines. For each baseline, light gathered at two observing points is directed along precisely controlled paths to a central combining point where the light interferes. The light paths must be controlled to within a small fraction of a wavelength of light in order to form the interference fringe at the combining point. Each measurement baseline represents one point in the $u,v$ plane (see Appendix A).

This precise alignment must be achieved during observations; yet between observations the vehicles undergo maneuvers (e.g., to initialize the formation, to move to a new $u,v$ point, or to observe a new target). A multi-level control strategy consisting of successively finer, but increasingly limited-range alignment mechanisms is flexible enough both to regulate large-scale formation maneuvers and to precisely align a separated-spacecraft interferometer. Each controller depends on the previous control layer to maintain the system state within that layer's workspace. When the finest control layer is active, the interferometer level of alignment is achieved. Figure 1.6 illustrates the multi-level control-system concept with three alignment levels. The outermost level initializes the formation to cm level using GPS-based sensing and spacecraft thrusters plus reaction wheels. The optical-alignment level measures relative position and velocity states much more accurately (micrometer level); it nulls relative-position rates using thrusters, and aligns the interferometer light paths using steered mirrors. Optical alignment then enables a fringe-tracking system to find interference fringes which are the desired science data.

The formation flying control system acts as the first control layer; it transforms the separate spacecraft into a virtual spacecraft bus. The instruments on each spacecraft then behave as if they are rigidly connected, without the extra mass and vibration associated with structural connections. In addition, the baselines are flexible and are limited only by the control system's range. However, the fuel cost is higher because thrusters — instead of structural connections — maintain the relative formation. Once the VSB is established, the remaining alignment issues are the same as aligning the optics on a rigidly-connected interferometer. Reference [12] provides a good discussion of these issues for space-based interferometry.
CHAPTER 1. INTRODUCTION

1.2.2 SSI-Observing Flow

Figure 1.7 depicts a typical SSI mission’s observation procedure for one observing target. The observation begins with the vehicles moving to the first observing baseline. The formation flying control system moves the vehicles into a desired relative configuration within an error bound (e.g., 1 or 2 centimeters). Once this alignment is achieved, an optical-alignment
layer takes over. Using light detectors, mirrors and lasers steered through the optical paths, the relative locations of the instruments are sensed very accurately (micrometer level). The optical elements are aligned with steering mirrors based on this sensing. The thrusters and reaction wheels can continue to null relative rates using this more-accurate sensing information prior to interferometer alignment.

Once the optics are aligned sufficiently and relative drift rates are minimized, the interferometer adjusts the optical pathlength (using delay lines) to find and track the interference fringe. Thruster firings will be stopped because the interferometer is working over too small a range to tolerate thruster disturbances. The fringe data will be recorded for some observing period (e.g., about 1 hour for ST3 [36], or about 2 hours for TPF [37]). The data can continue to be recorded as long as the spacecraft remain in formation (to some tolerance).
If this “lock” is lost, the alignment sequence is repeated — including a corrective maneuver using the thrusters.

When enough data has been collected, the optical layers disengage, and the outermost layer (the formation flying control) moves the vehicles to the next observing baseline configuration. When data has been taken at all baselines, an image of the target can be synthesized, and the vehicles can be configured to observe the next target.

A quiet observing scheme (without thruster firing during data collection) requires careful relative alignment. To illustrate this problem, consider an example based on some early ST/3 design parameters (from Reference [38]). The spacecraft mass is 150 kg, and pulsed plasma thrusters (PPTs) provide a minimum impulse bit of 60 μNs. Therefore the minimum ΔV provided by thrusters is 0.4 μm/s. Using relative-rate information from the optical-alignment layer, the control system should be able to reduce the relative drift rate to μm/s-level (about twice the minimum ΔV). If the alignment window is 1 cm wide and the vehicles are near the center, then the formation remains aligned for ~80 minutes (~5 × 10^{-3} m / 1 × 10^{-6} m/s). This is sufficient to allow 60 minutes of data (the ST3 observing requirement) to be taken before lock is lost, with some margin for error. This example illustrates the utility of the VSB control concept. Future SSI mission designs will trade-off quiet time, alignment-window size, and thruster choice to achieve required observing capabilities. The following section examines three planned spacecraft formation flying missions to which the VSB control approach applies.

1.2.3 Formation Flying Mission Requirements

Design studies for three planned spacecraft formation flying missions produce a list of system requirements. This thesis addresses the requirements for these missions with new control-system concepts, tools and insights.

TechSat 21 (described in References [39, 40]) is an SAR mission with clusters of spacecraft in low Earth orbit (LEO). Multiple alignment layers will be required for the precise relative sensing necessary to synthesize the radar images. GPS is a likely candidate for the outermost sensing layer, since it is available in LEO. The relative-state sensing must be accurate without excessively high communications or processing requirements. Thus effective estimator design and an efficient estimation architecture are key to the success of this mission.
CHAPTER 1. INTRODUCTION

Reference [17] specifies the alignment requirements for Space Technology 3\textsuperscript{3}. The relative range must be aligned within 1 cm over baseline separations from 100 m to 1 km. The relative angles between spacecraft will be sensed to ±1 arcmin, which corresponds to ±30 cm at 1 km separation. The optical-metrology system will be able to search within this range. Once the metrology system is active, inter-spacecraft angles can be determined to ~1 arcsec, which corresponds to 0.5 cm at 1 km separation. Interferometer alignment layers reduce the error until the light pathlength is controlled to ~5 nm. Efficient relative control is required for coarse alignment, for moving the vehicles to new baselines, and for retargeting maneuvers.

Reference [37] defines a candidate five-spacecraft configuration for the Terrestrial Planet Finder mission. Spacecraft separations during observations will vary from 10 m to 1000 m. The alignment system will be based on ST3, but has not been designed yet. Fuel-efficient relative control and a control architecture that efficiently distributes the relative control task among the vehicles will be required. Autonomous fuel-balancing and other formation management functions will be critical to achieve maximum mission life.

These three missions illustrate how the VSB approach to control will enable new space-based science applications. Chapter 7 describes how the present research maps into the designs of these missions. To efficiently form the VSB, a formation flying control system must address the challenges of this new multi-spacecraft control problem:

1. Cooperative relative-state estimation
2. Fuel-efficient relative-state control
3. Autonomous inter-vehicle coordination
4. Informed selection of centralized or distributed estimation and control architectures

This thesis presents a formation flying control system that meets these challenges. Insights are provided for the formation flying problem, and performance and requirements trade-offs are identified for the control- and estimation-architecture design problems. The formation flying control system has been experimentally validated on a three-vehicle testbed. The following section describes the pieces of the formation flying control system in more detail.

\textsuperscript{3}as of March, 1998
1.2.4 Formation Flying Control System

The formation flying control system senses the relative states (inter-spacecraft ranges and rates) and moves the spacecraft by means of their thrusters in order to initialize, maintain, and reconfigure the virtual spacecraft bus. The architectures for distributing the relative-state sensing and cooperative control tasks among the vehicles are part of the sensing and control problems. In addition, a fleet-wide coordinator manages cooperative formation tasks such as balancing fuel use between vehicles. The sensing, control, and coordination subsystems are described in this section.

The control of the formation’s position (e.g., its heliocentric orbital location) is a separate problem, called the absolute control problem. In cases where the orbital dynamics will strongly influence the relative motions of the spacecraft (i.e. in LEO), an absolute control system may be used instead of a formation flying control system to keep the spacecraft in carefully selected orbits that will achieve the desired relative formation. This approach, also called constellation control, is discussed further in References [41, 42, 43, 44]. Absolute control can be implemented with standard, well-understood techniques; it is not the focus of this research.

Synthetic-aperture science missions will employ three-axis stabilized spacecraft (i.e., actively pointed along all three axes using reaction wheels). The spacecraft attitude-control systems for an SSI mission are responsible for pointing the spacecraft so that light from the observing target can be collected and then steered towards a combining spacecraft where the interference fringes are formed. The collecting apertures will not be steerable over large ranges; thus the attitude-control systems on the collector spacecraft are responsible for pointing the collecting apertures at the target. This fixes the task for two of the spacecraft axes and can be addressed with conventional attitude-control techniques. The third axis will be pointed to allow light to be redirected to the combiner spacecraft. With the virtual spacecraft bus and optical-alignment layers active, this task can also be accomplished with conventional attitude-control techniques. Thus the attitude-control systems will hold the collector spacecraft orientations fixed with respect to the observing target and to the combiner spacecraft. This control system can be implemented with standard, well-understood techniques and therefore is not the focus of this research.

Thus, the absolute control and attitude control for formation flying spacecraft are not developed in this thesis because these systems can be based on traditional spacecraft control methods. However, relative control, relative estimation, and coordination for spacecraft
CHAPTER 1. INTRODUCTION

formation flying are new problems requiring new solutions. This thesis develops potential solution approaches as parts of the formation flying control system.

Relative-State Estimation

Accurate knowledge of inter-spacecraft relative positions and relative velocities is critical for precise formation flying. Thus a global, accurate sensing system is needed. Global positioning system (GPS) sensors exhibit these properties. In particular, differential carrier phase GPS (DGPS) techniques are well suited to measuring the distance between two GPS antennas. DGPS can be applied to the formation flying problem to track relative positions as well as spacecraft attitudes.

DGPS hardware has been developed already, including: receivers, antennas, and even transmitters ([45]) which can be leveraged for this application. However, the GPS phase differences still must be translated into useful position and attitude information. In addition, velocity and rotation rates can be derived from the phase measurements. This function is accomplished through a Kalman filter described in Chapter 3. Additional issues associated with DGPS (e.g., tracking-loop design, multi-path mitigation, integer initialization) are discussed in Chapter 3, but these are not the focus of this research.

Estimation Architecture

The distributed nature of the formation flying problem necessitates the choice of an estimation architecture. This defines how the spacecraft cooperate to solve the relative-estimation problem. One choice is to collect all measurements (i.e., GPS phase differences) at a central processor, perform a batch estimation, and then distribute the results to each vehicle. An opposite approach is to allow each vehicle to perform the subset of the estimation problem relevant to that vehicle. The goal is to estimate the relative states with accuracy sufficient for the control system to maintain the formation within a desired tolerance. However, the estimation-architecture choice strongly affects the inter-vehicle communication bandwidth and computational power needs of the control system, which impact spacecraft hardware requirements. In addition, some architectures provide more flexibility and robustness. Thus the mission designer must choose an estimation architecture that meets performance requirements without excessive hardware costs. This issue becomes more important and more
complex as the number of vehicles in the formation grows. Chapter 3 compares some estimation architectures; and highlights the trade-offs associated with this systems-engineering decision.

Cooperative Control

One challenge in designing control laws for a formation of spacecraft is the use of on-off thrust actuators for relative-position control. Due to this nonlinearity in the system, linear control techniques (e.g., LQR) are of limited utility. This challenge can be addressed by designing pulse-width modulation or thrust-mapping schemes that effectively cause the actuator to behave linearly in a time-averaged sense ([46]). However, these approaches often waste fuel and cannot compensate for the maximum thrust limit on the thrusters ([47]). Thus, new multi-vehicle relative control laws that directly account for the on-off nature of the thrusters were sought.

Efficient relative stationkeeping that maintains the formation (formationkeeping) requires cooperative control. That is, it requires two or more vehicles cooperating to regulate the relative states. Chapter 4 derives an analytic fuel-time-optimal cooperative control law for two vehicles and a numerically-optimized control algorithm for three or more vehicles using on-off thrusters.

Control Architecture

The formationkeeping problem can be solved centrally, distributed to local controllers on each vehicle, or the formation can be divided into sub-formation groups that maintain formation both internally and relative to each other. Any of these strategies can be used to achieve VSB control. This control architecture choice involves balancing fuel efficiency, system complexity, and computing intensity.

The control-architecture design parallels the estimation-architecture design. The performance level must be balanced with hardware and design constraints. A successful design will meet the required performance using available spacecraft hardware without sacrificing robustness and flexibility. Obviously, this systems-engineering problem is very mission dependent. Chapter 4 presents a range of choices, defines some critical performance metrics and compares several designs in simulation.
CHAPTER 1. INTRODUCTION

Coordinator

Current space missions often assign monitoring functions such as fault detection and maneuver planning to ground controllers. Ground controllers are also responsible for monitoring system performance, and may modify the system behavior by uploading revised control and estimation “gains” to the spacecraft. Such functions become more challenging for formation flying missions due to the number of spacecraft involved. In addition, new goals such as fuel balancing arise for multi-spacecraft missions.

A mission controller on the ground could perform these functions for a simple, short-term formation flying mission (e.g., the ORION mission described in Reference [48]). But an autonomous agent onboard the spacecraft allows faster operation as well as reduced operator workload and down-link requirements. Chapter 5 applies decentralized control methods to the formation management problem, introducing a new coordinator function which operates at a low bandwidth to manage fleet-level tasks and to enhance inter-vehicle cooperation. An autonomous fuel-balancing coordinator algorithm is presented as an example.

1.2.5 The Science Instrument

The formation flying control system keeps the vehicles in formation and within the workspace of the optical-alignment control layer. The optical-alignment control system is responsible for aligning the optical elements on each vehicle so that they behave as a unified instrument. Chapter 6 describes an experimental demonstration of a simplified, multi-level alignment system that regulates the pointing of a laser between two active vehicles to less than a millimeter. This demonstration illustrates the multi-level control concept that will enable SSI and separated-spacecraft SAR missions. Reference [17] outlines a realistic SSI optical-alignment system, designed for the three-vehicle NMI mission. Reference [15] presents a preliminary design of the optical alignment for another proposed SSI mission and Reference [49] analyzes the optical-alignment system for a proposed monolithic space interferometry mission. Reference [50] describes experimental results from a similar alignment system on a monolithic interferometer testbed.

The combination of the coarse and mirror control layers brings the optical elements into alignment with precision equivalent to that of a monolithic interferometer with small disturbances. Optical delay lines, articulated mirrors and internal metrology beams are used to stabilize the observing-target fringe for measurement. This applied-physics control
Problem has been studied extensively (e.g., see References [51, 52]). Examples of space-based monolithic-interferometer alignment system designs are found in References [50, 13].

1.2.6 Problem Statement Summary

A multi-level control approach addresses the separated-spacecraft interferometry control problem. Successively finer control loops achieve nm-level alignment between the separate elements of the instrument (on separated spacecraft) which allows interference fringes to be measured as part of the aperture-synthesis process. Each control level depends on the one above it to keep the system state within its workspace. The outermost control level is the formation flying control system; it is the focus of this research.

The formation flying control system performs three functions: relative-state estimation, cooperative control for formationkeeping, and fleet-wide coordination. The DGPS-based estimator provides accurate estimates of the inter-vehicle states without excessive computation and/or communication requirements. The cooperative control performs fuel-efficient formationkeeping using fixed-level on-off thrusters on all of the spacecraft in concert. A coordinator performs fleet-wide tasks such as fuel balancing. In addition, both the estimator and control problems include architecture designs to distribute the task among the vehicles.

These capabilities will be important to the success of upcoming formation flying missions. Chapters 3, 4, and 5 develop solutions that address these problems and explore the design spaces for the estimation- and control-architecture problems. Chapter 6 presents experimental demonstrations of formation flying on a three-vehicle testbed. This testbed is described in Chapter 2.

1.3 Related Research

The research in this thesis draws on and contributes to the fields of spacecraft formation flying, distributed control and GPS-based relative sensing. This section references important results from related research efforts in these three fields.

1.3.1 Spacecraft Formation Flying

Advances in computing, sensing and small-satellite design as well as the need for expanded capabilities are behind the recent surge in spacecraft formation flying research. Reference [53] defines a control framework specific to the spacecraft formation flying control
problem which is parallel to the multi-level VSB control implementation described in this thesis. Reference [54] investigates a fuel-balancing approach for a cluster of spacecraft in Earth orbit. This approach only applies to pre-planned orbital station-change maneuvers (not to relative control).

Several research efforts have developed spacecraft control laws for formation flying. References [55, 56, 41] present control laws for two-vehicle leader/follower formation flying in Earth orbit. Reference [57] presents a control approach for a many-vehicle formation in deep space using leader/follower control with a dynamically chosen leader (there is no discussion of optimality). Reference [58] derives a stabilizing control approach for a many-vehicle formation in Earth orbit based on nearest-neighbor tracking. Reference [42] compares absolute and relative techniques for constellation control in Earth orbit. References [44, 21] investigate orbit design and relative-control techniques for spacecraft formations in Earth orbit.

All of the references in this section present computer-simulation results only.

1.3.2 Distributed Control

This spacecraft formation flying research draws on prior work in distributed control for terrestrial systems. The focus of these efforts is driving multiple, independent systems to achieve an overall goal. The control approach options for this problem range from centralized to fully decentralized with many possible intermediate control structures. Reference [35] investigates a simple decentralized (leader/follower) control architecture for a two-aircraft formation. Reference [34] compares centralized and decentralized control of a five-aircraft formation in simulation. Reference [32] describes several control strategies for a formation of mobile robots including several variations on the nearest-neighbor tracking approach. Reference [33] simulates a group of mobile robots with each robot tracking its two nearest neighbors. Reference [24] organizes a large formation of automobiles into platoons. Each platoon has internal leader/follower control and the platoons regulate their positions relative to each other as well. Reference [27] compares centralized, leader-referenced, and nearest-neighbor control architectures for a formation of four military vehicles. Of these references, only [24, 27] include experimental results.

The control architectures described in these references have been adapted to the spacecraft formation flying problem and compared in both simulation and experiment. These
ideas have also been applied to the estimation architecture problem. The new results presented here quantify the trade-offs of control- and estimation-architecture design choices for the spacecraft formation flying application.

1.3.3 GPS-Based Relative Sensing

This research investigates the use of GPS as a relative sensor. The first experimental use of GPS for physically-simulated spacecraft rendezvous was reported in Reference [59]. This thesis includes follow-on work published in References [60, 61]. Reference [62] describes a complementary research effort including relative and absolute estimation algorithm development and results based on simulated and post-processed spaceborne GPS data using the widelane measurement technique. Reference [63] describes experimental results and an integer-initialization technique for two blimps flying in an indoor GPS environment. This testbed allows for motion in three dimensions; however there is no truth sensor, so assessment of GPS accuracy is challenging.

Reference [64] describes the status of spaceborne GPS sensing and lists many near-term missions that will take advantage of GPS. References [65, 66, 67] document the development of a GPS-derived relative sensor that can operate independently of the Earth-orbiting NAVSTAR GPS satellites. This new sensor, called the Autonomous Formation Flyer is targeted at SSI missions such as ST3 and TPF. Reference [68] presents an alternate approach to GPS-based relative sensing outside the NAVSTAR constellation based on pseudolite augmentation.

1.4 Contributions of This Research

This thesis presents the theoretical and experimental results of research carried out between 1994 and 2000. The research results include several important contributions to the field of spacecraft formation flying.

1. *Demonstrated important advances in carrier-phase differential GPS-based relative-state estimation for formation flying.*

Quantified the performance impacts of several new advances to GPS-based relative-state sensing for formation flying, including: the first implementation of trio-based relative estimation instead of pair-based relative estimation, including velocity states in the estimation problem, and increasing the filter propagation rate. All advances
were demonstrated in simulation, and many were experimentally verified. In addition, the improvement in estimation accuracy due to the addition of an extended Kalman filter has been quantified experimentally.

2. Derived new fuel-time-optimal relative-control laws for spacecraft formations of any size.
   An analytic solution to the fixed-thrust, two-vehicle fuel-time-optimal control problem first was derived. Then, a fast, numerically optimized control approach was derived for the three-or-more-spacecraft problem.

3. Developed a spacecraft-formation coordinator to address fleet-wide performance goals (such as fuel balancing).
   A fuel-balancing coordinator was added to the spacecraft formation flying control system to enhance inter-vehicle cooperation and fleet-level performance. The coordinator is shown in simulation to quickly achieve fuel-balance (within 1%).

4. Quantified the performance benefits and implementation costs of estimation- and control-architecture design options for spacecraft formation flying missions.
   The problem of distributing the estimation and control functions requires estimation and control architectures. Distributed, centralized and hierarchical architectures were compared in simulation. The results of these studies quantify realistic performance and implementation cost trade-offs for the control- and estimation-architecture designs, which are potential inputs to system-wide mission-design trade studies.

5. First-ever experimental demonstrations of a precise formation flying control system.
   Experimentally demonstrated successive loop closure with GPS-based sensing for the outer loop and an optical-alignment system for the inner loop on a physical simulation of two spacecraft. Experimentally demonstrated a three-vehicle virtual spacecraft-bus maneuver with cm-level relative control on three physically-simulated spacecraft. Both of these capabilities will be necessary for SSI missions.

6. Mapped these new research results into the design spaces of three near-term spacecraft formation flying missions.
   Identified and spelled out potential applications of this research to important formation flying missions planned for the near-term by both NASA and the Department of Defense (Terrestrial Planet Finder, Space Technology 3, and TechSat 21).
1.5 Overview of this Thesis

This chapter has motivated the formation flying problem and its application to separated-spacecraft interferometry, and presented a control-system framework to enable virtual spacecraft bus missions. Chapter 2 describes the Stanford University Aerospace Robotics Laboratory’s formation flying testbed in some detail, including the techniques used to identify the system behavior. Chapter 3 describes a GPS-based solution to the relative-sensing problem and presents an estimation architecture design study. Chapter 4 derives new low-level control algorithms for multi-vehicle relative stationkeeping and presents a control architecture trade study. Chapter 5 describes an autonomous coordinator methodology, including an example fleet-level fuel-balancing application. Chapter 6 describes two formation flying experiments that demonstrate multi-level control for precise alignment and cm-level formation flying of three physically-simulated spacecraft during a maneuver. Chapters 3 and 4 also present experimental results. Chapter 7 identifies potential applications of this research to three near-term formation flying missions. Chapter 8 suggests possible topics for further work building on or related to this research.
Chapter 2

The Formation Flying Testbed

A formation flying testbed (FFTB) was used to validate the sensing and control tools derived in this thesis. The testbed (shown in Figure 2.1) consists of three free-flying space robots that operate in an indoor GPS environment. The robots were constructed for various research purposes in the Aerospace Robotics Laboratory (ARL). All three robots were operated in concert for the first time for the formation flying experiments in this research. Experiments on the formation flying testbed are complemented by high-fidelity three-vehicle simulations (including GPS sensing) and supplemented by simplified simulations of larger formations.

The free-flying space robots were designed to simulate the drag-free, zero-gravity dynamics of space to high fidelity in a plane. Each robot possesses onboard processing, power, propulsion, wireless communications, and GPS receivers. The test environment has two independent sensing systems: an indoor GPS system (consisting of six ceiling-mounted transmitters broadcasting GPS signals) and an overhead camera-based vision sensing system.

This chapter presents the FFTB and discusses how the dynamic parameters of the system were quantified so that control laws could be optimized for testing. In addition, the formation flying simulation codes are briefly described.

2.1 Free-Flying Space Robots

The first free-flying space robot was constructed in 1989 by Marc Ullman ([69]). Two additional vehicles were added later. The three free-flying space robots are very similar in
design. Each robot floats on a 0.08 mm air cushion. The original vehicle uses compressed-air thrusters as the only actuators. The other two vehicles are based on the original design with the addition of reaction wheels for more precise orientation control.\footnote{The first vehicle’s manipulator system (shown in Figure 2.1) was removed for the experiments described in this thesis.}

The three free-flying space robots have been used for several research projects in the ARL. Many improvements have been made to the original systems including faster processors, an improved vision-based sensing system, numerous software upgrades, and the addition of an indoor GPS sensing system. The three free-flying robots simulate the drag-free motion of vehicles in space, so together they comprise an excellent testbed for spacecraft formation flying research.
The vehicles are briefly described in this section; more detailed descriptions appear in ARL theses for the vehicle design ([69]), software architecture ([70]), vision system ([71]), and indoor GPS system ([72]).

2.1.1 Robot Structure

Each vehicle fits inside a 0.5 m diameter cylinder 0.8 m in height. The mass of each robot is roughly 80 kg. The robots are designed with modular layers divided by functionality (from bottom to top): air cushion, compressed air supply, thrusters, reaction wheel, power and analog electronics, digital electronics, GPS antennas and a unique LED pattern recognizable by the overhead vision system.

2.1.2 Computer System

Each vehicle has a Motorola MVME2604-2131B 200 MHz PowerPC 604 processor single-board computer running ControlShell software under the VxWorks operating system. For each experiment, the control software runs onboard the vehicles. The software runs sensing and control loops at a 60 Hz sample rate.

2.1.3 Real-Time Control Software

The software that drives the vehicles for all experiments was developed within the ControlShell programming environment. This development tool takes care of most of the implementation details (e.g. task and memory management, scheduling, etc.) which allows the programmer to focus on the sensing and control algorithms.

In addition, because ControlShell was developed in the ARL ([70]), a large body of ControlShell code was already available. This code base was leveraged extensively for this project.

2.1.4 Communications

Each vehicle uses a BreezeNet Access Point wireless ethernet device to communicate with the other vehicles and the user (via a wireless ethernet Station Adapter linked to the ARL network). Throughput capability is 2 Mbps.
2.1.5 Electrical Power System

Each vehicle carries two rechargeable batteries. The batteries power all onboard computers and actuators. The system typically yields up to 20 minutes of operation before recharging is necessary.

2.1.6 Reaction Wheel

Two of the three vehicles have reaction wheels for orientation control. The reaction wheel provides torque via angular-momentum transfer. The wheel inertias are roughly 1/33 of the vehicle inertias; they can deliver about 0.5 N·m of torque and have enough momentum storage capacity that momentum dumping was not required during the formation flying experiments.

2.1.7 Drag-Free Motion

Each vehicle floats on an air cushion about 0.08 mm thick over a smooth granite plate. The 9 ft by 12 ft (2.74 m by 3.66 m) granite plate was installed in 1986 and at the time was smooth to within 0.033 mm (laboratory grade AA). The plate is carefully balanced.
Each vehicle has eight compressed-air thrusters (two each in $+x$, $-x$, $+y$, and $-y$) which can provide translational thrust as well as torque. The diameter of one thruster (shown) is 1 cm.

to provide a level surface for the vehicles. Each vehicle also must be carefully balanced\(^2\) so that its air cushion is evenly distributed. An uneven air cushion produces a constant disturbance force in the vehicle frame which is quite noticeable after ten or twenty seconds even for a small imbalance.

The air-cushion heights are adjustable through pressure regulators on the vehicles (regulator \#5 in Figure 2.2). Zero-drag behavior was obtained for the experiments in this research only when the air-cushion flow was increased well above the design level (which indicates that the table and vehicle surfaces should be re-polished).

### 2.1.8 Propulsion System

Each vehicle uses compressed-air thrusters for propulsion (see Figure 2.3). The thrusters were designed to deliver 0.9 Newton of force at 100 psi ([69]). The air-pressure level is controlled by regulator \#3 in Figure 2.2. The thrusters were operated at 50 psi for this

\(^2\)Robot balancing is accomplished by placing weights around the perimeter of the base plate.
research in order to increase the precision of the control system. The force delivered by each thruster at this pressure was not measured directly. Instead, *system identification* tests experimentally measured the acceleration produced by the thrusters (see Section 2.2).

### 2.1.9 Pneumatic Gripper System

Each free-flying space robot includes a pair of pneumatic grippers mounted on two-link robot arms (see Figure 2.1). These were not used for the formation flying experiments. In fact, the manipulator systems were locked down on two vehicles and removed from the third to eliminate dynamic disturbances caused by the links.

