

CONTROLLED AND UNCONTROLLED MOTION IN THE
CIRCULAR, RESTRICTED THREE-BODY PROBLEM:
DYNAMICALLY NATURAL SPACECRAFT FORMATIONS

by

Ralph Ramos Basilio

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(AEROSPACE ENGINEERING)

August 2007

UMI Number: 3287118

Copyright 2007 by

Basilio, Ralph Ramos

All rights reserved.



UMI Microform 3287118

Copyright 2007 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Epigraph

In this life we get only those things for which we hunt, for which we strive, and for which we are willing to sacrifice.

-George Matthew Adams

Dedication

This work is dedicated to my loving wife, Eleanor, and to our son, Andrew, who is and will always be our pride and joy.

Acknowledgements

Completing a challenging graduate degree program was only possible through the guidance, encouragement, assistance, and patience of a number of individuals. My sincere thanks go to:

- Paul Newton, Professor, Department of Aerospace and Mechanical Engineering, for supervising this body of work and understanding my need to balance my academic interests with a professional career and family responsibilities;
- Peter Baxendale, Professor, Department of Mathematics, and Larry Redekopp, Professor, Department of Aerospace and Mechanical Engineering, for serving on both my dissertation and guidance committees;
- Henryk Flashner, Professor, and Firdaus Udwadia, Professor, both of the Department of Aerospace and Mechanical Engineering, for serving on my guidance committee;
- Eleanor, my darling wife, and Andrew, our son, for their enduring patience, support, understanding, and for representing the three of us during those family events and outings I missed out on, because of my academic commitments;
- Greg and Teresita Basilio, my parents, Joseph Basilio, my brother, Pat Basilio, his wife and my sister-in-law, Matthew, their son and my nephew, Sergio and Suerte Vasco, my in-laws, Elizabeth Vasco-Belen, my other sister-in-law, Glenn Belen, her husband, Christina Belen, their daughter, and Glenn Belen, Jr., their son for understanding why and allowing me to miss out on those family events and outings;
- Marrietta Penoliar, Student Programs Advisor, Department of Aerospace and Mechanical Engineering, for guiding me through much of my entire graduate school experience;
- Elliot Axelband, Associate Dean for Research Development, Viterbi School of Engineering, Conrad Newberry, Professor Emeritus, Naval Postgraduate School, and Richard Stanton and Mike Jahan, my former managers at the Jet Propulsion Laboratory, California Institute of Technology who all believed that I could complete a challenging

graduate engineering program while working full-time and demonstrating that conviction by writing and submitting letters of recommendation;

- Martin Lo of the High Capability Computing and Modeling Group, Jet Propulsion Laboratory, California Institute of Technology for his support, encouragement in this endeavor, and assistance in identifying and developing a research topic;
- Randy Paffenroth, formerly a Staff Scientist in the Applied and Computational Mathematics Department, California Institute of Technology for tutoring me on the inner workings of the AUTO 2000 continuation and bifurcation analysis software tool; and finally,
- Mohamed Abid, Richard “Jim” Aragon, Cindy Bevans, Ron Boain, Dave Braun, Shannon Brown, Randy Coffey, Mike Davis, Ben Dominguez, Angie Dorsey, Chuck Eidem, Steve Durden, Diane Evans, Lee Fu, Mike Gallagher, Jim Graf, Steve Greenberg, Dave and Bobbi (Grable) Gregorich, David Guarino, Natalie Guzman, Mary Ann Hall, Gail Hammitt, Paul Hernandez, Debbie Higuera, Eastwood Im, Bill Kert, Ami Kitiyakara, John Kreigenhofer, Bruce Krohn, Ron Kruid, Try Lam, Thomas Livermore, Scott Michel, Alex Nicolson, Mark Rokey, Dawn Skinner, Deborah Vane, Parag Vaze, K. Charles Wang, Ray Welch, John Wirth, Charlie Yamarone, and others not mentioned – my managers and colleagues, past and present, who allowed me to further my education without making me feel guilty that I did not spend ‘all’ of my extra off-hours on CloudSat, OSTM (Ocean Surface Topography Mission), and/or OCO (Orbiting Carbon Observatory) related work.

The Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration, provided tuition re-imbursement.

Table of Contents

Epigraph.....	ii
Dedication.....	iii
Acknowledgements	iv
List of Tables	viii
List of Figures.....	x
Acronyms and Abbreviations.....	xv
Abstract.....	xvi
Preface.....	xvii
Chapter 1: Introduction.....	1
1.1 Synchronization.....	1
1.1.1 Coordinated Action	1
1.1.2 Coordinated Motion	2
1.2 Astrodynamics and Spacecraft Formation Flying	3
1.2.1 Current Methods and Scientific Applications	4
1.2.2 Future Possibilities and the Thesis.....	6
1.3 The Structure of the Ph.D. Dissertation.....	7
Chapter 2: Two-Body Systems, Modeling, and Analysis	8
2.1 Two-Body Systems	8
2.1.1 General Two-Body Problem	8
2.1.2 Central Force Motion	11
2.2 Computer Simulations and General Analysis.....	13
Chapter 3: The Circular, Restricted Three-Vortex Problem	19
3.1 Equations of Motion.....	21
3.2 Equilibrium Points.....	23
3.3 Level Curves of the Hamiltonian.....	23
3.4 Phase-Lock Controller.....	28
3.5 Formation Establishment.....	44
3.5.1 Resonant Frequency Approach.....	45
3.5.2 Controller Method	48
3.6 Proof-of-Concept Problem and Solution	52

Chapter 4: The Circular, Restricted Three-Body Problem.....	57
4.1 Equations of Motion.....	58
4.2 Equilibrium Points.....	62
4.3 Jacobi Integral.....	70
4.4 Periodic Orbit Generation	72
4.5 General Stability and Other Periodic Orbit Characteristics	86
4.6 Problem No. 1: Lyapunov (Planar) Case.....	99
4.6.1 Periodic Orbits.....	99
4.6.2 Phase-Lock Controller.....	101
4.6.3 Formation Establishment	108
4.6.3.1 Resonant Frequency Approach.....	108
4.6.3.2 Controller Method.....	109
4.6.4 Example Problem and Solution	117
4.7 Problem No. 2: Three-Dimensional Case.....	118
4.7.1 Periodic Orbits.....	119
4.7.2 Phase-Lock Controller.....	120
4.7.3 Formation Establishment	124
4.7.3.1 Resonant Frequency Approach.....	124
4.7.3.2 Controller Method.....	124
4.7.4 Example Problem and Solution	130
Chapter 5: Evaluation and Assessment.....	132
5.1 Verification & Validation.....	132
5.2 Limitations.....	133
5.3 Lessons Learned	134
5.4 Potential Scientific Applications.....	135
5.5 Recommendations for Future Work	135
Chapter 6: Conclusions.....	136
Glossary	138
Bibliography	139
Appendices.....	141
Appendix A: MATLAB and AUTO 2000 Computer Tools	142
Appendix B: MATLAB Scripts, Function Files, and Programs.....	149
Appendix C: AUTO 2000 Program Files	188
Appendix D: Catalogue of Periodic Orbits Around the Earth-Moon L4 Point.....	197
Appendix E: Floquet Theory and the Monodromy Matrix	212
Appendix F: 2005 SIAM Dynamical Systems Conference Presentation Charts	216
Appendix G: 2006 SIAM PDE Conference Presentation Charts.....	223
Appendix H: Dissertation Defense Presentation Charts	232

List of Tables

Table 2.1. The relationship between the magnitude of eccentricity, e , and the shape of the orbit.....	12
Table 3.1. Equilibrium point coordinates for the case with equal strength primary vortices.....	23
Table 3.2. Attributes of the two periodic orbits, P_1 and P_2 , surrounding the second vortex.....	27
Table 3.3. Circular, restricted three-vortex problem: Controller no. 1 parameter values.....	41
Table 4.1. The coordinates for the five equilibrium points in the earth-moon system.....	64
Table 4.2. The coordinates for the five equilibrium points in the Saturn-Titan system.....	65
Table 4.3. The eigenvalues for the five equilibrium points in the earth-moon system.....	68
Table 4.4. The eigenvalues for the five equilibrium points in the Saturn-Titan system.....	69
Table 4.5. A description for each of the four AUTO 2000 input files.....	73
Table 4.6. Earth-moon L4 equilibrium point periodic orbit families.....	74
Table 4.7. Earth-moon L5 equilibrium point periodic orbit families.....	77
Table 4.8. Planar case: Floquet multipliers for the inner most periodic orbit in Figure 4.21.....	88
Table 4.9. Planar case: Floquet multipliers for the second smallest periodic orbit in Figure 4.21.....	88
Table 4.10. Planar case: Floquet multipliers for the middle periodic orbit in Figure 4.21.....	88
Table 4.11. Planar case: Floquet multipliers for the second largest periodic orbit in Figure 4.21.....	89
Table 4.12. Planar case: Floquet multipliers for the outer most periodic orbit in Figure 4.21.....	89
Table 4.13. General attributes for six planar periodic orbits.....	90
Table 4.14. Green's theorem-derived area bounded by each planar periodic orbit.....	91
Table 4.15. Three-dimensional case: Floquet multipliers for orbit no. 126.....	97
Table 4.16. Three-dimensional case: Floquet multipliers for orbit no. 128.....	97
Table 4.17. Three-dimensional case: Floquet multipliers for orbit no. 130.....	97
Table 4.18. Three-dimensional case: Floquet multipliers for orbit no. 132.....	98
Table 4.19. Three-dimensional case: Floquet multipliers for orbit no. 134.....	98
Table 4.20. Stokes' theorem-derived area bounded by each three-dimensional periodic orbit.....	98
Table 4.21. Planar case: Controller no. 1 parameter values.....	105

Table 4.22. Planar case: Staging, transfer, and total time to reach the desired orbit.....	107
Table 4.23. Planar case: Controller no. 2 parameter values	112
Table 4.24. Three-dimensional case: Controller no. 1 parameter values.....	121
Table 4.25. Three-dimensional case: Staging, transfer, and total time to reach the desired orbit	122
Table 4.26. Three-dimensional case: Controller no. 2 parameter values.....	125

List of Figures

Figure 2.1. A sketch of a two-body system located in three-dimensional physical space.....	9
Figure 2.2. A MATLAB plot of orbit traces for a two-body system with equal masses.....	13
Figure 2.3. A plot of the barycenter speed for a two-body system with equal masses.....	14
Figure 2.4. An orbit trace using ode23 with reltol=1.0E-03 and abstol=1.0E-06.....	14
Figure 2.5. An orbit trace using ode23 with reltol=1.0E-05 and abstol=1.0E-08.....	15
Figure 2.6. An orbit trace using ode45 with reltol=1.0E-03 and abstol=1.0E-06.....	15
Figure 2.7. An orbit trace using ode45 with reltol=1.0E-05 and abstol=1.0E-08.....	16
Figure 2.8. A MATLAB plot of the differences in barycenter speed.....	16
Figure 2.9. A plot of potential, kinetic, and total energy of a two-body system.....	17
Figure 3.1. A hurricane churning the air and water around the Atlantic Ocean.....	19
Figure 3.2. A vortex moving only in a flat, two-space surface.....	20
Figure 3.3. The primaries, V_1 and V_2 , lying on the real axis of the rotating coordinate frame.....	21
Figure 3.4. Level curves of the Hamiltonian surrounding one of the two primary vortices.....	24
Figure 3.5. A plot of characteristic distance versus orbit period.....	24
Figure 3.6. A plot of characteristic distance versus angular rate.....	25
Figure 3.7. A plot of characteristic distance versus angular rate using the log-log scale.....	25
Figure 3.8. A plot of characteristic distance versus energy level.....	26
Figure 3.9. A plot of two periodic orbits that formed the foundation for the study.....	27
Figure 3.10. A test particle under the influence of a time independent controller.....	28
Figure 3.11. The same plot shown in Figure 3.10b, but using a different scale.....	29
Figure 3.12. A test particle under the influence of a time dependent controller where $\kappa=1.0$	30
Figure 3.13. A test particle under the influence of a time dependent controller where $\kappa=4.0$	31
Figure 3.14. A state diagram for both the time independent and time dependent cases.....	32
Figure 3.15. A test particle moving along a periodic orbit between the P_1 and P_2 orbits.....	32
Figure 3.16. A test particle being placed on the same periodic orbit as another.....	33

Figure 3.17. The trajectory of a test particle under the influence (driven by) controller no. 1.....	34
Figure 3.18. Test particle trajectories where controller no. 1 is active.....	35
Figure 3.19. A state diagram for controller no. 1	36
Figure 3.20. Test particle no. 1 shown moving in a periodic orbit about V_2	36
Figure 3.21. Test particle no. 1 shown moving under the influence of the controller	37
Figure 3.22. Test particle no. 1 shown moving in a new periodic orbit	37
Figure 3.23. A plot of the energy level as a test particle moves along the transfer trajectory.....	38
Figure 3.24. Four test particles, P_1 - P_4 , each traveling along their respective initial orbits.....	39
Figure 3.25. A plot of the trajectory needed to place the first test particle, P_1 , on the new orbit	39
Figure 3.26. Controller no. 1 trajectories at four different initial starting positions	40
Figure 3.27. The trajectory showing a test particle leaving and arriving on a tangent	41
Figure 3.28. Transfer trajectory curves for each of the four test particles.....	42
Figure 3.29. The staging times and transfer trajectory times for each of the four test particles.....	42
Figure 3.30. The staging times and transfer trajectory times for each of the four test particles.....	43
Figure 3.31. The actual and desired test particle positions at $t=1.48$ units of time	44
Figure 3.32. The flowchart for the script file, i.e. main_script_v5_1.m.....	45
Figure 3.33. A plot showing how many times each particle must traverse their periodic orbits.....	47
Figure 3.34. Motion of a test particle using a scale factor for controller no. 2	49
Figure 3.35. Motion of a test particle using a time element for controller no. 2.....	49
Figure 3.36. Motion of a test particle using a trigonometric function for controller no. 2.....	50
Figure 3.37. Trajectory/orbit plots for four different scale factor values for controller no. 2	50
Figure 3.38. A plot of scale factor versus orbit period plus curve-fit residuals	51
Figure 3.39. A trajectory/orbit plot for a scale factor value of 2.6	51
Figure 3.40. The initial states and periodic orbits for four test particles	52
Figure 3.41. Test particle formation establishment timeline using controller no. 2	53
Figure 3.42. Initial positions and periodic orbits for four test particles	54

Figure 3.43. Four test particles traveling in controlled (controller no. 1) motion.....	55
Figure 3.44. Four test particles traveling in controlled (controller no. 2) motion.....	55
Figure 3.45. Initial and final states and periodic orbit for four test particles	56
Figure 4.1. The geometry of the circular, restricted three-body problem.....	57
Figure 4.2. The five equilibrium points of the circular, restricted three-body problem	64
Figure 4.3. An earth-moon L4 family w/initial orbit period = 6.283185 [XY planar projection].....	75
Figure 4.4. An earth-moon L4 family w/initial orbit period = 6.283185 [XZ planar projection]	76
Figure 4.5. An earth-moon L4 family w/initial orbit period = 6.283185 [YZ planar projection]	76
Figure 4.6. An earth-moon L5 family w/initial orbit period = 6.283185 [XY planar projection].....	78
Figure 4.7. An earth-moon L5 family w/initial orbit period = 6.283185 [XZ planar projection]	79
Figure 4.8. An earth-moon L5 family w/initial orbit period = 6.283185 [YZ planar projection]	79
Figure 4.9. An earth-moon L4 family w/initial orbit period = 21.070352 [XY planar projection].....	80
Figure 4.10. An earth-moon L4 family w/initial orbit period = 21.070352 [XZ planar projection]	80
Figure 4.11. An earth-moon L4 family w/initial orbit period = 21.070352 [YZ planar projection]	81
Figure 4.12. An earth-moon L5 family w/initial orbit period = 21.070352 [XY planar projection].....	81
Figure 4.13. An earth-moon L5 family w/initial orbit period = 21.070352 [XZ planar projection]	82
Figure 4.14. An earth-moon L5 family w/initial orbit period = 21.070352 [YZ planar projection]	82
Figure 4.15. Orbit no. v. orbit period for each L4 equilibrium point periodic orbit.	83
Figure 4.16. Some orbit traces of the earth-moon L4 equilibrium point periodic orbit family.....	84
Figure 4.17. A MATLAB plot of average distance from the L4 equilibrium point.....	84
Figure 4.18. A MATLAB plot of average distance from the L5 equilibrium point.....	85
Figure 4.19. A plot of average distance from the L4 equilibrium point as a function of period	85
Figure 4.20. A plot of average distance from the L5 equilibrium point as a function of period	86
Figure 4.21. Five periodic orbits around the earth-moon L4 equilibrium point.	87
Figure 4.22. An arbitrary state vector for a MATLAB initial value problem.	87
Figure 4.23. A linear plot of orbit frequency versus area.....	92
Figure 4.24. A semi-log plot (1 of 2) of orbit frequency versus area	92

Figure 4.25. A semi-log plot (2 of 2) of orbit frequency versus area.....	93
Figure 4.26. A log-log plot of orbit frequency versus area.....	93
Figure 4.27. A linear plot of minimum distance versus orbit period.....	94
Figure 4.28. A semi-log plot (1 of 2) of minimum distance versus orbit period	94
Figure 4.29. A semi-log plot (2 of 2) of minimum distance versus orbit period.	95
Figure 4.30. A log plot of minimum distance versus orbit period.....	95
Figure 4.31. Five periodic orbits around the earth-moon L4 equilibrium point	96
Figure 4.32. An attempted MATLAB initial value problem periodic orbit plot.....	96
Figure 4.33. Planar case: Spacecraft, S1 through S4, initial state vectors.....	100
Figure 4.34. Planar case: Spacecraft, S1 through S4, uncontrolled motion animation.....	101
Figure 4.35. Planar case: A spacecraft placed in controlled motion using controller no. 1.....	102
Figure 4.36. Planar case: Two spacecraft traveling in uncontrolled motion	103
Figure 4.37. Planar case: A spacecraft moves from an inner to an outer periodic orbit.....	103
Figure 4.38. Planar case: Desired spacecraft formation, i.e. equally spaced in time	104
Figure 4.39. Planar case: Spacecraft staging points, i.e. controlled motion initial condition.....	105
Figure 4.40. Planar case: Individual spacecraft transfer trajectories	106
Figure 4.41. Planar case: Spacecraft transfer trajectories plotted together.....	106
Figure 4.42. Planar case: Spacecraft initial conditions and desired formation	107
Figure 4.43. Planar case: Schematic of actual versus desired spacecraft relative positions.....	108
Figure 4.44. Planar case: Example spacecraft trajectories that close on themselves	109
Figure 4.45. Planar case: Example spacecraft trajectories that do not close on themselves.....	110
Figure 4.46. Planar case: Curve-fit of orbit period versus controller no. 1 λ_1 parameter	110
Figure 4.47. Planar case: Curve-fit of orbit period versus controller no. 1 λ_2 parameter	111
Figure 4.48. Planar case: Spacecraft 2 (S2) initial and final controller no. 2 states.....	112
Figure 4.49. Planar case: Spacecraft 2 (S2) controlled motion (controller no. 2) trajectory	113
Figure 4.50 Planar case: Spacecraft 1 (S1) initial and final controller no. 2 states.....	113

Figure 4.51. Planar case: Spacecraft 1 (S1) controlled motion (controller no. 2) trajectory	114
Figure 4.52. Planar case: Spacecraft 3 (S3) initial and final controller no. 2 states.....	114
Figure 4.53. Planar case: Spacecraft 3 (S3) controlled motion (controller no. 2) trajectory	115
Figure 4.54. Planar case: Spacecraft controller no. 2 trajectories.....	115
Figure 4.55. Planar case: Schematic of spacecraft final states.....	116
Figure 4.56. Planar case: Plot of spacecraft final states	116
Figure 4.57. Planar case: Spacecraft controlled motion (controller no. 1) animation.....	117
Figure 4.58. Planar case: Spacecraft controlled motion (controller no. 2) animation.....	118
Figure 4.59. Three-dimensional case: Spacecraft, S1 through S4, initial state vectors	119
Figure 4.60. Three-dimensional case: Spacecraft, S1 - S4, uncontrolled motion animation.....	120
Figure 4.61. Three-dimensional case: Spacecraft transfer trajectories plotted together	122
Figure 4.62. Three-dimensional case: Spacecraft initial conditions and desired formation	123
Figure 4.63. Three-dimensional case: Schematic of actual versus desired positions	123
Figure 4.64. Three-dimensional case: Spacecraft 2 (S2) initial and final controller no. 2 states	126
Figure 4.65. Three-dimensional case: Spacecraft 2 (S2) controlled motion trajectory	126
Figure 4.66. Three-dimensional case: Spacecraft 3 (S3) initial and final controller no. 2 states	127
Figure 4.67. Three-dimensional case: Spacecraft 3 (S3) controlled motion trajectory	127
Figure 4.68. Three-dimensional case: Spacecraft 1 (S1) initial and final controller no. 2 states	128
Figure 4.69. Three-dimensional case: Spacecraft 1 (S1) controlled motion trajectory.....	128
Figure 4.70. Three-dimensional case: Spacecraft transfer trajectories plotted together	129
Figure 4.71. Three-dimensional case: Schematic of spacecraft final states	129
Figure 4.72. Three-dimensional case: Spacecraft final states.....	130
Figure 4.73. Three-dimensional case: Spacecraft controlled motion (controller no. 1) animation.....	131
Figure 4.74. Three-dimensional case: Spacecraft controlled motion (controller no. 2) animation.....	131

Acronyms and Abbreviations

AVI	Audio-Visual Interleaved
BSD	Berkeley Software Distribution
BVP	Boundary-Value Problem
CDE	Common Desktop Environment
CR3BP	Circular, Restricted Three-Body Problem (co-planar case)
ESA	European Space Agency
HCW	Hill-Clohessy-Wiltshire (Equations)
IVP	Initial Value Problem
KAM	Kolmogorov-Arnold-Moser
LISA	Laser Interferometer Space Antenna
MATLAB	Matrix Laboratory
NASA	National Aeronautics and Space Administration
NOAA	National Oceanic and Atmospheric Administration
ODE	Ordinary Differential Equations
OS	Operating System
POSIX	Portable Operating System

Abstract

Spacecraft formation flying involves operating multiple spacecraft in a pre-determined geometrical shape such that the configuration yields both individual and system benefits. One example is an over-flight of the same spatial position by spacecraft in geocentric orbit with the intent to create a complementary data set of remotely sensed observables. Another example is controlling to a high degree of accuracy the distance between spacecraft in heliocentric orbit to create a virtual, large-diameter interferometer telescope. Although Keplerian orbits provide the basic framework for general and precision spacecraft formation flying they also present limitations. Spacecraft are generally constrained to operate only in circular and elliptical orbits, parabolic paths, or hyperbolic trajectories around celestial bodies. Applying continuation methods and bifurcation theory techniques to the circular, restricted three-body problem - where stable and unstable periodic orbits exist around equilibrium points - creates an environment that is more orbit rich. After surmounting a similar challenge with test particles in the circular, restricted three-vortex problem in fluid mechanics as a proof-of-concept, it was shown that spacecraft traveling in uncontrolled motion along separate and distinct planar or three-dimensional periodic orbits could be placed in controlled motion, i.e. a controller is enabled and later disabled at precisely the proper positions, to have them phase-locked on a single periodic orbit. Although it was possible to use this controller in a resonant frequency/orbit approach to establish a formation, it was clearly shown that a separate controller could be used in conjunction with the first to expedite the formation establishment process. Creation of these dynamically natural spacecraft formations or multi-spacecraft platforms will enable the ‘loiter, synchronize/coordinate, and observe’ approach for future engineering and scientific missions where flexibility is a top-level requirement and key to mission success.

Preface

The motivation for this research is the firm belief that even with collaboration among research and development centers, academia, and industry as well as partnerships between nations scarce budgets and competing interests will remain realities. Continuous innovations in science, engineering, and technology are necessary for mankind to continue to expand the frontiers of space. It is hoped that this body of work will in some way contribute to the actual formulation, development, and implementation of scientific missions that until now exist only in the imagination.

Chapter 1: Introduction

The use of words such as disorder, randomness, and chaos is commonplace these days not only in science and mathematical circles, but also in casual conversation. Are we to believe that we, the world as we know it, and the universe are doomed to a fate of absolute entropy? There is anecdotal evidence to suggest otherwise, i.e. there appears to be a tendency for order/synchronization to emerge from chaos. If we expand this to the field of astrodynamics - the study of the motion of artificial satellites, spacecraft, and rockets - we can also cite a number of examples where coordinated motion yields scientific or societal benefits. How are these types of coordinated motion implemented? Can a novel implementation approach be developed to enable new, different types of scientific missions? These questions are examined and serve as the impetus in the identification and formation/structure of the thesis described later in this chapter.

1.1 Synchronization

1.1.1 Coordinated Action

For hundreds if not thousands of years, people had been fascinated with the pulsing rhythm of fireflies, bioluminescent beetles of the Lampyridae family described by Eisner [10]. However, they could not understand the mechanism behind the spectacle, i.e. when fireflies congregated in large numbers they would flicker on and off in unison as if they were members of an orchestra following the lead of a conductor. In compiling information on the general subject of synchronization, Strogatz

[26] stated that research conducted in the early 1960s finally concluded that the fireflies, each with its own internal oscillator, would adjust themselves to the rhythm of those in close proximity – a natural feedback control loop of sorts. Strogatz goes on to describe how pacemaker cells in the heart synchronize themselves with one another rather than to rely on a single lead or master cell. This allows the overall system to be more robust and resilient to individual cell death. These are but two examples in nature where many behave as one. Since these examples involve intrinsic qualities and do not generally involve spatial translation or rotation we will simply call them instances of coordinated action.

1.1.2 Coordinated Motion

Coordinated motion is another type of synchronization that clearly results in system benefits. Fish that swim in schools and wildebeest that travel in herds are less likely to fall victim to predators, thus supporting the old adage that there is safety in numbers. Another familiar example of formation flying is a flock of Canadian Geese traveling in a V-formation. Numerous studies have shown that this type of formation allows for each of the birds behind the lead to benefit from the upwash, i.e. upward moving vortices resulting from turbulent air created by the bird in front. When the birds alternate the lead position the collective group is able to remain aloft longer than they would have if each had to fly alone. Organisms, it appears, tend to aggregate themselves relative to environmental conditions and to other similar individuals. Acebron [1] states that the Kuramoto model, where each member of a given population is represented as a phase oscillator, helps explain the latter from a more mathematical perspective. Each oscillator is described as a phase on a circle. On one side of the spectrum the oscillators are different, so that they all move at different speeds and are permanently disorganized. The order parameter value in this case is “0”. On the other side of the spectrum, where the oscillators are identical and there is no coupling or perturbation, each oscillator would move at the same speed as the others – perfect synchronization. In this case, the order parameter value is “1”.

Strogatz [26] points out that there is a critical order parameter value that must be achieved before any level of synchronization can take place. Said differently, the individuals must be similar enough for coordinated motion to emerge from what seems like chaos. As human beings, our driving desire to better understand the world in which we are part has given rise to emulating behaviors of organisms from coarse computerized simulations of flocks of birds by Reynolds [23] to detailed mathematical modeling of schools of fish by Leonard [16]. Since these cases involve spatial translation and/or rotation we will call these examples of controlled motion.

1.2 Astrodynamics and Spacecraft Formation Flying

Although Johannes Kepler developed the three fundamental laws governing the motion of the planets in the early 1600s, Hill [13] produced the definitive reference on the relative motion of celestial bodies while studying the earth-moon system in the late 1800s. He became fascinated with one of the most interesting problems of the time. He sought to understand the reason for the discrepancy between the predicted motion of the lunar perigee and actual observations. Some thought that higher-order terms in the approximation should not have been disregarded while others believed that there were other forces and torques acting on the moon that had not yet been accounted for. Although Hill assumed a circular orbit for the moon, Barrow-Green [6] states that Hill's novel approach was to introduce solar perturbations to vary the motion of the moon and then allow the motion to vary again by introducing the eccentricity of the lunar orbit. In the 1960s, Clohessy and Wilshire [7] followed up on this investigation. They determined that the motion of a second (follower) satellite relative to the first (leader) satellite could be described by a system of nonlinear differential equations. They noted that in a special case where the orbit of a lead satellite is circular, the orbit radius and the angular rate become time-invariant. Therefore, any non-linear coupling would be insignificant as long as the distance between the two satellites was much smaller than the orbit radius. One could then solve for the position of a second satellite (follower) relative to the first

(leader) through a system of linearized equations of motion known as Hill's or the Hill-Clohessy-Wiltshire (HCW) equations. In these equations, the acceleration terms are assumed to be zero. In addition, there are two limitations that must be noted. First, the orbits of both satellites must be nearly circular, and second, the orbit periods must be identical. The equations produce satisfactory results as long as the assumptions are valid and the limitations are not in violation.

1.2.1 Current Methods and Scientific Applications

The aerospace community has come to realize the significance and value in multi-spacecraft operations. The concept of distributing the functionality traditionally found on a single spacecraft to two or more operating in a coordinated manner yields advantageous system benefits. Gurfil and Kasdin [11] state that this model allows for lower life-cycle costs, enhanced performance, the flexibility to adapt to changing mission objectives and operational conditions, and improved fault tolerance. Atkins and Penneçot, [3] state that formation flying may increase data coverage and Tollefson [28] states that reductions in overall launch costs could be realized. Based on the earlier work of Hill, Clohessy, and Wiltshire on the relative motion of satellites, research and development centers, academia, and industry over the past forty years have devoted much time and directed resources to the general subject of spacecraft formations. These are some specific instances:

- The Jet Propulsion Laboratory, California Institute of Technology, under contract with NASA (National Aeronautics and Space Administration) has responsibility for several instruments and spacecraft that are currently or are planned to be a part of the A-Train Constellation. Basilio et al. [4] describe the concept for general formation flying in this specific case as over-flights of the same spatial position by multiple spacecraft in geocentric orbit with the intent to create a complementary data set of remotely sensed observables. Boain [5] states that frequent, periodic maneuvers are required to maintain the spacecraft formation, since environmental forces and torques such as solar pressure,

atmospheric drag, and the planet's equatorial bulge (that produces an out-of-plane force causing a gyroscopic orbit precession) complicate any attempt at precision spacecraft formation flying. Active control systems would be needed to sense and compensate for these external effects.