### 2.1.10 Overhead Vision System

An overhead vision system measures FFTB vehicle positions. This system was originally designed by Vince Chen ([71]). It includes downward-looking CCD cameras mounted about 2.5 m above the granite plate. The cameras each have infra-red filters (Kodak #87), and the room is shaded so that IR sources (such as reflected sunlight) are minimized. Each vehicle has a unique pattern of three IR LEDs mounted on top. The cameras image these LED patterns, then custom-designed *PointGrabber* boards convert the images into lists of detected points. These points are processed by pattern-matching software and filtered to yield position and velocity information for each vehicle. This information is transmitted to the vehicles via the wireless ethernet. Vision processing takes place on two Motorola MVME-167 68040 processor single-board computers using code developed within the Aerospace Robotics Laboratory.

In 1999, a third camera was added and the system was recalibrated to provide nearly complete coverage of the granite surface plate (this procedure is described in Reference [73]). The performance of the vision system is estimated to be better than 0.5 cm within each camera’s field of view with a 1-2 cm (worst case) alignment error between cameras. The vision system provides measurement updates at 60 Hz.

### 2.1.11 Indoor GPS System

An indoor GPS system was added to the testbed by Kurt Zimmerman to test GPS sensing for space-vehicle rendezvous (see Reference [72]). The GPS system employs six ceiling-mounted transmit antennas that broadcast signals generated by GPS pseudosatellites (or
pseudolites) originally designed at Stanford ([45]). Pseudolites generate GPS-like signals at the L1 frequency which can be recognized by standard GPS receivers.

Each vehicle has a Trimble TANS GPS receiver and four patch antennas. The receivers have modified internal code so that they pass differential GPS phase measurements to the vehicle’s main processor via a serial link. Given the vehicle’s initial positions (e.g., from the overhead vision system), these phase measurements can then be processed to track the vehicle with cm-level location accuracy in the GPS pseudolite-constellation frame of reference (the tracking process is described in more detail in Chapter 3). The pseudolite reference frame was designed to coincide with the vision system reference frame. The GPS receivers provide phase measurements at 10 Hz.

2.2 System Identification

The compressed-air system is shown in Figure 2.2. Precise control was desired for this research, so the thrusters’ minimum impulse bit\(^3\) was reduced by adjusting the thruster pressure (regulator #3 in Figure 2.2) to 50 psi (instead of the 100 psi design level). The corresponding acceleration was carefully modelled for each FFTB vehicle using system identification techniques as described in this section. The resulting acceleration models are used for the experiments in Chapter 6.

This section also identifies two non-ideal thruster behaviors. First, the acceleration per thruster is affected by the number of currently-active thrusters. Second, the thruster valves tend to display “capacitance” effects during short thruster on-times and off-times. These effects are characterized, and nonlinear thruster maps that partly compensate for them are discussed.

2.2.1 Analytical Thruster Model

Initially, the robots were assumed to behave as drag-free inertias with ideal on-off, fixed-thrust actuators. The thrusters are designed to provide 0.9 N thrust at 100 psi. As initially designed, the mass of a free-flying space robot was 70 kg ([69]). Using these values, the predicted acceleration per thruster is

\(^3\)The minimum impulse bit is the integral of thrust over the minimum thruster on-time.
Because this estimate is based on the original design, it was not reliable and required testing. Measuring the robot mass and thruster force would have been challenging and unnecessary. Instead the acceleration provided by the thrusters was computed based on the dynamic effects which were tracked by the overhead vision system. This technique measures the acceleration per thruster, so lower air-pressure levels — which yield smaller accelerations — were tested as well. Reducing the acceleration per thruster enables finer formation flying control. The following sections describe three sets of experiments that identify accurate thruster models, including an acceleration-per-thruster\(^1\) value for each vehicle.

2.2.2 System Identification Experiments

The translation dynamics of the robot along one axis are modelled in state-space form as

\[
\begin{bmatrix}
    x_{k+1} \\
    v_{k+1}
\end{bmatrix} =
\begin{bmatrix}
    1 & \Delta t \\
    0 & 1
\end{bmatrix}
\begin{bmatrix}
    x_k \\
    v_k
\end{bmatrix}
+ \begin{bmatrix}
    \frac{1}{2} a \Delta t^2 & \frac{1}{2} d \Delta t^2 \\
    a \Delta t & d \Delta t
\end{bmatrix}
\begin{bmatrix}
    u_k \\
    1
\end{bmatrix}
\]  

(2.2)

where \(x\) is position, \(v\) is velocity, \(\Delta t\) is the sample period (1/60 second), \(k\) is the current sample period, \(a\) is the acceleration per thruster, and \(u\) is the number of thrusters firing in the positive direction (that is, \(u\) will take an integer value between \(-2\) and \(+2\)). \(d\) is included as a constant disturbance acceleration which captures constant disturbance force effects (e.g., due to an unbalanced air cushion); ideally it should equal zero.

Given this model form, a system identification experiment can be performed to find a value for \(a\) in this way: a known thrust program \(u(t)\) is executed, and the vehicle’s state \(x(t), v(t)\) is observed and recorded. Then a \textit{prediction-error identification method (PEM)} algorithm is used to find the values for \(a\) and \(d\) that minimize the prediction error. The prediction error is the difference between the observed state history \(x(t), v(t)\) and the state history predicted by the model (including \(a\) and \(d\)) for the thrust sequence \(u(t)\). For more details, see References [74, 75].

\(^1\)The acceleration per thruster was assumed equal for all thrusters on the same vehicle.
Table 2.1: Estimated acceleration (m/s²) per thruster \( a \), estimated disturbance acceleration \( d \), and resulting accumulated errors (m and m/s) at four air-pressure levels.

<table>
<thead>
<tr>
<th>Pressure (psi)</th>
<th>( a )</th>
<th>( d )</th>
<th>Position Error</th>
<th>Velocity Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>0.0281</td>
<td>-0.00001</td>
<td>43.60</td>
<td>1.21</td>
</tr>
<tr>
<td>100</td>
<td>0.0220</td>
<td>-0.00002</td>
<td>23.28</td>
<td>1.49</td>
</tr>
<tr>
<td>50</td>
<td>0.0095</td>
<td>0.000001</td>
<td>20.18</td>
<td>1.18</td>
</tr>
<tr>
<td>25</td>
<td>0.0057</td>
<td>-0.00001</td>
<td>7.82</td>
<td>2.01</td>
</tr>
</tbody>
</table>

**Air-Pressure-Level Experiments**

The acceleration per thruster was measured at four levels of air pressure. At each pressure level, an identification experiment was performed. A robot was started at rest with its orientation controlled to be constant (using the reaction wheel\(^5\)). The vision system’s position and velocity measurements were recorded for 16 seconds. During this time, a 10 second sequence of thruster commands was executed. The thruster sequence cycled both forward thrusters on for 2 samples, then off for 5 samples (each sample is 1/60 second long).

A prediction error technique was used with the system model in Equation 2.2 to find the \( a \) and \( d \) values that minimize the velocity-error vector.

Figure 2.4 presents the velocity history data for these four experiments. For reference, the analytical model-predicted velocity history (i.e. predicted by the model in Equation 2.2 with the \( a_0 \) value listed in Section 2.2.1) is repeated on each graph. Table 2.1 lists the acceleration per thruster \( a \) found by PEM for each case along with the drag \( d \) and position- and velocity-model errors which are calculated as

\[
\text{error} = \sum_{k=1}^{n} |\hat{x}_k - x_k|
\]  

(2.3)

where \( n \) is the number of samples recorded, \( \hat{x}_k \) is the model-based expected state (position or velocity) value at sample \( k \), and \( x_k \) is the measured state value at sample \( k \). This error metric is the sum of the state errors from every sample during the experiment.

Notice that the velocity error is minimized; this was the goal of the PEM algorithm. A larger position error results because errors in the velocity estimate are integrated, and

\(^5\)Vehicle 1 has no reaction wheel. Its orientation was left uncontrolled for this experiment; the resulting angular motion during the experiment was minimal.
the velocity measurements lag the position measurements in time (the vision system does not measure velocity directly). The results define how the acceleration per thruster varies with air-pressure level. Based on these results, 50 psi was selected for the experiments in this research. This value reflects a (partly subjective) balance that achieves a fine level of control without underpowering the robots.

Figure 2.4: Velocity versus time at four air-pressure levels. Measured and *PEM*-predicted values are shown (solid) with the analytic model-predicted values (dash-dot) for comparison.
Thruster Characteristics Experiments

In practice, the free-flying space robot thrusters do not behave as on-off, constant thrust actuators. Instead, the acceleration per thruster depends also on other factors: how many thrusters are on, how long they are on and how long they are off between firings, as further experiments quantified.

![Graph showing acceleration per thruster versus off-time for 2 and 4 thrusters.](image)

**Figure 2.5:** Acceleration per thruster \( (a) \) versus off-time for 2 thrusters (*) and for 4 thrusters (×).

Two series of experiments similar to those described in the previous section were performed using vehicle 1 (Huey)\(^6\) at 50 psi. For the first series, the on-time was fixed at one pulse (1/60 second) and the off-time was set to: 0, 1, 2, 3, 6, and 12 time steps. These cases were run with two and four active thrusters; that is, for the 2-thruster experiments only the forward thrusters are pulsed, but for the 4-thruster experiments both the forward and side thrusters are pulsed on. The side thrusters have little or no direct effect on the forward motion, they just decrease the available air supply. The identified acceleration per thruster \( (a) \) is plotted in Figure 2.5. The acceleration per thruster grows by a factor of 3 from continuous firing (off-time = 0) to low-frequency pulsing. This variation demonstrates that the thrusters have a capacitance effect: off-times less than \(~5/60 \text{ second}\) do not allow the thrusters time to fully “recharge”.

Similarly, the thruster on-time affects the acceleration as shown in Figure 2.6 which shows acceleration per thruster versus on-time. The off-time is 12 samples for all points.

---

\(^6\)The vehicle names are: Huey, Louie, and Heavenly.
and the following on-times were tested: 1, 2, 3, 6, 12, and 24 time steps. Ignoring a point that is probably in error (marked), this experiment shows that the acceleration per thruster drops as the pulse length increases. This is due to non-ideal behavior of the thruster valve; the initial thrust is higher than the steady-state thrust, which distorts the impulses for short on-times (less than \(\approx 5/60\) second).

The other non-ideal thruster effect is apparent in both experiments: the 4-thruster acceleration is roughly two-thirds of the 2-thruster acceleration. The causes of these nonlinear behaviors are air flow effects inside the compressed-air system and the non-ideal nature of the thrust-valve actuators. With knowledge of these nonlinear thruster behaviors, corrections can be made in some cases. In particular, if the planned thrust program is known in advance (e.g., for pre-computed trajectories), corrections can be applied by adding thruster pulses to compensate for short off-times or by deleting thruster firings to compensate for short on-times. In addition, thrust programs can be adjusted to account for the acceleration difference between 2 and 4 thrusters firing simultaneously. These correction strategies were implemented on the FFTB for the experiment in Chapter 6.

For cases where the thrust program is known, two corrections are applied to the baseline continuous 4-thruster acceleration value. If only two thrusters are firing, the thruster on-time is reduced by one third to compensate for the increased thrust. If the on-time is less than 6 samples, one sample is subtracted which results in an impulse that more closely approximates the desired effect. These corrections improve the feedforward performance,
as demonstrated in Section 2.2.3. No correction was implemented for short off-times, simply because the trajectories in this research rarely command short off-times.

For cases where the thrust program is not known in advance (i.e. for typical feedback control operation), the model of the vehicle dynamics in Equation 2.2 requires a single acceleration value. This model will be the basis of the feedback control law design. The feedback controller will generally be thrusting continuously in one or both axes (e.g. during a correction maneuver, an aperture filling maneuver, or a retargeting maneuver). Therefore, the acceleration value for continuous two-axis (4 thruster) thrust, which is the minimum-thrust value, is chosen. Using the minimum-thrust model is important because the bang-off-bang controllers described in Chapter 4 can more easily correct for underestimated accelerations than for overestimated accelerations. That is, the effective acceleration can be reduced by the controller (e.g., by switching thrusters off and on), but cannot be increased beyond the full-thrust value. The following section documents the acceleration values for each of the three robots.

**Acceleration Identification Experiments**

A final series of identification experiments was performed to estimate each robot's acceleration per thruster. Figure 2.7 shows the experiment output for Huey (as an example). A baseline operating mode of four thrusters firing continuously at 50 psi was chosen. This is the case when the vehicle accelerates in both the x- and y-axes simultaneously.

Table 2.2 lists the acceleration per thruster for the three free-flying space robots. The error metrics used in Table 2.1 are repeated for comparison. The acceleration per thruster is smaller than in Table 2.1 because the 4-thruster, continuous firing mode is tested here. The reduction in model error is due to a combination of factors, including: the reduced accelerations result in reduced overall motion, the orientation controller was improved, and the continuous firing results in motion that is easier for the vision system to track.

**2.2.3 Experimental Validation of the Revised Model**

Figure 2.8 shows data from an experiment designed to validate the thruster accelerations and thruster corrections described in the previous sections. The three vehicles execute a formation trajectory program that lasts 53 seconds. The controller commands feedforward (open-loop) forces only. This test measures how close the vehicle dynamics are to the identified models. Three plots are shown in the table-fixed vision system reference frame's
Figure 2.7: Identification experiment data for vehicle 1 (Huey) firing 4 thrusters continuously for 10 seconds. Measured and model-predicted values are shown with resulting errors (position and velocity) below.

x- and y-axes for each of three vehicles. The dotted line is the expected behavior, the solid line is the behavior without the thruster corrections, and the dashed line is with the corrections.

Notice that for all cases, the runs with thruster corrections follow the desired trajectory more closely than the runs without thruster corrections. The scale of each plot is different because each vehicle follows a different path. Table 2.3 lists the desired path lengths and
CHAPTER 2. THE FORMATION FLYING TESTBED

Table 2.2: Estimated acceleration (m/s²) per thruster, estimated disturbance acceleration \(d\), and resulting accumulated magnitude of the errors (m and m/s) for each FFTB vehicle.

<table>
<thead>
<tr>
<th>Vehicle</th>
<th>Acceleration</th>
<th>(d)</th>
<th>Position Error</th>
<th>Velocity Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Huey)</td>
<td>0.00325</td>
<td>-0.000002</td>
<td>2.71</td>
<td>1.09</td>
</tr>
<tr>
<td>2 (Louie)</td>
<td>0.00310</td>
<td>0.000002</td>
<td>5.85</td>
<td>0.87</td>
</tr>
<tr>
<td>3 (Heavenly)</td>
<td>0.00386</td>
<td>-0.000002</td>
<td>1.74</td>
<td>1.34</td>
</tr>
</tbody>
</table>

Table 2.3: Thruster model validation experiment results.

<table>
<thead>
<tr>
<th>Vehicle/Axis</th>
<th>Path Length cm</th>
<th>Error without Correction cm</th>
<th>Error with Correction cm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td></td>
</tr>
<tr>
<td>1 (Huey)/x</td>
<td>79</td>
<td>71</td>
<td>3</td>
</tr>
<tr>
<td>1 (Huey)/y</td>
<td>216</td>
<td>39</td>
<td>33</td>
</tr>
<tr>
<td>2 (Louie)/x</td>
<td>58</td>
<td>25</td>
<td>13</td>
</tr>
<tr>
<td>2 (Louie)/y</td>
<td>179</td>
<td>22</td>
<td>1</td>
</tr>
<tr>
<td>3 (Heavenly)/x</td>
<td>42</td>
<td>47</td>
<td>23</td>
</tr>
<tr>
<td>3 (Heavenly)/y</td>
<td>79</td>
<td>42</td>
<td>9</td>
</tr>
</tbody>
</table>

The path-length errors for runs with and without thruster corrections. The error values are listed with the percent error (compared to the total path length). The two worst-case results come from the shortest path lengths, where smaller disturbance effects (such as drag due to dust on the table, and imperfectly balanced air cushions) are most evident. Table 2.3 illustrates the improvement due to the thrust corrections. The acceleration-per-thruster models are also validated by this experiment.

In general, the percent errors achieved are excellent for motion control with no feedback for this system. Adaptive schemes or more complex models would further improve performance, but the model developed here is sufficient for the cm-level formation flying demonstrations in Chapter 6. Accurate thruster models are critical for precise relative control on the formation flying testbed; this calibration issue will be equally important for spacecraft formation flying missions.
Figure 2.8: Position (x and y) versus time for three vehicles following a trajectory with feedforward (open loop) control only: desired path (dotted), vehicle data without (solid) and with (dashed) thrust correction maps. Note that dotted and dashed lines coincide on the Louie-y plot.

2.2.4 System Identification Summary

This section has identified models of the free-flying space-robot translational dynamics. PEM techniques identified the acceleration-per-thruster values for each robot. The accelerations are based on the minimum-thrust case because it is easier for the bang-off-bang
controllers described in Chapter 4 to correct for underestimated accelerations than for overestimated accelerations. Also, thruster corrections applicable to pre-computed trajectories were derived based on experiments that characterized the thruster behaviors for short on- and off-times. Both the acceleration models and the thruster corrections were experimentally validated as described in Section 2.2.3. These models are used for the experiments described in Chapter 6.

2.3 Testbed in Simulation

The ability to simulate the dynamics of a spacecraft formation is important for two reasons. First, algorithms can be tested and refined in simulation prior to experiments. Experiments on the FFTB require a significant time investment because of the complexity (3 free-flying robots and 7 computers must be readied for each experiment) and the desired level of precision (the table must be cleaned and the robot air cushions need to be balanced before every set of experiments). Second, those cases that cannot be implemented on the testbed must rely on simulation testing only. For example, the control architecture discussion in Section 4.5 relies on results from a six-vehicle formation simulation.

The following sections briefly describe two simulation codes that were developed for this research. The first code simulates the FFTB including the indoor GPS constellation. The second code simulates the dynamics of six spacecraft with a simplified sensing model.

2.3.1 High-Fidelity FFTB Simulation

A Matlab simulation code was created that simulates the three-vehicle FFTB. The code models the vehicles as three simple $1/s^2$ plants with noise. It generates GPS measurements for each vehicle by computing all the correct GPS phase observables (based on the geometry of the vehicles with respect to the ceiling-mounted six-transmitter pseudolite array)\footnote{This function is based on code created by Kurt Zimmerman ([72]) and modified for three vehicles by John Carl Adams ([76]).} and then adding noise. Alternately, experimental data (GPS phase measurements and vehicle positions from the vision system) can be loaded by the code to test estimation techniques in a post-processed mode (using the vision sensing data as truth). Thrusters are simulated as ideal on-off actuators. The code was used to test GPS estimation techniques, formation flying control algorithms, and a fuel-balancing coordinator.
Table 2.4: Noise variances for three-vehicle simulation code.

<table>
<thead>
<tr>
<th>Noise parameter</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plant Position</td>
<td>$1 \times 10^{-6}$ (mm)$^2$</td>
</tr>
<tr>
<td>Plant Velocity</td>
<td>$1 \times 10^{-6}$ (mm/s)$^2$</td>
</tr>
<tr>
<td>Single-Difference</td>
<td>4.151 (mm)$^2$</td>
</tr>
<tr>
<td>Double-Difference</td>
<td>30.9926 (mm)$^2$</td>
</tr>
</tbody>
</table>

Table 2.5: Noise variances for six-vehicle simulation code.

<table>
<thead>
<tr>
<th>Noise parameter</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plant Position</td>
<td>$1 \times 10^{-6}$ (mm)$^2$</td>
</tr>
<tr>
<td>Plant Velocity</td>
<td>$1 \times 10^{-6}$ (mm/s)$^2$</td>
</tr>
<tr>
<td>Relative Position Sensing</td>
<td>3.24 (mm)$^2$</td>
</tr>
<tr>
<td>Relative Velocity Sensing</td>
<td>$1.0 \times 10^{-2}$ (mm/s)$^2$</td>
</tr>
</tbody>
</table>

The noise values include plant noise and GPS phase noise. This research focuses on GPS single-differenced and double-differenced phase measurements as the sensing observables. These measurement types are described in Chapter 3. The noise on these measurements is simulated as zero-mean white noise with the variances shown in Table 2.4. These values are typical of those observed on the FFTB. However the noise on the actual measurement signals is dominated by multipath noise which is difficult to model accurately and is not white. Chapter 3 describes an experiment that shows that the simulation yields results comparable to experimentally observed performance. The plant noises are also modelled as zero-mean white noise with the variances listed in Table 2.4 which approximate the free-flying space robot behavior.

2.3.2 Six-Vehicle Simulation

A six-vehicle simulation is used for the control architecture study in Section 4.5. This simulation is based on the three-vehicle code; however, the GPS sensing is replaced by simple relative-position and relative-velocity measurements with noise that approximates the level of performance expected from GPS-based relative sensing.
2.4 Summary

This chapter has described the formation flying testbed, established accurate models of the FFTB vehicles (and documented the modelling process), and described the two simulation codes used in this research (one for three-vehicle and one for six-vehicle simulations). Descriptions of the experimental system are available in more detail in ARL thesis References [69, 70, 71, 72]. The acceleration-per-thruster values for the free-flying space robots used in this research are listed in Table 2.2. The GPS noise values and the vehicle dynamics noise values used in the three-vehicle simulation of the FFTB are listed in Table 2.4. The sensing and plant noises used in the six-vehicle simulation are listed in Table 2.5.
Chapter 3

GPS Sensing for Spacecraft Formation Flying

Global Positioning System (GPS) technology can provide accurate, global relative sensing for spacecraft formation flying. This chapter presents an overview of GPS and how it can be applied to the spacecraft formation flying problem. Estimation algorithms were investigated in simulation and experimentally on the FFTB in cooperation with Tobé Corazzini. The initial relative-sensing experiments by Zimmerman (Reference [59]) are extended to three vehicles and refined to enable cm-level control with relative-position estimation errors on the order of 1 cm. The estimation architecture design problem — which must be addressed for formations larger than two vehicles — is introduced and explored.

The results in this chapter extend Kurt Zimmerman’s research ([72]) in important ways. The two motivations of the present work were: to improve estimator performance on the formation flying testbed and to determine the related implementation costs, and to identify the issues and costs related to extending relative sensing to a three-vehicle formation. The FFTB (described in Chapter 2) includes an indoor, six-transmitter GPS environment with three vehicles. The distances separating the vehicles and between the vehicles and the transmitters are all on the order of meters. Thus, the results in this section are primarily relevant to relative-sensing applications with similar transmitter-receiver, and receiver-receiver separations. Potential applications of these results include: GPS self-constellations (briefly described in Section 3.3.2), formations of widely-separated spacecraft in Earth orbit, and pseudolite-augmented formation flying applications. Other GPS-based relative-sensing problems (e.g., satellite clusters in LEO, or NAVSTAR-based relative sensing for terrestrial
applications) may take advantage of the far-constellation assumption which simplifies the estimation problem. See Reference [77] for an analysis of the error associated with this assumption for the relative-sensing application.

3.1 The Relative-Sensing Application of GPS

Conceived in 1973, the Global Positioning System is a satellite-based navigation system that nominally provides position measurements with accuracies on the order of 10 meters (see Reference [78]). The system was built by the United States Department of Defense but has found many civilian applications worldwide.

GPS receivers track signals from orbiting NAVSTAR satellites. Each GPS signal includes the satellite ephemeris, and time information generated by a very-accurate clock on the satellite. These signals are decoded to provide pseudorange measurements. With four or more tracked signals, a receiver solves the combined position and time problem to triangulate its position and to determine the precise time. An intentional degradation of the GPS signal called Selective Availability (S/A) can be imposed by the U.S. government to restrict the full accuracy of the system to authorized users only\textsuperscript{1}.

The carrier-phase differential GPS (DGPS) technique improves GPS sensing accuracy by circumventing S/A and other error sources in the basic GPS measurement (see Reference [78]). DGPS measures relative position. The relative position between two GPS antennas is estimated to a high degree of accuracy based on tracking the relative phases of the GPS carrier waves. The carrier wavelength is only 19 cm, so the accuracy of this technique is high (e.g., less than one centimeter if the wave is tracked within 5\%). DGPS is highly applicable to spacecraft formation flying which requires precise relative-position sensing over a large field of view.

There is an integer number of wavelengths difference between the phases measured at the two receivers. This integer ambiguity is not directly observable and must be estimated or calibrated by an additional sensor. Nevertheless, DGPS has been applied successfully to numerous sensing problems, including: automatic precision landing of commercial aircraft [79], autonomous model-helicopter control [80], precision farming [28] and attitude estimation for spacecraft in Earth orbit [81].

\textsuperscript{1}S/A was turned off on May 1, 2000.
CHAPTER 3. GPS SENSING FOR SPACECRAFT FORMATION FLYING

One limitation of GPS sensing is its reliance on the twenty-four NAVSTAR satellites in Earth orbit. This precludes the use of conventional GPS indoors or in deep space. This limitation has been overcome by the use of “pseudo-satellites” or pseudolites: transmitters that emit GPS-like signals (see Reference [45]). Pseudolite technology enables the use of GPS sensing where the NAVSTAR spacecraft are not visible. Reference [63] presents a pseudolite-based demonstration of GPS sensing for blimp formation flying indoors. Reference [68] describes how pseudolites can augment the NAVSTAR signals for spacecraft formation flying. Reference [82] describes a pseudolite-based positioning system for navigation on Mars. The FFTB includes a six-pseudolite indoor GPS system that tests GPS sensing for spacecraft formation flying.