- The Intelligent Servosystems Laboratory, University of Maryland, has recently completed more specific work on the subject of spacecraft formation flying from dynamics to control laws. Zhang and Krishnaprasad [31] developed/obtained control laws by introducing a Lyapunov function in the space developed for a system of two spacecraft operating in geocentric orbit. The desired formation is achieved asymptotically by the controlled dynamics of the each spacecraft.
- The School of Aeronautics and Astronautics, Purdue University has also been involved in the spacecraft formation flight. Marchand and Howell [17] have completed an investigation where a transition from the three-body problem to the more general, complete n -body ephemeris, so that environmental forces and torques such as gravitational perturbations and solar radiation pressure can be accounted for.
- The joint ESA (European Space Administration) and NASA LISA (Laser Interferometer Space Antenna) concept mission [9] is one of the most pertinent investigations to date in regards to dynamically natural formations. Unlike previous space missions where gravity wave detection was limited in scope and duration, the primary objectives of the five-year LISA mission will be to detect gravity waves generated by binary stars in the Milky Way Galaxy and large black holes in distant galaxies. The space antenna consists of three spacecraft flying in an equilateral triangle formation with a separation distance of five million kilometers or about 13 times the distance from the earth to the moon. Each spacecraft will carry a payload of sensitive instruments to measure changes in distance between free-floating test masses. After launch, it will take approximately thirteen months for each spacecraft to reach its operational heliocentric orbit and be placed in its requisite position to create the desired formation. Once the formation has

been established, the main on-board propulsion system on each spacecraft will be jettisoned preventing any active control of the formation for the next five years. Only micronewton thrusters will be used to compensate for small disturbance forces and torques to keep the test masses floating freely in space.

- The School of Aeronautics and Astronautics, Stanford University was involved in the Orion mission. Robertson et al., [24] state that the results of the spacecraft formation flying development mission will eventually help to enable controlling to a high degree of accuracy the distance between spacecraft in heliocentric orbit to create a virtual, large-diameter, interferometer telescope. This special case is called precision spacecraft formation flying.

1.2.2 Future Possibilities and the Thesis

In all but one of the examples described in the previous section, spacecraft formation flying is performed in the basic framework of Keplerian orbits - circular or elliptical orbits, parabolic paths, or hyperbolic trajectories. Although multi-body physics was being used as a backdrop for the investigation of spacecraft formation flying at the School of Aeronautics and Astronautics, Purdue University, the work performed by both the Control and Dynamical Systems Division and Applied & Computation Mathematics Department, California Institute of Technology made it possible to explore a novel approach to spacecraft formation flying. Firstly, Koon et al. [15] performed a study of dynamical systems and its actual application to single-spacecraft mission design (e.g. Genesis) that is based on the circular, restricted three-body problem that Henri Poincaré first investigated in the late 1880s. This served to define the fundamental equations of motion in this realm. Secondly, Paffenroth, et al. [25], through the use continuation and bifurcation methods, were able to densely foliate periodic orbits about the equilibrium points creating an environment that is more orbit rich than that defined by Kepler's laws. The next logical step was to take what was learned from both studies

to demonstrate the feasibility of spacecraft formation flying in regions that were previously overlooked, i.e. near equilibrium points. Three assertions are made. First, spacecraft traveling along separate and distinct planar or three-dimensional periodic orbits can be placed in controlled motion, i.e. a controller is enabled and later disabled at precisely the proper positions, to have them phase-locked on a single periodic orbit. Second, when used in a resonant frequency/orbit approach the controller can be used to establish a formation. Finally, a separate controller can be used in conjunction with the first to expedite the formation establishment process. Creation of these dynamically natural formations or multi-spacecraft platforms will enable the ‘loiter, synchronize/coordinate, and observe’ approach for future engineering and scientific missions where flexibility is a top-level requirement and key to mission success.

1.3 The Structure of the Ph.D. Dissertation

Chapter 2 provides background information on the two-body problem and the specialized case of central force motion that Johannes Kepler studied as well as the results of initial computer simulations that proved useful for subsequent modeling and analysis. Chapter 3 is the proof-of-concept for the eventual solution. The simulation and analyses of test particle motion in the circular, restricted three-vortex problem provided the basic framework for developing the methodology necessary to solve the more complex celestial mechanics problems. Techniques and lessons learned were directly carried over to Chapter 4 where spacecraft phase-locking and formation establishment in the circular, restricted three-body problem for both planar and three-dimensional cases were examined. An evaluation of the solution techniques, general lessons learned, and recommendations for future work are documented in Chapter 5. Conclusions are given in Chapter 6. Finally, the appendices contain general information on the computer applications used, software source code, background material on general stability of periodic orbits, as well as several sets of briefing charts on the topic of dynamically natural spacecraft formations.

Chapter 2: Two-Body Systems, Modeling, and Analysis

Background information on the two-body problem and the specialized case of central force motion that Johannes Kepler studied is described in this chapter. The results of initial computer simulations that proved useful for subsequent modeling and analysis are provided as well.

2.1 Two-Body Systems

2.1.1 General Two-Body Problem

The two-body problem is a classical problem involving the use of Newton's law of universal gravitation. Danby [8] considers it the fundamental problem in celestial mechanics. To a first-order, the motion of the planets about the sun exemplifies the situation in which two bodies move only under mutual gravitational attraction. In reality, perturbations - caused by the force of attraction of a third body, for example - may alter the motion of one or both of the bodies, but are not considered a significant factor here. Consider two bodies of uniform mass, i.e. the center-of-mass of each is located at the geometrical center of each body, located in inertial space as shown in Figure 2.1. From Newton's law of universal gravitation, we know the following:

$$\mathbf{F} = m\mathbf{a} = -\frac{Gm_1m_2}{r^2} \cdot \frac{\mathbf{r}}{r} \quad 2.1$$

Here, G is the universal constant of gravitation, m_1 is the mass of the first body, m_2 is the mass of the second body, and r is the magnitude of the vector \mathbf{r} . The force acting on m_1 and m_2 are shown respectively as,

$$\mathbf{F}_1 = m_1 \mathbf{a}_1 = -\frac{Gm_1 m_2}{r^2} \cdot \frac{(\mathbf{r}_1 - \mathbf{r}_2)}{r} \quad 2.2$$

$$\mathbf{F}_2 = m_2 \mathbf{a}_2 = -\frac{Gm_1 m_2}{r^2} \cdot \frac{(\mathbf{r}_2 - \mathbf{r}_1)}{r} \quad 2.3$$

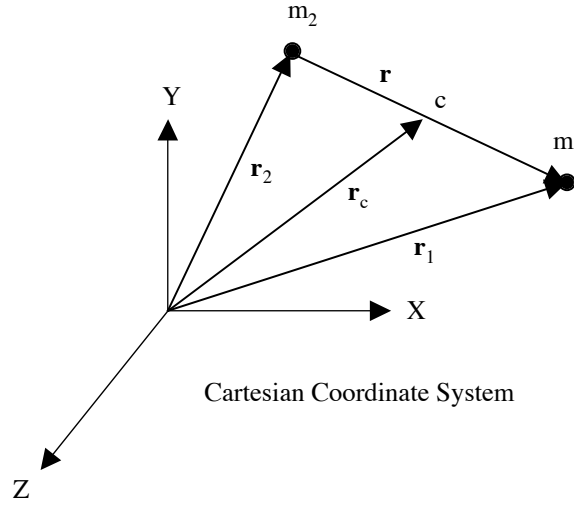


Figure 2.1. This is a sketch of a two-body system located in three-dimensional physical space.

Now sum the forces acting on the two-body system.

$$\mathbf{F}_1 + \mathbf{F}_2 = m_1 \mathbf{a}_1 + m_2 \mathbf{a}_2 = m_1 \ddot{\mathbf{r}}_1 + m_2 \ddot{\mathbf{r}}_2 = \left(-\frac{Gm_1 m_2}{r^2} \cdot \frac{\mathbf{r}_1 - \mathbf{r}_2}{r}\right) + \left(-\frac{Gm_1 m_2}{r^2} \cdot \frac{\mathbf{r}_2 - \mathbf{r}_1}{r}\right) \quad 2.4$$

This leads to:

$$m_1 \ddot{\mathbf{r}}_1 + m_2 \ddot{\mathbf{r}}_2 = 0 \quad 2.5$$

Both $\ddot{\mathbf{r}}_1$ and $\ddot{\mathbf{r}}_2$ must be equal to zero for this to be true. One can only conclude that $\ddot{\mathbf{r}}_1$ and $\ddot{\mathbf{r}}_2$ is of the form,

$$\mathbf{r} = c_1 t + c_2 \quad 2.6$$

Here, c_1 and c_2 are constant coefficients. Since the end point of \mathbf{r}_c is along the line formed by the end points of \mathbf{r}_1 and \mathbf{r}_2 , \mathbf{r}_c must also be of the form given in equation 2.6. Therefore, $\ddot{\mathbf{r}}_c$ is also equal to zero. This result is important, since it means that the center-of-mass of the system, c , does not accelerate and plays an important part in development of the governing equation of motion. Let's return to Figure 2.1. Through inspection, one can see that the following relationships are true:

$$m_1(\mathbf{r}_1 - \mathbf{r}_c) + m_2(\mathbf{r}_2 - \mathbf{r}_c) = 0 \quad 2.7$$

$$\mathbf{r}_2 = \mathbf{r}_1 - \mathbf{r} \quad 2.8$$

$$\mathbf{r}_1 - \mathbf{r}_c = \left(\frac{m_2}{m_1 + m_2}\right)\mathbf{r} \quad 2.9$$

$$\mathbf{r}_2 - \mathbf{r}_c = \left(-\frac{m_1}{m_1 + m_2}\right)\mathbf{r} \quad 2.10$$

Equating the forces acting on the two bodies results in the following:

$$\mathbf{F}_1 = -\mathbf{F}_2 = -m_2\ddot{\mathbf{r}}_2 = -m_2\ddot{\mathbf{r}}_c + \frac{m_1 m_2}{(m_1 + m_2)}\ddot{\mathbf{r}} \quad 2.11$$

Since $\ddot{\mathbf{r}}_c = 0$,

$$\mathbf{F}_1 = \frac{m_1 m_2}{(m_1 + m_2)}\ddot{\mathbf{r}} \quad 2.12$$

or

$$\frac{Gm_1m_2}{r^2} \cdot \frac{\mathbf{r}}{r} = \frac{m_1m_2}{(m_1+m_2)} \ddot{\mathbf{r}} \quad 2.13$$

Through simplification the following result is obtained:

$$\frac{G(m_1+m_2)}{r^2} \cdot \frac{\mathbf{r}}{r} = \ddot{\mathbf{r}} \quad 2.14$$

or

$$\ddot{\mathbf{r}} + \frac{\mu}{r^3} \mathbf{r} = 0 \quad 2.15$$

where $\mu = G(m_1+m_2)$.

2.1.2 Central Force Motion

The result in equation 2.15 is a second-order, non-linear ordinary differential equation. The equation describes the motion of m_1 about m_2 and vice-versa. However, there is a special, simplified case where one of the masses orbits the other while the other remains fixed in inertial space. This is the central-force motion problem that Johannes Kepler studied and developed three laws of celestial mechanics for in the early 1600s. The first of these is described as follows: planets move in elliptical orbits with the sun at one focus. The orbital elements that help to define/describe the orbit of m_1 about the fixed mass, m_2 , are as follows:

- Semi-major axis, a : A constant that defines the largest dimension of the orbit
- Inclination, i : The angle between the vector normal to the orbit plane and the vector normal to the plane of interest
- Eccentricity, e : A constant that describes the shape of the orbit
- Longitude of the ascending node, Ω : The angle at which the orbit plane crosses the plane of interest

- Argument of periapsis, ω : The angle measured from the longitude of the ascending node to the periapsis as measured on the orbit plane
- Time of periapsis passage, T : The time when the orbiting object was at periapsis

Of the six orbital elements it is the orbit eccentricity, e , which defines the only possible paths for m_1 and m_2 in a two-body problem. The unit-less magnitude of the eccentricity vector, e , is important, since it defines the orbit shape. How the magnitude and orbit shapes are related is shown in Table 2.1 below.

Eccentricity, e	Orbit Shape
0	Circle
$0 < e < 1$	Ellipse
1	Parabola
$e > 1$	Hyperbola

Table 2.1. Relationships between the magnitude of eccentricity, e , and the shape of the orbit are shown in this table.

It is interesting to note that the orbit shapes are all conic sections, i.e. each orbit can be defined as the intersection of a plane with a right circular cone. These conic sections then create a family of curves. What has just been described is the first of Kepler's Laws. This forms the basis for the other two, both of which are only briefly described below.

Kepler's second law states that in a central-force motion problem, where m_1 orbits m_2 and m_2 remains fixed in inertial space, the line joining the two masses sweeps out equal areas in equal time. For this to be true, the velocity of m_1 must be inversely proportional to its distance from m_2 or,

$$v_{m_1} \propto 1/d_{(m_1-m_2)} \quad 2.16$$

Finally, Kepler's third law states that the square of the orbit period for m_1 is proportional to the cube of the semi-major axis,

$$P^2 \propto a^3 \quad 2.17$$

2.2 Computer Simulations and General Analysis

Equation 2.15, the second-order general equation of motion for the two-body case, was phrased as a system of first-order differential equations and coded into a function file, `two_body_func.m`, for use in MATLAB [see Appendix A]. A MATLAB script initially using a standard ODE (Ordinary Differential Equation) solver, i.e. `two_body_init_v5.m`, was then written that called this function file. Both of these files are included in Appendix B. One of the plots created by executing these files is shown in Figure 2.2 below.

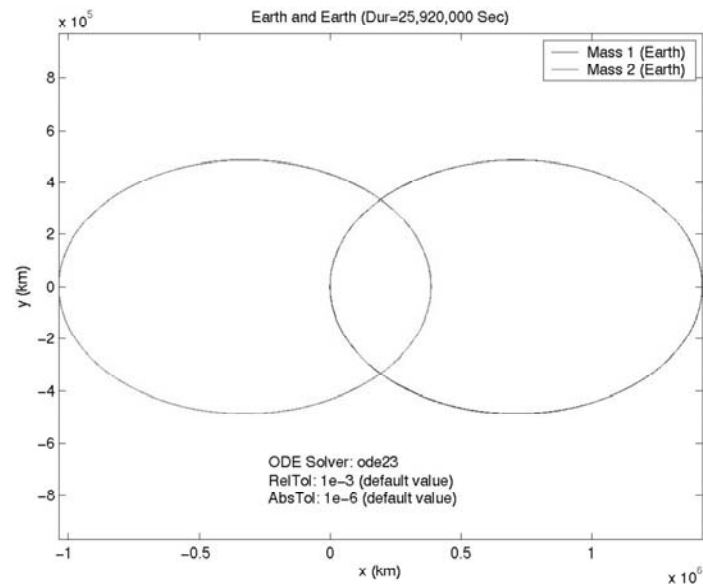


Figure 2.2. This is a MATLAB plot of orbit traces for a two-body system where the two masses are identical (mass 1 = mass 2 = Earth mass).

In this particular case, the two bodies are of identical mass. Therefore, the orbit traces should be identical, and the inertial speed of the barycenter should be zero. The former can easily be seen in this figure and the latter can be seen in Figure 2.3. The speed of the barycenter is less than 1 mm/sec. For a two-body system where each mass is equal to the mass of the Earth, this is essentially 0. It was also possible to do some investigation of ODE solvers. Specifically, we examined the 2nd and 3rd, 4th and

5th, and 7th and 8th-order Runge-Kutta techniques as well as relative and absolute tolerance levels. Figure 2.4 shows an orbit trace using the 2nd and 3rd-order Runge Kutta technique (a.k.a. ode23), and the default values for relative (reltol) and absolute (abstol) error tolerance of 1.0E-03 and 1.0E-06, respectively. At first, it looks like the orbit trace shows good accuracy.

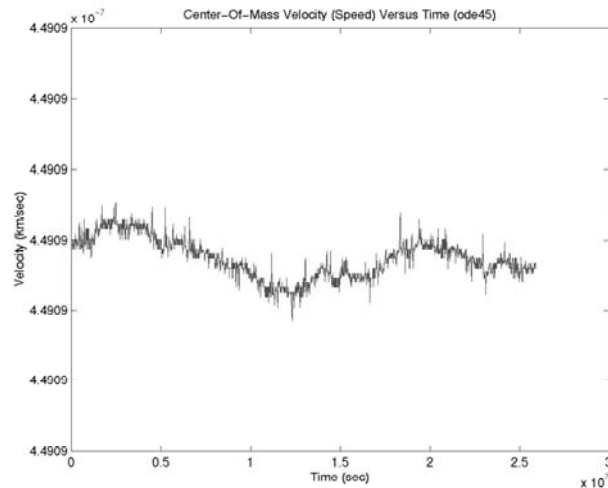


Figure 2.3. This is a MATLAB plot of the speed of the barycenter over time for a two-body system where the masses are identical (mass 1 = mass 2 = Earth mass).

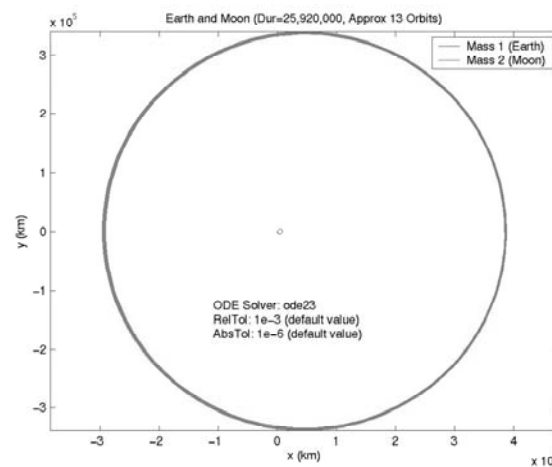


Figure 2.4. This is a MATLAB plot of an orbit trace using ode23 with default values for reltol and abstol of 1.0E-03 and 1.0E-06, respectively.

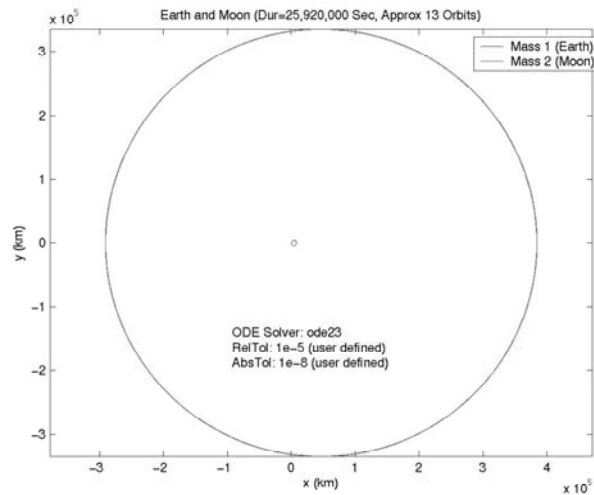


Figure 2.5. This is a MATLAB plot of an orbit trace using ode23 with user-defined values for reltol and abstol of 1.0E-05 and 1.0E-08, respectively.

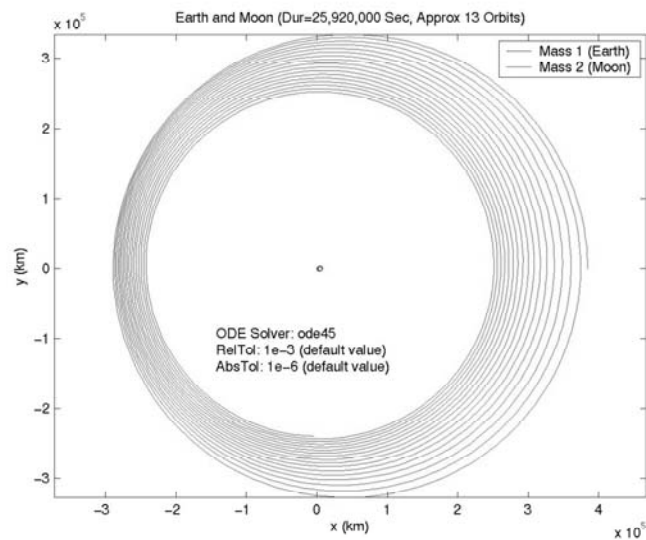


Figure 2.6. This is a MATLAB plot of an orbit trace using ode45 with default values for reltol and abstol of 1.0E-03 and 1.0E-06, respectively.

However, when reltol and abstol are changed to 1.0E-05 and 1.0E-08, respectively, one can see in Figure 2.5 there is improved accuracy. This is indicated by the “thinner” orbit trace/line.

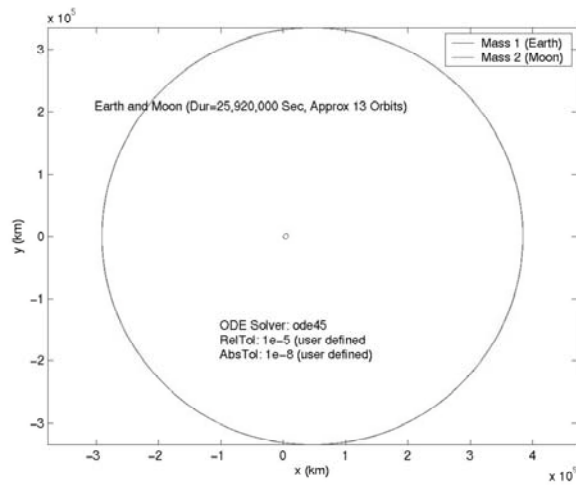


Figure 2.7. This is a MATLAB plot of an orbit trace using ode45 with user-defined values for reltol and abstol of 1.0E-05 and 1.0E-08, respectively.

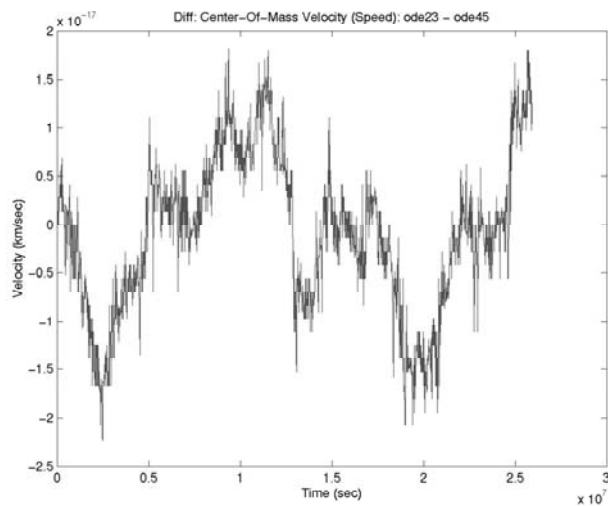


Figure 2.8. This is a MATLAB plot of differences in barycenter speed using default and user-defined values for reltol and abstol.

How the values for reltol and abstol affect the accuracy of the solution or orbit trace is more clearly evident in Figures 2.6 and 2.7. Figure 2.6 shows an orbit trace using the 4th and 5th-order Runge Kutta technique (a.k.a. ode45), and the default values for reltol and abstol of 1.0E-03 and 1.0E-06, respectively. Here it is clearly evident that the solution does not show good accuracy, because of the multiple orbit traces. However, when reltol and abstol are changed to 1.0E-05 and 1.0E-08, respectively, one can see in Figure 4.7 much better accuracy. This is indicated by the single orbit trace/line. It is interesting to note that although reltol and abstol have a profound effect on the orbit trace, in actuality the position components of the state vector for each mass, it appears that the default values are sufficient for other attributes. Taking a look at Figure 2.8, it is clearly evident that the difference in the barycenter center speed calculations using default and user defined values for reltol and abstol are insignificant, i.e. the differences are at least ten orders of magnitude less than the values themselves. Finally, potential, kinetic, and total energy for a two-body system were calculated. It can be seen in Figure 2.9 that the total energy of the system is a conserved quantity, which is a requisite for any Hamiltonian system.

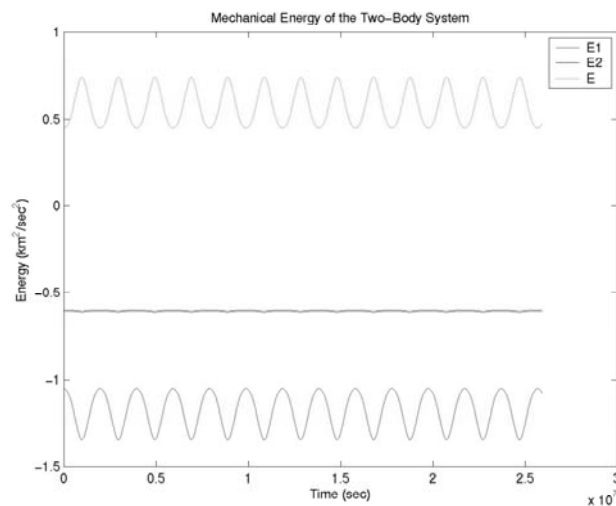


Figure 2.9. This is a MATLAB plot of potential, kinetic, and total energy of a two-body system.

An examination of the simple two-body problem served as the initial ‘spade work’ in understanding the accuracy of MATLAB numerical solution techniques and the affects of the relative and absolute tolerance values used in the calculations. A decision was made to forgo the 2nd & 3rd and 4th & 5th-order approach in favor of the 7th & 8th-order Runge-Kutta method for even better accuracy. A pre-existing MATLAB script, ode78.m, was used as function call for the two-body case as well as the circular, restricted three-body and circular, restricted three-vortex problems described in subsequent chapters [see appendix B]. In addition, reltol and abstol were changed to 1.0E-05 and 1.0E-08, respectively.

Chapter 3: The Circular, Restricted Three-Vortex Problem

The three-body problem involves investigating the behavior or motion of three mutually attracting bodies. The circular, restricted three-body problem is a specialized case where one of the three bodies is of negligible mass and does not influence the behavior of the other two bodies. However, the two primary bodies directly drive its motion. Since the late nineteenth century considerable research has been conducted on the circular, restricted three-body problem in the field of celestial mechanics. In order to solve this complex problem, the circular, restricted three-vortex problem in fluid mechanics was studied as a proof-of-concept. Specifically, can a number of test particles traveling in different periodic orbits of this realm be controlled such that they can be placed in the same (equal period) orbit? Additionally, can the same or a different controller be used to fix the relative positions of the test particles such that a virtual or dynamically natural formation can be established? It will be shown that the answer to both questions is, ‘yes’.

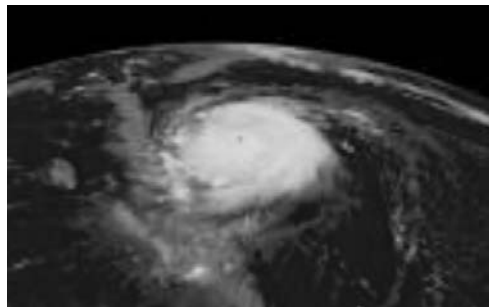


Figure 3.1. A hurricane churning the air and water around the Atlantic Ocean, but headed in a northwesterly direction to the warm waters of the Gulf of Mexico. [Credit: National Oceanic and Atmospheric Administration (NOAA)]

The 2005 Atlantic hurricane season was one for the record books. There were at least twenty-four named storms in a season that resulted in much devastation and loss of life in the states bordering the Gulf of Mexico (see Figure 3.1). A hurricane is a tropical cyclone or rotary circulation with sustained wind speeds. We will call the hurricane a vortex. Motion of an object within or in close proximity to a hurricane is determined by its distance from and the strength of the vortex. Take for example a soccer ball dropped from an aircraft traveling at a high altitude. If the aircraft is 1,000 miles away from the vortex center, the motion of the soccer ball will not be influenced by it at all. If it is dropped 100 miles from the vortex center, the rotating air will most certainly influence the motion of the soccer ball, i.e. transfer horizontal energy or motion. However, the ball will have negligible influence on the motion of the vortex itself. We will call the relatively mass-less soccer ball a test particle. Since dealing with vortices on sphere can be more complicated than we really desire for this study, we will make some simplifying assumptions. The first of which is to assume only planar motion. See Figure 3.2 below.

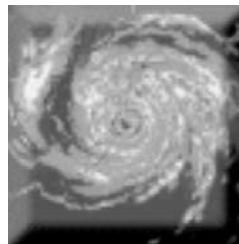


Figure 3.2. A vortex moving only in a flat, two-space surface (real component and imaginary component of a complex number) helps to simplify the problem. [Credit: National Oceanic and Atmospheric Administration (NOAA)]

Now we will place two vortices of equal strength, which we will call the primaries, on a complex Cartesian coordinate frame with one axis defined as the real and the other the imaginary axis. In order to fix the locations of both primaries along the real axis, we will allow the coordinate frame to rotate about the origin at the same rate of rotation that each primary will experience around the center of

vorticity. The test particle defined earlier will actually be the third vortex. Being of negligible mass and insignificant vortex strength, the test particle identified above meets the definition for the third vortex in the circular, restricted three-vortex problem, i.e. it's motion is influenced/driven by the primaries, but itself exerts no influence on the primaries. Figure 3.3 is a simple schematic of the circular, restricted three-vortex problem.

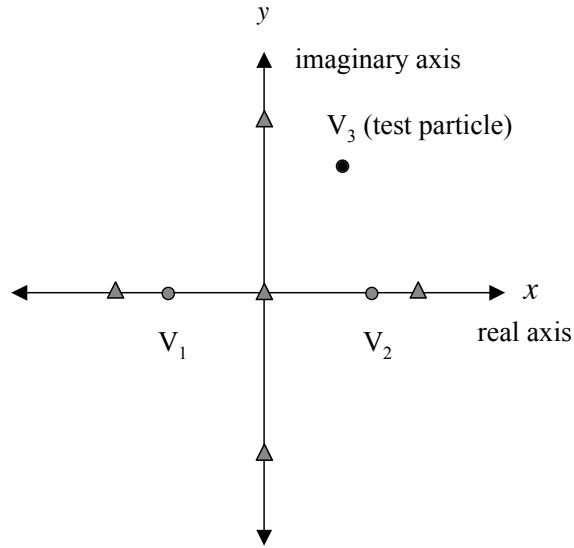


Figure 3.3. The primaries, V_1 and V_2 , lie on the real axis of the rotating, complex Cartesian coordinate frame. The third vortex, V_3 , will be referred to as the test particle. The triangles represent stable and unstable equilibrium points associated with this problem.