3.2 Estimation Problem

The fundamental DGPS measurements are carrier-wave phase differences. The relative vehicle positions are recovered from these measurements through an estimator. For the FFTB case, the estimator solves for each vehicles’ absolute attitude and position in the GPS reference frame. The attitudes must be computed to establish the orientation of each vehicle’s antenna array. The resulting position estimates contain errors that are mostly common mode. Thus, by differencing the absolute-position estimates of two vehicles, a relative-position estimate is obtained which is more accurate than either of the absolute-position estimates. Vehicle i’s state is defined as

\[ X_i = \begin{bmatrix} p_{ix} \\ p_{iy} \\ p_{iz} \\ \epsilon_{i1} \\ \epsilon_{i2} \\ \epsilon_{i3} \\ \epsilon_{i4} \end{bmatrix} \]

(3.1)

where \( p \) is the position in \( x, y, \) and \( z \) in the GPS reference frame, and \( \epsilon \) is the four-element quaternion definition of vehicle \( i \)’s attitude.
The carrier phase measured at antenna \( j \) of vehicle \( i \) from pseudolite \( k \) is

\[
\phi_{ijk} = |D_{ijk}| + c\tau_{vi} + c\tau_{pk} + \lambda K_{ijk} + v_{ijk}
\]

(3.2)

where the vector \( D_{ijk} \) is simply a vector from the phase center of the pseudolite antenna to the phase center of the receive antenna. The vector \( D_{ijk} \) is defined in Figure 3.1. \( \lambda \) is the GPS-carrier wavelength (which equals 19 cm). Knowledge of the \( D \) vectors allows the locations of each antenna to be computed in the GPS reference frame. This leads easily to the vehicle state \( X_i \), since the antenna locations \( B_{ij} \) in the vehicle frames are known. Thus the position and attitude problems are coupled and must be solved together. \( v_{ijk} \) is the measurement noise. The terms \( c\tau_{vi} \) and \( c\tau_{pk} \) represent the portions of the phase measurement incurred by clock drift at vehicle \( i \)'s receiver and at pseudolite \( k \) respectively. These are the dominant error terms in the measurement equations. Differencing over multiple measurements is used to eliminate these terms.
The intra-vehicle single differences contribute primarily to determination of the attitude of each vehicle. These measurements are obtained by differencing between a “master” antenna \((j = 1)\) and the remaining antennas \((j = 2, 3, 4)\) of vehicle \(i\). For a measurement from pseudolite \(k\)

\[
\Delta \phi_{ijk} = |D_{i1k}| - |D_{ijk}| + \lambda M_{ijk} + v_{i1k} - v_{ijk}
\]

\(\forall k\) and \(\forall j \neq 1\). In this expression, the \(M_{ijk}\) are the intra-vehicle integers

\[
M_{ijk} = K_{i1k} - K_{ijk}
\]

The inter-vehicle double differences contribute primarily to the determination of the relative positions between the vehicles. Given \(n\) pseudolites, there are \(n - 1\) independent double differences between pseudolites \(k_1\) and \(k_2\). The inter-vehicle double differences are calculated in order to eliminate the remaining effects due to clock differences \(c(\tau_a - \tau_b)\). Only phase measurements from each vehicle’s master antenna \((i = 1)\) are used. So, for vehicles \(a\) and \(b\)

\[
\nabla \Delta \phi_{abk_1k_2} = |D_{a1k_1}| - |D_{b1k_1}| - (|D_{a1k_2}| - |D_{b1k_2}|) + \lambda N_{abk_1k_2} + v_{abk_1k_2}
\]

\(k_1 \in [1, \ldots (n - 1)]\), \(k_2 = k_1 + 1\). In this expression, \(N_{abk_1k_2}\) refers to the inter-vehicle integers

\[
N_{abk_1k_2} = K_{a1k_1} - K_{b1k_1} - (K_{a1k_2} - K_{b1k_2})
\]

and \(v_{abk_1k_2}\) is the collected noise term

\[
v_{abk_1k_2} = v_{a1k_1} - v_{b1k_1} - (v_{a1k_2} - v_{b1k_2})
\]

The double-difference measurements are coupled to the states of the two vehicles used in each pairing, so all measurements must be combined to resolve the formation state. From Equations 3.3, 3.5, and the quaternion constraint \((\epsilon_{i1}^2 + \epsilon_{i2}^2 + \epsilon_{i3}^2 + \epsilon_{i4}^2 = 1)\), the complete set of measurements is related to the vehicle states through one matrix equation. For the
three-vehicle problem this equation is

\[
\begin{bmatrix}
\Delta \phi_{1jk} \\
0 \\
\Delta \phi_{2jk} \\
0 \\
\Delta \phi_{3jk} \\
0 \\
\nabla \Delta \phi_{12k,k_2} \\
\nabla \Delta \phi_{23k,k_2}
\end{bmatrix}
= \begin{bmatrix}
h_1(X_1) \\
h_2(X_2) \\
h_3(X_3) \\
h_c(X_1) \\
h_c(X_2) \\
h_c(X_3)
\end{bmatrix}
+ \lambda
+ \begin{bmatrix}
M_{1jk} \\
0 \\
M_{2jk} \\
0 \\
M_{3jk} \\
0
\end{bmatrix}
+ \begin{bmatrix}
v_{1jk} - v_{1jk} \\
0 \\
v_{2jk} - v_{2jk} \\
0 \\
v_{3jk} - v_{3jk} \\
0
\end{bmatrix}
\]

(3.8)

where \( h_i \) is a set of nonlinear functions of \( X_i \), \( h_{12} \) is a set of nonlinear functions of the given formation states, and \( h_c \) is a quaternion constraint function. Given the phase measurements and the integer ambiguities, the optimal estimate of the formation states \( X_i \) can be solved in real-time using an estimator. Estimator algorithms are discussed in Section 3.4.

### 3.3 GPS Issues

The focus of this research is the estimation problem: how to derive relative-state estimates from DGPS measurements. However, other issues — not the focus of this research — also impact GPS sensing for spacecraft formation flying. This section lists these issues and solution approaches applicable to spacecraft formation flying.

#### 3.3.1 Multipath

The phase of the GPS signal is the observable for DGPS sensing. The solution to the estimation problem assumes the signal is arriving directly from the transmitter (i.e., the pseudolite). However, GPS signals reflected from nearby surfaces are also tracked by the receiver. These reflected signals, often arriving with SNRs similar to the direct signal, contain an extra phase shift due to the longer indirect path. These phase shifts degrade estimation accuracy. This phenomenon is called *multipath* (see Reference [78] for further details).

In space, the only surfaces that can reflect the GPS signal are those on the spacecraft themselves, so this problem may be minor for spacecraft formation flying missions. The multipath problem can be addressed through careful antenna placement (i.e. with limited or
no visibility to potential multipath sources) or by characterizing the error and compensating for it. Reference [83] discusses the first approach. Reference [84] presents a multipath-correction-map technique. For this research, multipath was treated as a noise input for experiments on the indoor testbed.

3.3.2 Self-Constellation

GPS-based sensing outside the range of the Earth-orbiting NAVSTAR constellation requires a GPS self-constellation. Transmitters and receivers carried by the vehicles allow solution of the constellation geometry using only signals internal to the formation (i.e., without external reference signals). Systems are currently being developed for spacecraft formation flying beyond Earth orbit (see References [68] and [65]), and for navigation on the surface of Mars (see Reference [82]).

3.3.3 Integer Initialization

The estimation problem in Equation 3.8 shows that the phase measurements include contributions from the vehicle states as well as from the carrier-phase integer ambiguities \( M, N \). Both the states and the integers must be solved. However, while the states are constantly changing, the integers remain fixed as long as the relevant signals are tracked. Thus before state estimation is possible, an integer-initialization process occurs. This is an inherent issue in carrier-phase differential GPS sensing (as discussed in Reference [78]). Several integer-initialization techniques that are applicable to spacecraft formation flying have been developed, as described below.

Initialization from External Knowledge

If the vehicle states are known, the integers can be found from phase measurements and Equation 3.8. Thus if the vehicles begin in a known location, the integers are initialized once and used as long as the GPS signals are continuously tracked. If lock is lost for a signal, all the integers related to that signal must be reinitialized. This initialization technique is used on the FFTB (using the vision sensing system to provide the initial state information). This technique could also be considered for space missions that begin with the spacecraft in a known configuration (e.g., before deployment from the launch vehicle), however when
CHAPTER 3. GPS SENSING FOR SPACECRAFT FORMATION FLYING

signals are lost and re-acquired during the mission, a re-initialization technique will still be required.

Motion-Based Initialization

Because the integers do not change as long as the corresponding GPS signals are tracked, measurements made at several epochs can be combined and solved as a nonlinear batch least-squares problem. As long as a sufficient number of GPS signals are tracked during initialization, the state history and integers can be solved for simultaneously. However, in order for the combined problem to have observability to the solution, the batch problem must contain sufficient independent measurements. Physically, this means the transmitter-receiver geometry must undergo change during the initialization. This motion-based initialization technique will converge to the correct solution with sufficient motion. However, for the spacecraft formation flying problem such motion may be expensive, particularly if this technique is used repeatedly for re-initialization when signals are lost.


Integer Search Methods

Integer search techniques explore the space of all possible integers for the set that best solves Equation 3.8. That is, for a state estimate \( X \) and carrier phase measurements \( \Phi \), integer-search techniques attempt to find the integer set \( I \) that minimizes the measurement residual \( W \) which is the difference between the right and left sides of Equation 3.8

\[
W = H(X) + \lambda I - \Phi
\]  

(3.9)

These techniques require a good initial estimate of the vehicle states (e.g., from pseudoranges). Integer search techniques are computationally expensive due to the large search space of integer combinations. They are prone to false results, particularly if the initial guess is not sufficiently accurate. The phase measurements used to solve Equation 3.9 do
not need to include motion. Reference [60] demonstrates two integer-search techniques for the spacecraft formation flying problem.

Widelaning

If the GPS receiver has access to both $L_1$ and $L_2$ GPS frequencies, then phase differences at the two frequencies can be combined to narrow the integer-search space. This technique still requires an integer-search method, but improves the speed and performance significantly as described in Reference [87].

Reference [88] presents an initialization technique applicable to open-pit mining applications that relies on pseudolites broadcasting GPS-like signals on four different frequencies simultaneously. Such a system will allow instant integer resolution for a pseudolite-only system with at least four transmitters in view by the receiver.

Operation at Higher Frequency

The integer-initialization issue can be avoided in deep space by using pseudorange-based estimation exclusively. A GPS-based sensor operating at a higher frequency can achieve cm-level positioning with pseudorange-only measurements. The Autonomous Formation Flyer (AFF) sensor in development at the Jet Propulsion Laboratory leverages this idea (References [65, 66, 67]). According to Reference [66], the AFF frequency is 20 times higher than GPS\(^2\). This results in pseudoranges with 1 cm uncertainty levels.

Integer Re-Initialization

Reference [68] proposes a promising integer-resolution algorithm for spacecraft formation flying. An initial integer solution is still required (e.g., via a motion-based technique), but then subsequent integer re-initializations (e.g., as signals are lost and re-acquired) are accomplished by a layered integer-tracking technique. This technique combines a motion-based integer-tracking algorithm with a more-accurate but slower integer-search technique. This method of integer tracking is shown to outperform the motion-based technique alone for cases of frequent integer re-initialization, such as will occur for self-constellations which have large signal strength differences or in LEO where GPS satellites move into and out of view quickly.

\(^2\)30 GHz with a chip rate of 100 MHz.
3.4 GPS-Based Estimator Experiments

Zimmerman demonstrated GPS-based relative sensing for formation flying with one active vehicle following one passive vehicle on the FFTB in 1995 (Reference [72]). This section describes the results of the present research, which sought improved estimation accuracy and a three-vehicle FFTB implementation. The performance improvements and corresponding computational costs are quantified and the related communication costs and robustness issues are discussed.

The integer part of Equation 3.8 is assumed known (i.e. initialized using one of the methods described in Section 3.3.3), so the estimators discussed in this section solve for the vehicle states $X$. This assumption is implemented through perfect integer knowledge in simulation and by initializing the integer estimates from the overhead vision system before each experiment. Thus the error contribution from integer estimation will be unrealistically small for the results in this section.

Equation 3.8, without the integer term is

$$\Phi = H(X) + V$$ (3.10)

where $\Phi$ is the vector of phase difference measurements from the right side of Equation 3.8, $H(X)$ is the vector of nonlinear functions of the vehicle states from the left side of Equation 3.8 and $V$ is the measurement noise. This equation is equivalent to the standard nonlinear estimation problem

$$z = h(x) + v$$ (3.11)

Techniques for estimating $x$ based on the measurements $z$ are briefly reviewed next. These estimators are tested in simulation for two- and three-vehicle cases and are experimentally demonstrated for formation flying control of two FFTB vehicles. Results are presented demonstrating the performance levels achieved for GPS-based relative sensing. The results validate GPS as a relative sensor for spacecraft formation flying and indicate the expected estimator performance and corresponding computational cost.
3.4.1 Estimation Algorithms

The experiments by Zimmerman (in Reference [72]) employ an iterated weighted least-squares algorithm (IWLS) to solve for the states $X$ that minimize the error at each measurement. That work also discusses using an extended Kalman filter (EKF) for this task; however at the time it was too computationally expensive to implement the full EKF for real-time use. The FFTB onboard computers were upgraded in the summer of 1998. This enabled an experimental implementation of an iterated EKF (IEKF). A non-iterated version of the EKF was also tested.

Each of these estimation algorithms is briefly reviewed below (based on Reference [89] which contains more complete descriptions of all three algorithms).

**Iterated Weighted Least Squares (IWLS)**

For the nonlinear estimation problem in Equation 3.11, expand $h(x)$ around the initial state estimate $x_0$ to find the gradient

$$H(x_0) \equiv \left. \frac{\partial h(x)}{\partial x} \right|_{x_0}$$

(3.12)

Then the estimate of $x$ that minimizes the estimation error $E$ weighted by the measurement-noise covariance matrix $R$

$$E = (z - h(x))^T R^{-1} (z - h(x))$$

(3.13)

can be found by iterating the expression

$$x(i + 1) = x(i) + \left[ H(x(i))^T R^{-1} H(x(i)) \right]^{-1} H(x(i))^T R^{-1} (z - h(x(i)))$$

$$x(i = 0) = x \text{ estimate from previous epoch}$$

(3.14)

until $x$ converges (or until a maximum number of iterations is reached). Note that $i$ is an iteration index; the state estimate is found by iteration at every time step.

This technique finds a new state estimate at each epoch based on that epoch’s GPS measurements only. Thus it is sensitive to measurement noise at every epoch. $R$ can be based either on the average expected phase-measurement noise, or on the measured SNR values of the corresponding phase measurements (as described in Reference [72]).
Extend Kalman Filter (EKF)

The extended Kalman filter finds the state estimate that minimizes the state-error covariance matrix. At every time step $k$, the state estimate $x$ and the error-covariance estimate matrix $P$ are propagated from the previous estimates based on a model of the vehicle dynamics including actuator inputs $u$

$$x_k = F \hat{x}_{k-1} + Gu_{k-1}$$

$$P_k = F\hat{P}_{k-1}F^T + Q$$

where $Q$ is the covariance of the dynamic noise on the states. Next, the Kalman gain is computed

$$K_k = P_kH^T(z_k|H(x_k)P_kH^T(x_k) + R_k)^{-1}$$

where $H(x)$ is as defined in Equation 3.12 and $R$ is the measurement-noise matrix. Then the state estimate and the covariance matrix are updated from the current measurement $z$

$$\hat{x}_k = x_k + K_k(z - h(x_k))$$

$$\hat{P}_k = (I - K_kH(x_k))P_k$$

which yields the current estimate of the state, $\hat{x}$.

The EKF includes a model of the vehicle dynamics. This allows velocity states to be added to the estimation problem (i.e., the state vector in Equation 3.1 can include translation and rotation rates even though they are not directly measured). The expected effects of thruster firings can be included in the estimation process to improve accuracy. Also, the state-propagation step can run at a higher rate to facilitate control at a higher bandwidth than the GPS measurement updates (e.g., on the FFTB, GPS updates arrive at 10 Hz but the control loops run at 60 Hz).

Iterated Extended Kalman Filter (IEKF)

The IEKF is the same as the standard EKF with the measurement update step performed repeatedly at each time step to allow the state estimate to converge. Thus the update
equations (3.17-3.19) are modified slightly

\[
K_k(i) = P_k H^T(\hat{x}_k(i)) H(\hat{x}_k(i)) P_k H^T(\hat{x}_k(i)) + R_k)^{-1} \\
\hat{x}_k(i + 1) = x_k + K_k(i)[z - h(\hat{x}_k(i)) - H(\hat{x}_k(i))(x_k - \hat{x}_k(i))] \\
\dot{P}_k(i + 1) = [I - K_k(i) H(\hat{x}_k(i))] P_k \\
\hat{x}_k(i = 0) = x_k
\]

(3.20) (3.21) (3.22) (3.23)

The IEKF typically converges more quickly than the EKF when the state estimate is poor (e.g., at initialization or after a large disturbance). However, there is more computational effort involved. The following sections examine this trade-off in more detail.

3.4.2 Estimator Accuracy Performance Metric

Based on analysis of the FFTB GPS noise sources, Zimmerman predicted relative-positioning errors with a standard deviation of 2.6 cm using a static filter (Reference [72], page 100). However, Zimmerman’s step-response experiment (ibid, page 82) shows significantly better relative-positioning performance (maximum estimation error of 2.5 cm).

For this research, experimental performance data is based on “best-case” runs; generally when the vehicles manage to maintain tracking of at least 5 pseudolites during the experiment. This philosophy is chosen partly because the FFTB represents a poor GPS signal environment (compared to the expected space environment), and partly because these experiments are designed to demonstrate the potential performance of GPS as a relative sensor for spacecraft formation flying.

The metric \( J \) represents the GPS-based relative-positioning error compared to the overhead vision system.

\[
J = \sqrt{\Delta x_{12(\text{vision})} - \Delta x_{12(\text{GPS})})^2 + (\Delta y_{12(\text{vision})} - \Delta y_{12(\text{GPS})})^2}
\]

(3.24)

where \( \Delta x_{12(\text{sensor})} \) and \( \Delta y_{12(\text{sensor})} \) are the difference in position between vehicle 1 and vehicle 2 measured in the \( x \)- and \( y \)-directions by the sensor.

This metric represents the radial distance between the relative-position vectors found by the GPS system and the truth sensor (the overhead vision system). \( J \) is the magnitude of \( \vec{E} \) in Figure 3.2. Note that the GPS-based state estimate includes the full 3D position, however the vision system only senses in the \( x \) and \( y \) axes. The error in the vision system for
relative-position sensing is thought to be quite small: generally better than 0.5 cm. Thus the mean of the error metric provides a good indication of the error in the GPS-based relative-position estimate. However, $J$ is not a good indicator of sensing bias. That is, $J$ cannot be used to determine if the error is zero-mean or if there is a bias offset. References [60] and [61] include this type of analysis for GPS sensing experiments on the FFTB.

### 3.4.3 Two-Vehicle Experiments

References [60] and [61] investigate the GPS-based relative-state estimation performance for the two-vehicle problem based on the FFTB. Table 3.1 combines the results from these tests\(^3\), including simulation data, post-processed experimental data, and real-time experimental data. The estimation algorithms listed in Section 3.4.1 are compared in terms of computation per sensing epoch and sensing performance based on the error metric $J$ in Equation 3.24. As noted in Table 3.1, the cases considered include a two-vehicle simulation without motion, post-processed experimental data with and without vehicle motion, and real-time estimation for two FFTB vehicles undergoing motion.

All test cases but the last line in Table 3.1 were presented in Reference [60]. The last case was presented in Reference [61]; it differs from other IEKF implementations in that

---

\(^3\)This research was accomplished cooperatively with Tobé Corazzini.
CHAPTER 3. GPS SENSING FOR SPACECRAFT FORMATION FLYING

Table 3.1: Relative-position sensing for two vehicles. Computational cost per step and estimation performance is listed for several FFTB cases, including: simulation, post-processed experimental data, and real-time experimental estimation. EKF is extended Kalman filter, IWLS is iterated weighted least squares, and IEKF is iterated extended Kalman filter.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Mean MFLOPS Per Step</th>
<th>Data Source</th>
<th>Vehicle Motion</th>
<th>Rel. Pos. Error $J$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean (cm)</td>
</tr>
<tr>
<td>EKF</td>
<td>0.43</td>
<td>Simulation</td>
<td>Fixed</td>
<td>1.02</td>
</tr>
<tr>
<td>IWLS</td>
<td>1.51</td>
<td>Simulation</td>
<td>Fixed</td>
<td>1.01</td>
</tr>
<tr>
<td>IEKF</td>
<td>1.01</td>
<td>Simulation</td>
<td>Fixed</td>
<td>0.98</td>
</tr>
<tr>
<td>EKF</td>
<td>0.54</td>
<td>Post-Processed</td>
<td>Fixed</td>
<td>0.57</td>
</tr>
<tr>
<td>IWLS</td>
<td>1.17</td>
<td>Post-Processed</td>
<td>Fixed</td>
<td>1.44</td>
</tr>
<tr>
<td>IEKF</td>
<td>0.99</td>
<td>Post-Processed</td>
<td>Fixed</td>
<td>0.57</td>
</tr>
<tr>
<td>EKF</td>
<td>0.54</td>
<td>Post-Processed</td>
<td>Moving</td>
<td>0.94</td>
</tr>
<tr>
<td>IWLS</td>
<td>1.18</td>
<td>Post-Processed</td>
<td>Moving</td>
<td>1.60</td>
</tr>
<tr>
<td>IEKF</td>
<td>1.11</td>
<td>Post-Processed</td>
<td>Moving</td>
<td>0.93</td>
</tr>
<tr>
<td>IEKF</td>
<td>*</td>
<td>Real-Time</td>
<td>Moving</td>
<td>0.78</td>
</tr>
</tbody>
</table>

* Computational cost not measured for this case.

velocity was also estimated and actuator histories were included in the filter for the first time (previous results ignored the vehicle dynamics). For this case, the vehicles were performing the estimation in real-time, thus the computational effort was not measured, however this case proves the feasibility of this technique for real-time (10 Hz) estimation on a real-time computer system. The addition of velocity-state estimates is important because it enables the use of the fuel-time-optimal control algorithms discussed in Chapter 4. A control loop was closed around the estimator of the last case for cm-level formation flying on the FFTB with GPS as the relative sensor (see Section 4.2.3).

The EKF algorithm performs a fixed number of operations at each step. The difference in EKF computation between the simulated and experimental GPS data reflects the extra processing required to compute the measurement-noise covariance matrix $R$ based on the measured SNRs.

It should be noted that the two-vehicle version of the full problem listed in Equation 3.8 is solved for all cases listed in Table 3.1. Thus each vehicles' full state (including attitude) is estimated. However, for simplicity only the relative-position estimation performance in $x$ and $y$ is reflected in Table 3.1.
CHAPTER 3. GPS SENSING FOR SPACECRAFT FORMATION FLYING

Verifying the GPS Simulation Environment

As mentioned in Section 2.3, simulation studies are an important component in testing GPS sensing for formation flying. It is clear from the first six lines of Table 3.1 that the simulation does not exactly capture the system characteristics. The simulation does not model the complex noise on the phase measurement signals, which is dominated by multipath noise. However, the simulation results are close enough to the experimental results to serve as an indicator of the expected performance of the real system. Comparing the simulation and post-processed results, the mean of $J$ is within 50% (differences less than 0.5 cm) for all three estimation algorithms, as is the average computation required per step.

Furthermore, the simulation results here are conservative for the EKF filters. So the simulated performance can be treated as a conservative estimate of the actual EKF performance in the FFTB environment for cases that cannot be tested experimentally.

Algorithm Comparison

The simulation code models system noise as zero-mean white noise with the covariance values listed in Table 2.4. Furthermore, the simulated vehicles remain fixed. As expected, the IWLS performs about the same as the EKF methods for the simulated cases. However, experimental data from the FFTB has both measurement and process noises that are more complex. The EKF’s more sophisticated estimation results in improved performance, while the IWLS does not perform as well. The iterated version of the EKF slightly outperforms the non-iterated EKF. This reflects the faster initial estimate convergence. However, this IEKF feature comes at a substantially higher computational cost.

Vehicle motion degrades the state estimates by $\sim 64\%$ for the EKF methods and $11\%$ for the IWLS. This estimator accuracy degradation is partly due to the large changes in SNR and multipath for small motions on the FFTB and partly because motion cannot be modelled without velocity states (which is why the EKF performance degrades more than the IWLS). Adding velocity states and actuator knowledge recovers $\sim 42\%$ of the motion-degraded error, as seen by the results in the last line.

The Kalman filter algorithms outperform the IWLS algorithm significantly in estimation accuracy. This result was experimentally confirmed with real-time moving-vehicle relative-positioning experiments reported in Reference [60] for the IWLS and Reference [61] for the IEKF. Comparing the relative-error distributions in $x$ and $y$, the range of the error is
reduced from \( \sim 4 \) cm for the IWLS to \( \sim 1.5 \) cm for the IEKF. Furthermore, the iterated version of the Kalman filter requires less computation than the iterated weighted least-squares algorithm, despite having a higher computational cost per iteration, because it converges to an estimate in fewer iterations.

The IEKF is appealing for its rapid convergence when the initial estimate is poor. This will be the case at initialization and also when integer estimates are updated (e.g., due to a new GPS transmitter coming into view). The IEKF requires about twice the computation at each step as the EKF does\(^4\).

The IEKF-based relative-state estimation achieves mean position accuracy better than 1 cm with 3-\( \sigma \) performance better than 2 cm. As noted at the start of Section 3.4, this promising experimental result was attained in the FFTB environment with perfect integer ambiguity knowledge (and thus zero error contribution from integer estimation). However, the results indicate that cm-level relative sensing is achievable with DGPS, which validates its potential as a spacecraft formation flying sensor.

### 3.4.4 Three Vehicles: Simulated Performance

The three-vehicle relative-state estimation problem introduces the additional design decision: whether the estimation is performed as one group of three, or as three groups of two. The three-vehicle group is called a trio; the two-vehicle groups are called pairs. This section compares these two estimation approaches in simulation for the three-vehicle FFTB setup. The simulation results quantify potential performance benefits for trio-based estimation compared to pair-based estimation, and the corresponding computational cost. Robustness and communication cost differences are also discussed.

Pairs estimate the relative state between the two vehicles based on the GPS measurements from those two vehicles. The trio approach performs estimation of all relative states as a single estimation problem. The trio estimator must gather all measurements, process the estimation problem, and distribute the results (if necessary), which implies significant computation and communication costs. The trio approach may also be vulnerable to a single-point failure in the central processor. The pair approach divides the problem into smaller, less costly estimation problems that can be solved in a distributed fashion. The

\(^4\)It is possible to reduce this computational cost by adjusting the convergence test and by removing expensive numerical steps. For example, \( H(x) \) does not need to be recomputed in Equations 3.20–3.23 for \( i > 1 \).
Table 3.2: Simulated relative-position sensing performance for three vehicles. Computational cost per measurement epoch and estimation accuracy are listed for sensing performed as three pairs and as one trio.

<table>
<thead>
<tr>
<th>Estimator Implementation</th>
<th>Mean MFLOPS Per Step</th>
<th>Relative Position Error $J$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Three Pairs</td>
<td>1.07</td>
<td>0.608</td>
</tr>
<tr>
<td>One Trio</td>
<td>3.38</td>
<td>0.497</td>
</tr>
</tbody>
</table>

pair approach is also more robust to failures because relative estimation can continue among functioning vehicles in the formation. These two estimation approaches are the only choices for a three-vehicle formation; but pairs and trios can be used as building blocks in a distributed estimation schemes for larger formations (as will be discussed in Section 3.5).

The two estimator schemes were tested in simulation using the IEKF algorithm with velocity-state estimation and actuator knowledge. The noise models remain the same as before. This simulation includes moving vehicles (as opposed to the simulations of Section 3.4.3, which model the vehicles as stationary). The estimation algorithm propagates the estimated state at 60 Hz (with GPS measurement updates at 10 Hz as before), which matches the operating rate of the FFTB vehicles. These changes lead to an improvement in predicted estimation accuracy. The runs simulate 60 seconds of three FFTB vehicles undergoing small motions with periodic open-loop thruster firings.

The relative-state estimation-accuracy results are listed in Table 3.2 for the pair and trio implementations. For the trio case, one vehicle is solving the three-vehicle state-estimation problem. The computational cost required for that vehicle is listed. The three-pairs case assumes each vehicle is estimating its state relative to one other vehicle, and that each of the three vehicles is doing estimation. The mean per-vehicle computational cost is listed.

Section 3.5.2 discusses the system engineering implications of centralized and distributed estimation, including robustness and implementation costs, in more detail.

**Estimator Implementation Comparison**

Clearly the trio implementation outperforms the three-pairs implementation in terms of estimation accuracy, but at a large cost in computational effort. The three-pairs implementation requires three problems to be solved at an average computational cost of $\sim 1$ MFLOPs per measurement update step per relative state. However these three problems
can be solved in parallel (e.g., on the three different vehicles). The trio implementation costs more than three times the computational effort per measurement update step. Therefore the processor performing trio estimation will need to be more than three times faster than each of the three separate processors performing pair estimations. This may be significant for near-term space missions due to the limited availability of highly-capable space-qualified processors.

As shown in Table 3.2, the trio implementation estimates the relative positions 18% better than the three-pairs implementation, with a 68% smaller standard deviation. This result is not necessarily intuitive, because the trio estimator is essentially adding the GPS measurements from a third vehicle into the solution of the relative state between two other vehicles and coming up with a more accurate result for the relative position between those two vehicles. However, a comparison of the numbers of unknowns and measurements for each problem provides some insight into this result. On the formation flying testbed, each pair of vehicles computes a set of 5 double-difference measurements from the six pseudolites, according to Equation 3.5. Because the double-difference measurements contribute primarily to position determination, consider the number of unknown position states and the double-difference measurements for each estimator implementation. Each pair estimator solves for 6 unknown position states (3 per vehicle) using 5 double-difference measurements. A third vehicle adds 3 more unknowns and 5 independent double-differences. Thus, the trio implementation solves for 9 unknown positions using 10 double-difference measurements, compared to a three-pairs implementation that solves for 18 unknowns using 15 measurements (as three pair-estimation problems). Therefore, the ratio of measurements to unknowns is better for trio estimation than for three-pairs estimation.