3.1 Equations of Motion

Newton [19] states that contrary to solving the circular, restricted three-body problem in celestial mechanics, solving the circular, restricted three-vortex problem in fluid mechanics is more straightforward. He gives the equation of motion for the third vortex of negligible strength as:

$$\dot{\xi}_3 = i\omega\xi_3 + \frac{i\Gamma_1}{2\pi} \cdot \frac{(\xi_3 - \xi_1)}{|\xi_3 - \xi_1|^2} + \frac{i\Gamma_2}{2\pi} \cdot \frac{(\xi_3 - \xi_2)}{|\xi_3 - \xi_2|^2} \quad 3.1$$

In equation 3.1, there is a special relationship between several of the parameters. This is shown in equation 3.2 below.

$$\Gamma_1 \xi_1 + \Gamma_2 \xi_2 = 0 \quad 3.2$$

In equations 3.1 and 3.2 as appropriate, ω is the orbit frequency of each primary about the center of vorticity; Γ_1 and Γ_2 are the vortex strengths of the two primaries; and ξ_1 , ξ_2 , and ξ_3 are each the vortex position states fixed in the rotating, complex Cartesian coordinate frame. Similar to what is typically done with the circular, restricted three-body problem, normalizing by choosing appropriate values for units of length and time simplifies the problem. Let the sum of the primary vortex strengths and the absolute value of the difference between the two primary position states be equal to 2π and 1, respectively. Then choose the primaries to lie on the real axis of the rotating, complex Cartesian coordinate frame. The system can be further simplified by defining the relationships in equations 3.3 through 3.5 below.

$$\xi_1 = -\lambda \quad 3.3$$

$$\xi_2 = 1 - \lambda \quad 3.4$$

$$\Gamma_2 = 2\pi\lambda \quad 3.5$$

The position state for ξ_3 is given as:

$$\xi_3 \equiv u + vi \quad 3.6$$

The Hamiltonian for this fully conserved system is given as:

$$H(u, v) = -\frac{1}{2}(u^2 + v^2) + (1 - \lambda) \log(\sqrt{(u + \lambda)^2 + v^2}) + \lambda \log(\sqrt{(u + \lambda - 1)^2 + v^2}) \quad 3.7$$

For expediency the MATLAB application was used to simulate the circular, restricted three-vortex problem. As with the general two-body problem the 7th-8th order Runge-Kutta technique was used to solve the ordinary differential equation given in equation 3.1 and the Hamiltonian given in equation

3.7. The MATLAB script, `main_script_v5_1.m`, and the function file, `three_vortex.m` are both provided in Appendix B.

3.2 Equilibrium Points

Equilibrium points in the circular, restricted three-vortex problem in fluid mechanics can be found using the same approach for the circular, restricted three-body problem in celestial mechanics. In addition, the `find_libration_point.m` MATLAB script can be employed to expedite problem solving. Both are described in detail in Chapter 4. In the case where the two vortices are of equal strength the equilibrium point locations are those shown in Table 3.1.

Equilibrium Point	X Coordinate	Y Coordinate
L1	0	0
L2	1.1180	0
L3	-1.1180	0
L4	0	0.8660
L5	0	-0.8660

Table 3.1. The coordinates for the five equilibrium points in rotating Cartesian coordinate frame for the case where the two primaries are of equal strength are given in this table. Note “L” stands for “libration”, which is synonymous with “equilibrium” point.

3.3 Level Curves of the Hamiltonian

Figure 3.4 shows a number of level curves of the Hamiltonian in the rotating, complex Cartesian coordinate frame for the symmetric case of the circular, restricted three-vortex problem. A test particle in this realm would traverse along a line of constant energy in the absence of external forces or torques. We will call this uncontrolled motion. We will concentrate on the area near the primary vortex located at $(0.5, 0.0i)$, denoted as V_2 , and call the level curves of the Hamiltonian that surround this vortex, periodic orbits. It is interesting to note that test particles on these periodic orbits

– orbits that resemble, but are not technically ellipses - traverse in a counterclockwise direction in the complex, rotating Cartesian coordinate frame.

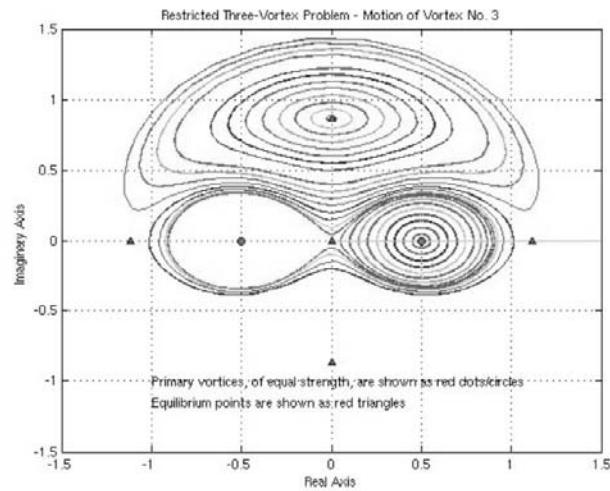


Figure 3.4. Level curves of the Hamiltonian surrounding one of the two primary vortices and one of the five equilibrium points for the symmetric case of the circular, restricted three-vortex problem are shown in this plot. The two equal primaries, shown as circular markers, are located at $(-0.5, 0.0i)$ and $(0.5, 0.0i)$, and the equilibrium points of the system are shown as triangles.

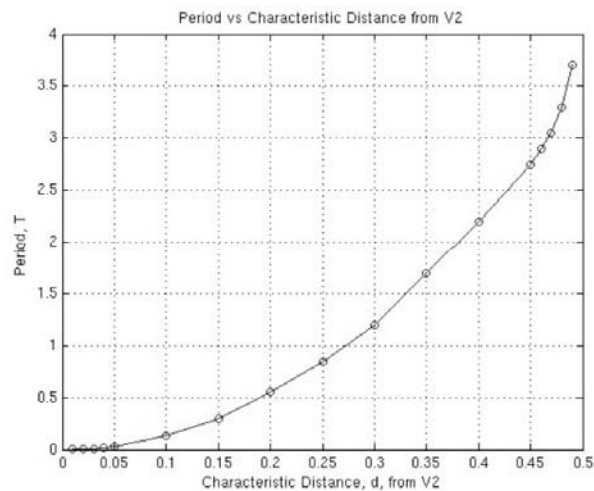


Figure 3.5. This is a plot of characteristic distance versus orbit period. Note that as characteristic distance, d , increases, so does the orbit period, T .

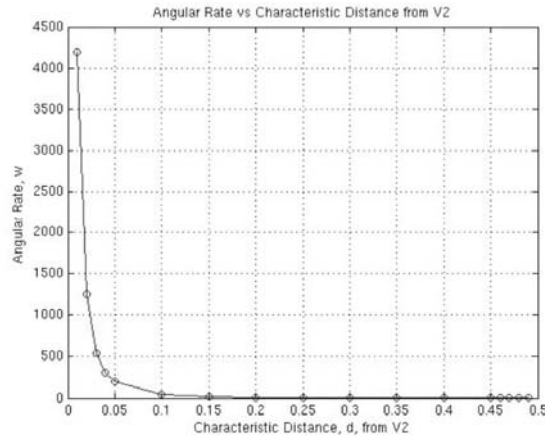


Figure 3.6. This is a plot of characteristic distance versus angular rate. Note that as characteristic distance, d , increases, the angular rate decreases.

We define characteristic distance, d , as the distance from the primary vortex at $(0.5, 0.0i)$ along the real axis in the negative direction to the point at which the periodic orbit crosses the real axis (refer to Figure 3.4). Plots of characteristic distance versus orbit period, T , and angular rate, r , are shown in Figure 3.5 and 3.6, respectively. One can easily see in Figure 3.5 that the orbit period increases as the characteristic distance increases, i.e. proportional relationships of one another.

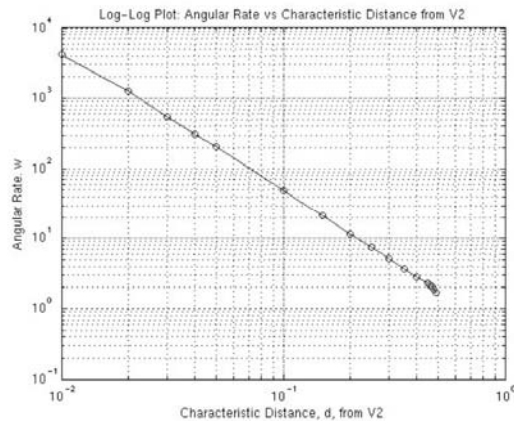


Figure 3.7. This is a plot of characteristic distance versus angular rate using the log-log scale. Note the linear curve in this scale that leads to a power law relationship.

This is similar to what is seen with two-body approximations, i.e. Kepler's third law where the square of the planetary orbit period is proportional to the cube of its semi-major axis. The angular rate used in producing Figure 3.6 is inversely proportional to the orbit period or simply $2\pi/T$. As expected, the angular rate decreases as the characteristic distance increases. A closer look at Figure 3.6 leads one to believe that there could be a special relationship between characteristic distance and angular rate. Plotting the same points, but now on a log-log scale produces the plot shown in Figure 3.7. One can easily see that this produces a linear curve, and therefore, compiles with the power law. It is relatively easy to show that the resulting relationship or empirical equation is:

$$r = 0.42d^{-2.034} \quad 3.8$$

Figure 3.8 is a plot of characteristic distance versus energy level or the value of the Hamiltonian, H . Note that in general, the value of the Hamiltonian increases as the characteristic distance increases.

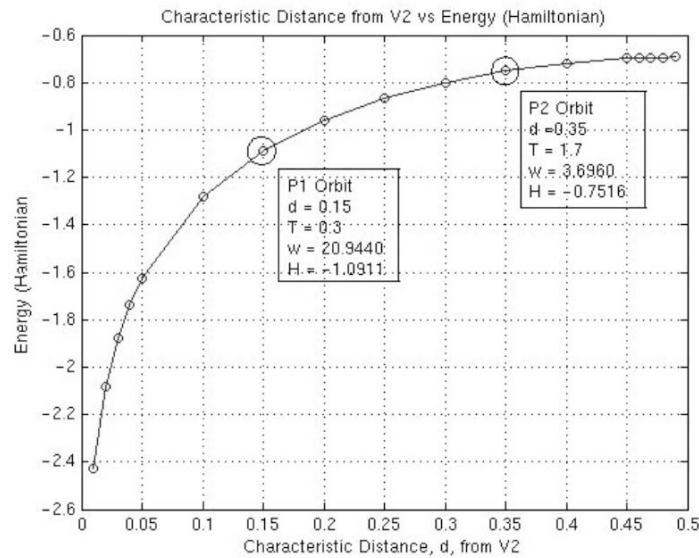


Figure 3.8. This is a plot of characteristic distance versus energy level, i.e. value of the Hamiltonian. Note that the two are proportional to one another. The energy levels for two periodic orbits, P_1 and P_2 , are singled out, since they were the focus of immediate attention.

A fifth order polynomial curve-fit produces a good approximation, i.e. $R^2 = 0.996$. This relationship, $d = f(H)$ rather than $H = f(d)$ is given as:

$$d = 0.659H^5 + 5.4193H^4 + 17.392H^3 + 27.307H^2 + 21.183H + 6.6785 \quad 3.9$$

Shown in Figure 3.9 are the attributes associated with two different periodic orbits, P_1 and P_2 . For convenience, this information is also shown in Table 3.2. These two periodic orbits formed the foundation for the investigation process, specifically, what controller(s) can be used to phase-lock a number of test particles and later establish a relative, virtual formation.

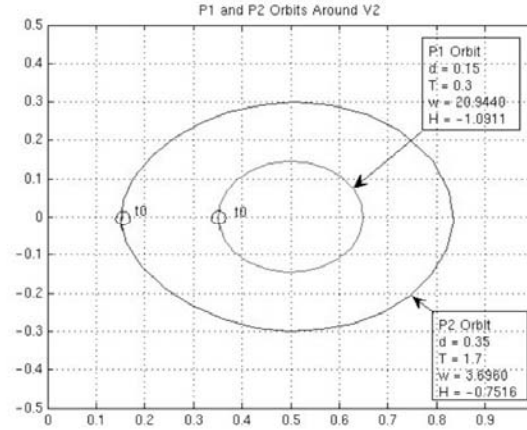


Figure 3.9. This is a plot of two periodic orbits that formed the foundation for the controller study, P_1 and P_2 . Test particles, not under the influence of external forces or torques, would traverse along each of the two orbits in a counterclockwise direction in the complex, rotating Cartesian coordinate frame.

Attribute	P_1 Orbit	P_2 Orbit
Characteristic Dist, d	0.15	0.35
Orbit Period, T	0.3	1.7
Angular Rate, r (or ω)	20.9440	3.6960
Hamiltonian, H	-1.0911	-0.7516

Table 3.2. Attributes for two periodic orbits, P_1 and P_2 , surrounding the second primary vortex, V_2 , are shown in this table for convenience.

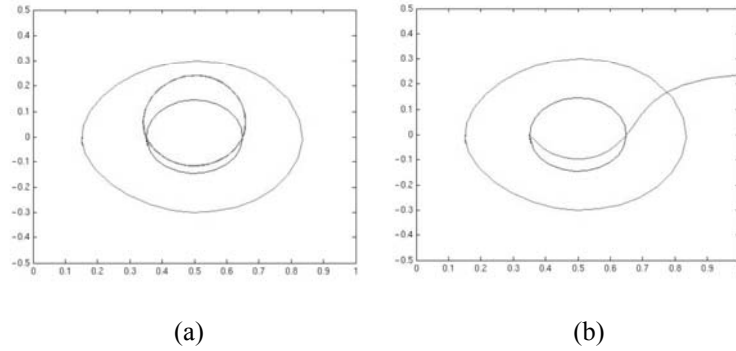
3.4 Phase-Lock Controller

The basic premise was to find a new term to add to the fundamental equation of motion given in equation 3.1. We call this new term a controller. Three different types of controllers were examined: (1) time independent, (2) time dependent, and (3) one where a trigonometric function is introduced, specifically, to produce a sinusoidal variation. A test particle influenced/driven by a controller is said to be in controlled motion.

Time-independent term – This is simply the addition of a scale factor term, κ , to equation 3.1 as shown below:

$$\dot{\xi}_3 = i\omega\xi_3 + \frac{i\Gamma_1}{2\pi} \cdot \frac{(\xi_3 - \xi_1)}{|\xi_3 - \xi_1|^2} + \frac{i\Gamma_2}{2\pi} \cdot \frac{(\xi_3 - \xi_2)}{|\xi_3 - \xi_2|^2} + \kappa \quad 3.10$$

Refer to Figure 3.10a. If a test particle is placed on the inner orbit resembling an ellipse it will traverse the periodic orbit in an uncontrolled motion state in the counterclockwise direction.



Figures 3.10a-3.10b. These are plots of two periodic orbits that formed the foundation for the controller study, P_1 and P_2 , the outer and inner periodic orbits resembling ellipses, respectively. A test particle originally on the P_2 orbit would be placed on the periodic orbit resembling a circle under controlled motion, i.e. under the influence of a time independent controller. The value of κ is 1.0 and 2.0 in (a) and (b), respectively. In both cases, the initial position for the test particle was (0.35, 0.0i).

If we let $\kappa = 1.0$ when the test particle is at $(0.35, 0.0i)$, the particle will now be in a controlled state and will follow the new periodic orbit. The orbit period is approximately 0.3 units of time, and this orbit is offset in the positive imaginary axis direction and more closely resembles a circle rather than an ellipse. Let's now let $\kappa = 2.0$, when the particle is again at $(0.35, 0.0i)$. The test particle will again be in a controlled state, but will follow a different trajectory as shown in Figure 3.10b. It seems obvious that the test particle would no longer be in a simple periodic orbit around V_2 , but is this absolutely true? If we propagate the motion of the test particle $\gg 0.3$ time units and change the scale of the complex, rotating coordinate frame, we get the plot shown in Figure 3.11.

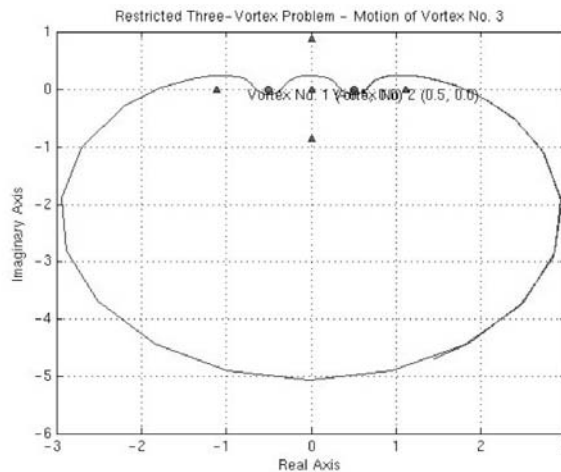


Figure 3.11. This is the same plot shown in Figure 3.10b with two changes: (1) test particle motion is propagated over a much longer period of time, and (2) the view of the rotating complex, Cartesian coordinate frame has been expanded.

One can now readily see that the test particle still traverses along a periodic orbit, albeit one with a large period and one that is oddly shaped. However, this is not necessarily the most important observation. If we return to Figure 3.10b, one can plainly see that the controlled motion trajectory crosses both the inner and outer ellipse-like orbits. The interesting insight is that placing a test particle in a controlled motion orbit/trajectory will allow it to move from one uncontrolled motion

orbit to another enabling phase-locking. This will be discussed further. However, simulations where $\kappa = 2.0, 3.0, \dots, 5.0$ produced results similar to that shown in Figures 3.10b and 3.11.

Time-dependent term – This is simply the addition of a product term, κt , to equation 3.1 as shown below:

$$\dot{\xi}_3 = i\omega\xi_3 + \frac{i\Gamma_1}{2\pi} \cdot \frac{(\xi_3 - \xi_1)}{|\xi_3 - \xi_1|^2} + \frac{i\Gamma_2}{2\pi} \cdot \frac{(\xi_3 - \xi_2)}{|\xi_3 - \xi_2|^2} + \kappa t \quad 3.11$$

Again, κ is a scale factor, and t is time. Refer to Figure 3.12. If a test particle is placed on the inner orbit, it will traverse the periodic orbit in an uncontrolled motion state in the counter-clockwise direction. If we now let $\kappa = 1.0$ when the test particle is at $(0.5, 0.0i)$, the particle will now be in a controlled state and will follow a new trajectory.

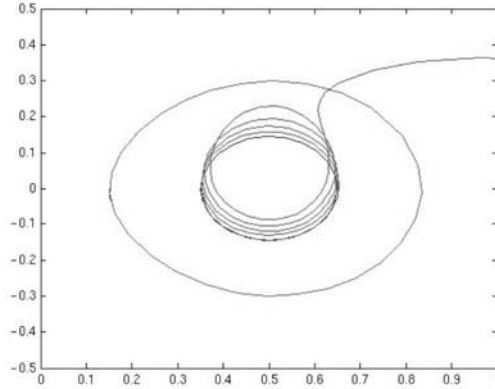


Figure 3.12. This is a plot of two periodic orbits that formed the foundation for the controller study, P_1 and P_2 , the outer and inner periodic orbits resembling ellipses, respectively. A test particle originally on the P_2 orbit, the inner orbit, would be placed on the trajectory shown when under controlled motion, i.e. under the influence of a time dependent controller where $\kappa = 1.0$. The initial position for this test particle is $(0.35, 0.0i)$.

One can readily see that the test particle no longer appears to be moving along a periodic orbit, but rather on a trajectory where each successive rotation is more and more offset in the positive imaginary

axis direction. After a certain number of revolutions, the test particle would ‘fly off’ in the positive real axis direction. Although it may be of some interest to some to determine if this new trajectory is periodic, it is not necessarily germane to our investigation.

What we are more interested in is a transfer trajectory, independent on whether or not it is periodic in the long run. Letting $\kappa = 2.0$ and 3.0 produces similar plots as that shown in Figure 3.12. However, if we let $\kappa = 4.0$, the plot shown in Figure 3.13 is produced. The test particle motion begins at $(0.35, 0.0i)$ and slowly moves away from the original periodic orbit.

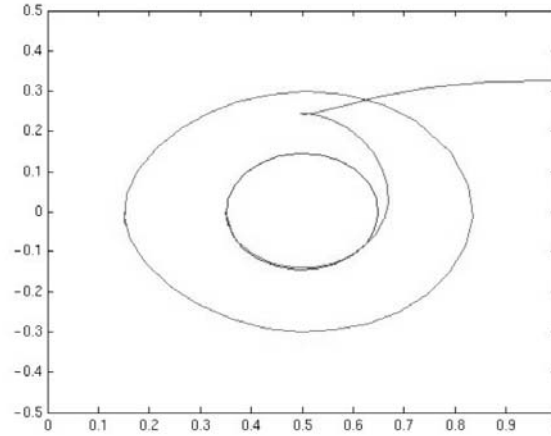


Figure 3.13. This is a plot of two periodic orbits that formed the foundation for the controller study, P_1 and P_2 , the outer and inner periodic orbits resembling ellipses, respectively. A test particle originally on the P_2 orbit, the inner orbit, would be placed on the trajectory shown when under controlled motion, i.e. under the influence of a time dependent controller where $\kappa = 4.0$. The initial position for this test particle is $(0.35, 0.0i)$. Note the ‘kink’ in the new trajectory.

At some point there is an abrupt change in the test particle motion, i.e. a ‘kink’ in the trajectory. One perspective on matters is that this presents a limitation to which new periodic orbit the test particle can be placed on. Is it reasonable to believe that the motion of a test particle can change so severely and not challenge stability requirements? Another interesting aspect this brings to light is the notion of

being able to turn ON and OFF the controller as necessary to achieve a desired result. One way to look at this is through the simple illustration shown in Figure 3.14.

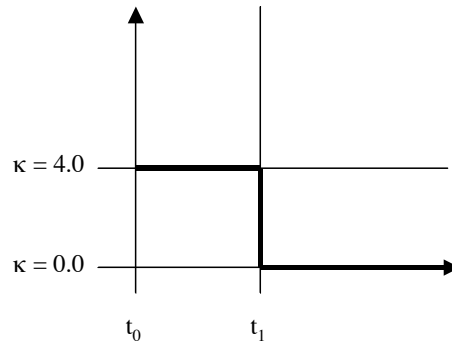


Figure 3.14. This is a sample controller state diagram for both the time-independent and time-dependent cases. At the initial time, t_o , the controller is turned ON and at some later time, t_1 , the controller is turned OFF thereby returning the governing equation of motion to its original, uncontrolled motion form.

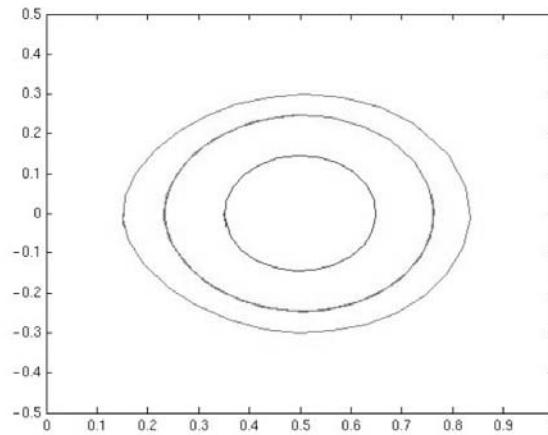


Figure 3.15. When the controller is turned OFF in close proximity or just prior to the ‘kink’, the test particle will be in an uncontrolled motion state and circumscribe a new periodic orbit about V_2 inside of P_1 and outside of P_2 as shown.

The controller is turned ON at the test particle initial condition/position $(0.35, 0.0i)$ by setting $\kappa = 4.0$. When $t = t_1$, the time corresponding to the ‘kink’ $(0.5009, 0.2465i)$, the controller is turned OFF by setting $\kappa = 0.0$. If we propagate the motion of the test particle in an uncontrolled state from this point, it will circumscribe a new, intermediate periodic orbit about V_2 in the counter-clockwise direction as shown in Figure 3.15. Therefore, if the desire were to place a test particle on the same periodic orbit as another for phase-locking the periodic orbit of the latter would have to be no further away than the ‘kink’ in the transfer trajectory. However, there is one other option, the same controller scheme can be used to bring a second test particle, say one that is traveling along an outer periodic orbit, to the new intermediate periodic orbit. This exact scenario is shown in Figure 3.16.

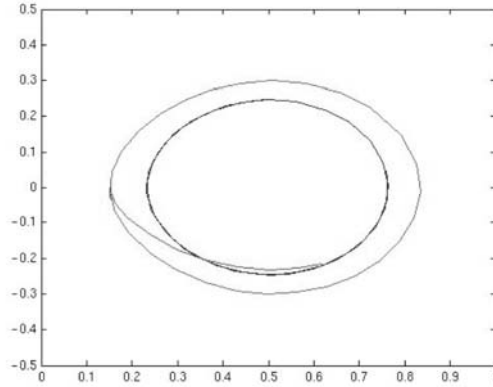


Figure 3.16. In this plot the inner orbit represents the new periodic orbit that a test particle was placed on after traversing along a transfer trajectory. A second test particle originally on the outer P_1 orbit would be placed on the trajectory shown when under controlled motion, i.e. under the influence of a time dependent controller where $\kappa=4.0$. The initial position for this test particle is $(0.15, 0.0i)$. The controller can then be turned OFF at some point where the transfer trajectory crosses the inner periodic orbit. At this point, the second test particle would also be in an uncontrolled motion state on the same intermediate periodic orbit as the first.

However, the new periodic orbit the first test particle was placed in is now the inner orbit. At the second test particle initial condition/position $(0.15, 0.0i)$ the controller is turned ON by setting $\kappa=4.0$.

When $t=0.3$, the time corresponding to what is actually the second orbit crossing (0.6167,-0.2165i), the controller is turned OFF by setting $\kappa=0.0$. At this point, the second test particle would also be in an uncontrolled motion state on the same intermediate periodic orbit as the first. In a situation where there were no external forces or torques, the two test particles would continue to follow this periodic or uncontrolled motion orbit indefinitely.

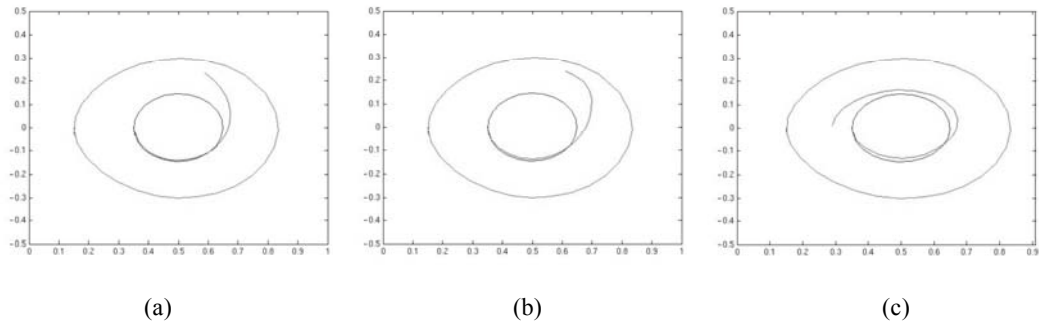
In a generally ‘trial and error’ process, the next step involved expanding the controller term to include sinusoidal variability by adding a trigonometric function. After a number of misses and near hits, the following relationship proved to yield the biggest benefit:

$$\dot{\xi}_3 = i\omega\xi_3 + \frac{i\Gamma_1}{2\pi} \cdot \frac{(\xi_3 - \xi_1)}{|\xi_3 - \xi_1|^2} + \frac{i\Gamma_2}{2\pi} \cdot \frac{(\xi_3 - \xi_2)}{|\xi_3 - \xi_2|^2} + \kappa \sin(\alpha\pi(t/T)) \quad 3.12$$

The term, α , is just a multiplier with a value of 1/2, 1, or 2. We will call the new term in the equation not just a controller, but controller no. 1. In three particular cases of interest, the parameter values were set as follows:

$$T = P_I \text{ orbit period} = 0.3; \kappa = 1; t = 0.3; \text{ and } \alpha = 1/2, 1, \text{ and } 2$$

The corresponding controller trajectory plots are shown in Figures 3.17a-3.17c.



Figures 3.17a-3.17c The trajectory of a test particle under the influence/drive by controller no. 1 is shown in each of the plots above. For all three cases, the values for T and κ are 0.3 and 1. For (a), (b), and (c), the value for α is 1/2, 1, and 2, respectively.

Figure 3.17b proved to be the most promising, since it appeared that the transfer trajectory went further than the one previously seen with the ‘kink’ in it. Next, the value for α was set to 1, but the value for κ was varied, i.e.

$$T = P_I \text{ orbit period} = 0.3; \kappa = 1.1, 1.2, \text{ and } 1.3; t = 0.3; \text{ and } \alpha = 1$$

This produced the series of plots shown in Figures 3.18a-3.18c. Figure 3.18c clearly shows that controller no. 1 can be used to move a test particle from P_1 to P_2 where the second test particle is moving. Additionally, this allows the test particles to be phase-locked in a shorter amount of time than the time dependent case, and also allows the test particle to ‘leave on a tangent’ and ‘arrive on a tangent’ avoiding abrupt changes that could arguably be considered unrealistic particle motion.