This simulation result for the FFTB includes a simplified model of the phase noise and does not reflect errors from integer estimation. The observed estimation improvement for centralized estimation is valid for the FFTB problem formulation, as described in Section 3.2. Potential applications may include pseudolite-based or pseudolite-augmented environments, such as GPS self-constellations, and widely-separated spacecraft formations in LEO.

For larger formations of this type, it is reasonable to expect the estimation accuracy to further improve with more vehicles added to a centralized estimation scheme. However, the

---

5Note the 6 states can be resolved from 5 measurements because of the coupling between the attitude and absolute-position problems.
computational cost will grow rapidly with the number of vehicles. Thus, even if this trend is confirmed experimentally, a distributed approach will still be necessary at some point. For larger formations, distributed approaches can use pairs or trios as building blocks for distributed estimation architectures. An example case is presented in the Section 3.5.

3.4.5 Summary

This section compared three relative-state estimation algorithms. Experimental results show a Kalman filter is more accurate and less computationally expensive than a weighted least-squares algorithm. Furthermore, an iterated version of the Kalman filter allows faster convergence from a poorly-known initial condition. Thus the IEKF is chosen as the best estimator algorithm for this application. Experimental results demonstrate cm-level estimation accuracy (mean 0.78 cm, standard deviation 0.32 cm) in real time at 10 Hz, given perfect knowledge of integer ambiguities. Chapter 4 experimentally demonstrates closed-loop control around this estimator, achieving cm-level control (less than 2 cm position error), which validates GPS sensing for formation flying.

Trio-based estimation is shown to be more accurate than pair-based estimation in simulation. This result indicates that GPS sensing accuracy improves as vehicles are added to the problem in the FFTB setup, the first time this accuracy improvement has been identified and quantified. This useful result may be applicable to some of the upcoming formation flying missions that will use GPS for relative sensing.

Finally, simulations indicate that trio-based estimation on the FFTB can achieve sub-cm level accuracy (mean 0.5 cm, standard deviation 0.2 cm). This result has not been experimentally verified, but is based on FFTB simulation results. The parallel integer-estimation problem will prevent this performance from being duplicated in experimental systems without perfect knowledge of the integer ambiguities. However this result does indicate that GPS-sensing alone — even with added noise from integer-ambiguity estimation — will be sufficient to bring a spacecraft formation into a tightly-aligned virtual spacecraft bus, potentially reducing the burden on the optical alignment layer (and thus reducing the mission cost).
CHAPTER 3. GPS SENSING FOR SPACECRAFT FORMATION FLYING

3.5 Estimation Architectures

A formation of three or more vehicles requires an estimation architecture. The estimation architecture determines how the formation estimation task is organized. Choices range from fully centralized, to fully distributed, with the number of intermediate choices growing as the formation grows. This section examines the performance-cost trade-off of the estimation architecture design problem, including robustness and implementation costs. This discussion applies to the FFTB setup, and may extend to similar relative-sensing applications of GPS.

Based on the simulation results in Section 3.4.4, the centralized solution yields the most accurate results for systems like the FFTB; however, the computational cost of centralized estimation grows quickly with the number of vehicles. Figure 3.3 illustrates the computational cost of centralized estimation versus the number of vehicles in the formation. This analysis is based on centralized estimation using the IEKF algorithm implemented on the FFTB with two to six vehicles.

The estimation architecture determines hardware requirements (through computational and communication requirements), performance (through estimation accuracy), and robustness (through the complexity and redundancy of the architecture). The task of choosing an estimation architecture obviously depends on the particulars of the mission. In this section, architecture choices for an example six-vehicle mission illustrate the performance trade-off and the resulting impact on mission requirements.

3.5.1 Example: Six-Vehicle Estimation Architectures

As an example, consider three possible estimation architectures (illustrated in Figure 3.4) for six vehicles on the formation flying testbed or in a similar GPS environment. The cases presented include: the simplest distributed architecture (case 1), the fully-centralized architecture (case 3), and one intermediate case (case 2). Predicted accuracy and computational cost numbers are based on the simulation results in Section 3.4.4. This example shows order-of-magnitude guidelines for performance vs. computation and connectivity. The three candidate estimation architectures are listed below.

1. Distributed (pair-based) Each of five “spoke” vehicles estimates its state relative to the center “hub” vehicle. Each spoke vehicle uses \( \sim 1 \text{ MFlops} \) per estimation step
and relative-position accuracy is $\sim 0.61$ cm (mean of $J$). Only the spoke states are estimated.

2. **Distributed (trio-based)** Each spoke vehicle solves the trio estimation problem for the relative states between itself, one other vehicle and the hub vehicle. This adds more phase observations to the estimation problem, which increases estimation accuracy as well as the computational cost of the problem. Each vehicle needs $\sim 3.4$ MFlops per estimation step, and relative-position accuracy is $\sim 0.50$ cm. Both spoke and “edge” states are estimated.

3. **Centralized** The phase measurements from all vehicles are combined into a single estimator on the hub vehicle which solves for all relative states simultaneously. This case was not implemented on our testbed. The computational cost of the estimation problem grows as $O(N^3)$ where $N$ is the number of vehicles. Extrapolating from the
CHAPTER 3. GPS SENSING FOR SPACECRAFT FORMATION FLYING

Figure 3.4: Three estimation architectures for a six-vehicle satellite cluster. 1) Distributed (pair-based), 2) Distributed (trio-based), and 3) Centralized.

$N = 2, 3$ cases, the computation cost for $N = 6$ will be on the order of $\sim 21$ MFlops\(^6\). Accuracy will improve over the other cases, but the level of improvement is not known.

3.5.2 Estimation Architecture Summary

A distributed estimation architecture based on multiple trio estimators appears to yield high estimation accuracy without extremely high computational cost. This estimation accuracy may justify the extra computational cost compared to the pair-based distributed architecture. On the other hand, a fully distributed pair-based architecture may be sufficient to serve as the outer-loop sensor for an optical metrology system, and is more flexible, has the

\(^6\)Note that the step from 1 MFlops to 3 MFlops required a processor upgrade for our testbed from 25MHz 68040s to 200 MHz PowerPC 604s.
minimum computational demand, and has no cross-link requirements other than that the hub vehicle must broadcast its GPS phase measurements to the other vehicles. The centralized architecture provides maximum estimation accuracy, but the computational cost is probably prohibitively expensive for near-term formation flying missions.

The architecture determines the estimation accuracy as well as the computational resources required. Also, simpler distributed architectures may be more robust than complex centralized architectures. The mission designer can improve the robustness of either architecture at the price of adding the computational resources necessary to provide backup capability; however the distributed architecture is inherently less vulnerable to single-point failures. Thus, in defining formation flying mission requirements, centralized estimation is more expensive in terms of required computation and communication hardware; and it may necessitate additional hardware costs to provide fault-tolerant performance.
CHAPTER 3. GPS SENSING FOR SPACECRAFT FORMATION FLYING

Note that the parallel control-architecture design problem (discussed in Section 4.5) has an impact on the mission requirements as well. If centralized control is used, then a distributed estimation architecture must collect the state estimates in one place (which requires additional communication resources). Similarly, distributed control requires a centralized estimation architecture to distribute the state estimates to the vehicles.

Communication cross-links must be able to send GPS phase measurements and state estimates between vehicles in a reliable manner at the desired estimation bandwidth (e.g., 60 Hz for the FFTB example in Section 3.4.4). As the size of the formation grows, the synchronization of data flow becomes important as well. That is, the centralized estimation architecture requires GPS information from all vehicles before estimation at each time step. This requirement may become more challenging as formation size grows. Thus distributed estimation architectures limit the design impacts from the estimation subsystem, particularly on cross-link communication demands.

The analysis presented here illustrates the estimation-architecture trade space for an example formation flying mission based on three-vehicle FFTB simulations. The accuracy versus computational cost of the three architectures is plotted in Figure 3.5. Note the accuracy for architecture 3 is not known but will be better than architecture 2. The mission designer must choose the estimation architecture that meets the accuracy and robustness needs without excessive computation and communication requirements. These results can serve as an input to the system-engineering trade studies that will be required for formation flying missions.

3.6 Conclusions

This chapter has explored the use of GPS-based relative sensing for spacecraft formation flying. Issues that must be addressed, including multipath mitigation and self-constellations for operation outside LEO, have been discussed. Three estimator algorithms were compared in simulation and experiment on the FFTB; the IEKF is the most appealing of these for the formation flying problem.

Centimeter-level relative-position estimation was experimentally demonstrated on the formation flying testbed. Simulation results demonstrate that centralized estimation provides better accuracy for three vehicles than distributed (pair-based) estimation on the FFTB. Furthermore, this accuracy improvement and the corresponding computational cost
have been quantified in simulation. A promising simulation result indicates that trio-based relative-position estimation may achieve accuracies on the order of 1 cm or less. This result does not include integer-estimation errors, and applies to the FFB GPS environment. However, this result is a promising indicator of the potential good relative-sensing accuracy achievable using DGPS. These experimental and simulation results validate GPS-based relative-state estimation as a viable cm-level sensor for spacecraft formation flying.

A six-vehicle example was used to illustrate the estimation-architecture design problem based on the simulation results in Section 3.4.4. Centralized estimation provides the best sensing accuracy, but the computational cost grows rapidly with the number of vehicles. Robustness and communication requirements demand additional capability, which translates to higher implementation costs as well. Distributed estimation achieves cm-level sensing with far less implementation costs. The best estimation architecture will be mission-dependent; but distributed estimation schemes will probably be preferred for near-term formation flying missions.
Chapter 4

Formation Flying Control

The on-off nonlinearity of spacecraft thrusters limits the utility of linear control-design techniques for spacecraft formation flying. Instead, a nonlinear technique is needed to achieve efficient control (see Reference [47]). The well known single-vehicle “bang-off-bang” control law achieves optimal performance according to a cost function that includes fuel and time (see, e.g., Reference [90]). This chapter presents a new cooperative version of this control law for relative stationkeeping by a two-vehicle formation. Simulation and experimental results on the FFTB verify the performance advantage of this new controller. A new numerically-optimized relative-control technique applicable to formations of three or more vehicles is also introduced. Simulation results demonstrate the advantage of this technique compared to other multi-vehicle relative-control strategies. Finally, the spacecraft formation flying control-architecture design problem is investigated. A six-vehicle simulation quantifies the performance trade-off between centralized, distributed and a new hierarchical control architecture.

4.1 Introduction

One challenge in designing control laws for a formation of spacecraft is the use of on-off actuators for position control. Due to this nonlinearity in the system, linear control techniques are of limited usefulness. Linear controllers combined with thresholding or pulse-width-modulation schemes produce sub-optimal behavior because they do not directly account for the saturation nonlinearity of the thrusters. Therefore, nonlinear control laws that directly account for the on-off nature of conventional spacecraft thrusters maximize efficiency. This
chapter derives two new nonlinear control laws for position control along one axis; the results are directly applicable to three-axis position control for spacecraft with independent thrusters on each axis.

4.2 Bang-Off-Bang Control

The weighted fuel-time-optimal control law for a single vehicle with on-off actuators (e.g., opposing thruster pairs aligned along a common axis) is the well known bang-off-bang control law (see Reference [90]). This control law produces optimal trajectories according to the cost function

\[ J = \int_{t_0}^{t_f} |1 + \lambda u(t)| \, dt \]  

(4.1)

The maneuver occurs during the interval \( t \in [t_0, t_f] \); \( \lambda \) is the penalty applied to fuel use. \( \lambda \) must be non-negative; for \( \lambda = 0 \), this control reduces to “bang-bang” control with thrusters firing throughout the trajectory. The control history \( u(t) \) is bounded by \( \pm 1 \) where \( 1 \) means maximum positive thrust (i.e. both positive thrusters firing\(^1\)) and \(-1\) means maximum negative thrust. The control is computed as

\[
u = \begin{cases}
+1 & \dot{x} < g_1(x) \text{ and } \dot{x} < g_2(x) \\
-1 & \dot{x} > g_1(x) \text{ and } \dot{x} > g_2(x) \\
0 & \text{otherwise}
\end{cases}
\]  

(4.2)

\[
g_1(x) = -\text{sgn}(x) \sqrt{\frac{2}{1 + 4\lambda}} |x| k_{\text{max}}
\]  

(4.3)

\[
g_2(x) = -\text{sgn}(x) \sqrt{2|x| k_{\text{max}}}
\]  

(4.4)

where, for a vehicle with two positive and two negative thrusters in each axis,

\[
k_{\text{max}} = \frac{\text{acceleration}}{\text{thruster}} \times 2 \text{ thrusters}
\]  

(4.5)

\(^1\)This convention is used throughout because translational thrusters are usually deployed in pairs to provide torque-free actuation. For example, the FFTB vehicles have pairs of thrusters oriented in each direction, for a total of eight thrusters.
This algorithm produces a fuel-time-optimal trajectory. Given the current state, and the model of the plant

\[ \ddot{x}(t) = k_{\text{max}}u(t) \]  

(4.6)

the optimal trajectory is completely determined for all time. Ideally, the trajectory would need to be computed only once. However, perturbations to the system, including noise and model errors, require the trajectory to be recomputed periodically. Therefore, the bang-off-bang solution is usually implemented as a feedback controller: at each time step, the optimal trajectory is computed and the current control action, \( u(0) \), is executed.

Also note that \( 1/s^2 \) systems never reach zero error exactly. To prevent the controller from correcting insignificantly small errors, the system requires a deadband function to disable the controller when the error is within an acceptable tolerance. This is true for any \( 1/s^2 \) system with on-off actuators. Section 4.2.2 discusses the deadband design used for the FFTB experiments.

### 4.2.1 Derivation/Review

Reference [90] contains a complete derivation of the bang-off-bang control algorithm. It is reviewed here because it serves as the foundation for the derivation of the two-vehicle weighted fuel-time-optimal controller developed in Section 4.3.

The single-vehicle continuous dynamics in one translation axis are described by

\[ \dot{x}(t) = v(t) \]  

(4.7)

\[ \dot{v}(t) = k_{\text{max}}u(t) \]  

(4.8)

where \( x \) is position, \( v \) is velocity, \( k_{\text{max}} \) is the two-thruster acceleration, and \( u(t) \) is the direction to fire (i.e. \( +1 \) means fire two thrust in the positive direction). Note that \( u(t) \) technically could be \( \pm 0.5 \) to signify firing only one thruster; however this never occurs as will be shown. The optimal control law transfers this system from an arbitrary initial state \((x(t_0), v(t_0))\) to the final state \((x(t_f) = 0, v(t_f) = 0)\) with a minimum value of the performance measure \( J \) defined by

\[ J(u(t)) = \int_{t_0}^{t_f} (1 + \lambda|u(t)|) \, dt \]  

(4.9)
The control history that minimizes $J(u(t))$ is the *optimal control*, denoted $u^*(t)$. The optimal control is found using calculus of variations techniques (described more fully in [90]). Define the *Hamiltonian*

$$
\mathcal{H}(\mathbf{x}(t), u(t), p(t)) \equiv g(\mathbf{x}(t), u(t), t) + p^T(t)[a(\mathbf{x}(t), u(t), t)]
$$  \hspace{1cm} (4.10)

where $p(t) = [p_1(t), p_2(t)]^T$ is the vector of *Lagrangian multipliers*, also called the *costate*, and where

$$
J(u(t)) = \int_{t_0}^{t_f} g(\mathbf{x}(t), u(t), t) dt
$$  \hspace{1cm} (4.11)

and

$$
\dot{\mathbf{x}}(t) = a(\mathbf{x}(t), u(t), t)
$$  \hspace{1cm} (4.12)

where $\mathbf{x}(t) = [x(t), v(t)]^T$ is the state vector.

For this problem, the Hamiltonian is

$$
\mathcal{H}(\mathbf{x}(t), u(t), p(t)) = 1 + \lambda |u(t)| + p_1(t)v(t) + p_2(t)k_{max}u(t)
$$  \hspace{1cm} (4.13)

The optimal actuator history $u^*(t)$ and the associated optimal state and costate histories ($\mathbf{x}^*(t)$ and $p^*(t)$, respectively) satisfy the necessary conditions for optimality

$$
\dot{\mathbf{x}}^*(t) = \frac{\partial \mathcal{H}}{\partial \mathbf{p}}(\mathbf{x}^*(t), u^*(t), p^*(t))
$$  \hspace{1cm} (4.14)

$$
\dot{p}^*(t) = -\frac{\partial \mathcal{H}}{\partial \mathbf{x}}(\mathbf{x}^*(t), u^*(t), p^*(t))
$$  \hspace{1cm} (4.15)

$$
\mathcal{H}(\mathbf{x}^*(t), u^*(t), p^*(t)) \leq \mathcal{H}(\mathbf{x}(t), u(t), p(t))
$$  \hspace{1cm} (4.16)

Equation 4.14 states that the optimal control and state histories satisfy Equations 4.7 and 4.8. Equation 4.15 is called the costate equation; it defines the costate $p(t)$ such that the coefficient of the variational term $\partial \mathbf{x}(t)$ is zero. Equation 4.16, called *Pontryagin’s minimum principle*, states that an optimal control $u^*(t)$ must minimize the Hamiltonian $\mathcal{H}$. A final necessary condition applies for problems with “open final time” (i.e. the duration of the trajectory is not constrained). This condition is

$$
\mathcal{H}(\mathbf{x}^*(t), u^*(t), p^*(t)) = 0
$$  \hspace{1cm} (4.17)

on the extremal (optimal) trajectory.
Applying Equation 4.16 to the Hamiltonian in Equation 4.13 (including only the terms that depend on $u(t)$),

$$\lambda |u^*(t)| + p_2(t)k_{\text{max}} u^*(t) \leq \lambda |u(t)| + p_2(t)k_{\text{max}} u(t)$$  \hspace{1cm} (4.18)

The definition of the absolute value is

$$|u(t)| = \begin{cases} u(t), & \text{for } u(t) \geq 0 \\ -u(t), & \text{for } u(t) < 0 \end{cases}$$  \hspace{1cm} (4.19)

so

$$\lambda |u(t)| + p_2(t)k_{\text{max}} u(t) = \begin{cases} |\lambda + p_2(t)k_{\text{max}}|u(t), & \text{for } u(t) \geq 0 \\ -|\lambda + p_2(t)k_{\text{max}}|u(t), & \text{for } u(t) < 0 \end{cases}$$  \hspace{1cm} (4.20)

Therefore, the value of $u(t)$ that minimizes the Hamiltonian depends on the values of $\lambda$ and $p_2(t)k_{\text{max}}$. If $(\lambda + p_2(t)k_{\text{max}})$ is negative (i.e. if $p_2(t)k_{\text{max}} < -\lambda$), the optimal control is $+u_{\text{max}}$. If $(-\lambda + p_2(t)k_{\text{max}})$ is positive (i.e. if $p_2(t)k_{\text{max}} > \lambda$), the optimal control is $-u_{\text{max}}$. If neither condition is true, the optimal control $u^*(t)$ is zero. The admissible control is bounded by one, so the optimal control $u^*(t)$ is

$$u^*(t) = \begin{cases} +1 & \text{for } p_2(t) < -\frac{\lambda}{k_{\text{max}}} \\ -1 & \text{for } p_2(t) > \frac{\lambda}{k_{\text{max}}} \\ 0 & \text{otherwise} \end{cases}$$  \hspace{1cm} (4.21)

Note that $u^*(t)$ could have multiple values when $p_2 = \pm \frac{\lambda}{k_{\text{max}}}$, so it can be defined to be 0 for these conditions. Given this definition, $u(t)$ will not have values other than 0, 1 or -1.

Equation 4.15 defines the time-derivatives of the costates

$$\dot{p}_1^*(t) = 0$$  \hspace{1cm} (4.22)

$$\dot{p}_2^*(t) = -p_1(t)$$  \hspace{1cm} (4.23)

which leads to the form of the costates

$$p_1^*(t) = c$$  \hspace{1cm} (4.24)

$$p_2^*(t) = -ct + d$$  \hspace{1cm} (4.25)
where $c$ and $d$ are constants. Immediately this implies that $p_5(t)$ changes linearly in time. It is clear from this and from Equation 4.21 that the optimal control history can switch at most two times. That is, $u^*(t)$ will follow one of the two sequences: $\{+1,0,-1\}$ or $\{-1,0,+1\}$, or a subset thereof. The next step is to determine when these switches occur. The first sequence $\{+1,0,-1\}$ is assumed for this part of the derivation (the result will apply to both sequences and all valid subsets). Define the two switch times as $t_1, t_2$ (where $t_0 < t_1 < t_2 < t_f$). Since $x(t_f) = 0, v(t_f) = 0$ and

$$u(t) = -1 \text{ for } t \in [t_2, t_f]$$

(4.26)

the state history during this part of the optimal trajectory is known

$$x(t_f) = 0 = x^*(t_2) + v^*(t_2)(t_f - t_2) - \frac{1}{2}k_{\text{max}}(t_f - t_2)^2$$

(4.27)

and

$$v(t_f) = 0 = v^*(t_2) - k_{\text{max}}(t_f - t_2)$$

(4.28)

then

$$(t_f - t_2) = \frac{1}{k_{\text{max}}}v^*(t_2)$$

(4.29)

substituting,

$$x^*(t_2) = -\frac{1}{k_{\text{max}}}v^*(t_2)^2 + \frac{1}{2}k_{\text{max}}\left(\frac{1}{k_{\text{max}}}v^*(t_2)\right)^2$$

(4.30)

or

$$x^*(t_2) = -\frac{1}{k_{\text{max}}}v^*(t_2)^2 + \frac{1}{2k_{\text{max}}}v^*(t_2)^2$$

(4.31)

which simplifies to

$$x^*(t_2) = -\frac{1}{2k_{\text{max}}}v^*(t_2)^2$$

(4.32)

which is the switching-condition test or switching line listed in Equation 4.4. Next, Equation 4.17 states that the Hamiltonian equals zero on the optimal trajectory, so evaluating at $t_1$

$$\mathcal{H} = 1 + p_1^*v^*(t_1) = 0$$

(4.33)

or

$$p_1^*(t) = -\frac{1}{v^*(t_1)} = c$$

(4.34)
CHAPTER 4. FORMATION FLYING CONTROL

Note that $v^*(t_1) = v^*(t_2)$ because thrusters are not fired between $t_1$ and $t_2$ (this is the drift portion of the trajectory). It is clear from Equations 4.21 and 4.25 that

$$p_2^*(t_1) = -\frac{\lambda}{k_{max}} = -ct_1 + d \quad (4.35)$$

$$p_2^*(t_2) = \frac{\lambda}{k_{max}} = -ct_2 + d \quad (4.36)$$

Subtracting yields

$$c(t_2 - t_1) = -2\frac{\lambda}{k_{max}} \quad (4.37)$$

and substituting Equation 4.34

$$(t_2 - t_1) = \frac{2\lambda v^*(t_1)}{k_{max}} \quad (4.38)$$

During the drift portion of the trajectory,

$$x^*(t_2) = x^*(t_1) + v^*(t_1)(t_2 - t_1) \quad (4.39)$$

substituting,

$$x^*(t_2) = x^*(t_1) + \frac{2\lambda}{k_{max}}(v^*(t_1))^2 \quad (4.40)$$

and substituting for $x^*(t_2)$ from Equation 4.32

$$-\frac{1}{2k_{max}}v^*(t_2) = x^*(t_1) + \frac{2\lambda}{k_{max}}(v^*(t_1))^2 \quad (4.41)$$

because $v^*(t_1) = v^*(t_2)$ this reduces to

$$x^*(t_1) = -\frac{(1 + 4\lambda)}{2k_{max}}v^*(t_1)^2 \quad (4.42)$$

which is the other switching line. Both switching lines (Equations 4.32 and 4.42) are rewritten for the general case in Equations 4.2–4.4.

Figure 4.1 shows the bang-off-bang control behavior in the phase plane (Reference [91] describes the phase plane analysis technique). This is a plot of velocity $v(t)$ versus position $x(t)$ in one axis. The two switching lines are plotted and one optimal trajectory is shown as an example. The switching lines divide the space into regions of optimal control.
CHAPTER 4. FORMATION FLYING CONTROL

Figure 4.1: Phase-plane portrait of single-vehicle bang-off-bang controller. Two switching lines and the optimal trajectory from an initial position of 0.2 and initial velocity -0.15 are plotted (the starting point is marked by a ’o’ on the plot).

behaviors \((u(t) = +1, -1 \text{ or } 0)\). The optimal trajectory switches when it reaches each line, demonstrating the bang-off-bang nature of the optimal control.

4.2.2 Deadband Design

With a deadband operating, the bang-off-bang controller becomes a “corrective” controller. That is, after initially moving quite close to the desired state it takes no action until the vehicle has drifted outside some bound. At that point, the optimal trajectory to return to zero is executed. Thus the vehicle state drifts within a deadzone, and the controller makes periodic corrections when the state drifts outside the zone. The frequency of these
corrections will depend not only on the disturbances acting on the system, but also on both the accuracy of the sensing and on the minimum impulse bit of the thrusters, which together determine how close to zero the controller can drive the state during each correction. The frequency also depends on the size of the deadband: smaller deadbands lead to more frequent corrections; however larger deadbands admit larger state errors. Thus deadband design is a trade-off between accuracy and control activity (which may interrupt virtual spacecraft bus observations as discussed in Section 1.2.2).

The experiments in this research use a simple deadband scheme. The controller acts until the measured velocity changes sign while the position is within the error bound (indicating that the system has been brought to rest with an acceptable position error), at which point the control is disabled. Control re-activates when the state leaves the deadband (in either position or velocity). The following logic implements this deadband strategy (a zero value of \( m \) disables the thrusters)

\[
\begin{align*}
\text{if} \quad (|x| > x_b \text{ or } |\dot{x}| > \dot{x}_b) \\
\text{then compute } u \text{ as usual and let } m = -\text{sgn}(\dot{x}) \\
\text{else} \\
\text{if } \text{sgn}(\dot{x}) = m \text{ then let } m = 0
\end{align*}
\]

\( x_b \) and \( \dot{x}_b \) create an error box around the origin in position/velocity space. For FFTB experiments, the position bound was \(~1–2\) cm and the velocity bound was \(~0.5–1.5\) cm/sec. This approach was sufficient for the demonstrations in this research.

### 4.2.3 Controller Experimental Performance

An experiment (published in Reference [61]) demonstrates cm-level control with only GPS-based relative-state estimation (as described in Chapter 3). The two-vehicle formation flying experiment employed the one-vehicle time-optimal controller. That is, the weight on fuel was set to zero for this experiment, resulting in bang-bang control. One vehicle maintained a desired absolute state in the GPS frame and the second vehicle regulated its state relative to the first vehicle using this controller. Figure 4.2 plots experimental data (relative position and relative velocity) in the phase plane.
Figure 4.2: Phase-plane portrait of relative controller experimental performance with GPS-based sensing only. This plot shows GPS-based estimated-state history. The estimate includes some time-lag plus multipath noise, distorting the expected parabolic trajectory shape (shown in Figure 4.1).

The deadband (±1.2 cm by ±0.9 cm/sec) is marked on the figure. Bang-bang control is a limiting case of bang-off-bang control where the two switching lines converge (eliminating the drift portion of the trajectory); the resulting single switching line is plotted. Note that it has been modified as follows to add damping to the system to compensate for estimation delays in this experimental setup (as discussed in Reference [61]): an offset was added to the line, and the maximum acceleration parameter was reduced such that the switching line
(from Equation 4.4) now obeys

\[ g_2(x) = -\text{sgn}(x)\sqrt{2|x|\left(k_{\text{max}} - k_1\right)} + \text{sgn}(x)k_2 \]

where \( k_1 = 0.5 \text{ cm/sec}^2 \) and \( k_2 = 1.2 \text{ cm} \) for this case. These modifications were chosen based on observed experimental performance; they add substantial damping that offsets the estimation delay, resulting in the successful performance shown in the figure.