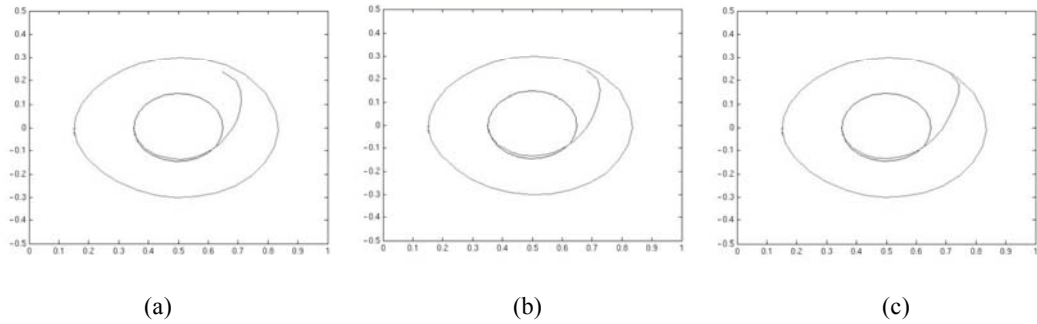


Figure 3.18a-3.18c. Test particle trajectories where controller no. 1 is active are shown in each of the three plots above. For all three cases, the values for T and α are 0.3 and 1. For (a), (b), and (c), the value for κ is 1.1, 1/2, and 1.3, respectively.

A piecemeal simulation of this phase-locking scenario is described here. Referring to the controller state diagram in Figure 3.19, the controller is in the OFF state until t_i . Up to this point in time, in the absence of any external forces or torques, the test particle traversed around V_2 in its initial periodic orbit as shown in Figure 3.20.

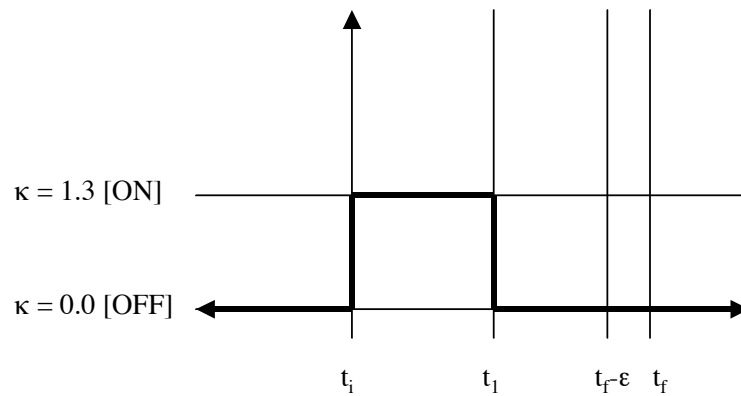


Figure 3.19. This is a state diagram for controller no. 1. The controller is turned ON from t_i to t_l , but remains OFF at all other times.

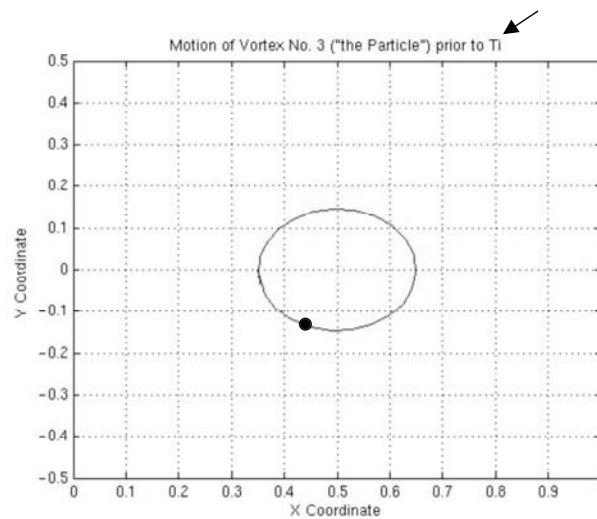


Figure 3.20. Test particle no. 1 moves in a periodic orbit about V_2 in the absence of any external forces or torques, i.e. uncontrolled motion.

At t_i the controller is turned ON and left ON until t_l . In this time interval, the test particle moves away from the initial periodic orbit and moves along a non-periodic trajectory as shown in Figure 3.21. At t_l , the controller is turned OFF.

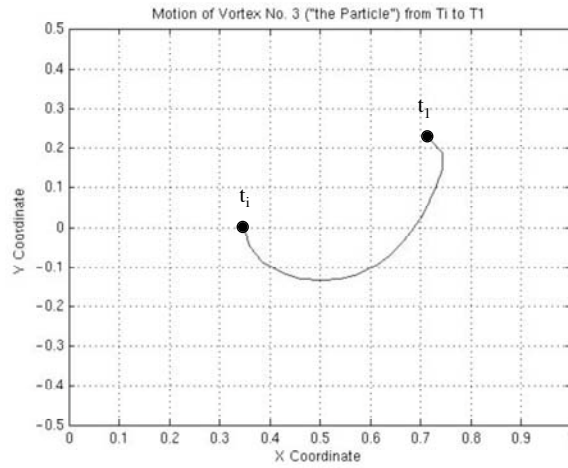


Figure 3.21. Test particle no. 1 is now under the influence of the controller. The particle leaves the initial periodic orbit and moves on a non-periodic trajectory until the controller is turned OFF.

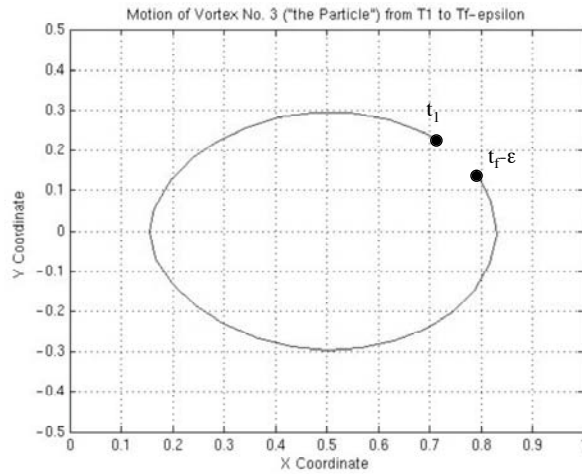


Figure 3.22. Test particle no. 1 moves in a new periodic orbit once the controller is turned OFF.

The test particle then returns to an uncontrolled motion state using the position at t_1 as the initial condition. This is shown in Figure 3.22. This confirms that it is possible to move a test particle from one periodic to another using a controller.

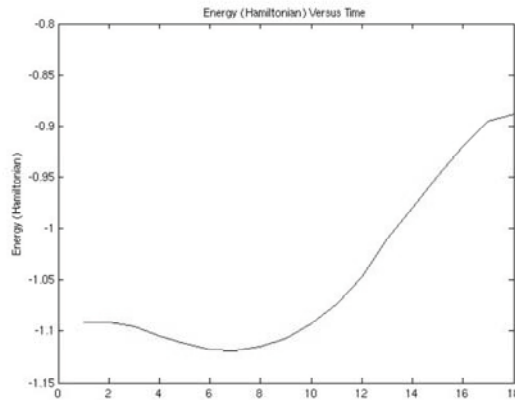


Figure 3.23. The energy level (i.e. Hamiltonian) varies as a test particle moves along the transfer trajectory. In this case, the test particle moves from a lower energy value (i.e. a periodic orbit in close proximity to V_2) to a higher energy value (i.e. a periodic orbit further away from V_2).

As stated a number of times earlier periodic orbits around V_2 are simply level curves of the Hamiltonian. Said differently, a test particle moving in a periodic orbit maintains a certain energy level. If we refer back to Figure 3.21 where a test particle moves along a non-periodic transfer trajectory, one question that arises pertains to the energy level. How does it vary from t_i to t_f ? The energy level generally moves from a lower to a higher value for this specific case as shown in Figure 3.23.

Now that a general investigation has identified a viable controller, which we've 'coined' controller no. 1, a more specific example can be studied. This example involves four test particles, P_1 - P_4 , each traveling along separate periodic orbits around V_2 . These test particles are to be moved to a single, lower energy periodic orbit for phase locking. Then a mechanism needs to be identified that adjusts the relative position of each test particle to maintain a virtual geometric shape or dynamically natural formation, in this case a rhombus or diamond shape. This scenario is illustrated in Figure 3.24. Initial conditions for the four test particles are also shown on the plot. The original concept was to follow what was done during the general investigation (i.e. turn ON the controller when the test particle is in

the ‘left most’ position, i.e. minimum value of the imaginary component of the complex number, of the uncontrolled, periodic orbit).

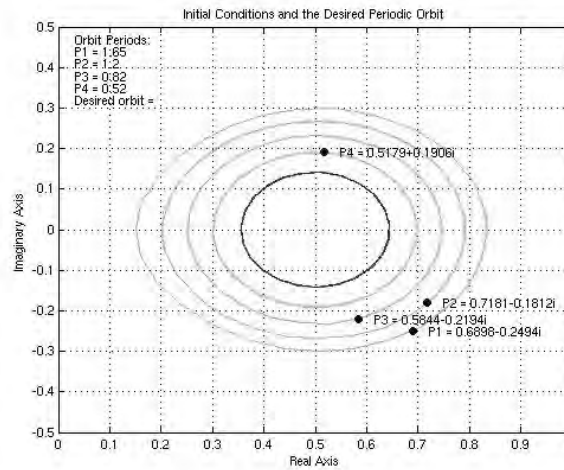


Figure 3.24. Four test particles, P_1 - P_4 , are each traveling along different periodic orbits. The desire is to eventually place each on single, lower energy periodic orbit for phase locking and then to adjust their relative positions to establish a desired shape or formation.

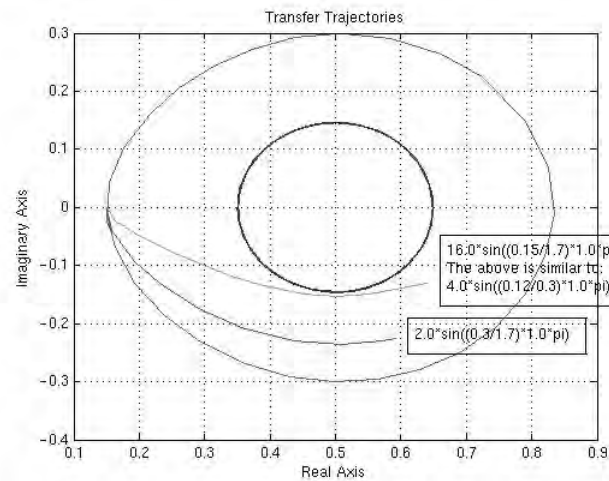


Figure 3.25. The transfer trajectory for placing the first test particle, P_1 on the new, desired periodic orbit requires $K=16.0$. Smaller values of K do not allow for the particle to reach the other periodic orbit. The controlled is turned ON when the particle is at (0.15,0.0i).

Figure 3.25 shows that a transfer trajectory to the new, desired orbit is achieved when $\kappa=16.0$ for the first test particle, P_1 . Even though the use of a sine function allows the test particle to ‘leave on a tangent’ and ‘arrive on a tangent’, it still appeared that the test particle would have to deal with an abrupt trajectory change almost immediately.

This prompted more examination and lead to the plot shown in Figure 3.26. The transfer trajectory on the left side of the plot is identical to the one shown in Figure 3.25 where $\kappa=16.0$. The one on right and the one at the bottom do not cross the new, desired periodic orbit, so are not valid. The one at the top, at first glance, doesn’t appear to be fruitful. However, when the controller parameter values were changed from $T=1.7$, $\kappa=16.5$, $t=0.130$, and $\alpha=1$ to $T=1.7$, $\kappa=2.3$, $t=0.330$, and $\alpha=1$, something interesting happens. The transfer trajectory shown in Figure 3.27 was produced. This is significant, because leaving from another extrema, i.e. maximum positive imaginary axis position of the periodic orbit (0.5,0.3i), and given a certain set of parameter values allows a test particle to follow a more natural or realistic type of motion as it moves from one periodic orbit to another.

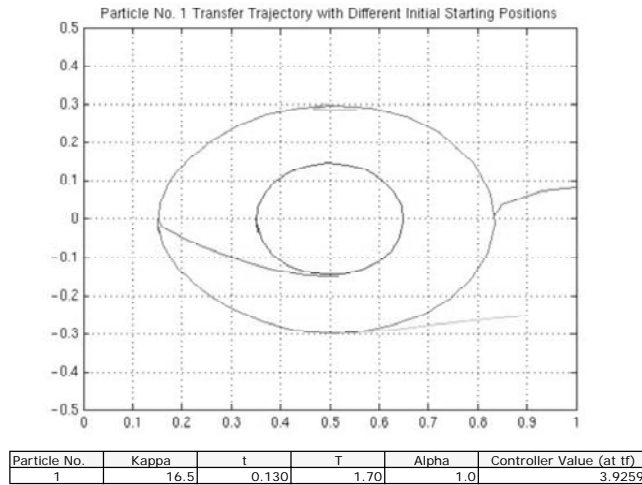


Figure 3.26. The controlled parameter values remained fixed as shown in the table below the plot, but the point at which the controller is turned ON varies. Here the spacing is $\pi/4$ or every 1/4 revolution.

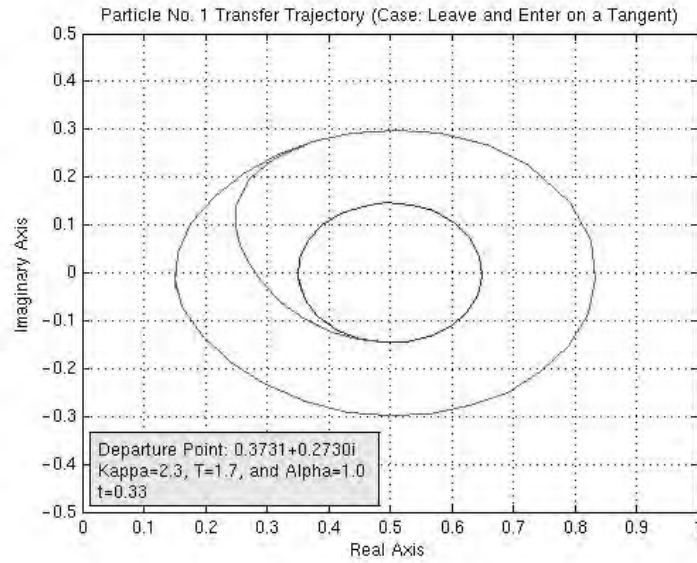


Figure 3.27. The transfer trajectory shown still allows a test particle to ‘leave on a tangent’ and ‘arrive on a tangent’. However, the general shape of the trajectory is such that there are no abrupt changes and is more consistent with shapes one expects of natural or realistic motion.

The appropriate controlled parameter values required to place P_1 and the other test particles on the new, desired periodic orbit are shown in Table 3.3. Individual transfer trajectory curves for P_1 - P_4 are shown in Figure 3.28.

Particle	κ	t	T	α
P_1	1.7	0.400	1.70	1
P_2	1.4	0.340	1.20	1
P_3	1.1	0.260	0.85	1
P_4	0.7	0.200	0.55	1

Table 3.3. Shown here are the controller no. 1 parameter values required to place each test particle from their uncontrolled periodic orbits to the new, desired periodic orbit.

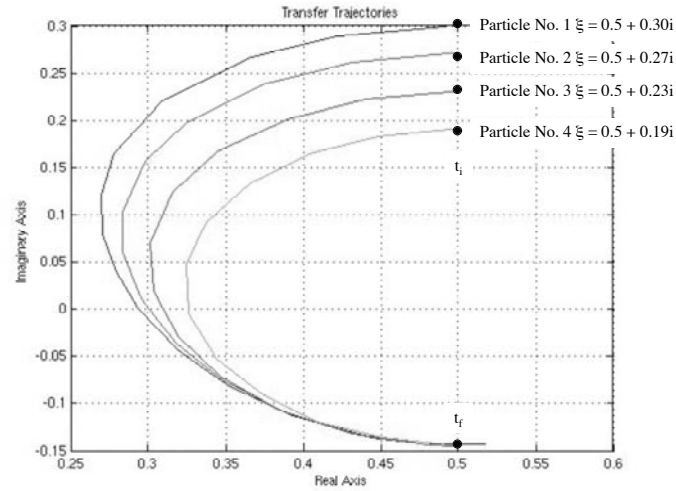


Figure 3.28. Transfer trajectory curves for each of the four test particles, P_1 through P_4 , are shown in the plot above. The general shape of each curve is such that there are no abrupt changes and is more consistent with shapes one expects of natural or realistic motion.

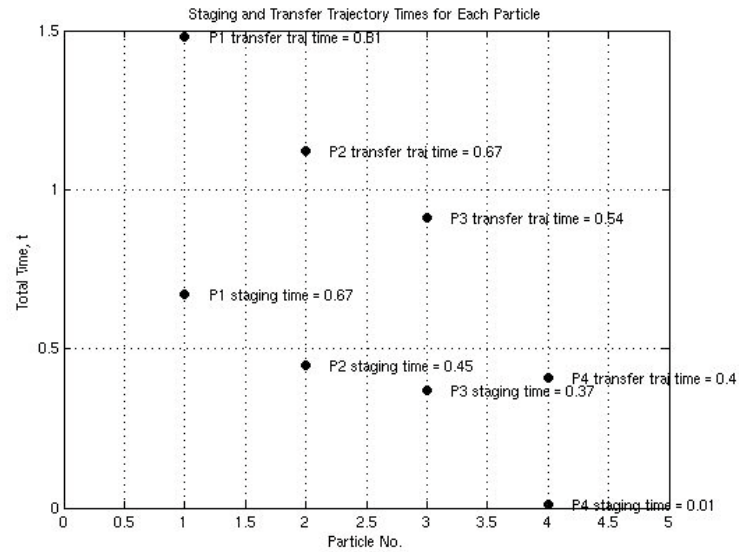


Figure 3.29. The staging times and transfer trajectory times for each of the four test particles, P_1 through P_4 , are shown in this plot. One can see that, P_1 , whose initial periodic orbit is furthest way from the new, desired periodic orbit requires the most amount of time for staging and transfer.

We will now introduce the notion of staging time. This is the amount of time required for a given test particle to traverse along the uncontrolled motion periodic orbit from its initial position to the extrema, i.e. maximum positive imaginary axis position of the periodic orbit, at which time controller no. 1 should be turned ON. Staging times and transfer trajectory times for each test particle are shown in Figure 3.29.

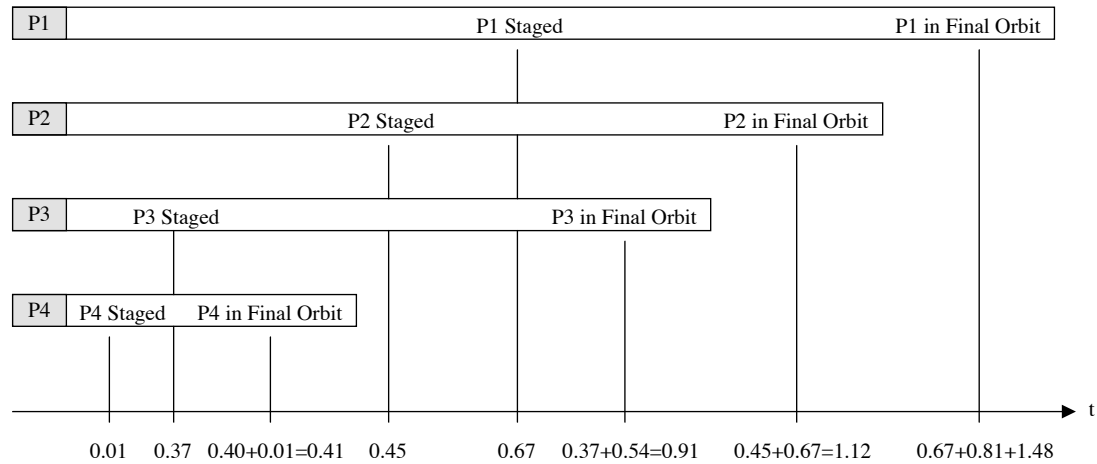


Figure 3.30. The staging times and transfer trajectory times for each of the four test particles, P_1 through P_4 , are shown in this plot, but this time on a common timeline. One can see that P_1 , whose initial periodic orbit is furthest way from the new, desired periodic orbit requires the most amount of time for staging and transfer.

A more meaningful chart might be Figure 3.30. It shows when each of the four test particles is staged and when it enters the new, desired periodic orbit on a single timeline. It shows that all four test particles are phase-locked at $t=1.48$ units of time. The next challenge is how to establish the desired naturally dynamic formation, since the relative positions of each test particle are not as desired (see in Figure 3.31).

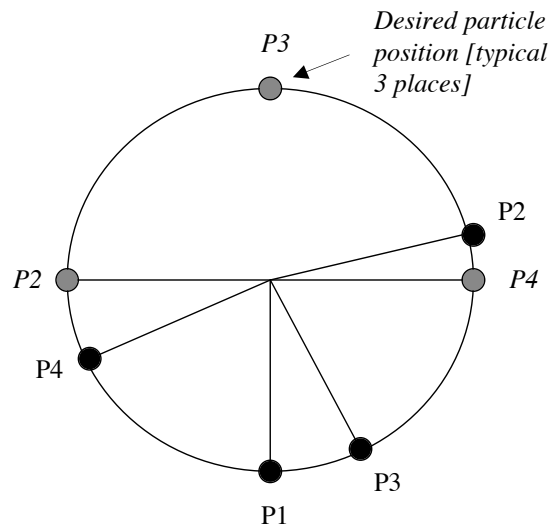


Figure 3.31. The actual test positions at $t=1.48$ units of time are shown in black. The desired virtual geometric shape is a rhombus or diamond with each of the test particles placed at each corner in the proper order (shown in grey shade above).

3.5 Formation Establishment

Two methods of formation establishment were examined. The orbit or resonant frequency approach was used to allow for both test particle phase-locking and formation establishment using a single controller, i.e. controller no. 1. In this case, an interactive MATLAB program that accepts user defined or random test particle initial conditions was written to define wait/traverse times, staging times, and transfer times as well as to produce desired plots. Since it was determined through many successive test cases that it would take a relatively long period of time to complete the test particle phase locking and formation establishment process, there was impetus to identify a second controller to use in conjunction with the first.

3.5.1 Resonant Frequency Approach

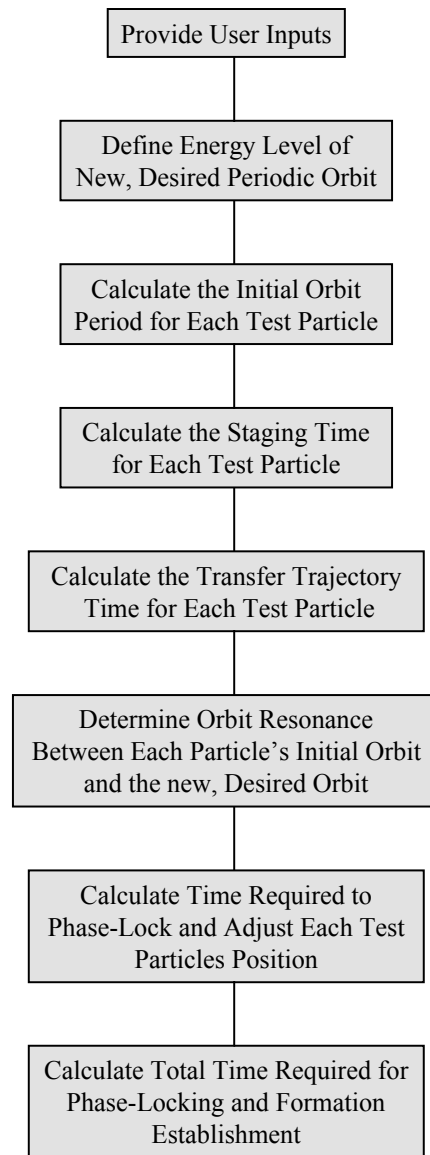


Figure 3.32. The flowchart for the script file, i.e. `main_script_v5_1.m`, is comprised of the eight main steps shown above.

One method for establishing the test particle formation actually requires that we implement a solution prior to the time the test particles are placed on the new, desired periodic orbit. In the general field of physics and waves, there is what is called resonant frequencies. In this case two or more waves have a special relationship where their frequencies are common multiples of one another. In celestial mechanics we talk of orbital resonances where, for example, two orbit periods share a least common multiple (i.e. orbit period of object A is x times the orbit period of object B, where x is an integer). In a similar manner we can determine how many orbits/revolutions a test particle must make on it's original orbit before it can be 'staged' and placed on a transfer trajectory, so that it arrives at precisely the correct time and position on the new, desired periodic orbit. In this case, a resonance exists between the test particles original orbit and the new, desired orbit. This particular method was used successfully to establish the desired test particle formation. Since we used the MATLAB application to assist, in a piecemeal manner, with problem solving and creating simulations, a decision was made to expand on this by creating a program with a user interface and capability to use the orbit resonance method (i.e. use of controller no. 1 exclusively) to quickly solve the problem of phase-locking and formation establishment. This required a single MATLAB script file, i.e. `main_script_v5_1.m`, and a single function file, i.e. `three_vortex.m`. A flowchart of the script file is given in Figure 3.32. The first step in this interactive program is for the user to provide inputs. These inputs are:

- Select the default set of or use the random number generator to define the test particle initial conditions
- Select the default or define the energy level for the new, desired periodic orbit
- Select the rhombus/diamond as the desired geometric shape or formation
- Select the default or define the acceptable formation error (e.g. ± 0.01 units of time)
- Select the option to produce plots or not
- Select the option to produce animations or not

The second step is to define the energy level of the new, desired periodic orbit. The third step is to calculate the initial orbit period for each test particle. This is done by propagating the position of the each test particle over time and then determining at which time the test particle returns to the initial position. The fourth step is to calculate the staging time for each test particle. This is easily done by calculating the time it takes for each test particle to travel from its initial position to the point at which the imaginary component of the orbit is at the maximum value. The fifth step is to calculate the transfer trajectory time for each particle. This is easily done by determining the time it takes for a test particle to travel from a position where the imaginary component is at the maximum value to its minimum value.

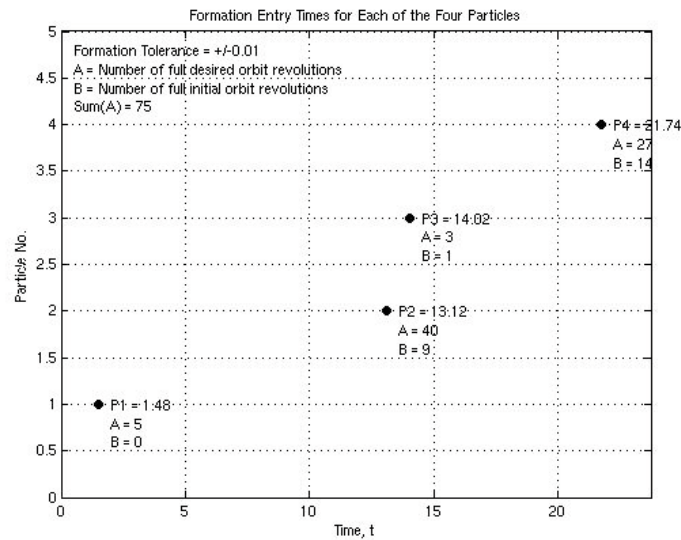


Figure 3.33. This charts shows how many times each particle must traverse their initial periodic orbits before being staged and placed on their respective transfer trajectories to arrive at the new, desired periodic orbit at precisely the right time and position. Note that it requires 75 revolutions of the new, desired periodic orbit for the formation to be established.

However, in order to do this, the problem determines the best controller parameter values to link the test particle's initial periodic orbit to the new, desired periodic orbit. The sixth step is the

most complicated. The program is required to determine the orbit resonance for a given test particle's initial periodic orbit and the new, desired periodic orbit. It must then stage the test particle at the appropriate revolution, cause it to move on the transfer trajectory, and then place it on the new, desired periodic orbit. For each test particle, the time it takes for orbital resonance, staging, and transfer are calculated, tracked, and reported at the end of the program and on a plot, if so desired. Finally, all appropriate times are added together to determine the overall time required for phase locking and formation establishment. The MATLAB script and function file source code is included in Appendix B. The `main_scrip_v5_1.m` file is on the order of 1,350 source lines of code and the `three_vortex.m` function file is on the order of 50 source lines of code.

Figure 3.33 shows many times each of the four test particles must traverse their original periodic orbits before being stages and placed on their respective transfer trajectories. Note that the formation is established at $t=21.74$ units of time.

3.5.2 Controller Method

In much the same manner used in Section 3.4, modifications to the fundamental equation of motion (equation 3.1) were identified and investigated. Rather than adding yet another (controller) term to the equation, a decision was made to see what would happen if one of the three existing terms were slightly modified. In the first case, a scale factor, λ , was added:

$$\dot{\xi}_3 = \lambda i \omega \xi_3 + \frac{i\Gamma_1}{2\pi} \cdot \frac{(\xi_3 - \xi_1)}{|\xi_3 - \xi_1|^2} + \frac{i\Gamma_2}{2\pi} \cdot \frac{(\xi_3 - \xi_2)}{|\xi_3 - \xi_2|^2} \quad 3.13$$

Setting $\lambda=0.9$ produces the plot shown in Figure 3.34. Although the test particle moves along a slightly small orbit, it returns to its initial position at (0.35,0.0i). The second case was to replace the scale factor, λ , in equation 3.13 with a time element, t . A test particle following this equation of motion produces the plot shown in Figure 3.35. Notice that each successive 'orbit' that the test

particle makes is slightly offset in the positive real axis direction. The third case involves adding a trigonometric time element/function to the first term rather than a scale factor or time element. A plot of the results is shown in Figure 3.36.

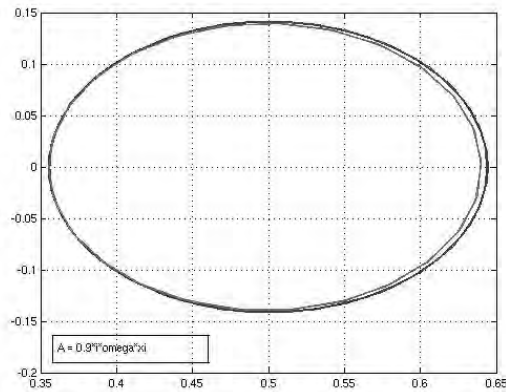


Figure 3.34. If a test particle starts at $(0.35, 0.0i)$ and follows the motion defined in equation 3.13, where $\lambda=0.9$, it circumscribes a periodic orbit that is slightly smaller than the original (i.e. $\lambda=1.0$).