The velocity ranges from +2 cm/s to -2 cm/s. But the position is cycling around the negative position error bound of \(-1.2 \text{ cm}\). The position is inside the position bound 80\% of the time and does not move more than 6 mm beyond the bound. The relative-position error therefore stays under 2 cm which was the goal of this experiment\(^2\). Similar performance is observed along the other translation axis. See Chapter 6 for more details on this experiment.

### 4.3 Control for Two Vehicles

A new two-vehicle weighted fuel-time-optimal controller has been derived (originally reported by Robertson et al. in Reference [92]). This cooperative controller uses the thrusters on both vehicles to regulate the inter-vehicle relative state. The controller is similar to the single-vehicle version discussed in the previous section, but it optimizes the more complicated two-vehicle cost function

\[ J = \int_{t_0}^{t_f} \left[1 + \lambda_1 |u_1(t)| + \lambda_2 |u_2(t)|\right] dt \]

where \( \lambda_1 \) and \( \lambda_2 \) are weights that penalize fuel use for each vehicle. For a given initial offset in relative position, the optimal trajectory is bang-off-bang (with different timing) for each vehicle. The switch times are determined by

\[ u_1 = \begin{cases} +1 & \dot{x} < g_1(x) \text{ and } \dot{x} < g_2(x) \\ -1 & \dot{x} > g_1(x) \text{ and } \dot{x} > g_2(x) \\ 0 & \text{otherwise} \end{cases} \]

\(^2\)There was no quiet-time goal for this experiment. The thrusters operated near-continuously to achieve the relative-separation performance.
The weights \( \lambda_1 \) and \( \lambda_2 \) balance fuel use between the vehicles and balance the combined fuel use against maneuver duration. For example, the relative values of the \( \lambda \)'s might be assigned in inverse proportion to the fuel remaining on each vehicle. A Matlab file that implements this controller is listed in Appendix B. The derivation of the two-vehicle controller is presented in the following section.

### 4.3.1 Two-Vehicle Optimal Control Law Derivation

The two-vehicle relative-state dynamics in one translation axis are

\[
\dot{x}(t) = \nu(t) \quad (4.59)
\]
\[ \dot{v}(t) = K_1u_1(t) + K_2u_2(t) \quad (4.60) \]

where \( x \) is the relative-position error and \( v \) is the relative velocity. These are defined by

\[
\begin{align*}
  x(t) &= x_1(t) - x_2(t) - x_{\text{desired}} \\
  v(t) &= v_1(t) - v_2(t) 
\end{align*}
\quad (4.61)
\]

and \( K_1, K_2 \) are the two-thruster accelerations for vehicles one and two, respectively. The positive-thrust directions of the two vehicles are defined to be in opposite directions for convenience in the derivation. Physically, this means that if the initial separation is positive, then positive thrust on either vehicle acts to increase the separation and negative thrust on either vehicle acts to decrease the separation. Thus the vehicle dynamics in the absolute sense are

\[
\begin{align*}
  \dot{x}_1(t) &= v_1(t) \\
  \dot{x}_2(t) &= v_2(t) \\
  \dot{v}_1(t) &= K_1u_1(t) \\
  \dot{v}_2(t) &= -K_2u_2(t) 
\end{align*}
\quad (4.63-4.66) \]

The optimal control law moves the system from an initial relative state \([x(t_0), v(t_0)]\) to the final relative state \([x(t_f) = 0, v(t_f) = 0]\) with a minimum value of the performance measure \( J \) defined in Equation 4.44.

The Hamiltonian is

\[
\mathcal{H}(x(t), u(t), p(t)) = 1 + \lambda_1 |u_1(t)| + \lambda_2 |u_2(t)| + p_1(t)v(t) + p_2(t)(K_1u_1(t) + K_2u_2(t)) \quad (4.67) \]

Applying Equation 4.16 yields conditions for the two \( u^*(t) \)s

\[
\begin{align*}
  u_1^*(t) &= \begin{cases} 
    +1 & \text{for } p_2(t) < -\frac{\lambda_1}{K_1} \\
    -1 & \text{for } p_2(t) > \frac{\lambda_1}{K_1} \\
    0 & \text{else} 
  \end{cases} \\
  u_2^*(t) &= \frac{\lambda_1}{K_1} - p_2(t) 
\end{align*}
\quad (4.68) \]
and

\[
 u_2^*(t) = \begin{cases} 
 +1 & \text{for } p_2(t) < -\frac{\lambda}{F_2} \\
 -1 & \text{for } p_2(t) > \frac{\lambda}{F_2} \\
 0 & \text{else}
\end{cases}
\] (4.69)

Equation 4.15 defines the time-derivatives of the costates

\[
 \dot{p}_1^*(t) = 0 \\
 \dot{p}_2^*(t) = -p_1(t)
\] (4.70) (4.71)

which leads to the form of the costates

\[
 p_1^*(t) = a \\
 p_2^*(t) = -at + b
\] (4.72) (4.73)

where \(a, b\) are constants. Immediately this implies that \(p_2^*(t)\) changes linearly in time. It is clear from this and from Equations 4.68 and 4.69 that the optimal control histories for both vehicles switch twice at most (similar to the single-vehicle case, e.g., shown in Figure 4.1).

Next, define four switch times: \(t_{ij}\) where \(i = 1, 2\) and \(j = 1, 2\). \(i = 1\) represents an off-switch (i.e., the start of the drift phase) and \(i = 2\) represents the on-switch that begins the terminal part of the trajectory. \(j\) is the vehicle number (e.g., \(t_{11}, t_{21}\) are the times at which vehicle one switches off and on, respectively). For convenience in the derivation, the assumption is made that vehicle two switches off before vehicle one and vehicle one switches on before vehicle two (that is, vehicle two has a longer drift period). As will be described, this assumption can be assured without loss of generality. Also, the derivation assumes that both \(u^*(t)\) histories follow the sequence \(\{+1, 0, -1\}\). It is clear that both vehicles follow the same order, because as the system approaches \(t_f\) the relative velocity must approach zero. The optimal (fastest) way to approach zero is for both vehicles to fire thrusters in the same direction (according to the relative dynamics defined by Equation 4.60)\(^3\). Once the solution is found for this thrust sequence, the optimal solution for the reverse case will be found through symmetry. Finally, it is clear from these assumptions that \(t_f > t_{22} > t_{21} > \)

\(^3\)The switching conditions in Equations 4.68 and 4.69 also validate this assumption.
\[ t_{11} > t_{12} > t_0, \] where the maneuver starts at time \( t_0 \). The four switching lines are sought. The following derivation works backwards in time, starting from \( t_f \).

**Switching Condition \( g_1 \)**

Consider the system dynamics during the time interval \( t \in [t_{22}, t_f] \)

\[
 u_1(t \in [t_{22}, t_f]) = u_2(t \in [t_{22}, t_f]) = -1 \tag{4.74}
\]

so

\[
x^*(t_f) = 0 = x^*(t_{22}) + v^*(t_{22})(t_f - t_{22}) - \frac{1}{2}(K_1 + K_2)(t_f - t_{22})^2 \tag{4.75}
\]

and

\[
v^*(t_f) = 0 = v^*(t_{22}) - (K_1 + K_2)(t_f - t_{22}) \tag{4.76}
\]

then

\[
(t_f - t_{22}) = \frac{v^*(t_{22})}{(K_1 + K_2)} \tag{4.77}
\]

substituting

\[
0 = x^*(t_{22}) + \frac{1}{(K_1 + K_2)}v^*(t_{22})^2 - \frac{1}{2}(K_1 + K_2)(\frac{v^*(t_{22})}{(K_1 + K_2)})^2 \tag{4.78}
\]

which simplifies to

\[
x^*(t_{22}) = -\frac{1}{2(K_1 + K_2)}v^*(t_{22})^2 \tag{4.79}
\]

which is the switching curve \( g_1 \) in Equations 4.45-4.58.

**Switching Condition \( g_2 \)**

During the interval \( t \in [t_{21}, t_{22}] \),

\[
u_1(t \in [t_{21}, t_{22}]) = -1 \tag{4.80}
\]

\[
u_2(t \in [t_{21}, t_{22}]) = 0 \tag{4.81}
\]

so

\[
x^*(t_{22}) = x^*(t_{21}) + v^*(t_{21})(t_{22} - t_{21}) - \frac{1}{2}K_1(t_{22} - t_{21})^2 \tag{4.82}
\]
and

\[ v^*(t_{22}) = v^*(t_{21}) - K_1(t_{22} - t_{21}) \]  (4.83)

The quantity \((t_{22} - t_{21})\) is found from the switching conditions in Equation 4.68 and 4.69 and from the known form of \(p^*_2(t)\) in 4.73

\[
\begin{align*}
-\alpha t_{22} + b &= \frac{\lambda_2}{K_2} \\
-\alpha t_{21} + b &= \frac{\lambda_1}{K_1}
\end{align*}
\]  (4.84, 4.85)

subtracting

\[
\alpha(t_{22} - t_{21}) = \frac{\lambda_1}{K_1} - \frac{\lambda_2}{K_2}
\]  (4.86)

and from the condition in Equation 4.17 at time \(t_{21}\) (when \(u_1^*(t_{21}) = u_2^*(t_{21}) = 0\))

\[
1 + \alpha v^*(t_{21}) = 0
\]  (4.87)

so

\[
a = -\frac{1}{v^*(t_{21})}
\]  (4.88)

thus

\[
(t_{22} - t_{21}) = \left(\frac{\lambda_2}{K_2} - \frac{\lambda_1}{K_1}\right)v^*(t_{21})
\]  (4.89)

The quantity \((t_{22} - t_{21})\) must be positive (otherwise the assumption on the order of switches is invalidated). To guarantee this, simply check whether \(\frac{\lambda_2}{K_2} < \frac{\lambda_1}{K_1}\) initially. If it is, reversing the vehicle numbering yields the correct result (as long as the resulting thrust histories are corrected for this change). The Matlab code in Appendix B provides an example of how to implement this logic.

Next, using the constant \(B\) defined in Equation 4.53,

\[
(t_{22} - t_{21}) = B v^*(t_{21})
\]  (4.90)

Substituting into Equations 4.82 and 4.83,

\[
\begin{align*}
x^*(t_{22}) &= x^*(t_{21}) + B v^*(t_{21})^2 - \frac{1}{2}K_1B^2 v^*(t_{21})^2 \\
v^*(t_{22}) &= v^*(t_{21}) - K_1B v^*(t_{21}) = (1 - K_1B)v^*(t_{21})
\end{align*}
\]  (4.91, 4.92)
At this point, note that negative values of the quantity \((1 - K_1 B)\) indicate that the relative velocity changes sign between \(t_{21}\) and \(t_{22}\). This would contradict what we know physically. That is, since \(u_1^*(t \in [t_{21}, t_{22}]) = -1\) and \(u_2^*(t \in [t_{21}, t_{22}]) = 0\), \(v\) decreases during \(t \in [t_{21}, t_{22}]\); \(v\) decreases during \(t \in [t_{22}, t_f]\) as well because \(u_1^*(t \in [t_f, t_{22}]) = -1\) and \(u_2^*(t \in [t_f, t_{22}]) = -1\). Therefore, the condition \((1 - K_1 B) < 0\) means that \(v^*(t_{21}) > v^*(t_f) = 0 > v^*(t_{22})\) and therefore \(t_{22} > t_f\). This simply means that the optimal trajectory reaches the final state without \(u_2^*(t)\) switching on. So if \((1 - K_1 B) < 0\) then the desired optimal switching condition is the one that drives the state to zero using only \(u_1\), which is

\[
x^*(t_{21}) = -\frac{1}{2K_1}v^*(t_{21})
\]

Practically, this condition can be achieved by applying an upper bound to \(B\)

\[
B \leq \frac{1}{K_1}
\]

(4.94)

Combining Equations 4.91 and 4.92 using the relation in 4.79 yields

\[
x^*(t_{21}) = Av^*(t_{21})^2
\]

(4.95)

where

\[
A = \frac{(1 - K_1 B)^2}{2(K_1 + K_2)} - B + \frac{B^2K_1}{2}
\]

(4.96)

This is the switching curve \(g_2\) in Equations 4.45–4.58\(^4\) (which, for \(B \neq 0\), must lie inside curve \(g_2\)).

**Switching Condition \(g_3\)**

From Equation 4.68,

\[
-at_{11} + b = -\frac{\lambda_1}{K_1}
\]

(4.97)

\[
-at_{21} + b = \frac{\lambda_1}{K_1}
\]

(4.98)

\(^4\)Note that \(A(B = \frac{1}{K_1}) = -\frac{1}{2K_1}\) as required.
subtracting
\[ (t_{21} - t_{11}) = -\frac{2\lambda_1}{aK_1} \] (4.99)
and from Equation 4.88
\[ (t_{21} - t_{11}) = \frac{2\lambda_1 v^*(t_{21})}{K_1} \] (4.100)
and because both vehicles are drifting during the interval \( t \in [t_{11}, t_{21}] \), it is clear that
\( v^*(t_{21}) = v^*(t_{11}) \), so
\[ (t_{21} - t_{11}) = -\frac{2\lambda_1 v^*(t_{11})}{K_1} \] (4.101)
Then from the system dynamics during the drift phase of the trajectory,
\[ x^*(t_{21}) = x^*(t_{11}) + v^*(t_{11})(t_{21} - t_{11}) \] (4.102)
and substituting for \((t_{21} - t_{11})\),
\[ x^*(t_{21}) = x^*(t_{11}) + \frac{2\lambda_1}{K_1} v^*(t_{11})^2 \] (4.103)
and from Equation 4.95
\[ x^*(t_{11}) = Av^*(t_{21})^2 - \frac{2\lambda_1}{K_1} v^*(t_{11})^2 \] (4.104)
again substituting \( v^*(t_{21}) = v^*(t_{11}) \)
\[ x^*(t_{11}) = (A - \frac{2\lambda_1}{K_1})v^*(t_{11})^2 \] (4.105)
which is switching curve \( g_3 \) in Equations 4.45–4.58 (which, for \( \frac{2\lambda_1}{K_1} > 0 \) must lie inside curve \( g_2 \)).

**Switching Condition** \( g_4 \)

Finally, during the time interval \( t \in [t_{12}, t_{11}] \)
\[ -at_{11} + b = -\frac{\lambda_1}{K_1} \] (4.106)
\[ -at_{12} + b = -\frac{\lambda_2}{K_2} \] (4.107)
subtracting
\[ t_{11} - t_{12} = \frac{1}{a} \left( \frac{\lambda_1}{K_1} - \frac{\lambda_2}{K_2} \right) \]  
(4.108)

substituting for \( a \) and defining the constant \( D \)
\[ t_{11} - t_{12} = Dv^*(t_{11}) \]  
(4.109)

where
\[ D = \frac{\lambda_2}{K_2} - \frac{\lambda_1}{K_1} \]  
(4.110)

The system dynamics during this interval are known
\[ x^*(t_{11}) = x^*(t_{12}) + v^*(t_{12})(t_{11} - t_{12}) + \frac{1}{2}K_1(t_{11} - t_{12})^2 \]  
(4.111)
\[ v^*(t_{11}) = v^*(t_{12}) + K_1(t_{11} - t_{12}) \]  
(4.112)

Now substituting 4.109 into 4.112
\[ v^*(t_{11}) = v^*(t_{12}) + K_1Dv^*(t_{11}) \]  
(4.113)

or
\[ v^*(t_{12}) = (1 - K_1D)v^*(t_{11}) \]  
(4.114)

Note that the velocities \( v^*(t_{12}) \) and \( v^*(t_{11}) \) may have opposite signs, so the constant \( D \), unlike \( B \), is not bounded.

Next define \( C \)
\[ C = (1 - K_1D) = 1 + \lambda_1 - \frac{\lambda_2K_1}{K_2} \]  
(4.115)

so
\[ v^*(t_{12}) = Cv^*(t_{11}) \]  
(4.116)

Now substituting 4.109 into 4.111
\[ x^*(t_{11}) = x^*(t_{12}) + v^*(t_{12})Dv^*(t_{11}) + \frac{1}{2}K_1D^2v^*(t_{11})^2 \]  
(4.117)

and from 4.116
\[ x^*(t_{11}) = x^*(t_{12}) + \frac{D}{C}v^*(t_{12})^2 + \frac{K_1D^2}{2C^2}v^*(t_{12})^2 \]  
(4.118)
then combining 4.105 and 4.116

$$x^*(t_{11}) = \frac{1}{C^2}(A - \frac{2\lambda_1}{K_1})v^*(t_{12})^2$$

(4.119)

setting these two expressions for $x^*(t_{11})$ equal

$$\frac{1}{C^2}(A - \frac{2\lambda_1}{K_1})v^*(t_{12})^2 = x^*(t_{12}) + \frac{D}{C}v^*(t_{12})^2 + \frac{K_1D^2}{2C^2}v^*(t_{12})^2$$

(4.120)

which simplifies to

$$x^*(t_{12}) = \left|\frac{1}{C^2}(A - \frac{2\lambda_1}{K_1}) - \frac{D}{C} - \frac{K_1D^2}{2C^2}\right|v^*(t_{12})^2$$

(4.121)

which is the final switching curve, $g_4$.

Switching line locations in the phase plane (position versus velocity space) are ambiguous due to the $v^2$ terms in the equations. The locations can be resolved as follows, based on knowledge of the system dynamics. This derivation assumes that both vehicles follow the thrust sequence \{+1, 0, -1\}, which is the case when the system approaches the terminal zero-error state from the negative-position, positive-velocity quadrant. The derived switching lines are valid for this case and their reflections about the origin apply to the opposite case (i.e. when zero is approached from the positive-position, negative-velocity quadrant). The first three switching lines must lie in the negative-position, positive-velocity space (because the drift-phase must have positive velocity or the position error will increases during drift). Thus, the first three switching lines follow the form $g_i(x) = -\text{sgn}(x)\sqrt{C_i|x|}$ for $i = 1 \ldots 3$ (that is, the velocity-coordinate has the opposite sign of the position-coordinate).

However, the quadrant in which the fourth switching line lies is not so restricted. Equation 4.116 reveals whether the velocity changes sign during the interval $t \in [t_{12}, t_{11}]$. If $C > 0$ the fourth switching line lies in the same quadrant as the others (negative-position, positive-velocity). However if $C < 0$, then the sign of the coefficient in Equation 4.121 determines whether the fourth switching line is in the negative-position, negative-velocity or positive-position, negative-velocity quadrant. Therefore, two logical tests are required to determine the sign of the fourth switching line. These tests are included in Equations 4.45–4.58 as $f_1, f_2$. 
Figure 4.3: Phase-plane portrait of two-vehicle cooperative controller. Shows four switching lines labelled $g_1$–$g_4$ and the optimal trajectory from $x(t_0) = 1$ m and $v(t_0) = -0.4$ m/s (the starting point is marked by a 'o' on the plot). The respective thruster settings are noted on the figure, for example at $x < 0, v = 0$, the thruster settings are $+/0$, i.e. $u_1 = +1, u_2 = 0$.

Figure 4.3 shows the optimal switching lines for a particular case ($\lambda_1 = 0.5, \lambda_2 = 3.0, k_1 = k_2 = 0.026$ m/s$^2$) plus an example trajectory (starting from $x(t_0) = 1$ m and $v(t_0) = -0.4$ m/s). The optimal actuator behavior is marked for each region of the phase plane in the form “$u_1^+/u_2^*$” and the switching lines are labelled. Figure 4.4 shows examples with the $g_4$ switching line in the three possible quadrants of the phase plane.
Figure 4.4: Phase-plane portraits of two-vehicle cooperative controller for $K_1 = K_2 = 0.013$, $\lambda_1 = 1.0$ and $\lambda_2 = 1.5, 2.5, 5$. Note how the $g_4$ switching line (solid line) changes location as $\lambda_2$ increases.

### 4.3.2 Simulation Results

Simulations for three relative-control strategies are presented here to highlight the benefit of the two-vehicle fuel-time optimal cooperative control law compared to linear control with thrust mapping, and the single-vehicle fuel-time-optimal controller. The simulation results are confirmed experimentally on two of the formation flying testbed vehicles in the following section.

Proportional plus derivative (PD) control with thrust-mapping is the first test case. The PD gains were designed using a linear system model, and then tuned through trial and error to provide good performance on the vehicles. A simple thrust mapping algorithm fires 2, 1 or 0 thrusters by comparing the desired acceleration (from the PD controller) to thresholds on the vehicle accelerations ($=1/4 K, 3/4 K$). In the second test case, each vehicle regulates the relative state independently, using single-vehicle fuel-time-optimal controllers (as described in Equations 4.2–4.4). In this case, each vehicle independently attempts to reduce the relative-state error. The cooperative controller is the third test case. Each controller’s input variables (gains or weights) were chosen to force vehicle 1 to expend 50% more fuel than vehicle 2 (e.g., because vehicle 1 had 50% more fuel at the time) for a maneuver of about 15.6 seconds. By fixing maneuver duration and fuel ratio, the total fuel use and the
Table 4.1: Two-vehicle relative-position offset response simulation (fuel numbers are normalized). Cost is found from Equation 4.44 with $\lambda_1 = 1.86, \lambda_2 = 1.99$.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Fuel$_1$</th>
<th>Fuel$_2$</th>
<th>Time (sec)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD + Thrust Mapping</td>
<td>1.00</td>
<td>0.66</td>
<td>15.63</td>
<td>40.04</td>
</tr>
<tr>
<td>2 Independent Controllers</td>
<td>0.45</td>
<td>0.31</td>
<td>15.60</td>
<td>27.07</td>
</tr>
<tr>
<td>Cooperative Controller</td>
<td>0.44</td>
<td>0.28</td>
<td>15.63</td>
<td>26.21</td>
</tr>
</tbody>
</table>

The results from the simulations are listed in Table 4.1.

This simulation of the formation flying testbed includes the GPS estimation system (as described in Section 2.3.1). GPS measurements are available at 10 Hz, and the estimator and plant run at 60 Hz. Each of the three control cases mentioned were run on two vehicles with an initial relative-position error of one meter. The fuel expended by each vehicle, the time to bring the error to zero, and the resulting cost (according to Equation 4.44) are recorded.

It is clear that the PD controller expends a large amount of extra fuel compared to the other controllers. The two vehicles running independent single-vehicle controllers perform much better than PD, but not as well as the cooperative controller. The cooperative controller reduces the cost $J$ by 3% in this simulation compared to two independent optimal controllers. The performance benefit of the cooperative controller compared to two independent single-vehicle fuel-time optimal controllers increases with the difference in $\lambda$'s (2% for $\lambda_1 = \lambda_2 = 1$ to 28% for $\lambda_1 = 0, \lambda_2 = 2$). This result confirms that the cooperative controller is the most efficient relative controller for a two-vehicle formation flying mission (such as Space Technology 3).

### 4.3.3 Experimental Results

The three cases in the previous section were tested experimentally on the FFTB. The overhead vision system (which has different noise and accuracy characteristics than the GPS system simulated) was used for this experiment. The vehicles' positions, velocities and fuel use were recorded. Figure 4.5 shows the position histories of both vehicles in the controlled axis for the three cases. Note that these relative controllers leave the absolute location of the formation uncontrolled. Thus the final states of the vehicles are different.
Figure 4.5: Position-versus-time experimental data for three different controllers of the relative spacing of a pair of vehicles. The initial relative separation is 2 m. The desired relative separation is 1 m, which is achieved in each case. Experimental data is shown for the PD controller (△), two independent fuel-time-optimal controllers (○), and the fuel-time-optimal cooperative controller (×). The latter is better: see Table 4.2.

for each test case. Figure 4.6 shows the relative-position error history for the experiment. Table 4.2 lists the performance of the controllers.

The experimental performance agrees with the simulation predictions. The primary source of differences was in the vehicle acceleration models. The controllers assumed that the acceleration-per-thruster parameters are all equal to 0.013 m/sec² as predicted in Section 2.2.1. As a result of using this optimistic model, all controllers required more fuel to accomplish the maneuver experimentally than predicted by the simulation. The normalized fuel value in Table 4.2 is ~50% higher than in Table 4.1. The two-vehicle fuel-time-optimal
controller outperforms the other two controllers: it uses far less fuel than the PD controller and less fuel for both vehicles and slightly less time than two independent fuel-time optimal controllers. The cooperative controller reduces the cost by 8% compared to two independent controllers. These results experimentally confirm the advantage of the cooperative controller as well as indicate the importance of accurate thruster models for relative control.

The system identification experiments described in Chapter 2 developed more accurate acceleration models for all three vehicles. These improved models were used for the formation flying maneuver experiment in Chapter 6.
4.4 Control for Three or More Vehicles

The number of relative position variables along each direction for a formation grows according to $\frac{1}{2}N(N - 1)$ where $N$ is the number of vehicles. Thus, as with the estimation problem, the complexity of the relative-control problem grows rapidly with the number of vehicles in the formation. This section presents options for implementing relative control on a three-vehicle formation.

The optimal trajectory according to

$$J = \int_{t_0}^{t_f} \left[ 1 + \lambda_1|u_1(t)| + \lambda_2|u_2(t)| + \lambda_3|u_3(t)| \right] dt$$  \hspace{1cm} (4.122)$$

from an initial relative-state error can be found using a numerical optimization technique called inverse dynamic optimization as described in Section 4.4.1. However, the high computational cost of this technique makes it unsuitable for real-time control of a spacecraft formation. A new, simple heuristic controller presented in Section 4.4.2 is easy to implement, however it produces sub-optimal trajectories. An analytic expression for the three-vehicle optimal relative-control law has not been found, but the form of the optimal control is derived in Section 4.4.3. This knowledge enables a simpler numerical algorithm to find the optimal trajectory. This new algorithm, presented for the first time here, is fast enough to be suitable for real-time control. An alternative to these centralized solutions is to implement a distributed control architecture. Section 4.5 discusses the control architecture design problem in more detail.

Table 4.2: Two-vehicle relative-position offset response experiment (normalized fuel). Cost is found from Equation 4.44 with $\lambda_1 = 1.86$, $\lambda_2 = 1.99$.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Fuel$_1$</th>
<th>Fuel$_2$</th>
<th>Time (sec)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD + Thrust Mapping</td>
<td>1.00</td>
<td>0.79</td>
<td>15.67</td>
<td>56.78</td>
</tr>
<tr>
<td>2 Independent Controllers</td>
<td>0.62</td>
<td>0.48</td>
<td>16.63</td>
<td>41.90</td>
</tr>
<tr>
<td>Cooperative Controller</td>
<td>0.53</td>
<td>0.44</td>
<td>16.20</td>
<td>38.43</td>
</tr>
</tbody>
</table>
4.4.1 Inverse Dynamic Optimization

The inverse dynamic optimization technique determines points on the optimal state histories through numerical optimization. The control histories that produce the optimal path are then found through numerical differentiation of the state-variable histories. This technique is described in Reference [93], pages 362–363.

The inverse dynamic optimization solution is discrete in time: discrete points in the optimal state and actuator histories are found. The separation in time between these points depends both on the number of points and on the length of the trajectory. Increasing the resolution in time requires computing more points on the trajectory, which greatly increases the amount of computation required. Furthermore, a good initial guess is required for convergence. If no information exists a priori about the optimal trajectory, several guesses may need to be checked before the optimal solution is found.

Figure 4.7 shows the optimal actuator histories for an example three-vehicle relative-control problem computed using inverse dynamic optimization with $\lambda_1 = 0.5$, $\lambda_2 = 1.0$, $\lambda_3 = 2.0$. Note that some actuator values lie between 0 (off) and 2 (on); this numerical effect occurs near switches due to the coarseness of the solution.

The relative-separation errors are reduced to zero from initial offsets of -0.8 m between vehicles one and two and -1.2 m between vehicles two and three. Figure 4.8 plots the corresponding relative-state trajectories. The duration of the optimal trajectory is 18.4 seconds. The time-resolution of the solution is 0.8 seconds/point. This solution requires $\sim 35$ GFlops and $\sim 2,110$ seconds to compute in Matlab on a 333 MHz UltraSPARC workstation. This high computational cost for even relatively coarse solutions eliminates the possibility of implementing real-time control based on the inverse dynamic optimization technique. The following two sections develop control alternatives for centralized, real-time control of spacecraft formations.

4.4.2 Heuristic (Sub-Optimal) Control Method

This section presents a new controller based on the idea of a relative-state-error "center of mass". This heuristic control law is a simpler alternative to computing the optimal solution using the inverse dynamic optimization technique. Each vehicle acts according to

$$u_i = f \left( \sum_{j=1, j \neq i}^{N} \frac{x_{ij}}{N-1} \right) \quad (4.123)$$
Figure 4.7: Optimal actuator histories for a three-vehicle relative-position offset response example, computed via the inverse dynamic optimization technique. The resulting motion is shown in Figure 4.8.

where $f(x)$ is the single-vehicle fuel-time optimal controller presented in Section 4.2 and $x_{ij}$ is the relative-state error between vehicles $i$ and $j$. This control method finds the average of the relative-state errors between one vehicle and the rest of the vehicles in the formation. Each vehicle acts to correct this error, which results in efficient control without a significant computational requirement. The trade-off is that each vehicle still needs full knowledge of the relative formation states, and the resulting control is sub-optimal to some degree.

This controller is physically appealing because each vehicle uses the efficient bang-off-bang control law to reduce its position error relative to the rest of the formation. λ’s allow fuel-use penalties per vehicle and formation-wide to be tuned as desired. The resulting
Figure 4.8: Relative-state histories (components in one inertial direction) for a three-vehicle relative-position offset response example, computed via the inverse dynamic optimization technique. Relative-position and relative-velocity errors are plotted for vehicles 1 and 2 (x) and 2 and 3 (o).

behavior is stable, computationally cheap, and efficient though not optimal according to Equation 4.122. The heuristic control law also extends easily to formations of any size.

4.4.3 Numerically Optimized Control

This section derives a new insight into the optimal relative-control problem, which enables a simplified, numeric solution technique. This insight reduces the computational cost of finding optimal control histories to such an extent that it can be used for real-time control.
Following the form of the two-vehicle optimal control derivation in Section 4.3.1, the three-vehicle system is described by

\[
\begin{align*}
\dot{x}_{12}(t) & = v_{12}(t) \\
\dot{v}_{12}(t) & = K_1u_1(t) - K_2u_2(t) \\
\dot{x}_{23}(t) & = v_{23}(t) \\
\dot{v}_{23}(t) & = K_2u_2(t) - K_3u_3(t)
\end{align*}
\]

where \(x_{ij}\) is the relative position and \(v_{ij}\) is the relative velocity defined by

\[
\begin{align*}
x_{ij}(t) & = x_i(t) - x_j(t) \\
v_{ij}(t) & = v_i(t) - v_j(t)
\end{align*}
\]

and \(K_i\) is the two-thruster acceleration for vehicle \(i\) (the \(K\)'s are redefined here so that they have the same direction sense in absolute terms for all three vehicles). The three-vehicle cost function is given by Equation 4.122. From Equation 4.10 the Hamiltonian is

\[
H(x(t), u(t), p(t)) = 1 + \lambda_1|u_1(t)| + \lambda_2|u_2(t)| + \lambda_3|u_3(t)| + p_1(t)v_{12}(t) + p_2(t)(K_1u_1(t) - K_2u_2(t)) + p_3(t)v_{23} + p_4(t)(K_2u_2(t) - K_3u_3(t))
\]

Then the optimal actuator histories \(u_{1,2,3}^*(t)\) obey

\[
u_1^*(t) = \begin{cases} 
+1 & \text{for } p_2(t) < -\frac{\lambda_1}{K_1} \\
-1 & \text{for } p_2(t) > \frac{\lambda_1}{K_1} \\
0 & \text{else}
\end{cases} 
\]

\[
u_2^*(t) = \begin{cases} 
+1 & \text{for } (p_4(t) - p_2(t)) < -\frac{\lambda_2}{K_2} \\
-1 & \text{for } (p_4(t) - p_2(t)) > \frac{\lambda_2}{K_2} \\
0 & \text{else}
\end{cases} 
\]

\[
u_3^*(t) = \begin{cases} 
+1 & \text{for } p_4(t) > \frac{\lambda_3}{K_3} \\
-1 & \text{for } p_4(t) < -\frac{\lambda_3}{K_3} \\
0 & \text{else}
\end{cases} 
\]
The time-derivatives of the costates are given by

\[
\begin{align*}
\dot{p}_1(t) &= 0 \\
\dot{p}_2(t) &= -p_1(t) \\
\dot{p}_3(t) &= 0 \\
\dot{p}_4(t) &= -p_3(t)
\end{align*}
\]

(4.134) (4.135) (4.136) (4.137)

which leads to the (now familiar) form

\[
\begin{align*}
p_1^*(t) &= a \\
p_2^*(t) &= -at + b \\
p_3^*(t) &= c \\
p_4^*(t) &= -ct + d
\end{align*}
\]

(4.138) (4.139) (4.140) (4.141)

from which it is clear that \( p_2^*(t), p_4^*(t) \) and \( (p_3^*(t) - p_3^*(t)) \) change linearly in time.

Thus the three actuator histories switch according to three linearly-changing parameters. Therefore, all three vehicles follow bang-off-bang trajectories. Based on this fact, a numeric optimization problem that solves for the switch-times can be used to find the optimal trajectory. This new technique, called switch-time optimization solves a small problem (only \( 2N + 1 \) parameters) yet returns the optimal trajectory exactly. Furthermore this derivation extends to spacecraft formations of any size, so this control algorithm is generally applicable to spacecraft formations of any size that require optimal relative control.

The thrust directions are an additional variable (i.e., each vehicle follows one of two thrust sequences: \( \{+1, 0, -1\} \) or \( \{-1, 0, +1\} \)), thus if the thrust sequences are not known a priori, all possible combinations must be tried. There are \( 2^N \) combinations to try, however each one is a small \( (2N + 1) \) problem. Furthermore, these problems can be solved in parallel, making it easy to distribute the problem (e.g., with each spacecraft assuming some of the computational burden). Thus this control technique is applicable even as \( N \) grows to 10, 20 or more vehicles.

The switch-time optimization technique was applied to the example presented in Section 4.4.1. The resulting thrust histories are plotted in Figure 4.9 and the corresponding relative-state histories are plotted in Figure 4.10. These results can be compared directly to Figures 4.7 and 4.8. The switch-time optimization technique fully resolves the optimal
trajectories. The solution requires 6.6 MFlops and $\sim 11.7$ seconds to compute in Matlab on a 333 MHz UltraSPARC workstation (compared with 35 GFlops and 2,110 seconds for inverse dynamic optimization). The switch-time optimizer reduces the computational cost of finding the optimal trajectory by four orders of magnitude compared to inverse dynamic optimization. Both examples used Matlab’s CONSTR function. Implementing the switch-time optimizer with the more efficient E04UCF optimization function from the Numerical Algorithms Group (part of Matlab’s NAG toolbox) further reduces the computation cost for this problem to $\sim 60,000$ Flops. This low computational cost level makes this optimization technique a candidate for real-time spacecraft formation flying control.
The following section takes advantage of this efficient solution technique to implement the switch-time optimization technique as part of a real-time receding-horizon controller.

**Switch-Time Optimization for Feedback Control**

The switch-time optimizer is fast enough that it can be implemented as a relative-state receding-horizon controller as follows. The optimizer outputs the optimal actuator plan $u^*(t \in [t_0, t_f])$ and the expected state history. When the measured state deviates from the planned trajectory, a new plan is generated (this setup is illustrated in Figure 4.11). The maximum relative-state deviation can be chosen based on relevant mission parameters, such
Figure 4.11: Switch-time optimizer configured for feedback control.

as the range of an optical inner-loop tracking system. The new plans generally follow the same thruster sequence, so they can be computed more quickly than the initial plan (which has to search all possible thrust-sequence combinations).

The switch-time optimizer receding-horizon controller was implemented in simulation. The absolute position and velocity histories for a 20-second correction maneuver are shown in Figure 4.12 for the three-vehicle example from Section 4.4.1. The initial vehicle separations are 1.8 m and 2.2 m. The desired separations are 1 m each. This example simulates the maneuver on the FFTB including noise, so replans are periodically required. Replanning events (marked on the plot) were triggered when any relative position deviated by more than 2 cm from the planned value. The initial plan takes \( \sim 1.2 \) seconds to compute and the replans (which assume the thrust sequences do not change) require an average of 0.15 seconds on a 333 MHz UltraSPARC computer in Matlab. The desired relative positions are achieved at 18 seconds. This controller is compared in simulation to two other relative-control techniques in the following section.

If the real-time replanning implementation is infeasible due to limited computational resources, the switch-time optimizer can still be utilized for trajectory planning. The output is the optimal state trajectory, which can be tracked by a conventional controller. The actuator history can be used in a feedforward mode. Either application of the optimizer benefits from the ability to quickly find the fuel-time optimal solution for regulating the relative states in a spacecraft formation of any size.
4.4.4 Simulation Results

The heuristic and numerically optimized relative controllers were compared in simulation to each other and to a distributed-control approach. (Experimental results comparing heuristic centralized control and distributed control on the FFTB are given in Chapter 6.) The simulations include a three-vehicle formation with initial relative-position errors. The three controllers tested are: the switch-time optimizer receding-horizon controller described in Section 4.4.3, the heuristic centralized controller described in Section 4.4.2, and a distributed controller. The distributed controller divides the relative-control problem into sub-problems.
that are addressed independently at the vehicle-level using a “follow-the-leader” strategy: each vehicle follows the next one (i.e., vehicle 1 follows vehicle 2, vehicle 2 follows vehicle 3, and vehicle 3 follows vehicle 1) using the single-vehicle bang-off-bang control law described in Section 4.2. The advantage of distributed control is ease of implementation: no significant computation or communication resources are required. In fact, a distributed controller could be combined with a distributed estimation architecture as described in Chapter 3 to form a minimum-resource control system that can be expanded to spacecraft formations of any size. The choice between centralized and distributed control constitutes the control architecture design problem, which is explored in Section 4.5.

The simulation includes centralized GPS-based relative estimation. The vehicles are started at separations of 1.8 m and 2.2 m (between vehicles one and two and two and three, respectively). The desired separations are 1 m. Figure 4.13 shows the absolute-position history for each controller. Note that these relative controllers leave the absolute location of the formation uncontrolled. Thus the final states of the vehicles are different for each test case. The cost of each controller’s correction maneuver according to Equation 4.122 with \( \lambda_1 = 0.5, \lambda_2 = 1.0, \lambda_3 = 2.0 \) is listed in Table 4.3. The heuristic controller has an advantage over the distributed controller because it has knowledge of all formation states. However the heuristic controller still costs 12% more than the switch-time optimizer. These results indicate that, as expected, there is a trade-off between performance, computational cost, and communication resources. If maximum performance is required, the switch-time optimizer appears to be the best choice. However, the heuristic controller provides relatively good performance without the computational cost associated with the switch-time optimizer.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Time</th>
<th>Veh 1 Fuel</th>
<th>Veh 2 Fuel</th>
<th>Veh 3 Fuel</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed</td>
<td>15.55</td>
<td>7.1</td>
<td>6.5</td>
<td>7.5</td>
<td>40.5</td>
</tr>
<tr>
<td>Heuristic</td>
<td>15.23</td>
<td>7.9</td>
<td>3.8</td>
<td>4.5</td>
<td>32.0</td>
</tr>
<tr>
<td>Switch-Time Optimizer</td>
<td>17.85</td>
<td>7.8</td>
<td>2.9</td>
<td>1.9</td>
<td>28.5</td>
</tr>
</tbody>
</table>
Figure 4.13: Simulated absolute position versus time for three vehicles under three relative-control laws: distributed (*), heuristic (o) and switch-time optimizer (△).

4.5 Control Architectures

A new design decision arises for spacecraft formation flying because multiple spacecraft must cooperate to achieve fleet-wide objectives such as formation keeping. The choice of control architecture determines how the formation keeping control task is organized among the spacecraft in a formation. Control-architecture designs meet relative-position control requirements but differ in performance, complexity and implementation cost. This section investigates the architecture design problem by comparing three example control architectures in simulation for a six-vehicle SSI mission. The results quantify the trade-off between performance and implementation cost, providing engineering insight applicable to upcoming formation flying missions.
4.5.1 Candidate Formation Flying Control Architectures

To illustrate the trade-offs involved in the control architecture selection process, three candidate architectures were compared: centralized, distributed and hierarchical.

A centralized control structure generates thrust commands based on knowledge of all relative states in the formation. A centralized architecture yields the best fuel/time performance, but requires a high level of inter-vehicle communication (or connectivity). Thruster commands can be computed using either the heuristic control law described in Section 4.4.2 or the receding-horizon control using the switch-time optimizer algorithm described in Section 4.4.3. The switch-time optimizer provides superior performance, but increases the computational cost of control. The heuristic controller was used for the centralized control architecture simulation in Section 4.5.2.

Distributed control architectures divide the formationkeeping task into local relative-control problems: each spacecraft generates thrust commands based on a small subset of the full relative-state information. Several distributed architectures exist including leader/follower and hub/spoke control structures (see, for example, [27, 94] for distributed control architecture examples). Distributed architectures avoid system complexity and require less connectivity, but the resulting fuel-time performance tends to be worse than centralized control.

In order to find which distributed control architecture was the best for the six-vehicle formationkeeping application, several simulations were run. These simulations included steady-state formationkeeping and recovery from initial relative-position errors, with fuel weights randomly distributed over several ranges. The results indicated that the most fuel-efficient distributed control architecture for this application is the hub/spoke architecture. In this case, one spacecraft (the hub) does not actively control its position and the remaining spacecraft regulate their states relative to the hub.

This research introduces a new control architecture that accomplishes formationkeeping using a hierarchical organization\(^5\). Three groups of two vehicles each regulate the relative-state within the group using the optimal two-vehicle cooperative control law described in Section 4.3. The three groups also control the inter-group relative-positions using the heuristic controller. Control commands from the two levels (intra-group and inter-group)

---

\(^5\)Note that the hierarchical control architecture presented here as one option for relative formationkeeping is distinct from the multi-level virtual spacecraft bus control strategy discussed in Chapter 1.
Figure 4.14: Three six-vehicle control architectures. Each vehicle with a "*" regulates its state relative to other spacecraft as indicated by the arrows.

are combined using the simple logic

\[
u = \begin{cases} 
+1 & (u_{\text{intra}}, u_{\text{inter}}) \in \{(+1, +1), (+1, 0), (0, +1)\} \\
0 & (u_{\text{intra}}, u_{\text{inter}}) \in \{(0, 0), (+1, -1), (-1, +1)\} \\
-1 & (u_{\text{intra}}, u_{\text{inter}}) \in \{(-1, -1), (-1, 0), (0, -1)\}
\end{cases}
\]

The hierarchical architecture is designed to provide an intermediate solution between fully centralized and fully distributed architectures.

The three control architectures are illustrated in Figure 4.14. Note the increasing connectivity (indicated by the increasing number and directions of arrows) from distributed to centralized. Increased connectivity probably translates into increased implementation cost. The fuel-time performance of the three control architectures are compared in simulation in the following section.

4.5.2 Control Architecture Simulation Results

The control architectures are compared for a simulated six-spacecraft formation performing an aperture-filling observing slew including 15 observing configurations with baselines between 20 m and 1 km. Each observing configuration requires 60 seconds of quiet (thrust-free) observing time with the relative states aligned within 2 cm. The observing slew was designed to simulate typical aperture filling maneuvers that will be required for SSI missions such as the Terrestrial Planet Finder.

Initial results showed a range of fuel performance differences for the distributed architecture compared to the centralized architecture. The trend is that both architectures are capable of accomplishing the observing goals, and the centralized control architecture
Figure 4.15: Simulation-based fuel/time trade curves for three control architectures for six vehicles performing 15-point stop-and-go observing slews: distributed (○), hierarchy (*), and centralized(●)
generally outperforms the distributed architecture in a fuel/time sense. The numbers vary depending on the variance of the fuel weights and on which vehicle is the hub vehicle for the distributed case. However, the cases where the distributed performance is close to the centralized performance lead to large fuel imbalances between the spacecraft.

Therefore, a fuel-balancing coordinator (described in Chapter 5) was added and the simulations were repeated. The coordinator minimizes the difference in fuel use among the vehicles by periodically adjusting the weights (λ's) on each vehicle's fuel use. The distributed architecture designates the vehicle with the highest λ as the hub vehicle (since this one does not move).
Three cases are tested for each architecture with different weights on total fuel use versus time. That is, the sum of the \( \lambda \)'s, which remains constant throughout the observing slew, is adjusted to vary the importance of total fuel use relative to the slew duration. Figure 4.15 plots the total fuel use and time to complete the observing slew for all cases for all three architectures. Fuel-time performance improves moving toward the lower-left of the plot.

All architectures successfully completed the observing slew. The centralized controller is the most efficient and the distributed controller is the least efficient for all fuel weights. In the case where time-performance is more important, the centralized architecture completes the slew in 88% of the time, using 77% of the fuel required by the distributed architecture. For the case where fuel efficiency is more important, the centralized architecture completes the slew in 96% of the time, using 82% of the fuel required by the distributed architecture. The centralized architecture is more efficient because it enables the appropriate vehicle or vehicles to act to correct formation errors, resulting in more efficient corrections compared to the other architectures.

The centralized architecture offers the best fuel-time performance. However, it requires knowledge of all relative states at 60 Hz (as implemented in the simulation), which may be a computational and/or connectivity challenge for formation flying missions. Fuel-time efficiency could be further improved by replacing the heuristic controller with the switch-time optimizer, at the cost of significantly increasing the computational requirement (this case was not simulated).

The hierarchical architecture provides a solution that is more efficient than the fully distributed control and reduces connectivity and computation cost compared to the fully centralized case. The distributed control architecture has the worst fuel-time performance, but is the easiest to implement and is straightforward to expand to larger formations (e.g., as spacecraft are added). Also, it is the most robust control architecture. The distributed control architecture responds gracefully to vehicle failures, leaving failed vehicles behind. Because there are controllers on each vehicle, a single-point failure disables one vehicle at most. However, because fuel use directly drives mission duration, the fuel-time efficiency offered by centralized control architectures probably outweighs the additional implementation costs and reduced robustness for near-term formation flying missions.

The fuel-time performance has been quantified for an example six-vehicle aperture filling maneuver. This simulation demonstrates that the control architecture design has a
significant impact on the fuel-time performance and thus on the overall capability of spacecraft formation flying missions. Experimental results (discussed in Chapter 6) agree with the performance trend identified here in simulation. A three-vehicle formation maneuver experiment on the FFTB demonstrates a 12% fuel performance decrease for the hub/spoke distributed architecture compared to the centralized control architecture (using the heuristic control law).

4.6 Conclusions

This chapter derived two new formation flying controllers: a cooperative control law that produces the fuel-time-optimal trajectory for two vehicles regulating their relative state, and a new switch-time optimizer that produces the fuel-time optimal trajectory for any number of vehicles regulating their relative states. The switch-time optimizer is fast enough to be implemented as a receding-horizon controller for real-time formationkeeping. The two-vehicle controller was demonstrated in simulation and experiment. The switch-time optimizer was demonstrated in simulation. Because fuel is a finite resource for space missions, both of these control techniques enable extended operational life for any formation flying mission compared to conventional control techniques.

A six-vehicle example was used to quantify the control-architecture design space in simulation. A centralized control architecture was shown to significantly outperform a distributed control architecture for a typical aperture-filling maneuver. The large difference in fuel-time performance found here seems to indicate that centralized control is well worth the added implementation cost and reduced robustness for near-term spacecraft formation flying missions. This chapter also introduced a new hierarchical control architecture which offers an intermediate choice that performs better than a fully-distributed architecture, but requires less connectivity than a fully-centralized architecture.
Chapter 5

Coordination

Formation flying’s group-level operational emphasis is new to spacecraft-controls engineering. Relative estimation and control tasks span multiple vehicles in the formation, and new tasks distributed across the fleet require coordination between vehicles. Thus, new tools are required to achieve efficient formation-level performance. Chapters 3 and 4 derived relative estimation and cooperative control techniques which contribute to efficient formation-level performance. This chapter introduces a new, higher-level formation coordinator which enables inter-spacecraft cooperation. The coordinator is an autonomous agent that translates fleet-wide goals into individual-spacecraft commands. In particular, it works through compact information variables such as fuel states and controller fuel weights. Using these information variables, the coordinator monitors the formation and tunes fleet-wide performance at a low bandwidth to realize fleet-wide operational goals. Tasks suitable for a coordinator include: fuel balancing, maneuver planning, trajectory monitoring and replanning, and fault detection and correction.

This chapter presents an autonomous formation coordinator and demonstrates a fuel-balancing application as an example. Section 5.1 defines the coordinator concept, which is based on ideas from decentralized control. Section 5.2 applies the coordinator to the fleet-level fuel-balancing task. Section 5.3 derives a new algorithm for fuel balancing that works through the fuel weights of the control laws derived in Chapter 4. Section 5.4 demonstrates the fuel-balancing coordinator in simulation.
CHAPTER 5. COORDINATION

5.1 Formation Coordinator

This research introduces a coordinator based on ideas from the field of decentralized control. In particular, the idea of a hierarchical control structure consisting of many low-level controllers operating according to constraints imposed by a higher-level coordinator via information variables is highly applicable to the spacecraft formation flying problem. Figure 5.1 illustrates this control structure; the coordinator optimizes a group-level cost function (e.g., minimizing fuel imbalances) by periodically adjusting the information variables according to some coordination algorithm.

Three examples of decentralized control references that describe hierarchical control structures and information variables are [95, 96, 97]. Reference [95] discusses multi-agent systems where communication and computation may both be limited. It introduces “a higher level coordinator ... The duty of the coordinator is to transmit coordinating parameters to the individual decision agents such that the system is coordinated in some sense.” Reference [96] presents the idea of an iterative two-level hierarchy for solving a problem composed of multiple interconnected dynamic subsystems. Reference [97] describes a similar method of decomposing a complex problem into subsystems that are solved separately. A second-level coordinator iteratively adjusts the subsystem problems using “pseudo-variables” until the optimal solution is found. These references describe problems...
that parallel the spacecraft formation flying problem. The hierarchy structure provides an efficient two-step solution procedure that makes fleet-wide tasks manageable.

A coordinator-based hierarchical control structure was introduced for spacecraft formation flying in Reference [98]. It provides an efficient solution technique for complex fleet-level tasks such as fuel balancing, with minimal resource (communication and computation) requirements. Reference [53] includes a similar, general concept for a spacecraft formation flying “Supervisor”; however, applications of this supervisor are not presented, and its effect on formation performance is not explored. The following section applies the coordinator concept to the fuel-balancing task.

5.2 Fuel Balancing

A separated-spacecraft interferometer’s capability is greatly reduced when one spacecraft runs out of fuel. Actively balancing fuel use among the formation vehicles maximizes the duration of the mission by ensuring that no spacecraft exhausts its fuel supply prematurely (i.e., while other spacecraft have significant amounts of fuel remaining). Fuel balancing is a good example of a critical fleet-wide performance objective that requires knowledge and action at the global level, but does not necessarily require a high-bandwidth implementation. Though the control laws derived in Chapter 4 produce fuel-efficient behavior, fuel balancing is not inherent to these controllers. An additional component — the coordinator — is required to manage fuel actively across the fleet.

In the hierarchical control solution of the spacecraft-formation fuel-balancing problem, the low-level control is accomplished by one of the control architectures described in Chapter 4. For example, this may be several leader-follower controllers or one centralized relative controller. In either case, the low-level controller implements fuel-time-efficient control. As discussed in Sections 4.3 and 4.4, the behavior of this low-level control is governed by the fuel weights ($\lambda$'s) which set control penalties for each spacecraft’s fuel use. Chapter 4 leaves the choice of $\lambda$’s free, thus they can be used as information variables by the coordinator. The low-level control continues to accomplish its relative-control task, while the coordinator achieves fleet-level fuel balancing by periodically adjusting the fuel weights according to fuel-usage data from each vehicle (another information variable).

Reference [54] studies the parallel problem of fuel-balancing for spacecraft clusters in low Earth orbit. That work applies to absolute formation flying control and relies on the
ability to compute fuel costs in advance for given maneuvers. The coordinator presented here requires no knowledge of the formation’s future actions. However, it does require full observability to the spacecraft fuel states \((f)\), access to the control law through through the \(\lambda\)’s, and knowledge of the global objective (in this case, minimizing the maximum fuel differential). Figure 5.2 illustrates the fuel-balancing coordinator. This coordinator is successfully implemented at a low operating bandwidth with only two information variables (one \(\lambda\) and one \(f\)) per spacecraft.

Chapter 4 provides the controller (or controllers for distributed control architectures) that occupies the lower level in the hierarchy illustrated in Figure 5.1. The following section presents a fuel-balancing algorithm for the coordinator level of the hierarchical control structure.

### 5.3 Fuel-Balancing Coordinator Algorithm

This section presents a new, relatively simple coordinator algorithm for fleet-level fuel balancing. This novel algorithm was developed through trial and error in FFTB simulations. The goal of the algorithm is to achieve stable, fuel-balanced formation behavior without unreasonably degrading the total-fuel and total-time performance. The objective of the
coordinator is to minimize the maximum rate of fuel use in the formation, thereby extending total science-mission life (with all spacecraft functional). Section 5.3.1 presents the coordinator algorithm and Section 5.3.2 characterizes its performance in simulation.

5.3.1 Algorithm Description

Each spacecraft tracks fuel use (e.g., by incrementing a counter every time a thruster fires). Fuel-used counts \( (f_i)'s \) are passed to the coordinator periodically. The coordinator then computes a new set of fuel weights \( (\lambda_i)'s \) based on the fuel counts from each spacecraft. The relative controller is then updated with the new \( \lambda_i)'s. The algorithm for assigning \( \lambda_i)'s based on the \( f_i)'s is

\[
\lambda_i = \Lambda \left[ L_1 \left( \frac{e_i}{E} \right) + \frac{(1 - L_1)}{N} \right] \tag{5.1}
\]

where

\[
E = \sum_{i=1}^{N} e_i \tag{5.2}
\]

and

\[
e_i = f_i - \min_i(f_i) \tag{5.3}
\]

where \( f_i \) is spacecraft \( i \)'s fuel-used count. \( e_i \) is the "extra" fuel used by spacecraft \( i \), compared to the minimum fuel used by any spacecraft in the formation. \( \Lambda \) is the total fuel-weight to be divided between the spacecraft \( \lambda_i)'s. Larger values of \( \Lambda \) cause the formation as a whole to conserve fuel, while smaller values result in faster control. \( L_1 \) takes a value between 0 and 1; it divides \( \Lambda \) into two parts. The first part \( (\Lambda L_1) \) is divided among the spacecraft in proportion to how much more fuel they have used compared to the minimum. Thus the spacecraft that has used the least fuel has no \( \lambda \)-contribution from this term. The second part \( (\Lambda(1 - L_1)) \) is divided equally among the spacecraft. Note that the following, by definition, is true

\[
\Lambda = \sum_{i=1}^{N} \lambda_i \tag{5.4}
\]

which ensures that the specified fuel weight is allocated across the fleet.

This algorithm is appealing because it assigns fuel penalties in proportion to fuel imbalances. This intuitive approach works with the control laws described in Chapter 4 to increase the control burden on vehicles with more fuel remaining, thereby equalizing fuel use over time. The \( L_1 \) parameter allows for a lower bound on fuel weight values to prevent
excessive swings in fuel use during each coordinator cycle. The total fuel weight ($\Lambda$) is conserved as an input to the formation control so that fuel-versus-time performance can still be tuned by ground controllers. Alternately, selection of $\Lambda$ could be assigned by another coordinator according to some other performance metric (e.g., to regulate the rate of data collection).