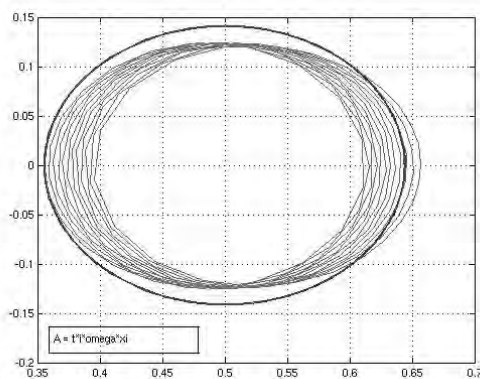


Figure 3.35. If a test particle starts at $(0.35, 0.0i)$ and follows the motion similar to that defined in equation 3.13, however, instead of a scale factor a time element is used, it moves further and further away from the initial position

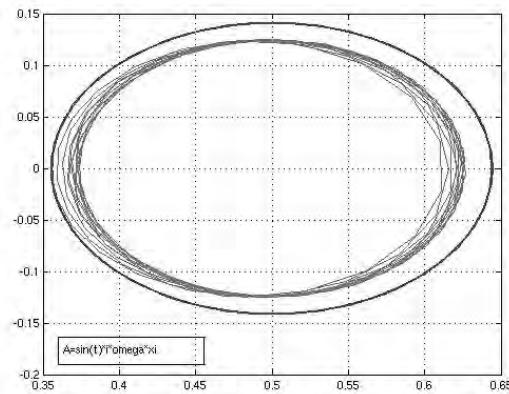


Figure 3.36. If a test particle starts at $(0.35, 0.0i)$ and follows the motion similar to that defined in equation 3.13, however, instead of a scale factor a trigonometric time element/function is used, it moves away from the initial position and appears to move along a slightly smaller orbit.

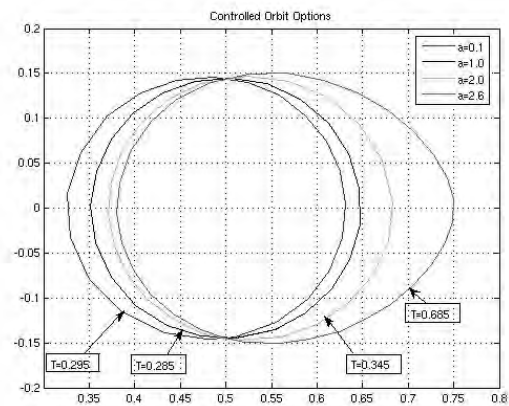


Figure 3.37. This is a plot of four trajectories/orbits for the four different scale factor values shown in the upper right-hand corner. The orbit period changes from a low of 0.285 units of time to a high of 0.685.

Of the three cases/controllers examined, it was the first that offered the most promise (e.g. a test particle returns to its initial position). However, does this type of controller change the orbit period? In order to adjust the relative position of each test particle in the new, desired periodic orbit,

we must be able to change the orbit period as an independent variable. Fortunately, the answer is ‘yes’. Referring to Figure 3.37, it is clear that the orbit period changes depending on the value of the scale factor used.

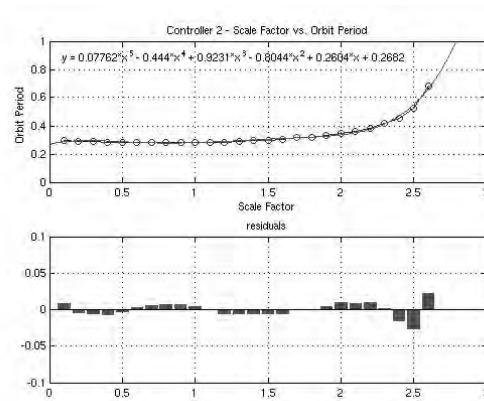


Figure 3.38. The first is a plot of scale factor versus orbit period. The data are curve-fit to a fifth-order polynomial. A plot of residuals is shown on the bottom. One can readily see that the curve provides a solution that is accurate to within 0.025 units of time for a selected scale factor value.

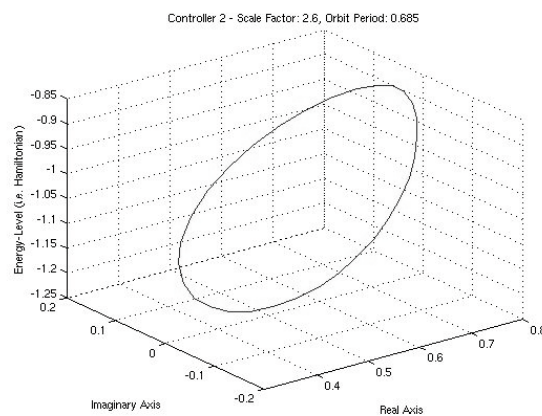


Figure 3.39. This is the trajectory/orbit for a scale factor value of 2.6. One can readily see that the energy value (or Hamiltonian) varies as a function of the orbit position. A particle traveling along this trajectory will eventually return to its original energy state.

The plot of scale factor versus orbit period is show in Figure 3.38. Notice that the rate of change is greatest for scale factors from approximately 1.5 to 2.6. Another interesting observation is that the energy level, i.e. Hamiltonian, varies over an orbit period. This is illustrated in the three ‘dimensional’ plot of Figure 3.39. This is similar to the inclined orbit in celestial mechanics.

3.6 Proof-of-Concept Problem and Solution

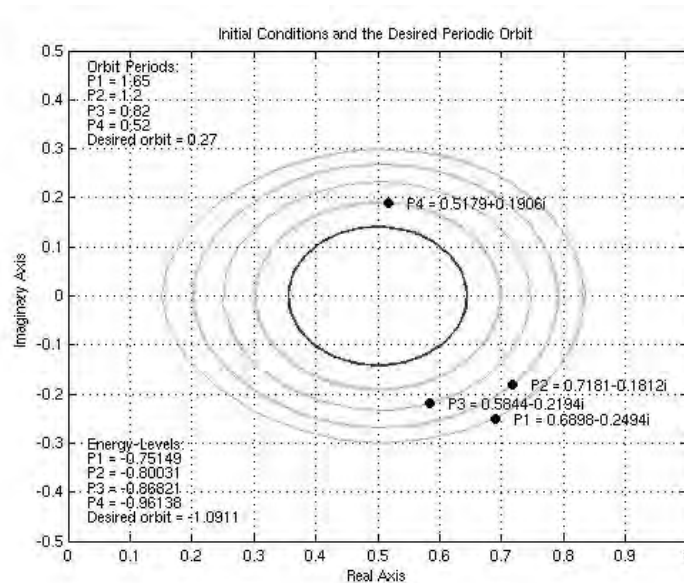


Figure 3.40. Four test particles, P_1 through P_4 , are each traveling along different periodic orbits. The desire is to eventually place each on single, lower energy periodic orbit for phase locking and then to adjust their relative positions to establish a desired shape or formation. In this case, a diamond or rhombus on the inner most periodic orbit.

Now we will examine how to use what we will now call controller no. 2 to adjust the relative positions of each test particle to form the desired dynamically natural formation (i.e. rhombus/diamond). Refer back to Figure 3.31. We will define the first test particle, P_1 , to be the master reference, i.e. formation seed, and is deemed to be in the correct position in the new, desired

periodic orbit. As started earlier, P_1 arrives on the new, desired periodic orbit at $t=1.48$ units of time. We will call this arrival position, “bottom dead center”. Each of the three other test particles is to be placed ‘behind’ P_1 (i.e. later in time) in quarter rev increments apart as shown in red. As time moves forward each test particle will move in a counter-clockwise direction. The next test particle to arrive at bottom dead center is P_4 . This occurs at $t=1.53$ units of time. We know where the desired position of P_4 is and how much time it takes for it to arrive at bottom dead center. Since we know the period of the new, desired orbit, we also know when it returns on each successive revolution. We will use controller no. 2 to place P_4 on a trajectory that allows it leave the new, desired periodic orbit, but also allows it to return to the bottom dead center position at precisely the right time to be in the proper position relative to P_1 . We do this by selecting the appropriate value for λ .

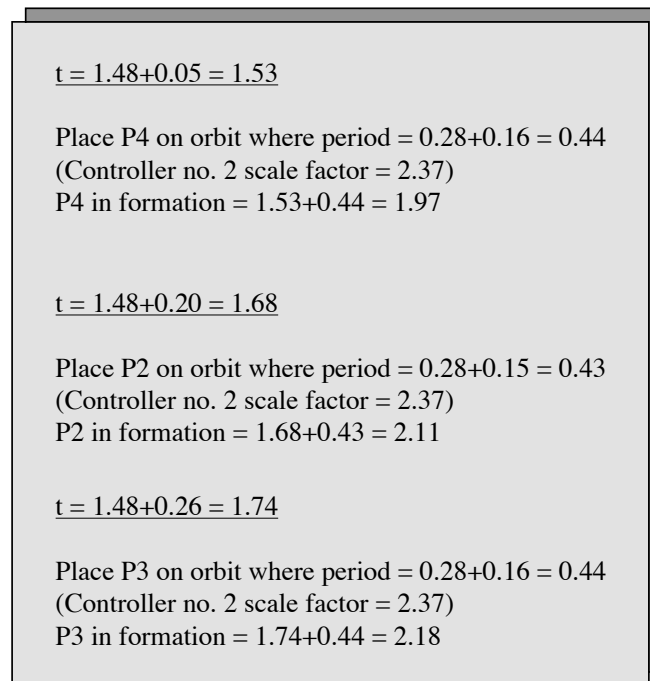


Figure 3.41. This is the timeline for establishing a formation with Controller No. 2. The first particle that is placed temporarily on an intermediate trajectory is P_4 . This is followed by P_2 , and finally, P_3 .

We follow the same procedure for P_2 and P_3 . The description of ‘what’ and ‘when’ is summarized in Figure 3.41. The most significant result is shown at the bottom where P_2 , the final test particle to be placed at the proper position, is shown to arrive at $t=2.18$ units of time. Comparing this to the 21.74 units of time it took to phase-lock and establish a dynamically natural formation using only Controller No. 1 in an orbit resonance method, this results in an order of magnitude reduction in total time required!

Three animations in Audio Video Interleaved (AVI) format were created (converted from MATLAB movies) to represent uncontrolled test particle motion, use of controller no. 1 to place each of the four test particles in the new, desired periodic orbit, and use of controller no. 2 to establish the desired dynamically natural formation.

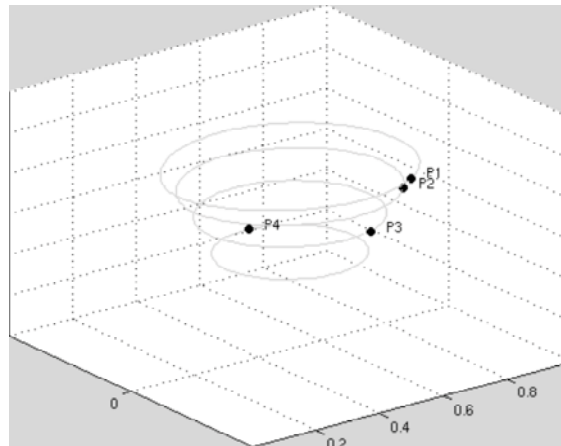


Figure 3.42. This is the first frame of the two hundred frames in the first animation. Each of the four test particles is shown in their initial periodic orbits. Particle motion is in the counter-clockwise direction as viewed from the Z or positive energy (i.e. Hamiltonian) axis direction.

Each of the three animations is a three ‘dimensional’ perspective of the problem: X is the real axis, Y is the imaginary axis, and Z is the energy (i.e. Hamiltonian) axis. This perspective provides more insight than what could be obtained through simple planar projections or models. The first frame of each animation is shown in Figures 3.42-3.44.

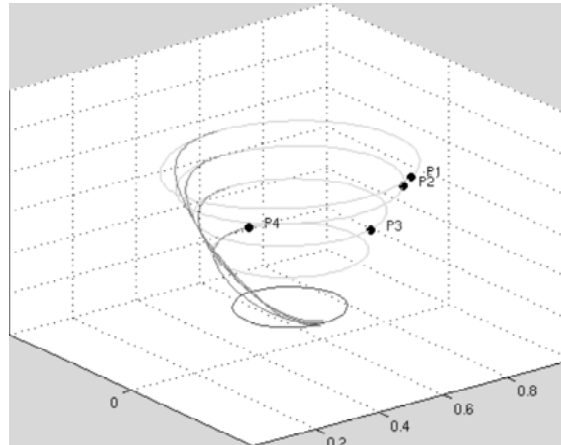


Figure 3.43. This is the first frame of the two hundred frames in the second animation. Each of the four test particles is shown in their initial periodic orbits shown in green. Particle motion is in the counter-clockwise direction as viewed from the Z or positive energy (i.e. Hamiltonian) axis direction. When each test particle reaches desired extrema they travel on the transfer trajectories and eventually arrive on the new, desired periodic orbit.

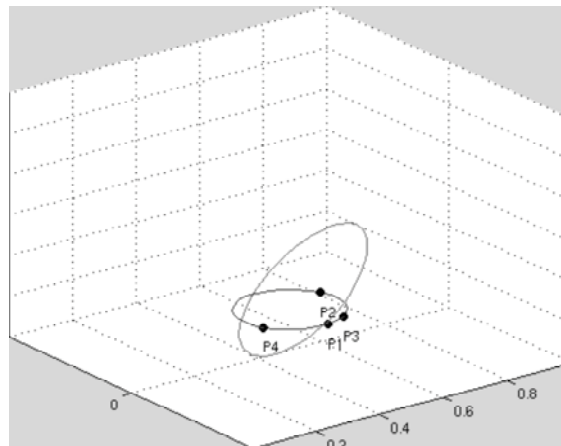


Figure 3.44. This is the first frame of the one hundred and sixty-five frames in the second animation. Each of the four test particles is shown in on the new, desired periodic orbit. Particle motion is in the counter-clockwise direction as viewed from the Z or positive energy (i.e. Hamiltonian) axis direction. When P_4 , P_2 , and P_3 reach the necessary initial position they switch to the intermediate periodic orbit and eventually return to the new, desired periodic orbit at precisely the correct time and position to establish the desired rhombus or diamond formation.

The initial and final states of the four test particles are shown in Figure 3.45. Where there once was just four test particles in their respective orbits, they were eventually phased-locked on a single new orbit and their relative positions adjusted to form a desired dynamically natural formation, specifically a rhombus/diamond.

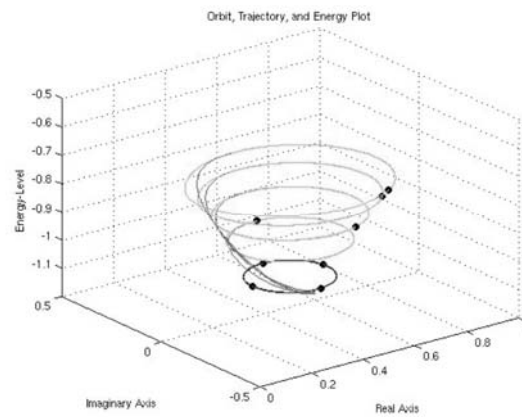


Figure 3.45. In this three-dimensional perspective, one can see the particle initial conditions and the particle final conditions (on the single periodic orbit defining the corners of a rhombus/diamond shape).

Chapter 4: The Circular, Restricted Three-Body Problem

The general three-body problem cannot be solved analytically. If one makes some simplifying assumptions approximate solutions to particular problems can be generated. A problem that has been often studied is the circular, restricted three-body problem in celestial mechanics. The circular, restricted three-body problem is a special case of the general three-body problem. In this case, the system is comprised of two bodies of significant mass that are in a circular orbit around the barycenter. A third body of insignificant mass is then introduced into the system. The third body does not influence the motion of the other two bodies, but they influence its motion. Refer to Figure 4.1. Let the mass, m_2 , of the lesser of the two significant bodies be equal to μ , and the mass of the greater m_1 , be equal to $(1 - \mu)$. Let m_3 be the mass of the third body.

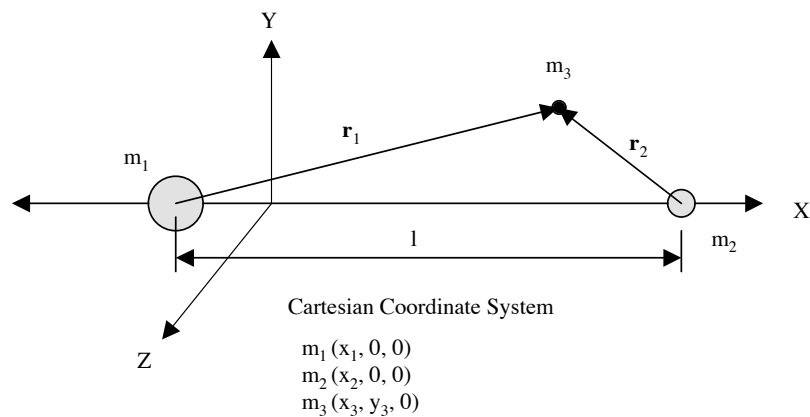


Figure 4.1. The geometry of the circular, restricted three-body problem is shown here.

4.1 Equations of Motion

The development of the equations of motion for the circular, restricted three-body problem is as follows. The angular velocity, \mathbf{w} , is equal to the mean motion, \mathbf{n} :

$$\mathbf{w} = \mathbf{n} = \sqrt{\frac{\mu}{a^3}} \hat{k} = k \sqrt{\frac{m_1 + m_2}{(x_2 - x_1)^3}} \hat{k} \quad 4.1$$

$$\mathbf{w} = \mathbf{n} = \sqrt{\frac{[(1-\mu) + \mu]}{1}} \hat{k} = \hat{k} \quad 4.2$$

The force acting on the mass of the third body must then be determined. This force is equal to the gravitational force of m_1 , gravitational force of m_2 , a coriolis force, and a centrifugal force. Kaplan [14] gives a general expression of this force. This is shown in equation 4.3. The coriolis force and the centrifugal force are shown in equation 4.4 and in equation 4.5, respectively.

$$\ddot{\mathbf{r}} = \ddot{\mathbf{r}}_0 + \ddot{\mathbf{r}}_b + 2\mathbf{w} \times \dot{\mathbf{r}}_b + \dot{\mathbf{w}} \times \mathbf{r} + \mathbf{w} \times (\mathbf{w} \times \mathbf{r}) \quad 4.3$$

$$2\mathbf{w} \times \dot{\mathbf{r}}_b \quad 4.4$$

$$\mathbf{w} \times (\mathbf{w} \times \mathbf{r}) \quad 4.5$$

The force acting on m_3 is then:

$$\mathbf{F}_3 = \mathbf{F}_1 - \mathbf{F}_2 - m_3(2\mathbf{w} \times \dot{\mathbf{r}}_b) - m_3[\mathbf{w} \times (\mathbf{w} \times \mathbf{r})] \quad 4.6$$

$$m_3 \ddot{\mathbf{r}} = -\frac{Gm_1m_2}{r_1^3} \cdot \mathbf{r}_1 - \frac{Gm_2m_3}{r_2^3} \cdot \mathbf{r}_2 - 2m_3\hat{k} \times \mathbf{r} - m_3\hat{k} \times (\hat{k} \times \mathbf{r}) \quad 4.7$$

$$\mathbf{a} = -\frac{Gm_1}{r_1^3} \cdot \mathbf{r}_1 - \frac{Gm_2}{r_2^3} \cdot \mathbf{r}_2 - 2\hat{k} \times \mathbf{v} - \hat{k} \times (\hat{k} \times \mathbf{r}) \quad 4.8$$

$$\mathbf{r} = x\hat{i} + y\hat{j} + z\hat{k}, \mathbf{v} = \dot{x}\hat{i} + \dot{y}\hat{j} + \dot{z}\hat{k}, \mathbf{a} = \ddot{x}\hat{i} + \ddot{y}\hat{j} + \ddot{z}\hat{k} \quad 4.9$$

$$\hat{k} \times \mathbf{v} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ 0 & 0 & 1 \\ x' & y' & z' \end{vmatrix} = -y'\hat{i} + x'\hat{j} \quad 4.10$$

$$\hat{k} \times \mathbf{r} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ 0 & 0 & 1 \\ x & y & z \end{vmatrix} = -y\hat{i} + x\hat{j} \quad 4.11$$

$$k \times (\hat{k} \times \mathbf{r}) = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ 0 & 0 & 1 \\ -y & x & 0 \end{vmatrix} = -x\hat{i} - y\hat{j} \quad 4.12$$

$$x''\hat{i} + y''\hat{j} + z''\hat{k} = -\frac{Gm_1}{r_1^3}[(x-x_1)\hat{i} + y\hat{j} + z\hat{k}] - \frac{Gm_2}{r_2^3}[(x-x_2)\hat{i} + y\hat{j} + z\hat{k}] - 2(-y\hat{i} + x\hat{j}) - (-x\hat{i} - y\hat{j}) \quad 4.13$$

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = -\frac{Gm_1}{r_1^3} \begin{bmatrix} (x-x_1) \\ y \\ z \end{bmatrix} - \frac{Gm_2}{r_2^3} \begin{bmatrix} (x-x_2) \\ y \\ z \end{bmatrix} - 2 \begin{bmatrix} -y' \\ x' \\ 0 \end{bmatrix} - \begin{bmatrix} -x \\ -y \\ 0 \end{bmatrix} \quad 4.14$$

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} + 2 \begin{bmatrix} -y' \\ x' \\ 0 \end{bmatrix} = -\frac{Gm_1}{r_1^3} \begin{bmatrix} (x-x_1) \\ y \\ z \end{bmatrix} - \frac{Gm_2}{r_2^3} \begin{bmatrix} (x-x_2) \\ y \\ z \end{bmatrix} - \begin{bmatrix} -x \\ -y \\ 0 \end{bmatrix} \quad 4.15$$

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} + 2 \begin{bmatrix} -y' \\ x' \\ 0 \end{bmatrix} - \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} = -\frac{Gm_1}{r_1^3} \begin{bmatrix} (x-x_1) \\ y \\ z \end{bmatrix} - \frac{Gm_2}{r_2^3} \begin{bmatrix} (x-x_2) \\ y \\ z \end{bmatrix} \quad 4.16$$

Recall that $m_1 = (1-\mu)$, and $m_2 = \mu$ and G is a constant. Also,

$$(x-x_1) = x - \mu \quad 4.17$$

$$(x-x_2) = x + (1-\mu) \quad 4.18$$

Equation 4.17 and 4.18 are true, since the distance from the barycenter to c is μ and $(1-\mu)$, respectively. Therefore,

$$x'' - 2y' - x = -\frac{(1-\mu)(x-\mu)}{r_1^3} - \frac{\mu[x+(1-\mu)]}{r_2^3} \quad 4.19$$

$$y'' + 2x' - y = \frac{-(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3} \quad 4.20$$

$$z'' = -\frac{(1-\mu)y}{r_1^3} - \frac{\mu z}{r_2^3} \quad 4.21$$

The equations of motion for the third body in a rotating coordinate system are shown in 4.22 through 4.24.

$$\ddot{x} - 2\dot{y} - x = -\frac{(1-\mu)(x-\mu)}{r_1^3} - \frac{\mu[x+(1-\mu)]}{r_2^3} \quad 4.22$$

$$\ddot{y} + 2\dot{x} - y = \frac{-(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3} \quad 4.23$$

$$\ddot{z} = -\frac{(1-\mu)y}{r_1^3} - \frac{\mu z}{r_2^3} \quad 4.24$$

Equations 4.22 through 4.24 represent the second-order, non-linear differential equations of motion for the third body. Developing a system of first-order linear equations (for the two-dimensional or planar case) from the non-linear, second-order equations of motion given in equations 4.22 and 4.23 begins as follows:

$$x = x_0 + \eta_1, \dot{x} = \dot{x}_0 + \eta_1, \ddot{x} = \ddot{x}_0 + \eta_1 \quad 4.25$$

$$y = y_0 + \eta_2, \dot{y} = \dot{y}_0 + \eta_2, \ddot{y} = \ddot{y}_0 + \eta_2 \quad 4.26$$

$$z = z_0 + \eta_3, \dot{z} = \dot{z}_0 + \eta_3, \ddot{z} = \ddot{z}_0 + \eta_3 \quad 4.27$$

$$r_1 = \sqrt{(x-\mu)^2 + y^2 + z^2} \quad 4.28$$

$$r_2 = \sqrt{(x+1-\mu)^2 + y^2 + z^2} \quad 4.29$$

$$r_1' = \sqrt{[x-(\mu+\eta_1)]^2 + y^2 + z^2} \quad 4.30$$

$$r_2' = \sqrt{[x+1-(\mu+\eta_1)]^2 + y^2 + z^2} \quad 4.31$$

$$(r_1)^{-3} = \left\{ \sqrt{[x-(\mu+\eta_1)]^2 + y^2 + z^2} \right\}^{-3} \quad 4.32$$

$$(r_2)^{-3} = \left\{ \sqrt{[x+1-(\mu+\eta_1)]^2 + y^2 + z^2} \right\}^{-3} \quad 4.33$$

$$(r_1)^{-3} = \left\{ x_0^2 - 2x_0\mu + \mu_2 + 2x_0\eta_1 + y_0^2 + 2y_0\eta_2 \right\}^{-3/2} \quad 4.34$$

$$(r_2)^{-3} = \left\{ x_0^2 - 2x_0\mu + 2x_0 + \mu^2 + 2x_0\eta_1 + 2\eta_1 + 1 - 2\mu - 2\mu\eta_1 + y_0^2 + 2y_0\eta_2 \right\}^{-3/2} \quad 4.35$$

$$(r_1)^{-3} = [(x_0 - \mu)^2 + y_0^2 + 2\eta_1(x_0 - \mu) + 2y_0\eta_2]^{-3/2} \quad 4.36$$

$$(r_2)^{-3} = [(x_0 + 1 - \mu)^2 + y_0^2 + 2\eta_1(x_0 - \mu + 1) + 2y_0\eta_2]^{-3/2} \quad 4.37$$

$$\alpha = (x_0 - \mu)2 + y_0^2 = r_{10}^2 \quad 4.38$$

$$\beta = 2\eta_1(x_0 - \mu) + 2y_0\eta_2 \quad 4.39$$

$$k = -\frac{3}{2} \quad 4.40$$

$$(r_1)^{-3} = (r_{10}^2)^{-3/2} - \frac{3}{2}(r_{10}^2)^{-5/2} [2\eta_1(x_0 - \mu) + 2y_0\eta_2] + \dots \quad 4.41$$

$$(r_1)^{-3} = (r_{10})^{-3} - \frac{3}{2}(r_{10})^{-5} [2\eta_1(x_0 - \mu) + 2y_0\eta_2] + \dots \quad 4.42$$

$$(r_2)^{-3} = (r_{20})^{-3} - \frac{3}{2}(r_{20})^{-5} [2\eta_1(x_0 - \mu) + 2y_0\eta_2] + \dots \quad 4.43$$

The first equation of motion is therefore,

$$\begin{aligned} (\ddot{x}_0 + \ddot{\eta}_1) - 2(\dot{y}_0 + \dot{\eta}_2) - (x_0 - \eta_1) = & -(1 - \mu)(x_0 + \eta_1 - \mu) \left\{ r_{10}^{-3} - \frac{3}{2} r_{10}^{-5} [2\eta_1(x_0 - \mu) + 2y_0\eta_2] \right\} \\ & - \mu(x_0 + \eta_1 + 1 - \mu) \left\{ r_{20}^{-3} - \frac{3}{2} r_{20}^{-5} [2\eta_1(x_0 + 1 - \mu) + 2y_0\eta_2] \right\} \end{aligned} \quad 4.44$$

The second equation of motion is therefore,

$$\begin{aligned} (\ddot{x}_0 - 2\dot{y}_0 - x_0) + (\ddot{\eta}_1 - 2\dot{\eta}_2 - \eta) = & -\frac{(1 - \mu)(x_0 + \eta_1 - \mu)}{r_{10}^3} - \frac{\mu(x_0 + \eta_1 + 1 - \mu)}{r_{20}^3} \\ & + \frac{3}{2} \frac{(1 - \mu)(x_0 + \eta_1 - \mu)}{r_{10}^5} \left\{ 2\eta_1(x_0 - \mu) + 2y_0\eta_2 \right\} \\ & + \frac{3}{2} \frac{\mu(x_0 + \eta_1 + 1 - \mu)}{r_{20}^5} \left\{ 2\eta_1(x_0 + 1 - \mu) + 2y_0\eta_2 \right\} \end{aligned} \quad 4.45$$

Simplifying, the first equation of motion becomes,

$$(\ddot{\eta}_1 - 2\dot{\eta}_2 - \eta_1) = \eta_1 \left\{ (1-\mu) \left[-\frac{1}{r_{10}^3} + \frac{3(x_0 - \mu)}{r_{10}^5} \right] + \mu \left[-\frac{1}{r_{20}^3} + \frac{3(x_0 + 1 - \mu)}{r_{20}^5} \right] \right\} \quad 4.46$$

$$+ \eta_2 \left[\frac{3(1-\mu)(x_0 - \mu)y_0}{r_{10}^5} + \frac{3(x_0 + 1 - \mu)y_0}{r_{20}^5} \right] \quad 4.47$$

and the second equation of motion,

$$(\ddot{\eta}_2 - 2\dot{\eta}_1 - \eta_2) = \eta_2 \left[\frac{3(1-\mu)(x_0 - \mu)y_0}{r_{10}^5} + \frac{3\mu(x_0 + 1 - \mu)y_0}{r_{20}^5} \right] \quad 4.48$$

$$+ \eta_1 \left[(1-\mu) \left(-\frac{1}{r_{10}^3} + \frac{3y_0^2}{r_{10}^5} \right) + \mu \left(-\frac{1}{r_{20}^3} + \frac{3y_0^2}{r_{20}^5} \right) \right] \quad 4.49$$

4.2 Equilibrium Points

Refer back to the standard equations of motion for the circular, restricted three-body problem, equations 4.22 through 4.24. If we wish to identify the equilibrium points the velocity and acceleration terms in the rotating Cartesian coordinate frame must be set to zero. This results in the following:

$$-x = -\frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x-1+\mu)}{r_2^3} \quad 4.50$$

$$-y = \frac{-(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3} \quad 4.51$$

$$0 = -\frac{(1-\mu)z}{r_1^3} - \frac{\mu z}{r_2^3} \quad 4.52$$

One can readily see in equation 4.52 that $z = 0$. Therefore, the equilibrium points must lie in the XY-plane. Setting $r_1 = r_2 = 1$ satisfies equations 4.50 and 4.51. This locates two of the equilibrium points

at the vertices of two symmetric and adjacent equilateral triangles where the two primaries are at the other two vertices. Finding the other equilibrium points is slightly more involved. Again, refer back to equations 4.22 and 4.24. Noticing that $y = 0$ satisfies equation 4.23 tells us that the other equilibrium points must lie along the X-axis. Setting $r_1 = r_2 = 1$ in equation 4.22, results in the following:

$$f(x) = x - \frac{(1-\mu)(x+\mu)}{(x+\mu)^3} - \frac{\mu(x-1+\mu)}{(x-1+\mu)^3} = 0 \quad 4.53$$

Expanding the above results in a quintic equation where the roots are the X-axis coordinates of the three equilibrium points. The Analytical Graphics, Inc. technical note [2] reduces this into individual equations, which are:

$$x^5 - (3-\mu)x^4 + (3-2\mu)x^3 - \mu x^2 + 2\mu x - \mu = 0 \quad 4.54$$

$$x^5 - (3-\mu)x^4 + (3-2\mu)x^3 - \mu x^2 - 2\mu x - \mu = 0 \quad 4.55$$

$$x^5 + (2+\mu)x^4 + (1+2\mu)x^3 - (1-\mu)x^2 - 2(1-\mu)x - (1-\mu) = 0 \quad 4.56$$

However, in equations 4.54 through 4.56, the variable x is defined as the equilibrium point distance from the closest primary body. The MATLAB program, `find_libration_points.m`, was written to expedite problem-solving and to explicitly identify the X and Y coordinates of all five equilibrium points in the rotating Cartesian coordinate frame (see Appendix B). The relative positions of the five equilibrium points with respect to the two primaries are shown in Figure 4.2. For the earth-moon system the mass ratio is $\mu = 0.012150$. The coordinates for the five equilibrium points are given in Table 4.1.

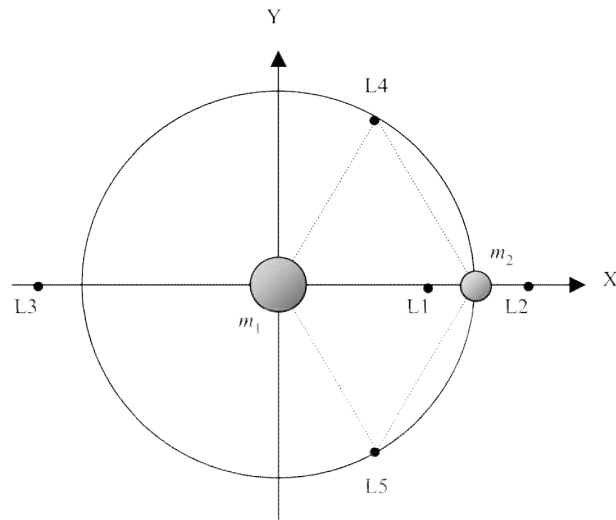


Figure 4.2. The five equilibrium points of the circular, restricted three-body problem in celestial mechanics are identified with respect to the two primary bodies in the plot above. Three equilibrium points (i.e. L1, L2, and L3) lie along the X-axis and the other two (i.e. L4 and L5) are at the vertices of the two equilateral triangles.

Equilibrium Point	X Coordinate	Y Coordinate
L1	0.8369	0
L2	1.1799	0
L3	-1.0051	0
L4	0.4879	0.8660
L5	-0.4879	-0.8660

Table 4.1. The coordinates for the five equilibrium points in rotating Cartesian coordinate frame for the earth-moon system are given in this table. Note “L” stands for “libration”, which is synonymous with “equilibrium” point.

For the Saturn-Titan moon system the mass ratio is $\mu = 0.000238$. The coordinates for the five equilibrium points are given in Table 4.2. Hamilton and Burns [16] describe the Hill Sphere as the gravitational sphere of influence of a body. It turns out that we can use this relationship to check the validity of the MATLAB script.

Equilibrium Point	X Coordinate	Y Coordinate
L1	0.9574	0
L2	1.0447	0
L3	-1.0001	0
L4	0.4998	0.8660
L5	-0.4998	-0.8660

Table 4.2. The coordinates for the five equilibrium points in rotating Cartesian coordinate frame for the Saturn-Titan moon system are given in this table. Note “L” stands for “libration”, which is synonymous with “equilibrium” point.

For a circular, restricted three-body system where the mass of one primary is significantly less than the other, as is the case with the Saturn-Titan moon system, the distance to L1 or L2 from the second primary can be approximated by:

$$d \approx \sqrt[3]{\mu/[3(1-\mu)]} \quad 4.57$$

For the Saturn-Titan moon system, $d = 0.0430$. Titan is located at (0.9998,0.0) and L1 at (0.9574,0.0). The distance between the two is 0.0424. Therefore, the MATLAB results were valid.

Having established that there are five equilibrium points in the circular, restricted three-body problem in celestial mechanics, the next step was to determine if they were stable or not through the use of simple linear stability theory. In order to check the stability of a given equilibrium point, a small displacement was introduced and a Taylor series expansion was carried out over the complete set of equations of motion. However, the higher-order terms were disregarded to simplify matters. Understanding what happens to the displacements over time then involved solving a fourth-order differential equation by determining the eigenvalues and eigenvectors of an associated characteristic equation. The structure of the eigenvalues will provide stability information, e.g. $\text{Re}(\lambda) < 0$ indicates stability.

Define the X-axis and Y-axis location of a given equilibrium point as x_e and y_e , respectively. Now assume that a spacecraft is first located at the equilibrium point, but is then slightly displaced from it. The spacecraft position will then be:

$$x = x_e + \delta x \quad 4.58$$

$$y = y_e + \delta y \quad 4.59$$

$$z = \delta z \quad 4.60$$

The δ terms in these three equations represent a small displacement in each respective direction. If we restricted ourselves to the XY-plane and perform a Taylor series expansion of a function $f(x, y)$ around the equilibrium point (x_e, y_e) , the following is obtained:

$$f(x, y) = f(x_e, y_e) + \delta x f_x(x_e, y_e) + \delta y f_y(x_e, y_e) + \dots \quad 4.61$$

Incorporating the displacement terms and the Taylor series expansion in the standard equations of motion shown in equations 4.22 and 4.23 and leaving the acceleration and velocity terms on one side results in the following:

$$\ddot{\delta x} - 2\dot{\delta y} = (U_x) + \delta x(U_{xx}) + \delta y(U_{xy}) + \dots \quad 4.62$$

$$\ddot{\delta y} + 2\dot{\delta x} = (U_y) + \delta x(U_{yx}) + \delta y(U_{yy}) + \dots \quad 4.63$$

Since we know that gravity potentials are

$$U_x = x - \frac{(1-\mu)(x-x_1)}{r_1^3} + \frac{\mu(x-x_2)}{r_2^3} \quad 4.64$$

$$U_y = y - \frac{(1-\mu)y}{r_1^3} + \frac{\mu y}{r_2^3} \quad 4.65$$

where the X-axis positions of the first and second primary bodies are $x_1 = -\mu$ and $x_2 = 1-\mu$, respectively, the associated partial derivatives of each are:

$$U_{xx} = 1 - \frac{(1-\mu)}{r_1^3} + \frac{\mu}{r_2^3} - \frac{3(1-\mu)}{r_1^5}(x-x_1)^2 + \frac{3\mu}{r_2^5}(x-x_2)^2 \quad 4.66$$

$$U_{yy} = 1 - \frac{(1-\mu)}{r_1^3} + \frac{\mu}{r_2^3} - \frac{3(1-\mu)}{r_1^5}y^2 + \frac{3\mu}{r_2^5}y^2 \quad 4.67$$

$$U_{xy} = U_{yx} = \frac{3(1-\mu)}{r_1^5} (x-x_1)y + \frac{3\mu}{r_2^5} (x-x_2)y \quad 4.68$$

Since the gravity potentials at the equilibrium point are zero, i.e. $U_x = 0$ and $U_y = 0$, and the higher-order terms are not of any concern, equations 4.62 and 4.63 simply to:

$$\ddot{x} - 2\dot{\delta}y = \delta x(U_{xx}) + \delta y(U_{xy}) \quad 4.69$$

$$\ddot{y} + 2\dot{\delta}x = \delta x(U_{yx}) + \delta y(U_{yy}) \quad 4.70$$

When an operator, $\Delta = \partial/\partial x = \partial/\partial y$, is introduced as appropriate, equations 4.69 and 4.70 become

$$\Delta^2 \partial x - 2\Delta \partial y = \partial x U_{xx} + \partial y U_{xy} \quad 4.71$$

$$\Delta^2 \partial y + 2\Delta \partial x = \partial x U_{yx} + \partial y U_{yy} \quad 4.72$$

Collecting like terms results in

$$(\Delta^2 - U_{xx})\partial x = (2\Delta + U_{xy})\partial y \quad 4.73$$

$$(\Delta^2 - U_{yy})\partial y = -(2\Delta - U_{xy})\partial x \quad 4.74$$

Now, operating on equation 4.73 with $2\Delta - U_{xy}$ and equation 4.74 with $2\Delta + U_{xy}$ the following relationships are developed

$$[\Delta^4 + (4 - U_{xx} - U_{yy})\Delta^2 + (U_{xx}U_{yy} - U_{xy}^2)]\partial y = 0 \quad 4.75$$

$$[\Delta^4 + (4 - U_{xx} - U_{yy})\Delta^2 + (U_{xx}U_{yy} - U_{xy}^2)]\partial x = 0 \quad 4.76$$

Since equations 4.75 and 4.76 are of the same form, it is obvious that ∂x and ∂y satisfy the fourth-order differential equation

$$\Delta^4 + (4 - U_{xx} - U_{yy})\Delta^2 + (U_{xx}U_{yy} - U_{xy}^2) = 0 \quad 4.77$$

Equilibrium Point	Real Part	Imaginary Part	Stable or Unstable
L1	0	2.3325	Unstable
	0	-2.3325	
	1.2337	0	
	-1.2337	0	
L2	1.6879	0	Unstable
	-1.6879	0	
	0	1.5971	
	0	-1.5971	
L3	0	1.8930	Unstable
	0	-1.8930	
	0.7569	0	
	-0.7569	0	
L4	0	0.9545	Stable
	0	-0.9545	
	0	0.2982	
	0	-0.2982	
L5	0	0.9545	Stable
	0	-0.9545	
	0	0.2982	
	0	-0.2982	

Table 4.3. The eigenvalues of the characteristic equation are complex numbers. The real and imaginary components of the eigenvalues for the five equilibrium points in the earth-moon system are shown above. One can see that L1-L3 are unstable, while L4 and L5 are stable.

To solve this equation we set $f = \varepsilon^{\lambda t}$. Therefore, equation 4.77 becomes

$$\lambda^4 + (4 - U_{xx} - U_{yy})\lambda^2 + (U_{xx}U_{yy} - U_{xy}^2) = 0 \quad 4.78$$

The roots of this equation are the eigenvalues needed to determine stability. The equation can be solved numerically, e.g. Newton-Rhapson method. However, to expedite problem solving, a MATLAB script, eigenvalues.m, was developed (see Appendix B). It uses the poly and roots functions to find the eigenvalues. The definitions of stability are:

- If any of the eigenvalues have imaginary parts, then the solution orbits around the equilibrium point and can be considered stable
- If any of the eigenvalues have a real part that is less than or equal to zero the solution is stable
- If any of the eigenvalues have a real part that is greater than zero the solution is unstable

Equilibrium Point	Real Part	Imaginary Part	Stable or Unstable
L1	0	2.3643	Unstable
	0	-2.3643	
	1.2640	0	
	-1.2640	0	
L2	2.2983	0	Unstable
	-2.2983	0	
	0	1.9451	
	0	-1.9451	
L3	0	1.8873	Unstable
	0	-1.8873	
	0.7495	0	
	-0.7495	0	
L4	0	0.9982	Stable
	0	-0.9982	
	0	0.0401	
	0	-0.0401	
L5	0	0.9982	Stable
	0	-0.9982	
	0	0.0401	
	0	-0.0401	

Table 4.4. The eigenvalues of the characteristic equation are complex numbers. The real and imaginary components of the eigenvalues for the five equilibrium points in the Saturn-Titan moon system are shown above. One can see that L1-L3 are unstable, while L4 and L5 are stable.

The eigenvalues associated with the five equilibrium points in the earth-moon system as well as the assessment of stability for each are provided in Table 4.3. The eigenvalues associated with the five equilibrium points in the Saturn-Titan moon system as well as the assessment of stability for each are provided in Table 4.4. While we state that L4 and L5 are stable equilibrium points for the earth-moon and Saturn-Titan moon systems, Valtonen and Karttunen [29] state that there is a limitation based on the mass ratio of the system being examined. The equilibrium points, L4 and L5 are only stable if $\mu < \mu_{\text{lim}}$, where

$$\mu_{\text{lim}} = \frac{1}{2} - \sqrt{\frac{23}{108}} \approx 0.0385 \quad 4.79$$

It just so happens that restricted three-body systems in the solar system meet this constraint. Systems outside of the solar system must be checked against this constraint to determine if the L4 and L5 equilibrium points are stable.

4.3 Jacobi Integral

The next key step is to develop the Jacobi integral. Although some would believe that this is the energy of the system, it actually represents the total energy plus the angular momentum of the system given the rotating Cartesian coordinate system. The first step is to multiply the equations of motion with the corresponding velocity component. This is shown in equations 4.80 through 4.82.

$$\dot{x} \left\{ \ddot{x} - 2\dot{y} - x = -\frac{(1-\mu)(x-\mu)}{r_1^3} - \frac{\mu[x+(1-\mu)]}{r_2^3} \right\} \quad 4.80$$

$$\dot{y} \left\{ \ddot{y} + 2\dot{x} - y = \frac{-(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3} \right\} \quad 4.81$$

$$\dot{z} \left\{ \ddot{z} = -\frac{(1-\mu)y}{r_1^3} - \frac{\mu z}{r_2^3} \right\} \quad 4.82$$

Multiplying through results in:

$$\dot{x}\ddot{x} - 2\dot{x}\dot{y} - \dot{x}x = -\frac{\dot{x}(1-\mu)(x-\mu)}{r_1^3} - \frac{\mu\dot{x}[x+(1-\mu)]}{r_2^3} \quad 4.83$$

$$\dot{y}\ddot{y} + 2\dot{x}\dot{y} - y\dot{y} = \frac{-\dot{y}(1-\mu)y}{r_1^3} - \frac{\mu\dot{y}y}{r_2^3} \quad 4.84$$

$$\dot{z}\ddot{z} = -\frac{\dot{z}(1-\mu)y}{r_1^3} - \frac{\mu\dot{z}z}{r_2^3} \quad 4.85$$

Summing both side of the equations:

$$(\ddot{x}\ddot{x} - 2\dot{x}\ddot{y} - \dot{x}\dot{x}) + (\ddot{y}\ddot{y} + 2\dot{x}\ddot{y} - y\ddot{y}) + \ddot{z}\ddot{z} =$$

$$\frac{(1-\mu)}{r_1^3}[-\dot{x}(x-\mu) - y\dot{y} - z\dot{z}] + \frac{\mu}{r_2^3}[-\dot{x}(x+1-\mu) - y\dot{y} - z\dot{z}] \quad 4.86$$

$$\ddot{x}\ddot{x} + \ddot{y}\ddot{y} + \ddot{z}\ddot{z} = x\dot{x} + y\dot{y} - \frac{(1-\mu)}{r_1^3}[(x-\mu)\dot{x} + y\dot{y} + z\dot{z}] - \frac{\mu}{r_2^3}[(x+1-\mu)\dot{x} + y\dot{y} + z\dot{z}] \quad 4.87$$

Recall that the distance from c to the center of m_1 is μ or $\mu = x_1$, the distance from c to the center of m_2 is $(1-\mu)$ or $(1-\mu) = x_2$, and $m_2 = \mu = m$, so:

$$\ddot{x}\ddot{x} + \ddot{y}\ddot{y} + \ddot{z}\ddot{z} = x\dot{x} + y\dot{y} - \frac{(1-m)}{r_1^3}[(x-x_1)\dot{x} + y\dot{y} + z\dot{z}] - \frac{\mu}{r_2^3}[(x-x_2)\dot{x} + y\dot{y} + z\dot{z}] \quad 4.88$$

Now,

$$\mathbf{r}_1 = (x-x_1)\hat{i} + y\hat{j} + z\hat{k}, \mathbf{r}_2 = (x-x_2)\hat{i} + y\hat{j} + z\hat{k} \quad 4.89$$

$$r_1\dot{r}_1 = (x-x_1)\dot{x} + y\dot{y} + z\dot{z}, r_2\dot{r}_2 = (x-x_2)\dot{x} + y\dot{y} + z\dot{z} \quad 4.90$$

$$d\left(\frac{1}{r_1}\right) = -\left(\frac{1}{r_1^2}\right)dr_1 = -\frac{\dot{r}_1}{r_1^2} = \frac{r_1\dot{r}_1}{r_1^3} = \frac{(x-x_1)\dot{x} + y\dot{y} + z\dot{z}}{r_1^3} \quad 4.91$$

$$x\dot{x} + y\dot{y} + z\dot{z} = d\left\{\frac{1}{2}[\dot{x}^2 + \dot{y}^2 + \dot{z}^2]\right\} \quad 4.92$$

$$x\dot{x} + y\dot{y} = d\left\{\frac{1}{2}[\dot{x}^2 + \dot{y}^2]\right\} \quad 4.93$$

Therefore,

$$x\dot{x} + y\dot{y} + z\dot{z} = x\dot{x} + y\dot{y} - \frac{(1-m)}{r_1^3}[(x-x_1)x + y\dot{y} + z\dot{z}] - \frac{m}{r_2^3}[(x-x_2)x + y\dot{y} + z\dot{z}] \quad 4.94$$

This results in equation 4.95 or in more simplified form, equation 4.96.

$$d\left[\frac{1}{2}(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)\right] - d\left[\frac{1}{2}(x^2 + y^2)\right] - (1-m)d\left(\frac{1}{r_1}\right) - md\left(\frac{1}{r_2}\right) \quad 4.95$$

$$\frac{1}{2}(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) - \frac{1}{2}(x^2 + y^2) - \frac{(1-m)}{r_1} - \frac{m}{r_2} = C \quad 4.96$$

Again, the Jacobi integral, C , is equal to the total energy plus the angular momentum of the system, because of the rotating Cartesian coordinate system.

4.4 Periodic Orbit Generation

Equations 4.22 through 4.24, the general equations of motion for the circular, restricted three-body case, were coded into a MATLAB function file, `three_body_v4_1.m`. A MATLAB script was then written that called this function file, i.e. `three_body_script_v5_1.m`. Both of these files are included in Appendix B. As stated earlier, even though MATAB has library functions for 2nd and 3rd order as well as 4th and 5th order Runge-Kutta routines a decision was made to utilize a custom 7th and 8th order approach for better accuracy. These MATLAB files formed the foundation for much of the computer simulations.

The governing equations of motion for the circular, restricted three-body problem shown in equations 4.22 through 4.24 were also augmented use in AUTO 2000 [see Appendix A]. The first of three steps was to phrase the computation of a periodic orbit as a two-point boundary value problem to (1) normalize the periodicity to a value of “1” and to solve for the unknown period, T . The next step was to discretize the system, so that Newton-Rhapson method could be used to find the solution. Finally, the damping, or “unfolding” parameter, λ , was introduced. This gives rise to a vertical Hopf bifurcation. Hilborn [12] states that a Hopf bifurcation signals the birth of a stable limit cycle, and is one of the most common two-dimensional bifurcations for models with a single control parameter. The equations of motion (as a system of first-order differential equations) are therefore transformed to the following set (equations 4.97 through 4.102).

$$\dot{x} = Tv_x + \lambda E_x \quad 4.97$$

$$\dot{y} = Tv_y + \lambda E_y \quad 4.98$$

$$\dot{z} = Tv_z + \lambda E_z \quad 4.99$$

$$\dot{v}_x = T[2v_y + x - \frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x-1+\mu)}{r_2^3}] + \lambda E_{v_x} \quad 4.100$$

$$\dot{v}_y = T[-2v_x + y - \frac{(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3}] + \lambda E_{v_y} \quad 4.102$$

$$\dot{v}_z = T[-\frac{(1-\mu)z}{r_1^3} - \frac{\mu z}{r_2^3}] + \lambda E_{v_z} \quad 4.103$$

As one can plainly see, the period, T , has been incorporated as scale factor to the baseline set of equations, and the unfolding parameter has been added as another term in each of the equations. Here, the energy, E , merely the Jacobi Constant, C , has also been added to each of the new terms. A list of the AUTO 2000 input and output files is shown in Table 4.5.

Input File Name	Description
c.3d	Parameter definitions and values
3d.c	Equations
compute_lagrange_points_0.5.auto	Script for computing equilibrium points
compute_periodic_orbits.xauto	Script for solving two-point BVP

Table 4.5. The four AUTO2000 input files are described above.

Each of the four input files can be found in Appendix C. The syntax for use of the compute_periodic_orbits.xauto script is as follows, i.e. this is the command that one would enter at the AUTO 2000 command prompt:

compute_period_orbits_xauto (option) (Lagrange Point of interest) (Mass ration of system)

Finally, output files are of the form: l1_mu_0.000233_period_3.040533_-_21.

The prefix denotes the selected equilibrium point. Imbedded in the file name are the mass ratio, period, and bifurcation point.

File Name
s.l4 mu 0.012150 period 21.070352
s.l4 mu 0.012150 period 21.070352~
s.l4 mu 0.012150 period 6.283185
s.l4 mu 0.012150 period 6.283185~
s.l4 mu 0.012150 period 6.283185 - 101
s.l4 mu 0.012150 period 6.283185 + 101
s.l4 mu 0.012150 period 6.283185 - 112
s.l4 mu 0.012150 period 6.283185 + 112
s.l4 mu 0.012150 period 6.283185 - 16
s.l4 mu 0.012150 period 6.283185 + 16
s.l4 mu 0.012150 period 6.283185 - 27
s.l4 mu 0.012150 period 6.283185 + 27
s.l4 mu 0.012150 period 6.283185 - 39
s.l4 mu 0.012150 period 6.283185 + 39
s.l4 mu 0.012150 period 6.283185 - 46
s.l4 mu 0.012150 period 6.283185 + 46
s.l4 mu 0.012150 period 6.283185 - 53
s.l4 mu 0.012150 period 6.283185 + 53
s.l4 mu 0.012150 period 6.283185 - 64
s.l4 mu 0.012150 period 6.283185 + 64
s.l4 mu 0.012150 period 6.283185 - 76
s.l4 mu 0.012150 period 6.283185 + 76
s.l4 mu 0.012150 period 6.283185 - 83
s.l4 mu 0.012150 period 6.283185 + 83
s.l4 mu 0.012150 period 6.283185 - 89
s.l4 mu 0.012150 period 6.283185 + 89
s.l4 mu 0.012150 period 6.582675
s.l4 mu 0.012150 period 6.582675~
s.l4 mu 0.012150 period 6.582675 - 53
s.l4 mu 0.012150 period 6.582675 + 53
s.l4 mu 0.012150 period 6.582675 - 55
s.l4 mu 0.012150 period 6.582675 + 55
s.l4 mu 0.012150 period 6.582675 - 61
s.l4 mu 0.012150 period 6.582675 + 61
s.l4 mu 0.012150 period 6.582675 - 70
s.l4 mu 0.012150 period 6.582675 + 70
s.l4 mu 0.012150 period 6.582675 - 79
s.l4 mu 0.012150 period 6.582675 + 79
s.l4 mu 0.012150 period 6.582675 - 84
s.l4 mu 0.012150 period 6.582675 + 84

Table 4.6. These are the earth-moon L4 equilibrium point periodic orbit families.

AUTO 2000 was used to identify earth-moon periodic orbits associated with the stable equilibrium points, L4 and L5. Table 4.6 identified the files produced for the L4 equilibrium point. The file names include the magnitude of the initial orbit period and also indicate whether or not there were bifurcations. Each file may contain multiple orbits and comprise a single orbit “family”. However, each of the orbits may actually have a different period than the initial orbit period. [Note: Since the period is normalized to one orbit of the moon about earth, an initial orbit period of 6.283185 means that that moon will make a little more than six full revolutions about the earth in the same time it will take a spacecraft to fully transit one of the orbits described in the file.] If there are any period-doubling effects, the file name will be appended with a two or three-digit suffix. It is interesting to note that the initial period of 21.070352 does not have any bifurcations. However, the initial period of 6.283185 has more than twenty bifurcations points. To give one a sense of the number of periodic orbits possible in a given L4 orbit family, the “egg-shaped” family for an initial orbit period of 6.283185 are shown in Figures 4.3 through 4.5. The XY planar projection is essentially a view from the celestial North Pole.

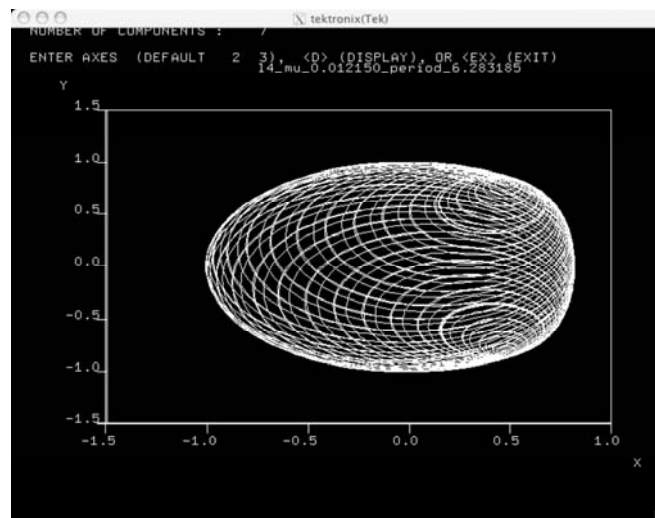


Figure 4.3. AUTO 2000 plot of the earth-moon L4 equilibrium point periodic orbit family. Initial orbit period = 6.283185. [XY planar projection]

The XZ planar projection can be regarded as a side-view of the system, and the YZ planar projection is a view from the axis joining the two primary bodies. One can see that there is symmetry in each of the views.

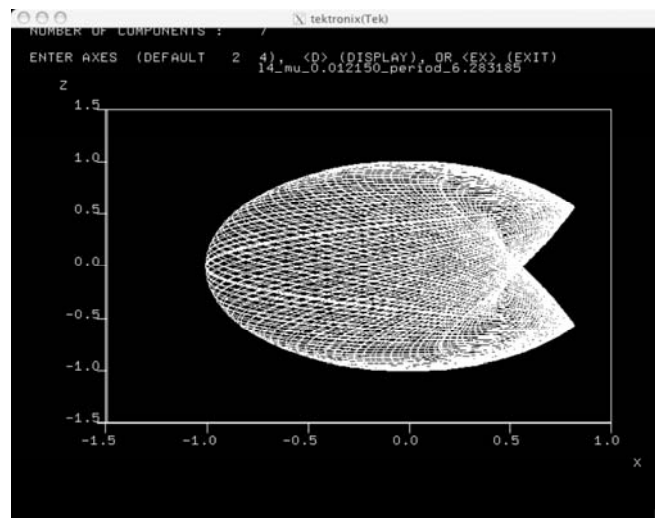


Figure 4.4. AUTO 2000 plot of the earth-moon L4 equilibrium point periodic orbit family. Initial orbit period = 6.283185. [XZ planar projection]

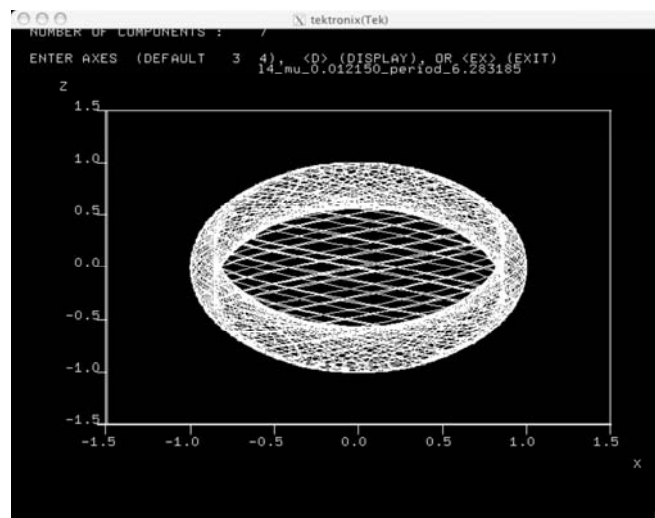


Figure 4.5. AUTO 2000 plot of the earth-moon L4 equilibrium point periodic orbit family. Initial orbit period = 6.283185. [YZ planar projection]

File Name
s.l5_mu_0.012150_period_21.070352
s.l5_mu_0.012150_period_21.070352~
s.l5_mu_0.012150_period_6.283185
s.l5_mu_0.012150_period_6.283185~
s.l5_mu_0.012150_period_6.283185_-104
s.l5_mu_0.012150_period_6.283185_+104
s.l5_mu_0.012150_period_6.283185_-115
s.l5_mu_0.012150_period_6.283185_+115
s.l5_mu_0.012150_period_6.283185_-19
s.l5_mu_0.012150_period_6.283185_+19
s.l5_mu_0.012150_period_6.283185_-30
s.l5_mu_0.012150_period_6.283185_+30
s.l5_mu_0.012150_period_6.283185_-42
s.l5_mu_0.012150_period_6.283185_+42
s.l5_mu_0.012150_period_6.283185_-49
s.l5_mu_0.012150_period_6.283185_+49
s.l5_mu_0.012150_period_6.283185_-56
s.l5_mu_0.012150_period_6.283185_+56
s.l5_mu_0.012150_period_6.283185_-67
s.l5_mu_0.012150_period_6.283185_+67
s.l5_mu_0.012150_period_6.283185_-79
s.l5_mu_0.012150_period_6.283185_+79
s.l5_mu_0.012150_period_6.283185_-86
s.l5_mu_0.012150_period_6.283185_+86
s.l5_mu_0.012150_period_6.283185_-92
s.l5_mu_0.012150_period_6.283185_+92
s.l5_mu_0.012150_period_6.582675
s.l5_mu_0.012150_period_6.582675~
s.l5_mu_0.012150_period_6.582675_-56
s.l5_mu_0.012150_period_6.582675_+56

Table 4.7. These are the earth-moon L5 equilibrium point periodic orbit families.