### 5.3.2 Algorithm Performance

Three examples presented in this section illustrate the behavior of the fuel-balancing coordinator and the effects of three input parameters: $\Lambda$, $L_1$, and update frequency.

#### Parameter $\Lambda$

The $\Lambda$ input to the fuel-balancing coordinator controls the trade-off between fuel use and speed of control response. Figure 5.3 illustrates how the total fuel weight parameter $\Lambda$ controls the balance between fuel\(^1\) and time. The example shows the results of the centralized control architecture combined with a fuel-balancing coordinator performing the 15-point observing sequence simulated in Section 4.5. Fuel weights are recomputed once per minute and $L_1 = 0.5$. Three values of $\Lambda$ are shown on the plot. As $\Lambda$ increases, fuel is increasingly conserved at the expense of longer durations. Longer maneuvers between observing configurations conserve fuel, but prolong the duration of the observing sequence. Thus $\Lambda$ is a useful and intuitive control “knob” for trading off fuel performance with time performance.

#### Parameter $L_1$

The $L_1$ input controls what portion of $\Lambda$ is used for the fuel balancing task. A zero value indicates equal fuel penalties for all spacecraft regardless of fuel imbalances. The maximum value of one assigns the fuel penalties completely in proportion to fuel imbalances. In this case, the vehicle with the most fuel remaining (least fuel used) receives a fuel penalty of zero which results in bang-bang control for that vehicle.

Figure 5.4 shows fuel histories for a simulated three-vehicle formation for three values of $L_1$. Over a 20-minute observing maneuver, the formation is commanded every 20 seconds to move to a new relative configuration. The heuristic control law (described in Section 4.4.2)

\(^1\)Fuel use is measured in terms of total thruster on-time.
regulates the relative positions according to the $\lambda$’s assigned by the coordinator. For this example, the coordinator operates once per minute with $\Lambda = 3$. Each spacecraft begins with $\lambda = 1$. An initial fuel imbalance is included in the simulation to emphasize the role of the coordinator: spacecraft one begins having already used 10 seconds more fuel than spacecraft two and three.

Three $L_1$ values were simulated: 0, 0.5, and 1. The fuel use in each case is summarized in Table 5.1. For the case of $L_1 = 0$, no action is taken to correct the initial fuel imbalance; equal fuel penalties ($= \Lambda/3$) are assigned to each vehicle. Note that the maneuver sequence puts more burden on spacecraft one. The top plot in Figure 5.4 shows that the imbalance
increases during the simulated maneuver; spacecraft one finishes having used 25 seconds more fuel than spacecraft two. At the end of the simulation, the percentages of total fuel used by the three spacecraft are: 38%, 31% and 31%.

For the case $L_1 = 1$, the imbalance is quickly eliminated. However, because the fuel weights are assigned completely in proportion to fuel imbalances, large changes in fuel weights occur at every coordinator cycle, even for small changes in fuel imbalances. At the conclusion of the simulated maneuver, the percentages of total fuel used by the three spacecraft are: 34%, 33% and 33%, but fuel use increases significantly (see Table 5.1) for
Table 5.1: Fuel used by three simulated formation flying spacecraft with a fuel-balancing coordinator for three $L_1$ values. Maneuver lasts 20 minutes in each case.

<table>
<thead>
<tr>
<th>$L_1$</th>
<th>Total Fuel</th>
<th>Fuel Used per Vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>seconds</td>
<td>vehicle #</td>
</tr>
<tr>
<td>0</td>
<td>333</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>0.5</td>
<td>356</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>459</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

all vehicles compared to the $L_1 = 0$ case. Thus extreme values of $L_1$ balance fuel very inefficiently.

The intermediate value $L_1 = 0.5$ accomplishes the fuel balancing goal more efficiently albeit more slowly compared to $L_1 = 1$. At the conclusion of the $L_1 = 0.5$ simulated maneuver the percentages of total fuel used by the three spacecraft are all 33%. The coordinator activity results in just 7% higher total fuel use than the case with $L_1 = 0$. But of course the critical performance metric is not total fuel use, but how long the formation is fully operational. In the $L_1 = 0$ case (i.e. equal fuel weights), spacecraft one uses 8.65 seconds of fuel more than the maximum fuel used by any spacecraft in the $L_1 = 0.5$ case. Thus the fuel-balancing coordinator achieves a 7% reduction in maximum fuel used by any spacecraft. In other words, without fuel balancing, one spacecraft would exhaust its fuel supply well before any of the spacecraft in the fuel-balancing coordinator case.

The time histories of the three $\lambda$'s are plotted in Figure 5.5 for the $L_1 = 0.5$ case. The lower limit (0.5 for each vehicle) is clear in the figure as is the fact that spacecraft one starts with a fuel deficiency so that its weight ends up higher, which places the burden of control on the other two spacecraft.
Another parameter inherent in the fuel-balancing-coordinator design is the frequency at which the coordinator updates the fuel weights. This section presents simulations of fuel-balancing coordinators operating at three update rates. The results of these simulations are shown in Figure 5.6.

The simulation parameters are the same as the previous section with $L_1 = 0.5$. The plots clearly show that fuel balancing is accomplished in all cases. Higher update rates require increased inter-vehicle communication bandwidth. However, a coordinator operating at a lower rate is slower to respond to new information. The 1/60 Hz update rate was implemented in the six-vehicle SSI observing maneuver simulation discussed in Section 5.4. Slower update rates (even updating after each observing sequence) will work for space applications as well, although the reduction in implementation cost may be small compared to a 1/60 Hz coordinator. Mission-specific factors will determine the choice of update rate. A 1/60 Hz coordinator was sufficient to demonstrate fuel balancing for the cases considered in this research.

5.3.3 Fuel-Balancing Coordinator Algorithm Summary

This section has demonstrated a new fuel-balancing algorithm that works in conjunction with a relative controller from Chapter 4 according to the hierarchical structure defined in
Figure 5.6: Fuel-use history for a three-spacecraft aperture-filling maneuver simulation with coordinators running at three different update rates and $\Lambda = 3$, $L_1 = 0.5$. These plots show that 1/120 Hz is an acceptable update rate, with little improvement from higher rates.

Section 5.1. Three input parameters to the fuel-balancing algorithm have been studied in simulation. $\Lambda$ controls the balance between formation-wide fuel use and speed of control. $L_1$ affects how rapidly fuel balancing is accomplished; a value of 0.5 was found to work well. The update frequency did not have a strong affect on performance; it can be chosen based on implementation cost and on desired responsiveness.

The fuel-balancing-coordinator algorithm meets the goals discussed in Section 5.3. The algorithm achieves stable, fuel-balanced formation behavior. The coordinator minimizes the maximum rate of fuel use by members of the formation, which extends total mission life (with all spacecraft functional). However, no performance or stability guarantees have
Table 5.2: Maximum spacecraft-pair fuel-difference percentage at conclusion of a six-spacecraft aperture filling maneuver simulation; results shown without and with a fuel-balancing coordinator.

<table>
<thead>
<tr>
<th>Control Architecture</th>
<th>$\Lambda$</th>
<th>Without a Coordinator</th>
<th>With a Coordinator</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Fuel Difference (%)</td>
<td>Fuel Difference (%)</td>
</tr>
<tr>
<td>Hub/Spoke</td>
<td>1.2</td>
<td>27</td>
<td>0</td>
</tr>
<tr>
<td>Hub/Spoke</td>
<td>6</td>
<td>27</td>
<td>1</td>
</tr>
<tr>
<td>Hub/Spoke</td>
<td>18</td>
<td>27</td>
<td>1</td>
</tr>
<tr>
<td>Hierarchy</td>
<td>1.2</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>Hierarchy</td>
<td>6</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>Hierarchy</td>
<td>18</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>Centralized</td>
<td>1.2</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>Centralized</td>
<td>6</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>Centralized</td>
<td>18</td>
<td>12</td>
<td>2</td>
</tr>
</tbody>
</table>

been presented, and systematic methods for choosing the three input parameters discussed in Section 5.3.2 are still under investigation. The following section describes a six-vehicle-formation simulated observing maneuver which employs a fuel-balancing-coordinator implementation based on the results of this section.

5.4 Fuel-Balancing Simulations

The six-spacecraft control architecture simulations in Chapter 4 include a fuel-balancing coordinator. A stop-and-go aperture-filling sequence (discussed in Section 4.5.2) was simulated for a six-vehicle formation for three different control architectures operating at three different values of $\Lambda$. The coordinator implements the algorithm in Equation 5.1 operating once per minute with $L_1 = 0.5$. Simulations were also run without a coordinator for comparison.

Table 5.2 lists the maximum spacecraft fuel differences at the conclusion of each simulation. The numbers are the difference, between spacecraft, of the minimum and maximum fuel used as a percentage of total fleet-wide fuel use. It is clear that the coordinator improves the fuel balance for the hub/spoke and centralized control architectures. The hierarchy control architecture does not benefit from the coordinator; the addition of the fuel-balancing coordinator does not significantly change the fuel differentials for this case. The impact of
the coordinator is reduced by the complexity of this control architecture. Two controllers (one intra-group and one inter-group) contribute to each spacecraft's control action. Further work is necessary to find the best way to achieve the desired fuel balancing behavior for the hierarchy control architecture. One solution may be to implement two coordinator levels, one for intra-group control and one for inter-group control.

Table 5.3: Fuel used by the vehicle using the most fuel, at the conclusion of the six-vehicle formation observing-sequence simulation without and with a fuel-balancing coordinator.

<table>
<thead>
<tr>
<th>Control Architecture</th>
<th>A</th>
<th>Without a Coordinator</th>
<th>With a Coordinator</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Maximum Fuel (sec)</td>
<td>Maximum Fuel (sec)</td>
</tr>
<tr>
<td>Hub/Spoke</td>
<td>1.2</td>
<td>1182</td>
<td>778</td>
</tr>
<tr>
<td>Hub/Spoke</td>
<td>6</td>
<td>800</td>
<td>565</td>
</tr>
<tr>
<td>Hub/Spoke</td>
<td>18</td>
<td>522</td>
<td>384</td>
</tr>
<tr>
<td>Centralized</td>
<td>1.2</td>
<td>800</td>
<td>753</td>
</tr>
<tr>
<td>Centralized</td>
<td>6</td>
<td>536</td>
<td>481</td>
</tr>
<tr>
<td>Centralized</td>
<td>18</td>
<td>346</td>
<td>322</td>
</tr>
</tbody>
</table>

Table 5.3 lists the maximum fuel used for the hub/spoke and centralized control architecture cases with and without fuel-balancing coordinators. This metric reflects the maximum fuel-use rate, which determines how long the mission operates before the first vehicle exhausts its fuel supply. Adding the fuel-balancing coordinator improves science mission duration by 26%–34% for the hub/spoke control architecture and by 6%–10% for the centralized control architecture. This practical performance benefit demonstrates the utility of a coordinator.

### 5.5 Summary

This chapter applies a hierarchical solution technique from the field of decentralized control to the spacecraft formation flying problem. A spacecraft formation coordinator enables fleet-wide cooperation to enhance performance and increase autonomy with minimal implementation costs. Potential coordinator applications include: fuel balancing, trajectory monitoring and replanning, observation scheduling, and fault detection and correction.

An autonomous fuel-balancing coordinator illustrates the utility of this approach. A new fuel-balancing algorithm that works with the low-level formation controllers developed in Chapter 4 was derived for this coordinator application. The fuel-balancing coordinator
was demonstrated in simulation for three- and six-vehicle formations. Fuel balancing is just one application of a coordinator; spacecraft formation flying missions can employ the coordinator technique to address any formation-level task. Autonomous management of formation-level tasks such as fuel balancing will be critical to the success of future spacecraft formation flying missions.
Chapter 6

Formation Flying Demonstrations

This chapter describes two FFTB experiments that demonstrate additional key capabilities required by upcoming spacecraft formation flying missions beyond those demonstrated in Chapters 3 and 4. The first experiment (Figures 6.1–6.3) demonstrates the multi-level alignment approach described in Section 1.2. The control uses GPS to bring a two-vehicle formation into coarse alignment and an optics layer to achieve fine alignment.

The second experiment (Figures 6.4–6.5) demonstrates a three-vehicle virtual-spacecraft-bus maneuver, maintaining coarse alignment while the formation undergoes a significant translation and rotation maneuver. This demonstration also experimentally compares distributed and centralized control architectures, verifying the simulations in Section 4.5.

6.1 Multi-Level Alignment

SSI missions must achieve extremely precise alignment between optical instruments on separate vehicles in order to measure interference fringe data (e.g., see Appendix A and [12, 37]). Section 1.2 describes a multi-level control approach that can achieve the required alignment using successive control layers of increasing precision but decreasing fields of view. A two-vehicle experiment on the FFTB demonstrates two potential layers of such a multi-level control system. The remaining alignment layers can then rely on standard interferometer-alignment techniques, such as those described in [51, 52, 50, 15, 17].

In the experiment, coarse alignment (cm-level) is achieved using GPS sensing and thrusters for relative stationkeeping. While the coarse-alignment system is active, a fine-alignment system can achieve mm-level alignment between instruments on the two vehicles.
Figure 6.1: Phase-plane portrait of relative state error in one axis with GPS-based sensing only (experimental data). Deadband region and control switching lines are also shown.

The fine-alignment control layer is a simple optics system consisting of a laser fixed on one vehicle and steered by that vehicle’s reaction wheel. The laser is directed at a target onboard the second vehicle, which detects the position of the laser using a CCD camera. The camera image is processed in real time to provide laser location updates at 60 Hz. This optical alignment system is a surrogate for the more-complex systems that will be used on SSI missions.

The coarse-alignment system uses GPS sensing only and leader-follower relative control. The vehicles are free-floating on the table, separated by 1.2 m. The laser vehicle follows the
target vehicle, which actively maintains its absolute location in the GPS frame of reference. Both position controllers use thrusters for actuation. Section 4.2.3 describes the relative control for this experiment. During ten seconds, the relative-position error orbits the lower deadband limit at $-1.2$ cm. The relative-position error is inside the position bound 80% of the time and remains under 2 cm. A phase-plane portrait containing 10 seconds of position and velocity data for translation in one axis is shown in Figure 6.1 (which repeats Figure 4.2).

The laser was tracked by the optical sensor as a metric (i.e. no feedback) during the first part of this demonstration. Figure 6.2 shows the laser position error during the same 10 seconds of data shown in Figure 6.1. The coarse controller keeps the laser on the 14 cm wide

Figure 6.2: Laser tracking performance with coarse control (GPS plus thrusters) only. (See footnote on following page.)
target, but the position of the dot moves freely within this range. This targeting behavior
is sufficient to keep the laser in the field of view of the more-precise optical sensor.1

With the coarse-control layer keeping the laser on the target, the fine-control layer was
activated. This control layer processes the target-vehicle's CCD-camera image of the laser
to find the pointing error. The reaction wheel on the laser-vehicle then corrects the laser
pointing. The two control layers are implemented independently for this experiment. Note
that the wheel torques are small and thus can be treated as disturbances by the coarse
loop. Furthermore, the optical sensor and reaction wheel are fast and accurate enough to
reject errors due to thruster activity. This experiment is designed to validate the multi-
level control approach, which achieves fine alignment via a series of increasingly-precise but
decreasing-workspace control systems. The goal of the fine-control layer is to significantly
improve pointing performance compared to the coarse loop alone.

The fine-control layer uses a simple proportional-derivative controller to generate reaction-
wheel torque commands based on the laser pointing error and rate. A typical performance
result with the fine-pointing loop engaged is plotted in Figure 6.3. Note the mean error
is only 0.2 cm, and the deviation from zero is less than 0.4 cm at all times. A constant
offset is seen in the laser position error which is due to bearing friction in the reaction wheel
(this could be easily corrected by adding an integral term to the control). The standard
deviation of the error (i.e., about the constant offset) is only 0.6 mm. This corresponds to a
factor of ~ 100 improvement (the size of the laser workspace is reduced from 14 cm to 1.2
mm) over the performance with only the coarse controller loop closed.

This experiment is the first demonstration of fine alignment achieved by multi-level
control of a simulated spacecraft formation using GPS sensing and an optical alignment
layer. This problem has a direct bearing on separated-spacecraft interferometer missions
which require precisely pointed, yet widely separated optical instruments. The optical
alignment system in this experiment is a surrogate for the more complex systems that
are being designed for these missions. Section 1.2.5 lists several references that include
detailed optical alignment system designs for SSI applications that will achieve the alignment

---

1The laser targeting error includes contributions from relative-position and orientation errors. The dead-
bands are designed so that when the errors are inside the deadbands, the laser usually falls on the target.
The relative-position-error deadband is ±1.2 cm. The orientation control is accomplished using thrusters
and GPS sensing only (the reaction wheel is used exclusively by the fine alignment control layer for this
demonstration). The orientation deadband is ±0.05 radians, which translates to ±0.6 cm in targeting error
at a 1.2 m separation. The field of view of the laser sensor is 14 cm wide. Thus when the error states are
within the deadbands, the laser almost always falls on the target.
precision necessary to generate interference fringes. These designs all depend on a spacecraft formation flying coarse-alignment control system. This experiment demonstrates one such control system using thrusters for actuation and GPS for sensing. This system keeps the relative state within the workspace of a fine-alignment control layer. The fine (but limited-range) optical alignment system improves alignment by a factor of 100.

6.2 Virtual Spacecraft Bus Maneuver

Reference [37] describes a preliminary design for the Terrestrial Planet Finder mission that acquires data while the spacecraft formation is in motion (see pages 108–109). The ability to combine relative and absolute control in this way will be important for SSI missions. The
experiments in this section were developed to demonstrate a virtual spacecraft bus maneuver with three FFTB vehicles maintaining tight relative control during a challenging maneuver. In these experiments, the vehicles move together as a rigid body. References [99, 100] study spacecraft formation flying rigid-body maneuvers in simulation, but this section presents the first experimental demonstration of this capability (using physically-simulated-spacecraft vehicles).

A 50-second formation maneuver including a 90° turn and 1.6 m translation was executed while the vehicles maintain an equilateral-triangle formation. Each side of the triangle is 1 m, regulated to within 2 cm. As described in Section 6.1, this level of alignment in turns allows finer control layers to align optical instruments as necessary.

The maneuver plan is generated using an optimal formation-maneuver-planning technique derived by Inalhan ([98]). The maneuver plan includes feedforward commands for all three vehicles and desired time histories for all vehicle states in absolute space. Feedback control of the absolute state is accomplished by one vehicle (Huey) which is assumed to have access to its absolute state. The absolute position control uses the single-vehicle fuel-time-optimal controller (described in Section 4.2). Centralized and distributed relative control architectures were compared to complement the simulations in Section 4.5. The relative- and absolute-controller λ's were chosen to emphasize relative control over absolute control for these experiments.

All controllers rely on models of the vehicles' thruster-acceleration parameters. This experiment uses the parameters listed in Table 2.2. Section 2.2 describes the system identification procedure used to estimate these parameters. These values were also used in the maneuver-planning algorithm.

Two relative-control architectures from Section 4.5 were tested. The overhead-vision sensing system is used for this experiment for both absolute and relative sensing. In both experiments, all three vehicles execute pre-computed feedforward thrust commands. The first experiment tested the distributed hub/spoke control architecture. The hub vehicle (Huey) regulates its absolute state according to the desired state history (part of the maneuver plan). Heavenly and Louie regulate their states relative to Huey. The relative state between Heavenly and Louie is not directly controlled.

---

2 Due to the large scale of the maneuver, the vehicles do not stay within the GPS coverage area on the testbed. Thus overhead vision sensing was used for this demonstration.
Table 6.1: Virtual spacecraft bus maneuver experiment results. Standard deviations of relative-position and relative-angle errors during the maneuver are listed for distributed and centralized control architectures. Total fuel use (in seconds of thrust) is also listed for each case.

<table>
<thead>
<tr>
<th></th>
<th>Distributed</th>
<th>Centralized</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Position Error (cm)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Huey-Louie</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td>Huey-Heavenly</td>
<td>1.1</td>
<td>1.9</td>
</tr>
<tr>
<td>Louie-Heavenly</td>
<td>3.1*</td>
<td>1.5</td>
</tr>
<tr>
<td><strong>Triangle Error (deg)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Louie-Huey-Heavenly</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>Huey-Louie-Heavenly</td>
<td>1.4</td>
<td>1.1</td>
</tr>
<tr>
<td>Louie-Heavenly-Huey</td>
<td>0.9</td>
<td>0.6</td>
</tr>
<tr>
<td><strong>Total Fuel (sec)</strong></td>
<td>125.4</td>
<td>110.8</td>
</tr>
</tbody>
</table>

* The length of this baseline was not directly controlled.

For comparison, a centralized control architecture was also tested. All three vehicles participate in the relative control using the heuristic multi-vehicle control law described in Section 4.4.2. Huey simultaneously regulates its absolute position and its relative position in the formation by computing independent absolute and relative control commands and then combining them in each vehicle-thruster axis. A simple combining algorithm assigns thrust commands $u$ based on the absolute and relative control commands ($u_{\text{abs}}$ and $u_{\text{rel}}$, respectively) according to

$$u = \begin{cases} 
+1 & \left(u_{\text{abs}}, u_{\text{rel}}\right) \in ((+1, +1), (+1, 0), (0, +1)) \\
0 & \left(u_{\text{abs}}, u_{\text{rel}}\right) \in ((0, 0), (+1, -1), (-1, +1)) \\
-1 & \left(u_{\text{abs}}, u_{\text{rel}}\right) \in ((-1, -1), (-1, 0), (0, -1)) 
\end{cases}$$

This algorithm was also used to combine feedforward thrust commands with feedback commands on all three vehicles. This scheme was sufficient for this demonstration, although a more sophisticated approach may improve performance.

Figure 6.4 plots the experimental results for the virtual spacecraft bus maneuver using the hub/spoke control architecture. The desired and actual paths are shown for all three vehicles in the x-y space of the testbed. The vehicle positions are shown connected at the start and end of the maneuver to highlight the equilateral-triangle formation. Note that the virtual spacecraft bus rotates by $90^\circ$ during the maneuver. The priority of the follower
Figure 6.4: Virtual-spacecraft-bus maneuver experiment with distributed relative control. The starting and ending positions are shown connected to illustrate the equilateral-triangle formation (dash-dot lines). Two intermediate points are marked (○’s and △’s). The starting triangle is on the right. At the start, Huey is at the bottom of the triangle and Heavenly and Louie are at the upper left and upper right, respectively. Position histories in the x-y space of the FFTB reference frame are plotted for all three vehicles (solid lines). The desired absolute positions are also plotted (dotted lines).

Table 6.1 lists formation errors as well as total fuel used for each control architecture during the virtual spacecraft bus maneuver experiments. Both architectures successfully maintain the formation during the maneuver. As shown in the table, the centralized architecture controls all three inter-spacecraft separations to within the 2 cm tolerance (for one
standard deviation of the errors). The distributed architecture directly controls two of the
three inter-spacecraft separations, these are also less than 2 cm (for one standard devia-
tion of the errors). As would be expected, the unregulated baseline (Louie-Heavenly) does
not perform as well (3.1 cm). This may be acceptable for SSI missions that use a collec-
tor/combiner approach; collector vehicles need to be aligned with the combiner vehicle (in
this case, Huey) more precisely than with other collector vehicles (e.g., see Reference [36]).

The triangle-error standard deviations are also listed in Table 6.1. These are the errors in
the angles between two triangle sides formed by pairs of vehicles. The standard deviations of
these errors are less than 2 degrees in all cases. Thus, both control architectures maintain the
size and shape of the desired equilateral-triangle formation during the rigid-body maneuver.
However the centralized architecture uses 12% less fuel than the distributed architecture,
which is consistent with the control architecture simulations in Section 4.5.
This experiment demonstrates that precise relative control can be achieved during a challenging rigid-body maneuver. This capability is important for upcoming SSI missions that require continuous data collection as the formation rotates to populate the image's $u, v$ plane.

6.3 Summary

The experiments in this chapter demonstrate key capabilities that will be required for upcoming spacecraft formation flying missions (e.g., the two NASA and one Air Force missions described in Chapter 7). The multi-level alignment experiment demonstrates a GPS-based coarse controller combined with a precise but limited-workspace optical control system to achieve precise alignment between separate simulated spacecraft. The virtual-spacecraft-bus maneuver experiments described here demonstrate the capability to combine relative and absolute control for simulated spacecraft which will enable rigid-body maneuvers and science observations “on-the-fly” for spacecraft formation flying missions. Also, the virtual spacecraft bus maneuver experimentally confirms the control-architecture simulation results in Section 4.5. Finally, the ability to maintain cm-level relative control during m-level motions on our testbed demonstrates the importance of accurate system modelling, as discussed in Chapter 2.
Chapter 7

Applications to Formation Flying Missions

This chapter discusses how some of the key findings in this research apply to upcoming spacecraft formation flying missions. The purpose of this study is to highlight the results in this research that could potentially contribute to key system engineering design choices in future spacecraft formation flying applications. Three missions are examined: Terrestrial Planet Finder (TPF), Space Technology 3 (ST3), and TechSat 21 (TS21). Each mission is currently under development, and preliminary reference designs are available. The features of this research that map into each mission’s design space are discussed in the following sections. Recommendations are made based on the results in this thesis and on the requirements listed for the reference designs. Of course, mission designers will make design decisions based on detailed examinations of all relevant parameters.

7.1 Terrestrial Planet Finder (TPF)

Reference [37] describes the TPF mission to search for planets orbiting stars other than our sun (“extra-solar planets’). It includes an illustrative mission design with a five-spacecraft SSI formation in an Earth-trailing orbit.
7.1.1 Control Architecture Design

The five-vehicle TPF mission is envisioned to act as a “rigid virtual truss” in space. Because five vehicles must cooperate to accomplish this goal, a distribution of control responsibilities must be devised. Section 4.5 identifies several control architectures that meet the relative-position control requirements. A centralized control structure yields the best fuel/time performance, but a simpler, distributed approach reduces system complexity. Section 6.2 experimentally identifies a 12% fuel-performance sacrifice for a distributed control architecture compared to a centralized control architecture, during a 50 second 3-vehicle virtual-spacecraft-bus maneuver.

Section 4.5 compares three control architectures for a simulated six-spacecraft formation performing an aperture-filling slew. This task is similar to the kind of tasks TPF will perform. A fuel-balancing coordinator (described in Chapter 5) was included for this simulation. The centralized architecture is the most efficient, completing the slews in 88–96% of the time using 77–82% of the fuel required by the distributed control architecture. The centralized architecture requires knowledge of all relative states at a high bandwidth (e.g., 60 Hz in this research), which may be a computational and/or connectivity challenge for TPF’s five-vehicle mission.

Recommendation

The cost of creating a robust, highly connected, fast computational network across the five vehicles must be compared to the benefit of implementing the centralized controller. The cost consists of any additional mass and power needed to implement fast, robust cross-link equipment. This cost may be outweighed by the fuel savings and/or added mission life provided by the centralized control architecture’s performance for TPF. Future missions with more vehicles and with requirements for more-flexible operation (e.g., allowing for adding vehicles to the formation during the mission timeline) will probably need to use distributed control architectures or hierarchical architectures to avoid the high connectivity cost associated with implementing a centralized control architecture.
7.1.2 Cooperative Control Algorithms

Fuel-efficient relative control will be of paramount importance for TPF. This requires both efficient maneuver design and efficient relative-state tracking. Chapter 4 derives a new fuel-time-optimal feedback controller for the two-vehicle case and a new multi-vehicle feedback-control algorithm. The two-vehicle case is a closed-form control law, while the multi-vehicle switch-time-optimized receding-horizon controller requires periodic optimization of a control plan (which has been rendered practical in this thesis by reducing the number of degrees of freedom in this plan to one plus two per vehicle). The switch-time-optimized receding-horizon controller is computationally expensive, but in three-vehicle simulations (see Section 4.4.4) it is more efficient than the heuristic centralized controller discussed in Section 4.4.2. This improved efficiency may justify the extra computational resources required.