Table 4.7 identified the files produced for the L5 equilibrium point. One can see that the initial orbit periods are identical to those associated with the L4 equilibrium point. However, it is interesting to note that the total number of files or number of files containing bifurcation points is not identical. One would initially think the opposite, since the L4 and L5 equilibrium points are at the same energy level, i.e. the value for the Jacobi Constant or constant of integration is the same for each point. Even though AUTO 2000 is using a flexible set of parameters to produce valid/real solutions, these solutions are highly unstable do not necessarily produce identical results. To give one a sense of the number of periodic orbits possible in a given L5 orbit family, the “clam shell” family for an initial

orbit period of 6.283185 are shown in Figures 4.6 through 4.8. Again, the XY planar projection is essentially a view from the celestial North Pole. The XZ planar projection can be regarded as a side-view of the system, and the YZ planar projection is a view from the axis joining the two primary bodies. Again, one can see that there is symmetry in each of the views.

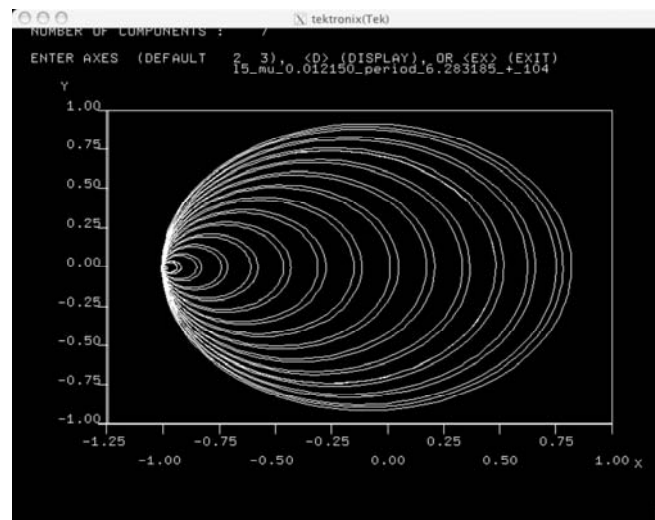


Figure 4.6. AUTO 2000 plot of the earth-moon L5 equilibrium point periodic orbit family. Initial orbit period = 6.283185. [XY planar projection]

What AUTO 2000 has essentially done in solving a two-point boundary value problem is to move off the initial orbit periods, solve for position and velocity along the length of the curve, i.e. orbit, determine if there is a closed orbit, and identify the period for this closed orbit. In both L4 and L5 equilibrium point cases, most of the orbit families are three-dimensional (i.e. most orbit points have Z-axis position and/or velocity components). A complete catalogue of periodic orbits is given in Appendix D. However, to simplify the initial investigation of this problem, the focus was on a two-dimensional (XY) or planar case first. This allowed for a relatively simple problem to be solved first before examining and also solving for the more complex three-dimensional case.

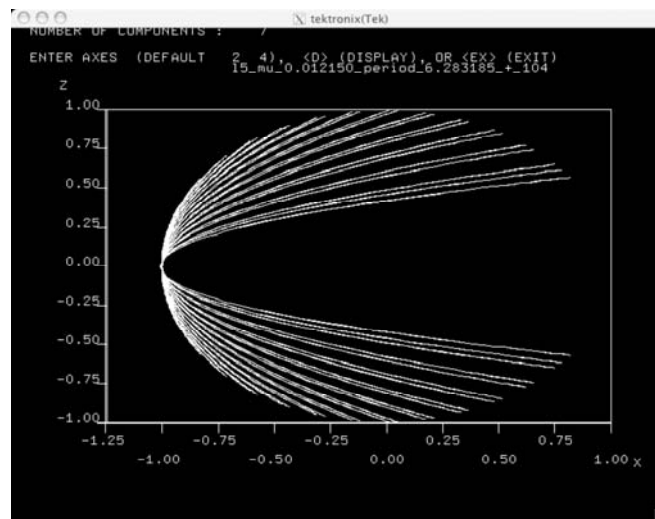


Figure 4.7. AUTO 2000 plot of the earth-moon L5 equilibrium point periodic orbit family. Initial orbit period = 6.283185. [XZ planar projection]

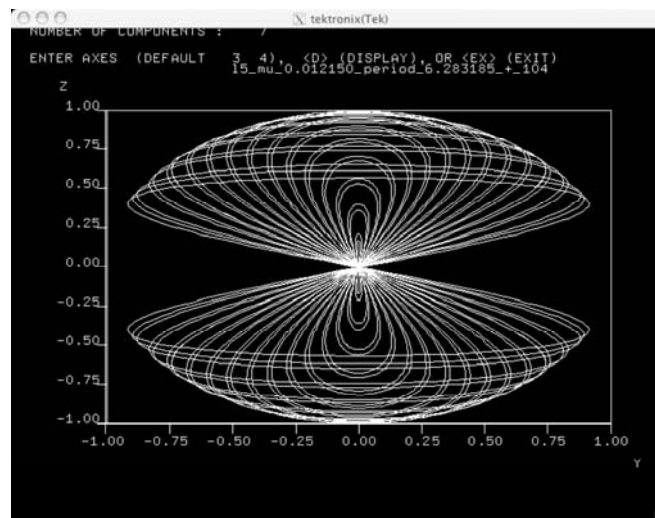


Figure 4.8. AUTO 2000 plot of the earth-moon L5 equilibrium point periodic orbit family. Initial orbit period = 6.283185. [YZ planar projection]

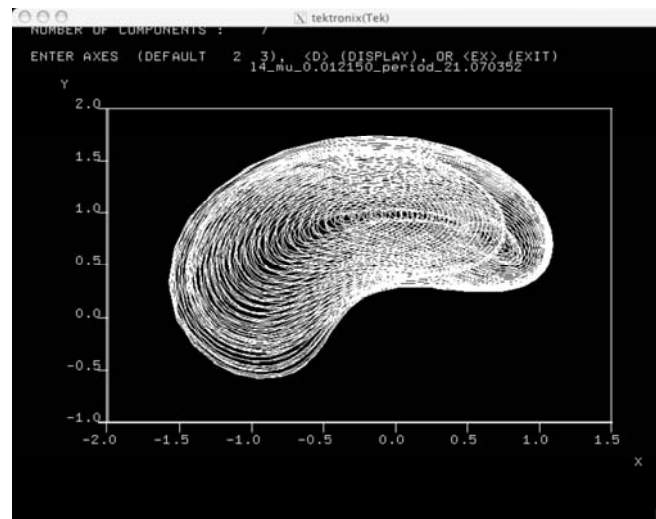


Figure 4.9. AUTO 2000 plot of the earth-moon L4 equilibrium point periodic orbit family. Initial orbit period = 21.070352. [XY planar projection]

Figure 4.9 through 4.11 represents an L4 “kidney bean” orbit family. All of the orbits in this family are planar and can be readily seen in Figures 4.10 and 4.11.

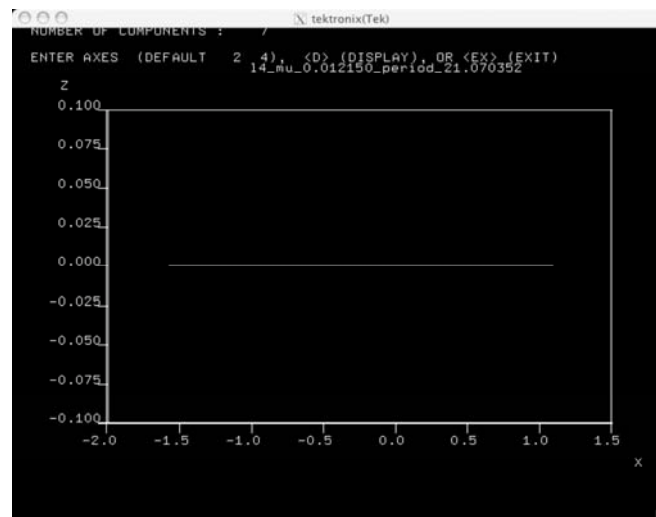


Figure 4.10. AUTO 2000 plot of the earth-moon L4 equilibrium point periodic orbit family. Initial orbit period = 21.070352. [XZ planar projection]

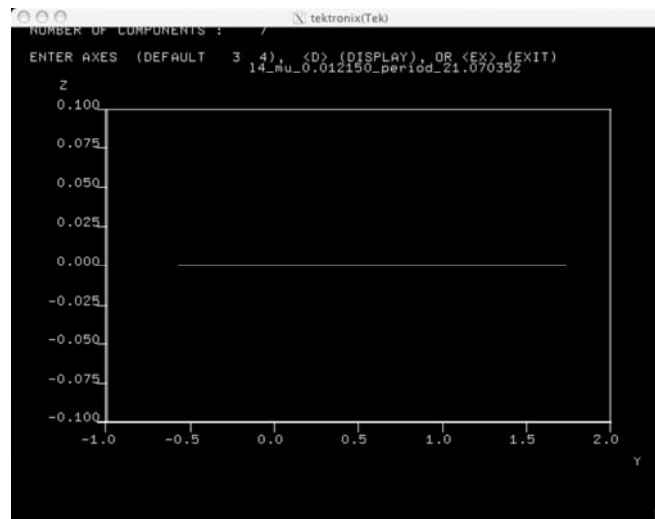


Figure 4.11. AUTO 2000 plot of the earth-moon L4 equilibrium point periodic orbit family. Initial orbit period = 21.070352. [YZ planar projection]

Figure 4.12 through 4.14 identifies an L5 “kidney bean” orbit family. All of the orbits in this family are planar and can be readily seen in Figures 4.13 and 4.14.

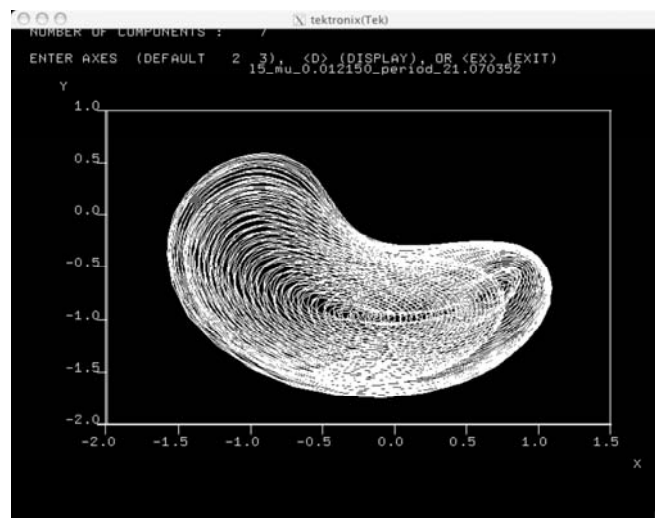


Figure 4.12. AUTO 2000 plot of the earth-moon L5 equilibrium point periodic orbit family. Initial orbit period = 21.070352. [XY planar projection]

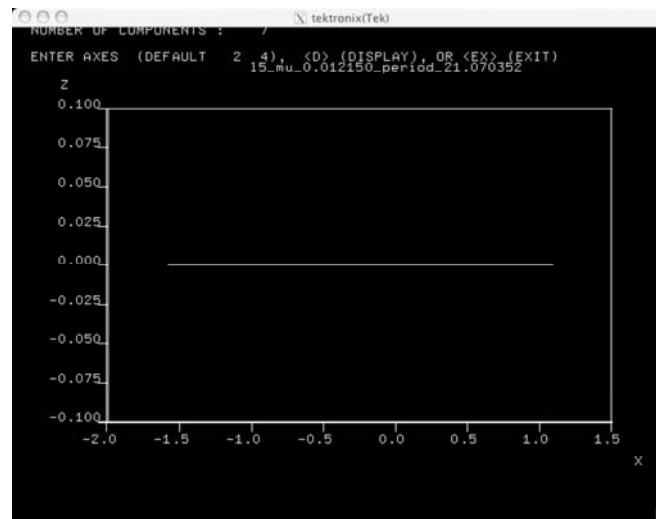


Figure 4.13. AUTO 2000 plot of the earth-moon L5 equilibrium point periodic orbit family. Initial orbit period = 21.070352. [XZ planar projection]

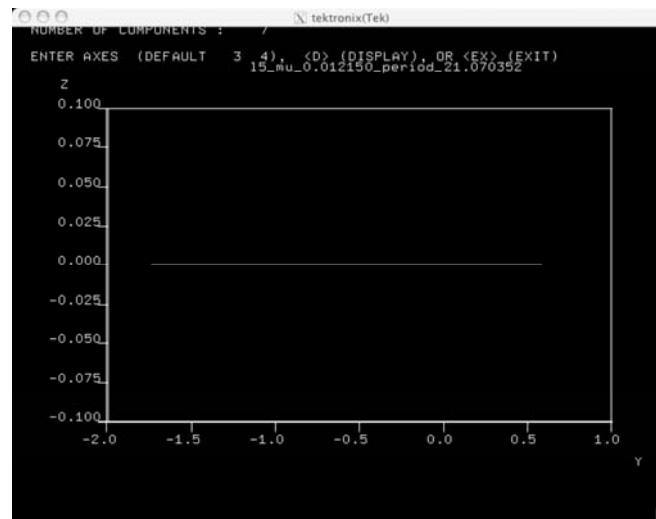


Figure 4.14. AUTO 2000 plot of the earth-moon L5 equilibrium point periodic orbit family. Initial orbit period = 21.070352. [YZ planar projection]

For each of the forty-seven orbits that comprise the L4 “kidney bean” orbit family, the periods range from approximately 21.07 to approximately 26.50. A plot is provided in Figure 4.15 and some

representative orbit shapes are shown in Figure 4.16. One can see that the periodic orbits do not conform to the conic sections that most individuals are accustomed to seeing, i.e. circle, ellipse, parabola, or hyperbola. Many of the trajectories have small loops or move about the L4 equilibrium point a number of times before the orbits close on themselves. It should be noted that the L5 “kidney bean” orbit family results are similar if not identical. The average distance from the respective equilibrium point for each L4 and L5 orbit family are shown in Figure 4.17 and 4.18, respectively. In addition, the average distance from the respective equilibrium point for each L4 and L5 orbit family as a function of orbit period are shown in Figure 4.19 and 4.20, respectively. One can see that the L4 and L5 plots are not identical. However, this has more to do with the method for determining these parameters of interest. Specific, finite points on each of the orbits were used to determine the average distance. If the element step size was infinitesimally small, the plots shown in Figures 4.17 and 4.18 would be identical, as would Figures 4.19 and 4.20.

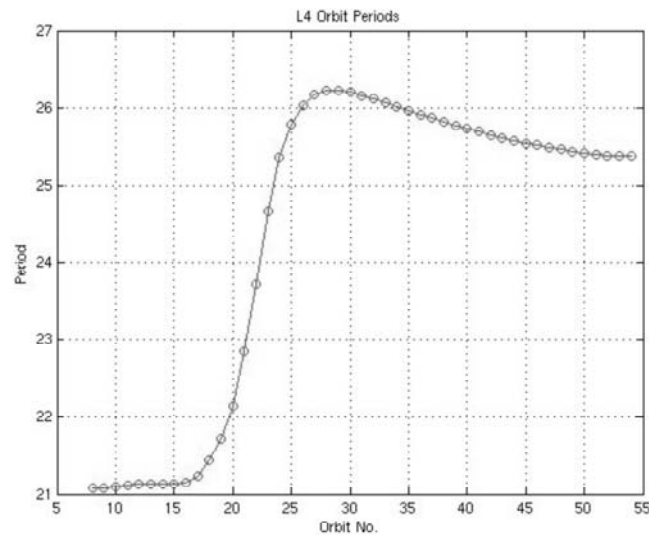


Figure 4.15. This is a MATLAB plot of actual periods for each L4 equilibrium point periodic orbit. Primary orbit period = 21.070352.

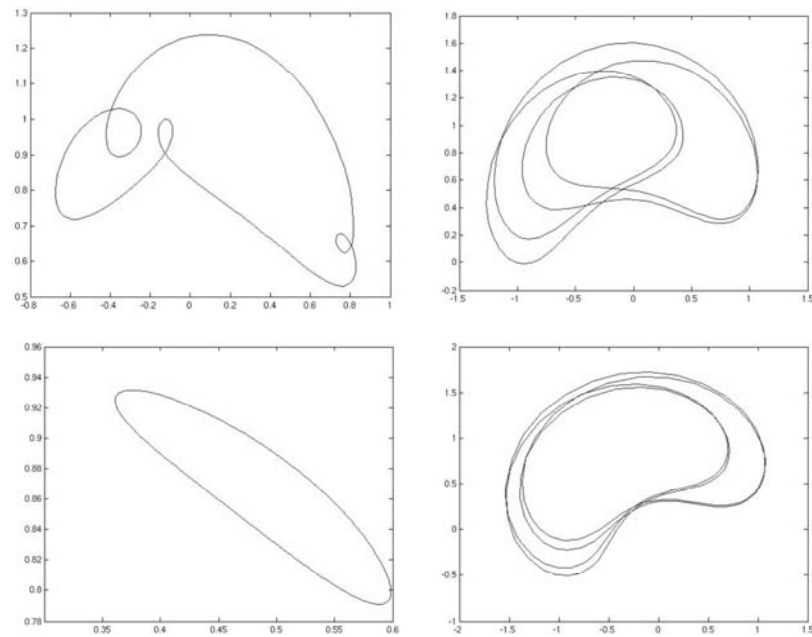


Figure 4.16. Orbit traces for the earth-moon L4 equilibrium point periodic orbit family. Primary orbit period = 21.070352. Clockwise from lower left: (a) 2.10944E+01, (b) 2.57786E+01, (c) 2.57317E+01, and (d) 2.54126E+01. [Note: The abscissa and ordinate scales are not identical for the four subplots.]

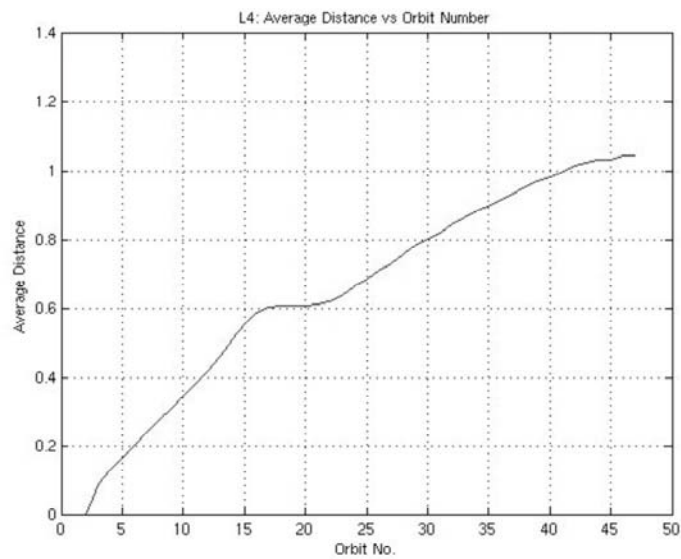


Figure 4.17. This is a MATLAB plot of average distance from the L4 equilibrium point for each periodic orbit. Primary orbit period = 21.070352.

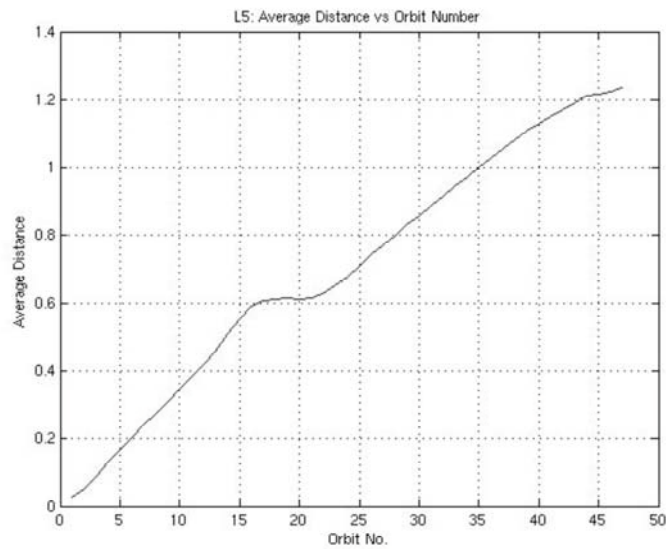


Figure 4.18. This is a MATLAB plot of average distance from the L5 equilibrium point for each periodic orbit. Primary orbit period = 21.070352.

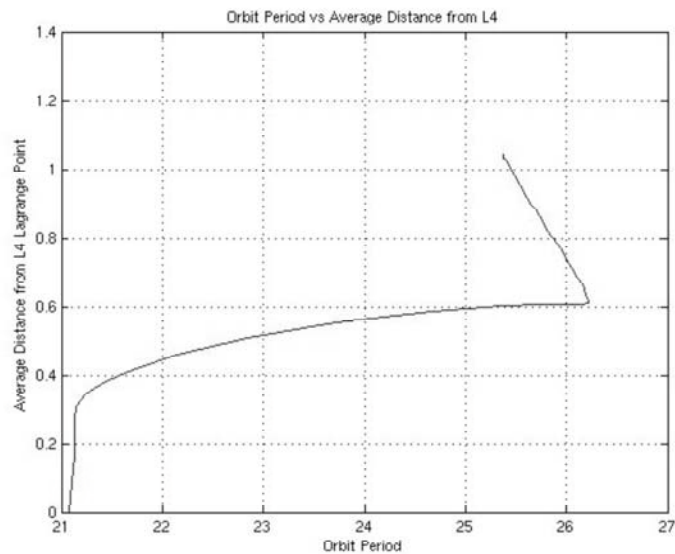


Figure 4.19. This is a MATLAB plot of average distance from the L4 equilibrium point as a function of period. Primary orbit period = 21.070352.

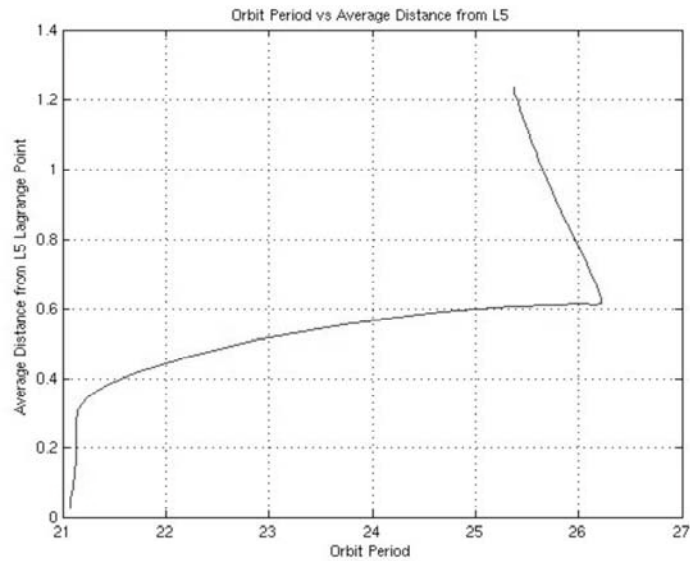


Figure 4.20. This is a MATLAB plot of average distance from the L5 equilibrium point as a function of period. Primary orbit period = 21.070352.

4.5 General Stability and Other Periodic Orbit Characteristics

Several Lyapunov (planar) orbits of the earth-moon L4 equilibrium point periodic orbit family generated in AUTO 2000 shown in Figure 4.9 were chosen for examination. These are shown in Figure 4.21 below. It will be shown later that four spacecraft each traveling along one of the four outer orbits can be phased-locked on the inner most periodic orbit and placed in a desired formation. The first question asked was whether or not MATLAB and AUTO 2000 could produce comparable results in generating periodic orbits. An arbitrary state vector, i.e. position and velocity, of the outer most periodic orbit was selected. It then became the initial condition for the MATLAB initial value problem. Propagating the point forward in time circumscribed the outer most trajectory shown in

Figure 4.22. As one can see, the trajectory closely resembles the closed periodic orbit generated by AUTO 2000 validating the MATLAB simulation.

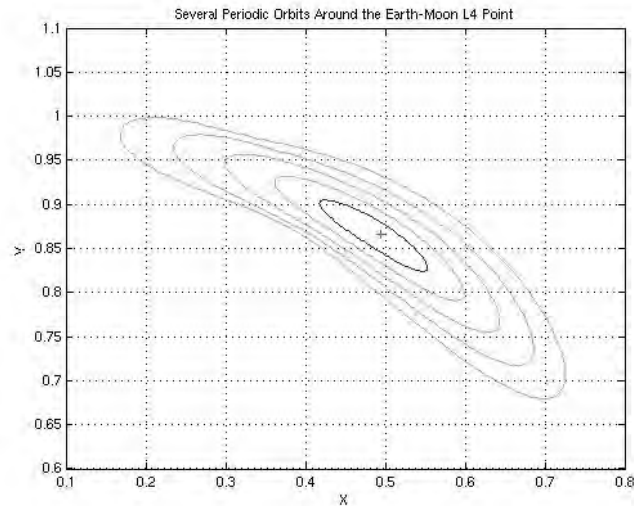


Figure 4.21. Five periodic orbits around the earth-moon L4 equilibrium point are shown in this plot.

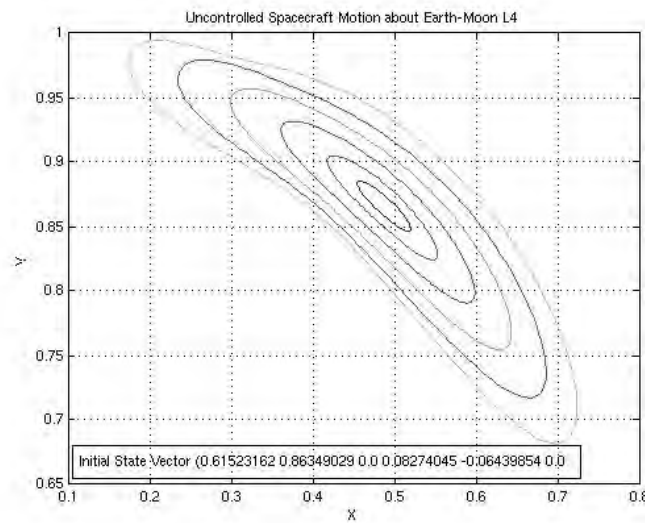


Figure 4.22. An arbitrary state vector, i.e. position and velocity, of the AUTO 2000 generated outer most periodic orbit shown in Figure 4.21 was used as an initial condition for a MATLAB initial value problem. One can clearly see that propagating the trajectory over time produces a closed periodic orbit.

The stability of periodic orbits can be found in a manner similar to that used for equilibrium points. Floquet theory and the monodromy matrix provide the means for analyzing each periodic orbit [see Appendix E]. As part of the periodic orbit generation process, AUTO 2000 produces a diagnostic file containing the Floquet multipliers, which are simply the eigenvalues of the characteristic or monodromy matrix. The Floquet multipliers for the five periodic orbits shown in Figure 4.21 are given in Tables 4.8 through 4.12.

Multiplier No.	Real Component	Imaginary Component
0	1.000000	0.00000
1	-0.6054816	0.795859
2	-0.6054816	-0.795859
3	0.3036922	-0.952770
4	0.3036922	0.952770
5	1.000000	0.00000

Table 4.8. Floquet multipliers for the inner most periodic orbit shown in Figure 4.21.

Multiplier No.	Real Component	Imaginary Component
0	1.000000	0.00000
1	-0.6055769	0.795787
2	-0.6055769	-0.795787
3	1.000000	0.00000
4	0.3036721	0.952777
5	0.3036721	-0.952777

Table 4.9. Floquet multipliers for the second smallest periodic orbit shown in Figure 4.21.

Multiplier No.	Real Component	Imaginary Component
0	1.000000	0.00000
1	-0.6056854	-0.795704
2	-0.6056854	0.795704
3	0.3036495	-0.952784
4	0.3036495	0.952784
5	1.000000	0.00000

Table 4.10. Floquet multipliers for the third or middle periodic orbit shown in Figure 4.21.

Multiplier No.	Real Component	Imaginary Component
0	1.000000	0.00000
1	-0.6058069	0.795612
2	-0.6058069	-0.795612
3	1.000000	0.00000
4	0.3036245	-0.952792
5	0.3036245	-0.952792

Table 4.11. Floquet multipliers for the second largest periodic orbit shown in Figure 4.21.

Multiplier No.	Real Component	Imaginary Component
0	1.000000	0.00000
1	-0.6059415	0.795509
2	-0.6059415	-0.795509
3	1.000000	0.00000
4	0.3035971	0.952801
5	0.3035971	-0.952801

Table 4.12. Floquet multipliers for the outer most periodic orbit shown in Figure 4.21.

Since the Floquet multipliers all have a modulus less than or equal to 1.0, the periodic orbits are stable, i.e. each multiplier when plotted on a complex coordinate system would either be on or within a unit circle with the center at the origin.