Recommendation

The switch-time-optimized receding-horizon control algorithm provides the best fuel/time performance for centralized relative control. The switch-time optimizer can also be implemented as a fuel-efficient maneuver planner, computing optimal formation-correction trajectories as needed. Either application of this new control tool will benefit SSI missions such as TPF.

7.1.3 Fuel-Balancing Coordinator

Fuel balancing will be critical for spacecraft formation flying missions such as TPF. Reference [37] (page 134) cites the need for "...a significant level of flight system autonomy, in order to coordinate the actions of the participating spacecraft." A fuel-balancing coordinator reduces the maximum fuel used per vehicle by 9% compared to a control system that does not attempt to balance fuel, based on a three-spacecraft simulation (see Section 5.4). In the six-spacecraft simulations in Section 4.5, the coordinator reduces fuel imbalances by 10%-30%. The coordinator works at a low bandwidth (e.g., once per minute) by assigning relative-control gains according to the fuel-balancing algorithm described in Chapter 5. Total fuel use is increased by this method as some vehicles expend extra fuel to compensate for those with less fuel remaining. However, this is a worthwhile trade, as all vehicles are needed to perform science operations.
CHAPTER 7. APPLICATIONS TO FORMATION FLYING MISSIONS

Recommendation

The benefit of a fuel-balancing coordinator makes it well worth the small added system complexity. The coordinator described in Chapter 5 has a low implementation cost because it operates at a low bandwidth using information variables to interface with the individual spacecraft. Other operations that could benefit from coordination (e.g., trajectory planning, observation scheduling, and fault detection) should be explored as well.

7.2 Space Technology 3 (ST3)

ST3 is a two-vehicle Earth-trailing mission that will perform interferometric observations on time scales of hours. Reference [36] describes an early design for this SSI mission. Efficient relative control will be important, as will fuel balancing. References [65, 67] describe a new radio-frequency sensor based on GPS technology called the Autonomous Formation Flyer (AFF). The AFF will perform the relative sensing for this mission (and possibly for future SSI missions such as TPF).

7.2.1 Cooperative Controller

Precise relative-position control (within a few cm) will be necessary during observations and during observing slews (if drift-mode observations are made). The rate of fuel use will determine the useful life of this mission. The two-vehicle cooperative control algorithm presented in Section 4.3 is ideally suited for this mission. This closed-form control law was designed for this application (two spacecraft with on/off thrusters in a low-disturbance environment). It allows ground controllers — or an onboard fuel-balancing coordinator as described in Chapter 5 — to balance fuel use between vehicles and fuel use versus maneuver time. And it is the optimal (most efficient) control law for a given weighted fuel-time metric, performing 8% better than two vehicles independently controlling the relative separation vector in experiments on our testbed (see Section 4.3.3). Note that this controller takes advantage of the uncontrolled common-mode state. That is, the formation will likely drift out of its desired orbit over time.

Also, this controller assumes $1/s^2$ plant dynamics, which is a reasonable approximation for this mission. For example, during a thirty-hour observing program with a worst-case 1 km vehicle separation aligned with the Sun-radial vector, the relative drift due to orbital
motion will be 5.8 cm and 0.6 degrees in bearing angle\(^1\). This motion can be treated as
a disturbance for the control system to reject or it can be included in the observing-slew
design.

**Recommendation**

The cooperative control law will yield the best relative-control performance for ST3 and
should be implemented. The ST3 vehicles will follow a parabolic relative trajectory during
each observing program; this control law works to ensure the vehicles follow this trajectory
efficiently. A periodic (e.g., after each observing program) maneuver will probably be
required to correct formation-state drift.

### 7.2.2 RF-Based Sensor Design

An advanced radio-frequency position sensor (called the AFF) is baselined for ST3. This
sensor is based on the principles of GPS, but operates at a higher frequency to improve
precision (e.g., see Reference [65]). The experimental work in this thesis with an opera-
tionally different system has verified the suitability of DGPS-based relative-sensing systems
for spacecraft formation flying. Lessons learned in this research may be applicable to the
AFF as well.

**Recommendation**

Recent AFF simulation results (Reference [67]) show excellent position estimation. The ex-
perimental experience documented in Chapter 3 suggests using a Kalman filter with velocity
states and tight coupling to the spacecraft dynamics (e.g., thruster activity) to improve
estimation performance — including relative-position estimation. On our testbed, this im-
provement was about a factor of two in position-estimate error variance (References [60, 61])
compared to a static estimation algorithm.

\(^1\)The attitude control system will probably be designed to compensate for this (worst case) angle drift,
rendering it a “rigid body” mode of the formation that will barely perturb the measurement’s \(u, v\) location.
7.3 TechSat 21 (TS21)

TS21 will be a cluster of Earth-orbiting spacecraft that can combine their measurements to perform synthetic aperture radar observations. Reference [39] presents some system design options and mission requirements. References [21, 44] discuss fuel-efficient orbit and cluster designs. The present research validates GPS-based relative sensing for formation flying.

7.3.1 Differential Carrier Phase GPS for Relative Sensing

The inter-vehicle relative positions will need to be known with great accuracy for TechSat 21. Carrier phase differential GPS yields relative-position estimates that may be augmented with a precise "inner loop" metrology system (e.g., an optical alignment package). Chapter 3 experimentally demonstrates relative-position sensing using DGPS. An extended Kalman filter propagating at 60 Hz with measurement updates at 10 Hz yields cm-level relative-position accuracies on a two-dimensional indoor formation flying testbed. The FFTB includes three simulated spacecraft equipped with Trimble TANS receivers, six Stanford-designed GPS pseudolites and a poor multi-path environment. In Earth orbit, the performance may improve due to reduced multi-path and the additional — well-known and predictable — orbital dynamics in low Earth orbit. However, a space-based system will need to compensate for rapid turnover in the reference NAVSTAR satellites; this issue has not been addressed in this research, but is under investigation elsewhere (e.g., see Reference [64]).

Recommendation

This research validates DGPS sensing for formation flying and recommends including an extended Kalman filter to achieve the best relative-position estimates. The results provide confidence that this level of performance can be duplicated in Earth orbit given further developments to address issues specific to that environment.

7.4 Future Formation Flying Missions

Missions that are still in the concept phase offer exciting prospects for imaging planetary systems around nearby stars. Such missions (e.g., see References [16, 101]) will require long baselines (10-10,000 of km) and many spacecraft (20 or more). Obviously such missions will
build on the experience of nearer-term missions like TPF; however some lessons from this research may provide germane hints to future designers. Distributed sensing and control will probably be required for such missions; this research has identified partly-distributed hierarchical estimation and control architectures (e.g., dividing the vehicles into groups or possibly groups of groups) that combine local control with group-level control. These architectures offer some centralized performance benefit at the local level with manageable computation and complexity costs for the fleet. The idea of a coordinator may also extend to this structure with either a global coordinator or a top-level coordinator over-seeing sub-coordinators to accomplish a uniform system-wide behavior for such highly-distributed systems.
Chapter 8

Conclusions

Spacecraft formation flying enables exciting new science missions such as gravity-wave detection, synthetic aperture radar, and separated-spacecraft interferometry. Formation flying technology also adds new capabilities to terrestrial systems such as aircraft, automobile, and mobile-robot formations. This thesis has derived and demonstrated new formation flying tools for sensing, control and coordination. These tools enhance the performance of formations as small as two vehicles. In addition, these tools are important for enabling the number of vehicles in a formation to grow beyond the 2–5 range of current mission designs.

The focus application of this research was the virtual spacecraft bus; the VSB concept enables advanced new missions, such as separated-spacecraft interferometry. New cooperative controllers, GPS-based relative sensing, and a fleet-level coordinator have been developed for this application and demonstrated in simulation and on a realistic testbed. The control-and estimation-architecture problems have been studied, and two first-time experimental demonstrations have validated key capabilities for virtual spacecraft bus control. Finally, potential applications of this research have been identified for three upcoming spacecraft formation flying missions.

The following section summarizes the research contributions of this thesis. Section 8.2 lists potential extensions to this work.

8.1 Research Summary

Chapter 3 identifies several methods for improving GPS-based relative sensing. Simulation and experimental results quantify the performance improvement from an extended Kalman
compared to a static (least-squares) filter, and from adding the vehicle dynamics (including velocity states and actuator models) to the estimation problem. For the first time, a performance improvement for GPS trio-based relative sensing has been identified and quantified compared to GPS pair-based relative sensing. GPS-based relative sensing has been experimentally demonstrated with accuracy better than 2 cm in an indoor pseudolite-based testbed environment, given perfect knowledge of the carrier-phase integer ambiguities (and this despite the indoor multipath environment which is considerably worse than the one in space). Simulations predict that centralized estimation for three vehicles will achieve even better relative-sensing accuracy on the FFTB.

Chapter 4 derives two new relative controllers for formation flying. The first is a two-vehicle fuel-time-optimal cooperative control law ideally suited to two-vehicle formation flying missions such as Space Technology 3. This control law was demonstrated in simulation and experiment. The second controller is a fuel-time-optimal multi-vehicle control algorithm. This algorithm features a four order of magnitude reduction in computational cost for the three-vehicle problem, compared to the standard inverse dynamic optimization technique. The numerically-optimized multi-vehicle algorithm can be implemented either as a trajectory planner or as a receding-horizon feedback controller. Receding-horizon control was demonstrated in simulation for three vehicles.

Chapter 5 applies ideas from decentralized control to solve the spacecraft formation fuel-balancing problem with a small implementation cost. A simple algorithm that interacts with the vehicles through compact information variables is shown to be effective for reducing fuel imbalances. This function is critical for maximizing SSI mission duration. Fuel balancing is just one potential application of a fleet-level coordinator. It can be used for any task that requires low-bandwidth, inter-vehicle cooperation based on global knowledge. Such tasks include: fault detection and correction, trajectory planning, and observation scheduling.

The estimation- and control-architecture designs will be critical to spacecraft formation flying missions. The architectures determine estimation accuracy and control performance as well as implementation costs and robustness; large trade spaces exist for both architecture-design problems. Chapter 3 compares estimation architectures in terms of accuracy, implementation cost and robustness for the FFTB environment. Chapter 4 discusses implementation costs and robustness, and compares centralized and distributed control architectures in simulation to characterize fuel-time performance. A new hierarchical control
architecture was also simulated. These simulation studies focused on a generic six-vehicle spacecraft formation flying mission.

The results of these studies indicate that distributed sensing and centralized control will be the best combination for near-term spacecraft formation flying missions such as Terrestrial Planet Finder and TechSat 21. Centralized sensing is too costly to implement, and distributed sensing does provide cm-level performance, which will be sufficient for formation flying missions using multi-level alignment systems. However, the simulation results clearly indicate that the fuel savings from centralized control will be worth the extra implementation costs (including redundant systems to ensure robustness) for near-term missions.

Two experiments described in Chapter 6 demonstrate key capabilities required by upcoming spacecraft formation flying missions. The multi-level alignment experiment demonstrates a GPS-based coarse controller combined with a precise but limited-field-of-view optical control system to achieve precise alignment between separate physically-simulated spacecraft. Combined with interferometer-alignment techniques (such as those discussed in [51, 52, 50, 15, 17]), this multi-level alignment approach will enable separated-spacecraft interferometer missions to achieve the nm-level alignment necessary to track interference fringes and to collect science data. A virtual-spacecraft-bus maneuver experiment demonstrates the ability to combine relative and absolute control to enable rigid-body formation maneuvers that may allow missions such as the Terrestrial Planet Finder to continuously collect science data as the formation moves through many observing configurations, significantly improving efficiency.

Chapter 7 maps the key findings in this research to three near-term formation flying missions. Identifying real-world applications for these new techniques underscores the importance of these research results and highlights some of the important design decisions faced by formation flying mission designers. The engineering insight gained in this thesis has the potential to benefit spacecraft formation flying missions such as TechSat 21, Space Technology 3, and Terrestrial Planet Finder as well as future missions.

In summary, the contributions made by this research are:

- Demonstrated and quantified important advances in GPS-based relative sensing (Chapter 3)
CHAPTER 8. CONCLUSIONS

- Derived and demonstrated two new fuel-time-optimal relative-control laws for spacecraft formation flying (Chapter 4)
- Applied decentralized control techniques to the formation coordination problem; a fuel-balancing coordinator demonstrates this technique (Chapter 5)
- Investigated control-architecture and estimation-architecture design problems via simulations and trade studies (Chapters 3 and 4)
- Experimentally demonstrated two key spacecraft formation flying capabilities in a realistic two-dimensional physically-simulated-spacecraft testbed (Chapter 6)
- Charted the potential applications of these research results into the design spaces of three near-term space missions (Chapter 7)

8.2 Future Work

Several ambitious extensions to this research could be carried out and tested on the Stanford formation flying testbed. Three-vehicle centralized GPS sensing could be implemented and tested to experimentally verify the simulation-predicted accuracy improvement from the trio estimator compared to the pair estimator. Similarly, the three-vehicle switch-time optimizer could be implemented in feedback mode and experimentally validated on the FFTB. This would require programming a fast numeric optimizer such as the Numerical Algorithms Group's NPSOL code (the basis of Matlab's E04UCF function) into ControlShell. The fuel-balancing coordinator algorithm should also be implemented on the FFTB. Combining these three research extensions with a three-vehicle optical alignment system would enable a complete, high-performance formation flying control-system demonstration. Such a challenging undertaking would also tie together all of the spacecraft formation flying research in this thesis.

Another interesting extension would be to implement an interferometer on the formation flying testbed. Expertise from applied physics could be combined with this formation flying research to demonstrate for the first time a physically-simulated separated-spacecraft interferometer. The process of constructing this elaborate demonstration would yield invaluable engineering insight. If it were successfully implemented, it would be the first demonstration of an interferometer formed from elements on separate, actively-controlled platforms. This
would be an excellent complement to the monolithic SIM interferometer testbed at NASA's Jet Propulsion Laboratory.

Though spacecraft formation flying is the focus application of this research, terrestrial applications could certainly benefit from insight into control and estimation architectures, a coordinator, and improved GPS-based relative sensing. It would be interesting to apply these results to terrestrial formation flying systems such as the intelligent vehicle highway system, autonomous aircraft formations, or groups of cooperating mobile robots.

Finally, Chapter 7 identifies several potential applications of this research to three planned spacecraft formation flying missions. It would be quite interesting and very useful to see NASA and the US Air Force execute the recommendations from that chapter.
Appendix A

Overview of Interferometry

Interference refers to the effect when two wavetrains are superimposed. If the two waves have the same frequency and amplitude, then the result when they are combined may be a wave with twice the amplitude (constructive interference) or the waves may completely cancel (destructive interference). The phase difference between the two waves determines which of these occurs. The phase difference is related to the difference in distance the two waves have travelled (i.e., the difference in the wave-path lengths). Thus interference measures differential path length extremely precisely (i.e., down to a small percentage of the wavelength of light).

Two waves from the same source (such as a star) can be directed over different paths and then interfered to find very precisely the direction in which the source lies. This is the principal behind an interferometer.

Figure A.1 shows a wave emanating from a single light source A. For simplicity, assume the light source emits light at a single amplitude and frequency. The wave meets a plane B with two small apertures. New waves emanate from each aperture (the apertures must be small, on the order of the wavelength of light from A, for this to occur), but each wave has the same frequency, amplitude, and phase as the original wave did when it encountered the aperture. When these new waves meet at plane C, they interfere. The resulting pattern on plane C is called an interference fringe. At point D, the two waves have travelled the same distance and thus they interfere constructively to produce light of the same frequency but twice the amplitude.

The light pattern on plane C is not uniform, however. In fact the light amplitude will vary from twice the original amplitude to zero as we move along plane C. This variation
occurs because of the phase difference in the light arriving from each slit in B. The phase difference depends on the differential path length travelled and on the wavelength of the light. The differential path length depends on the separation between the apertures and on the angle between the plane B and the line from the center of B (midway between the apertures) through the point in C. Thus this method can be used to determine the angle very precisely if the wavelength of light is known.

If the plane B is not perpendicular to the line from the midpoint of B through source A then this angle has the same effect on the phase as that described above. Thus an interferometer consisting of an aperture plane B and a viewing plane C observing a particular wavelength of light can determine very exactly the angular position of a light source A. Wavelengths can be measured independently by the use of optical filters. An observing target (e.g., a planet) can be described as a collection of point sources, the effects of which all contribute (superpose) to each fringe measurement. By performing a two-dimensional Fourier transform on a set of fringe measurements from one observing target, the set of points can be reconstructed (like pixels on a computer screen) to form an image of the
source. The resolution of the image is defined by the angular size of each point in the image which is related to the maximum separation of the apertures.

In mathematical terms, we may describe the light of frequency \( f \) emanating from source A as a wave

\[
y = a \sin(kx - \omega t) \tag{A.1}
\]

where \( a \) is the amplitude of the light, \( x \) is the distance from A, \( t \) is time, \( k \) is the wave number \((= 2\pi/\lambda)\) and \( \omega \) is the angular frequency \((= 2\pi f)\). The slits in plane B act as light sources and the waves emanating from the slits can also be described as waves

\[
y_1 = a \sin(kx_1 - \omega t - \phi_1) \tag{A.2}
\]

and

\[
y_2 = a \sin(kx_2 - \omega t - \phi_2) \tag{A.3}
\]

where \( \phi \) is a phase shift introduced by the distance \( d \) travelled from A to the slit \((\phi = kd)\). Note that in Figure A.1 the slits are equidistant from A so \( \phi_1 = \phi_2 \). Now, on the assumption that superposition occurs, the light \( y_c \) at some point on plane C is just the sum of the two light waves

\[
y_c = a \left[ \sin(kx_1 - \omega t - \phi_1) + \sin(kx_2 - \omega t - \phi_1) \right] \tag{A.4}
\]

which can be rewritten using the trigonometric equation for the sum of the sines of two angles

\[
y_c = 2a \cos(k \frac{x_2 - x_1}{2}) \sin(k \frac{x_1 + x_2}{2} - \omega t - \phi_1) \tag{A.5}
\]

or

\[
y_c = 2a \cos(\frac{\pi(x_2 - x_1)}{\lambda}) \sin(k \frac{x_1 + x_2}{2} - \omega t - \phi_1) \tag{A.6}
\]

Which is a light wave of the same frequency as A with a phase shift caused by the distance travelled from A to C. If the lengths of the paths from the slits in B to the point we’re interested in are equal, the amplitude of the resulting light is twice that of the light from either slit. This is constructive interference. However if the path lengths from the slits differ by one half of one wavelength \((\lambda)\), the amplitude term reduces to zero \((\cos(\pi/2) = 0)\) and no light appears. This is destructive interference.

This effect allows very small changes in length (the path-length difference) to be detected very reliably because the resulting amplitude of the light output varies from zero to twice the
light amplitude being measured for a path-length change of just one half of one wavelength (e.g., 650 nanometers for red light). Interferometers take advantage of this effect to create extremely sensitive instruments. As mentioned before, an interferometer may consist of a plane with two apertures B plus an instrument observing plane C, perhaps containing a photodetector. The angle between the instrument plane and the light source (e.g., a star) can be measured with a very high resolution. The theoretical limit on the resolution (based on Rayleigh’s criterion) is

\[ r_\perp = \frac{1.22 \lambda}{b} \]

where baseline \( b \) is the separation between the apertures. This is the angular distance from the center of a point source at which the first interferometric null appears (see Reference [102] for a more complete explanation).

The technique described above can be applied to any electromagnetic wave. The first interferometers built for astronomical imaging observed the (long baseline) radio frequency using large arrays of radiotelescopes (see Figure 1.2). Imaging at optical wavelengths using an interferometer is briefly summarized in the next section.

**Visual Observations with Interferometers**

Because of superposition, light from a cluster of point sources combines predictably. By measuring the interfered light over an array of baselines, an image of the original cluster may be synthesized. Similarly, a source that emits light at more than one frequency (color) can be imaged at each light frequency independently. So, if we model a distant target to be imaged (e.g., an extrasolar planet) as a collection of point sources emitting light at many frequencies, an interferometer can locate the color and intensity of each point source. These can then be combined to form an image of the target much like the pixels on a computer screen combine to form an image. Next, we summarize the mathematical technique that converts an array of interference fringe measurements to an array of pixels in an image. More complete discussions appear in References [103, 104].

Consider again the interferometer in Figure A.1. It is redrawn in Figure A.2 with some changes. We have rigidly connected planes B and C and added four photodetectors on plane C. The detectors are placed to introduce known phase differences between the light fringes measured by each detector. In fact, this effect can be achieved more efficiently using
modern optical elements, but the principal is the same. Also, source A is now placed at a small offset angle $\theta$ from the instrument boresight.

The interference fringe described by Equation A.6 has an amplitude $G$ which is proportional to the cosine of the phase difference $\phi$ between the light arriving from the two apertures

$$G \propto \cos(\phi) \quad (A.8)$$

For the setup in Figure A.2, $\phi$ consists of two parts. The first part $\phi_1$ depends on the position $x$ of the photodetector in plane C. It can be approximated from geometry and a small angle approximation

$$\phi_1 \approx k \frac{bx}{y} \quad (A.9)$$
The second part $\phi_2$ is due to the angle to target $A$; it can be similarly approximated

$$\phi_2 \approx kb\theta$$ \hspace{1cm} (A.10)

The detectors measure fringe intensity which is equal to the square of the amplitude

$$I = G^2$$ \hspace{1cm} (A.11)

At a detector equidistant from the apertures ($x = 0$, so $\phi_1 = 0$), let $Y_1$ be the complex amplitude of the light arriving from one aperture; then the light from the other aperture is phase shifted

$$Y_2 = Y_1 \exp(j(\phi_1 + \phi_2))$$ \hspace{1cm} (A.12)

and

$$G = \Re(Y_1 + Y_2)$$ \hspace{1cm} (A.13)

so

$$I_1 = |Y_1 + Y_1 \exp(jkb\theta)|^2$$ \hspace{1cm} (A.14)

which can be written as

$$I_1 = 2Y_1Y_1^*|1 + \cos(jkb\theta)|$$ \hspace{1cm} (A.15)

Three more detectors spaced apart in half-wavelength increments ($\phi_1 = \pi/2, \pi, 3\pi/2$) provide the intensity measurements

$$I_2 = 2Y_1Y_1^*|1 + \sin(jkb\theta)|$$ \hspace{1cm} (A.16)

$$I_3 = 2Y_1Y_1^*|1 - \cos(jkb\theta)|$$ \hspace{1cm} (A.17)

$$I_4 = 2Y_1Y_1^*|1 - \sin(jkb\theta)|$$ \hspace{1cm} (A.18)

When these four intensity measurements are combined to form the visibility $V$ according to

$$V \equiv (I_1 - I_3) + j(I_2 - I_4)$$ \hspace{1cm} (A.19)

the result is the useful form

$$V = 4Y_1Y_1^* \exp(jkb\theta) = |G|^2 \exp(jkb\theta)$$ \hspace{1cm} (A.20)
Now instead of a single light source, consider an image (e.g., composed of a continuum of sources) which may be described by a brightness distribution $B(\theta)$. The amplitude of the light arriving from a direction $\theta$ depends on $B(\theta)$

$$|G(\theta)|^2 = B(\theta) d\theta$$  \hspace{1cm} (A.21)

Then the visibility $V$ is the sum of the visibility from every angle $\theta$

$$V = \int_\theta |G(\theta)|^2 \exp(j kb\theta) \hspace{1cm} (A.22)$$

and for a visibility measurement at baseline $b$

$$V(b) = \int_\theta B(\theta) \exp(j kb\theta) d\theta \hspace{1cm} (A.23)$$

which is the one-dimensional Fourier transform of the brightness distribution $B(\theta)$ with respect to the spatial variable $u$ (where $u = kb$). Thus, a set of visibility measurements $V$ at many values of $u$ (i.e., at many baselines $b$) can be inverse-transformed mathematically to form the brightness distribution $B(\theta)$ which corresponds to an angular image map (i.e., a picture of the imaging target).

This technique can be expanded to two dimensions (where the spatial variables $u, v$ correspond to the baseline lengths on two axes with angles $\theta, \psi$ to the target area). More complete explanations appear in References [102, 103, 104]. The key idea is that a set of interference fringe measurements may be mathematically transformed into a visual image. The angular resolution of the resulting image is inversely proportional to the maximum baseline at which measurements were taken. The greater the number of measurements taken, the more detail appears in the picture. During a measurement, the baseline must be known to a high degree of precision and it must remain fixed.
Appendix B

Two-Vehicle Optimal Controller Implementation

The following Matlab function implements the two-vehicle fuel-time optimal control law derived in Chapter 4.

function [thrust,coeff1,coeff2,coeff3,coeff4]=twovehftopt(state,accel,lambda);

% Andrew Robertson
% 5/10/00
% This MATLAB function implements the two-vehicle fuel-time
% optimal control law in one axis.
% % INPUTS
% state = [vehicle 1 velocity, veh. 1 position, veh. 2 vel, veh. 2 pos]
% accel = [vehicle 1 acceleration/thruster, veh 2 acc/thruster]
% lambda = [veh 1 lambda, veh 2 lambda]
% % OUTPUTS
% coeff1-coeff4 = are the coefficients of the four switching lines
% thrust = [veh 1 thrust, veh 2 thrust]
% where thrust is {-1, 0 or +1}
APPENDIX B. TWO-VEHICLE OPTIMAL CONTROLLER IMPLEMENTATION

if (\(\lambda_1/\text{accel}(1)\)) > (\(\lambda_2/\text{accel}(2)\)),
   flag1=1;
else
   flag1=0;
end;

if flag1,
   state=[state(3) state(4) state(1) state(2)]’;
   lambda=[\lambda_2 \lambda_1];
   accel=[\text{accel}(2) \text{accel}(1)];
end;

\(x=\text{state}(2)-\text{state}(4)\);
\(\dot{x}=\text{state}(1)-\text{state}(3)\);

k1=2*\text{accel}(1);  \% Assuming two thrusters in each direction
k2=2*\text{accel}(2);
\lambda_1=\lambda_1;
\lambda_2=\lambda_2;

B=(\lambda_2/k2)-(\lambda_1/k1);

if B>(1/k1), \% upper bound on B
   B=1/k1;
end

A= - ((1-k_1*B)^2)/(2*(k_1+k_2)) - B + ((B^2)*k_1/2);
C=1+\lambda_1-\lambda_2*(k_1/k2);

flag2=1; \% Determine the quadrant of switching line 4
if C<0,
flag2=-1;
end;

if C==0,  % Prevents numeric errors (divide by zero)
    C=eps;
end;

coeff1=(2*(k1+k2));
coeff2=abs(1/A);
coeff3=abs(1/(A-2*lambda1/k1));

D=(lambda2/k2)-(lambda1/k1);

flag3=1;
if ((A-2*lambda1/k1)/C^2 -D/C -(k1*D^2)/(2*C^2))>0
    flag3=-1;
end;

coeff4=flag3*flag2*abs(1/((A-2*lambda1/k1)/C^2 -D/C -(k1*D^2)/(2*C^2)));

if flag3==1,
    if coeff4<coeff1,  % switching line 4 bounded by sw. line 1
        coeff4=coeff1;
    end;
end;


if ((g1>xdot) & (g2>xdot)),
    u1=+1;
else
    if ((xdot>g1) & (xdot>g2)),
        u1=-1;
    else
        u1=0;
    end;
end;

if ((g3>xdot) & (g4>xdot)),
    u2=+1;
else
    if ((xdot>g3) & (xdot>g4)),
        u2=-1;
    else
        u2=0;
    end;
end;

if flag3=-1, % Switch test reverses meaning in this quadrant
    if (u2~=0),
        u2=0;
    else
        u2=-sign(xdot);
    end;
end;

thrust=[u1 u2];

if flag1,
    thrust=[-thrust(2) -thrust(1)];
end;
Bibliography


BIBLIOGRAPHY