Since each of the orbits resembled ellipses, a question arose as to whether or not the standard equation for the area of an ellipse, see equation 4.104 below, would be a good approximation of the actual area.

$$a = \pi AB \quad 4.104$$

The area, a , is merely equal to the product of π , the semi-major axis, A , and the semi-minor axis, B . Table 4.13 provides the maximum and minimum distances from the L4 equilibrium point. These distances are synonymous with the semi-major and semi-minor axes, respectively. Green's theorem was used to calculate the actual area of the circumscribed by each orbit. Green's theorem is given as

$$\oint_C Pdx + Qdy = \iint_R \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dA \quad 4.105$$

In order to use Green's theorem to determine the area bound by a closed curve, P and Q must be selected such that

$$\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} = 1 \quad 4.106$$

Equation 4.105 then becomes

$$\oint_C Pdx + Qdy = \iint_R dA \quad 4.107$$

However this equation can be expressed as

$$\iint_R dA = x \int_C dy = -y \int_C dx = \frac{1}{2} \int_C xdy - ydx = Area \quad 4.108$$

Orbit No.	Minimum Distance	Maximum Distance	Period	Angular Rate
1 (Inner Most)	0.0042011	0.043135	21.0730	0.2982
2	0.012078	0.084334	21.0811	0.2980
3	0.023331	0.14577	21.0994	0.2978
4	0.035131	0.21119	21.1196	0.2975
5	0.046156	0.27794	21.1334	0.2973
6 (Outer Most)	0.055076	0.34547	21.1365	0.2973

Table 4.13. General attributes of the six periodic orbits shown in Figure 4.22 are provided in this table.

The closed curve, C , could then be parameterized as follows

$$C \left\{ \begin{array}{l} x_1 + (x_2 - x_1)t \\ y_1 + (y_2 - y_1)t \end{array} \right\}, 0 \leq t \leq 1$$

$$dx = (x_2 - x_1)dt$$

$$dy = (y_2 - y_1)dt$$

$$\frac{1}{2} \int_C xdy - ydx = \frac{1}{2} \int_C \{ [x_1 + (x_2 - x_1)t](y_2 - y_1)dt - [y_1 + (y_2 - y_1)t](x_2 - x_1)dt \}$$

$$\begin{aligned}
\frac{1}{2} \int_C xdy - ydx &= \frac{1}{2} \int_C \{ [x_1(y_2 - y_1) + (x_2 - x_1)(y_2 - y_1) - [y_1(x_2 - x_1) + (y_2 - y_1)(x_2 - x_1)] \} dt \\
\frac{1}{2} \int_C xdy - ydx &= \frac{1}{2} \int_C (x_1y_2 - x_1y_1 - y_1x_2 + y_1x_1) dt \\
\frac{1}{2} \int_C xdy - ydx &= \frac{1}{2} \int_C (x_1y_2 - y_1x_2) dt \\
\frac{1}{2} \int_C xdy - ydx &= \frac{1}{2} (x_1y_2 - x_2y_1)
\end{aligned} \tag{4.109}$$

Equation 4.109 was encoded in a simple MATLAB script to expedite the problem solving process. Table 4.14 shows the area of each periodic orbit shown in Figure 4.22 as determined by the ellipse formula and Green's theorem. One can clearly see that the former provides a good approximation for the area bound by a periodic orbit.

Orbit No.	Period	Frequency	Ellipse Formula	Green's Theorem
1 (Inner Most)	21.0730	3.3539	0.000569	0.000848
2	21.0811	3.3552	0.0032	0.0036
3	21.0994	3.3581	0.0107	0.0110
4	21.1196	3.3613	0.0233	0.0228
5	21.1334	3.3635	0.040	0.0386
6 (Outer Most)	21.1365	3.3640	0.0598	0.0586

Table 4.14. The area for each of the six periodic orbits shown in Figure 4.22, using the ellipse formula and Green's theorem are shown in this table. One can clearly see that the ellipse formula provides a good approximation.

It was asked whether or not there are any special relationships between the orbit frequency and the area for the six periodic orbits being examined, e.g. power law relationship. Figure 4.23 is a simple linear plot of the two values. One can see that the curve is non-linear. Figures 4.24 and 4.25 are semi-log plots, and Figure 4.26 is a log-log plot. In each of these three cases as well, there doesn't seem to be a simple relationship.

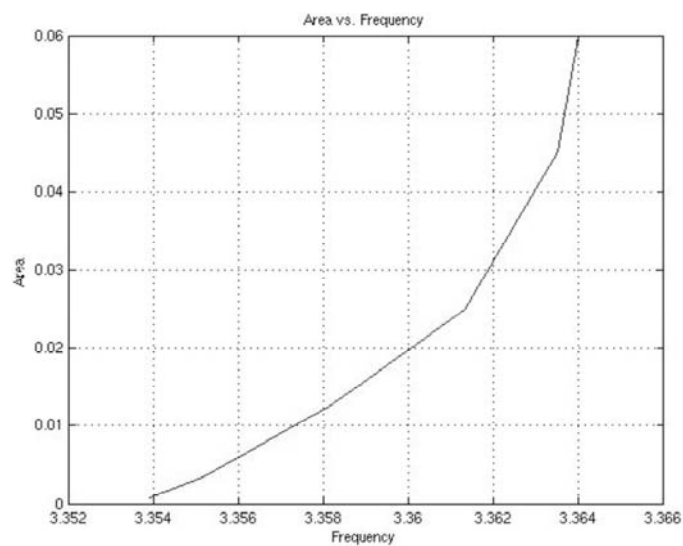


Figure 4.23. A linear plot of orbit frequency versus area is shown above.

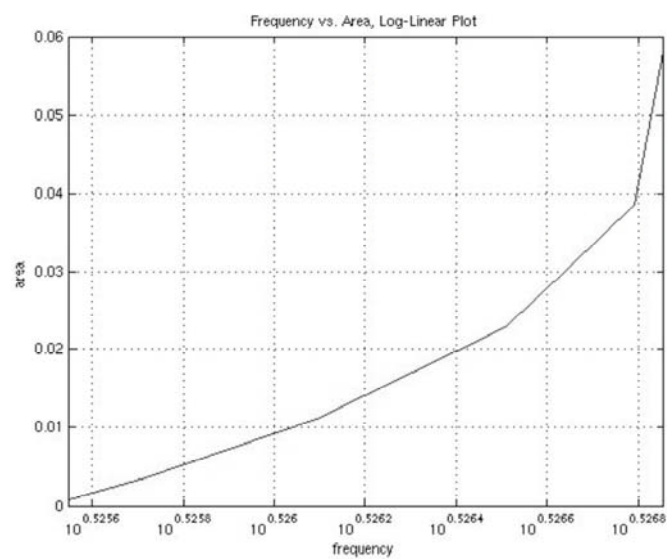


Figure 4.24. A semi-log plot of orbit frequency versus area is shown above.

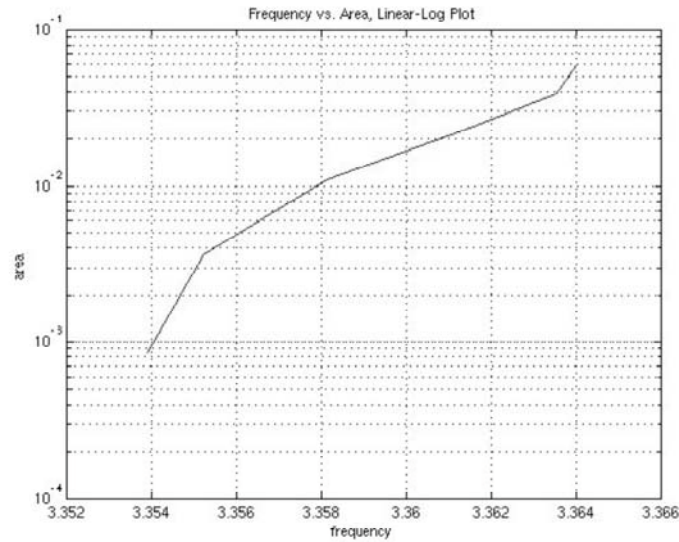


Figure 4.25. Another semi-log plot of orbit frequency versus area is shown above.

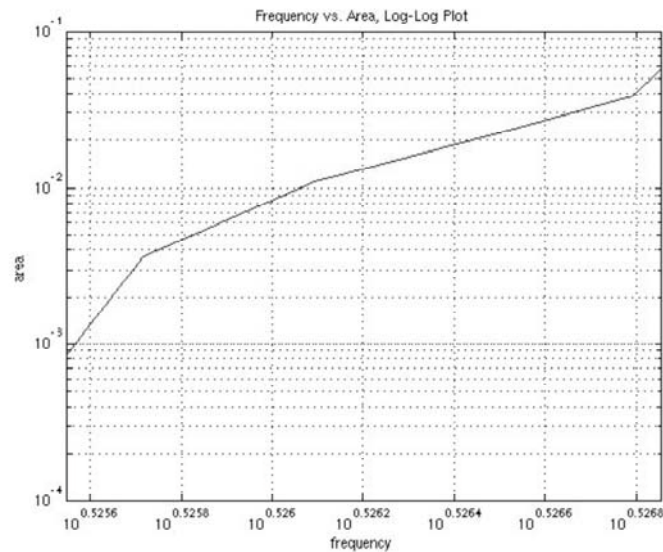


Figure 4.26. A log plot of orbit frequency versus area is shown above.

Finally, it was asked whether or not there are any special relationships between characteristic distance, i.e. minimum distance, and orbit period for the six periodic orbits being examined, e.g. power law relationship.

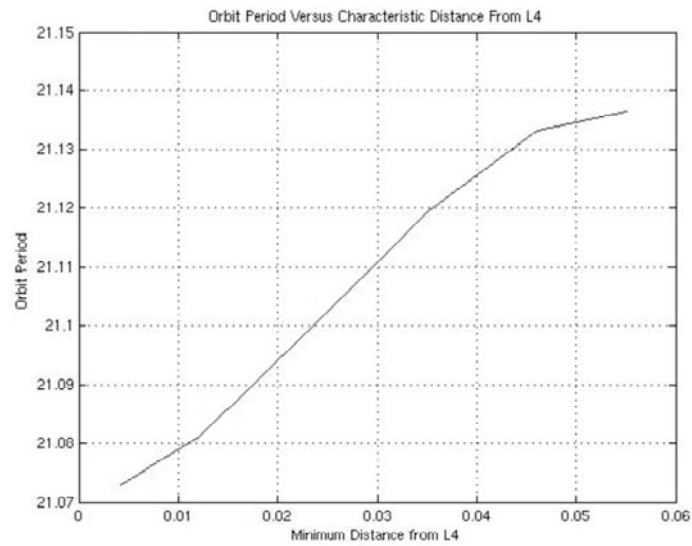


Figure 4.27. A linear plot of minimum distance versus orbit period is shown above.

Figure 4.27 is a simple linear plot of the two values. One can see that the curve is non-linear. Figures 4.28 and 4.29 are semi-log plots, and Figure 4.30 is a log-log plot. In each of these three cases as well, there doesn't seem to be a simple relationship.

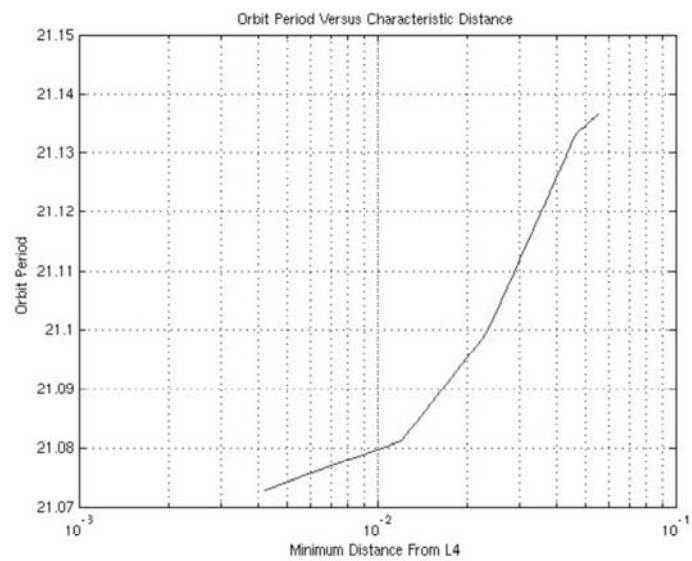


Figure 4.28. A semi-log plot of minimum distance versus orbit period is shown above.

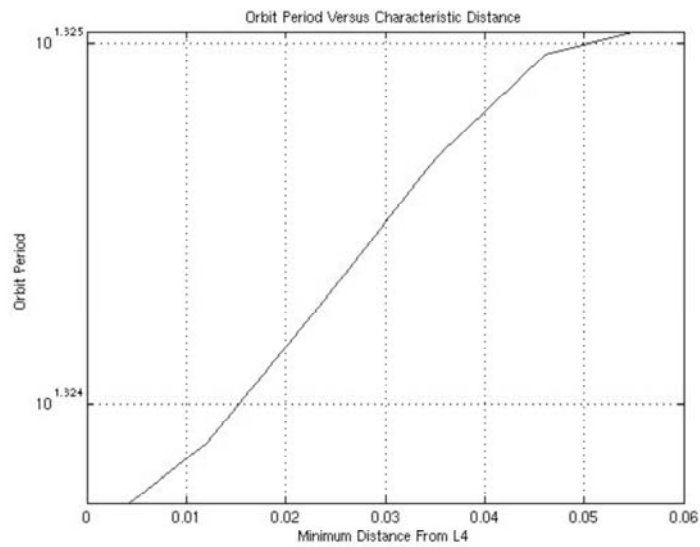


Figure 4.29. Another semi-log plot of minimum distance versus orbit period is shown above.

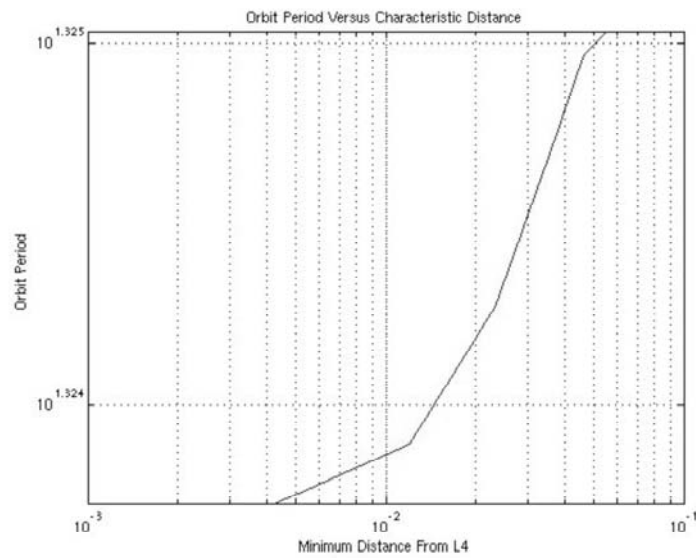


Figure 4.30. A log plot of minimum distance versus orbit period is shown above.

The five earth-moon L4 periodic orbits examined for the three-dimensional case were taken from the $l4_mu_0.012150_period_6.283185_+16$ periodic orbit family (see Appendix D) and are

shown in Figure 4.31. The orbit identification numbers from the outer to inner most are: 126, 128, 130, 132, and 134.

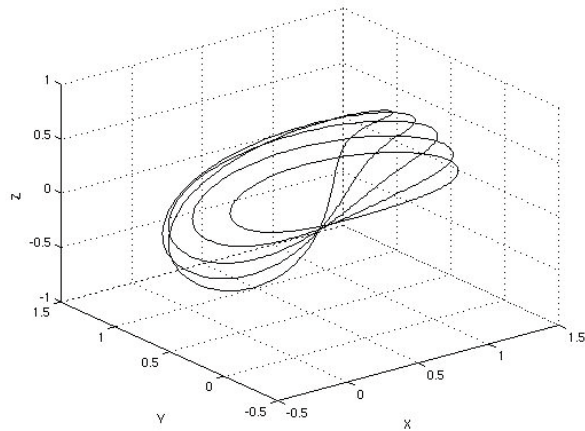


Figure 4.31. Five periodic orbits around the earth-moon L4 equilibrium point are shown in this plot.

As with the Lyapunov (planar) case, it will be shown later that four spacecraft each traveling along one of the four outer orbits can be phased-locked on the inner most periodic orbit and placed in a desired formation.

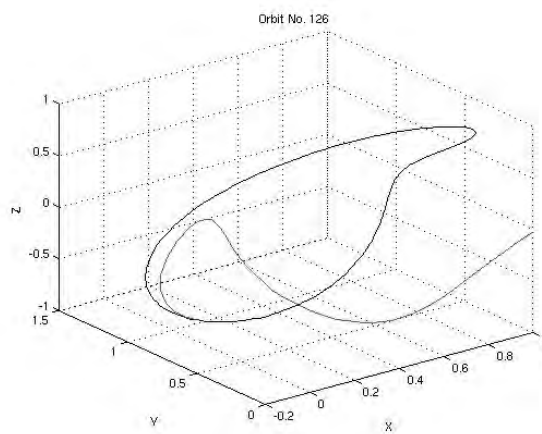


Figure 4.32. Beginning with an arbitrary state vector for the outer most periodic orbit, it is clear that a MATLAB initial value problem is unable to replicate the periodic orbit generated by AUTO 2000.

A question was asked regarding whether or not MATLAB and AUTO 2000 could produce comparable results in generating periodic orbits. An arbitrary state vector, i.e. position and velocity, of the outer most periodic orbit was selected. It then became the initial condition for the MATLAB initial value problem. Propagating the point forward in time produced the trajectory shown in Figure 4.32. As one can see, the trajectory the MATLAB initial value problem is unable to replicate the AUTO 2000 two-point value problem produced periodic orbit. Given this, the Floquet multiplier values were checked to determine if the periodic orbits are stable or unstable. The Floquet multipliers are shown in Tables 4.15 through 4.19.

Multiplier No.	Real Component	Imaginary Component
0	1.000000	0.00000
1	0.2753528	-0.961343
2	0.2753528	0.961343
3	1.000000	0.00000
4	0.9050145	0.00000
5	1.104955	0.00000

Table 4.15. Floquet multipliers for the inner most periodic orbit (no. 126) shown in Figure 4.31.

Multiplier No.	Real Component	Imaginary Component
0	1.000000	0.00000
1	0.2788031	-0.960348
2	0.2788031	0.960348
3	1.000000	0.00000
4	0.9029416	0.00000
5	1.107491	0.00000

Table 4.16. Floquet multipliers for the second smallest periodic orbit (no. 128) shown in Figure 4.31.

Multiplier No.	Real Component	Imaginary Component
0	1.000000	0.00000
1	0.2823921	0.959299
2	0.2823921	-0.959299
3	1.000000	0.00000
4	0.9008275	0.00000
5	1.110090	0.00000

Table 4.17. Floquet multipliers for the third or middle periodic orbit (no. 130) shown in Figure 4.31.

Multiplier No.	Real Component	Imaginary Component
0	1.000000	0.00000
1	0.2861243	-0.958193
2	0.2861243	0.958193
3	1.000000	0.00000
4	0.8986698	0.00000
5	1.112756	0.00000

Table 4.18. Floquet multipliers for the second largest periodic orbit (no. 132) shown in Figure 4.31.

Multiplier No.	Real Component	Imaginary Component
0	1.000000	0.00000
1	0.2900044	0.957025
2	0.2900044	-0.957025
3	1.000000	0.00000
4	0.8964660	0.00000
5	1.115491	0.00000

Table 4.19. Floquet multipliers for the outer most periodic orbit (no. 134) shown in Figure 4.31.

For each three-dimensional orbit, there is at least one Floquet multiplier with a modulus greater than 1.0. Therefore, all five periodic orbits are unstable, i.e. at least multiplier when plotted on a complex coordinate system would either be outside the unit circle with the center at the origin. While the MATLAB initial value problem method produces valid controlled motion trajectories and planar periodic orbits (at least the five being examined), it cannot be generally used to produce three-dimensional periodic orbits.

Orbit No.	Period	Frequency	Area
126	6.27470	1.0014	1.3533
128	6.25227	1.0049	1.4603
130	6.20294	1.0129	1.4373
132	6.09248	1.0313	1.2363
134	5.87161	1.0701	0.9572

Table 4.20. The area, derived from Stokes' theorem, for each earth-moon L4 equilibrium point periodic orbit is shown in this table.

Finally, similar to what was done using Green's theorem from planar orbits, Stokes' theorem was used to calculate the area bounded by each periodic orbit. Results are shown in Table 4.20.

4.6 Problem No. 1: Lyapunov (Planar) Case

In the previous chapter, it was determined that phase-locking and formation establishment in the circular, restricted three-vortex problem in fluid mechanics is possible. A controller was used in conjunction with a resonant frequency (or orbit resonance) approach to produce the desired result. This controller was essentially an additional term to the standard equation of motion. It was also shown that the same controller could be used in combination with another, i.e. this time a scale factor was incorporated in the standard equation of motion, to expedite the entire process. The next step was to carry forward these methods to the Lyapunov (planar) case of the circular, restricted three-body problem in celestial mechanics.

4.6.1 Periodic Orbits

The two-dimensional or planar orbits examined in Section 4.5, was used to develop the two controllers needed. Again, it should be noted that contrary to the circular, restricted three-vortex problem in the fluid mechanics, spacecraft motion is clockwise on the XY-plane when viewed from the positive Z-axis. Shown in Figure 4.21, are the five periodic orbits that were used in the study. The four spacecraft travel along the four outer periodic orbits, while the inner most periodic orbit is the desired orbit for phase-locking and formation establishment.

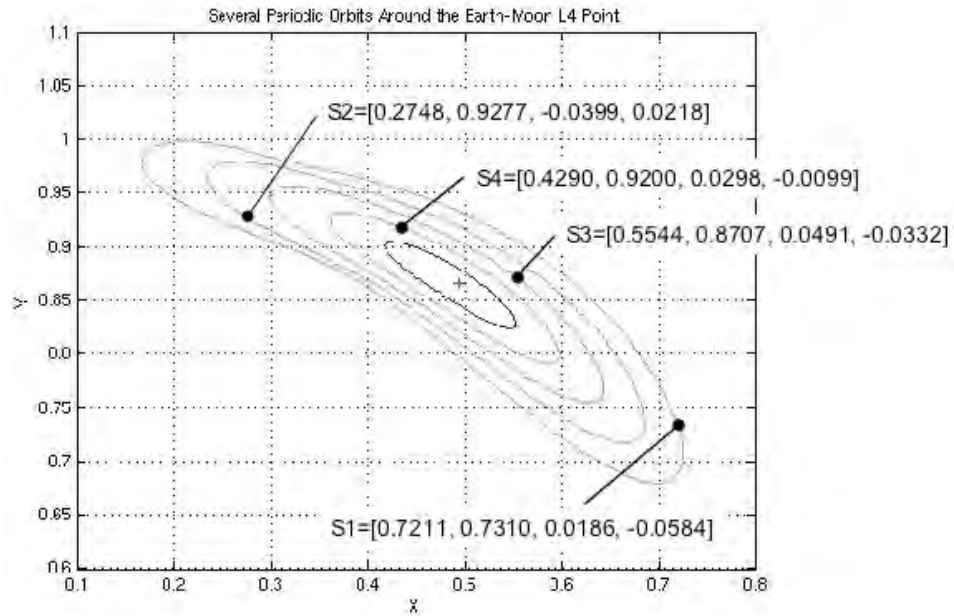


Figure 4.33. The initial condition state vector for each of the four spacecraft is shown in this plot. The terms given in each state vector are, in the order shown, X-axis position, Y-axis position, velocity in the X-axis direction, and velocity in the Y-axis direction.

In Figure 4.33, initial conditions, selected randomly, for each of the four spacecraft are shown in the plot. The state vectors for each are given in the form $[x, y, u, v]$, where x and y are the positions along and u and v are the velocities in direction of the X-axis and Y-axis, respectively. As done in the circular, restricted three-vortex problem in fluid mechanics, animations were created for the circular, restricted three-body problem in celestial mechanics. The first of these is represented in Figure 4.34.

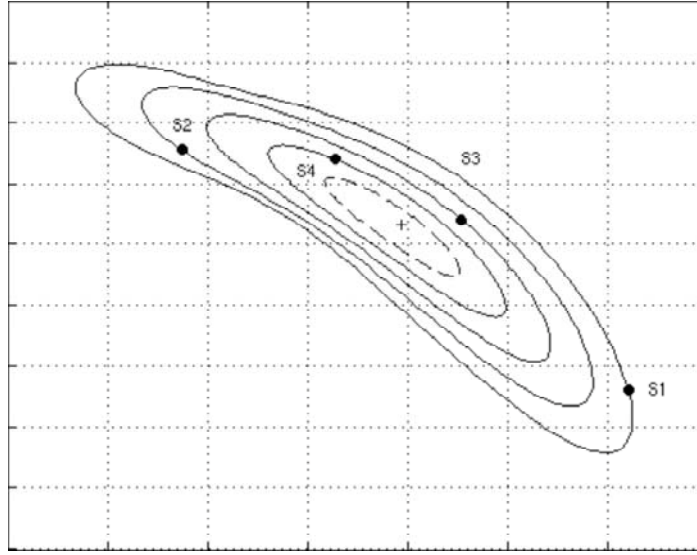


Figure 4.34. This is the first frame of the two hundred frames in the first animation. Spacecraft motion is clockwise when viewed from the Z-axis direction. Since each spacecraft is only driven by the standard equations of motion, i.e. uncontrolled motion state, they will continue to traverse the periodic orbits they are on until a controller is enabled or turned ON.

4.6.2 Phase-Lock Controller

Equations 4.22 and 4.23 are the two standard equations of motion for the planar case of the circular, restricted three-body problem in celestial mechanics. The controller term was added to each resulting in the following equations:

$$\ddot{x} - 2\dot{y} - x = -\frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x-1+\mu)}{r_2^3} + \kappa \sin\left(\frac{\alpha\pi t}{T}\right) \quad 4.110$$

$$\ddot{y} + 2\dot{x} - y = \frac{-(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3} + \kappa \sin\left(\frac{\alpha\pi t}{T}\right) \quad 4.111$$

As in the circular, restricted three-vortex problem in fluid mechanics, κ is a term scale factor; α is equal to 1/2, 1, or 2; t is the time or the sine function angle multiplier; and T is the base orbit period.

The term here as well will be referred to as controller no. 1. Now refer to Figure 4.35. A spacecraft is initially traveling along the outer periodic orbit. At a somewhat arbitrary position, the controller is turned ON. After a series of ‘trial and error’ guesses, it was determined that if κ and T were set to -0.027 and 21.070352 , the spacecraft would traverse a trajectory that is tangent to the adjacent periodic orbit, which is precisely what was desired. This allows a spacecraft to move from an outer to an inner periodic orbit. Can the same controller be used to move a spacecraft from an inner to an outer periodic orbit?

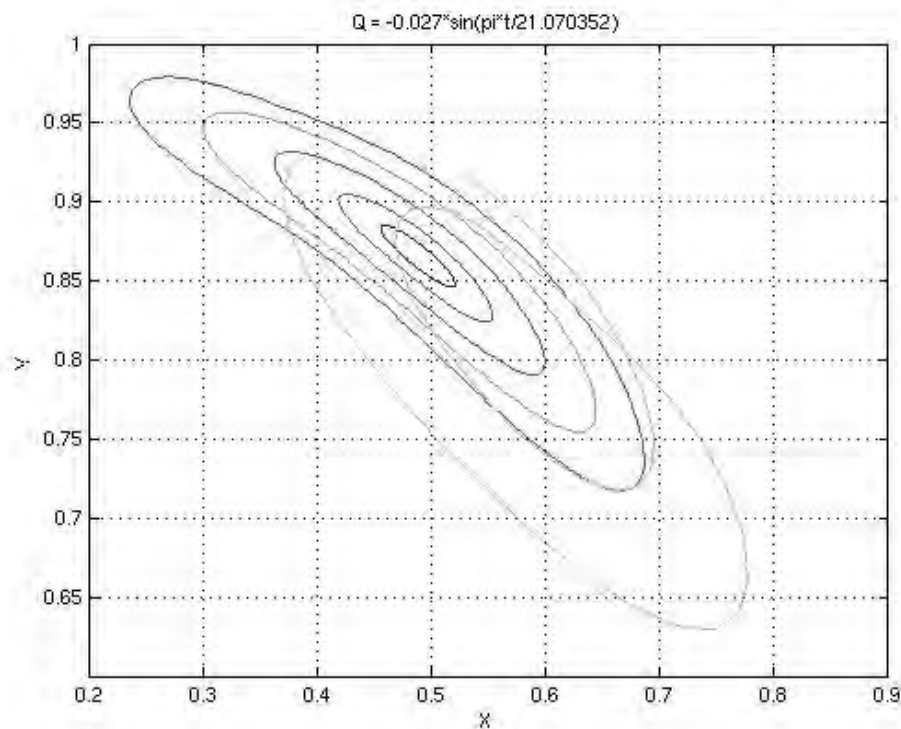


Figure 4.35. A spacecraft traveling along a periodic orbit is placed in controlled motion at a somewhat arbitrary initial position. For this specific case, the values for κ and T were set to -0.027 and 21.070352 , respectively. However, t was left as a variable. One can see that the spacecraft traverses a trajectory that is tangent to the adjacent periodic orbit.

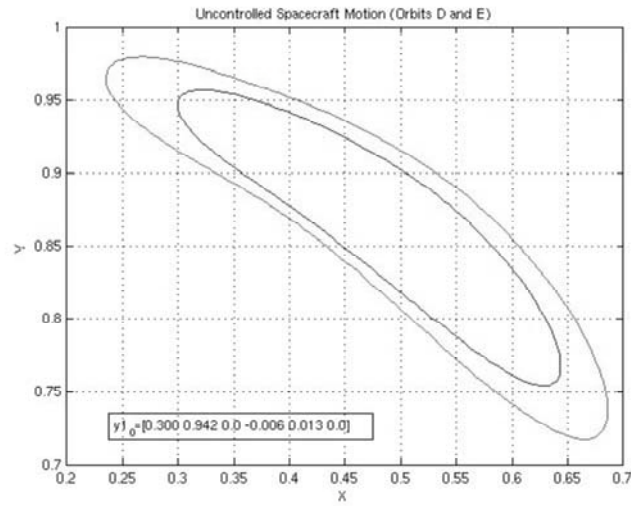


Figure 4.36. In the plot above two spacecraft travel along each periodic orbit subject only to the standard equations of motion. This is called the uncontrolled motion case.

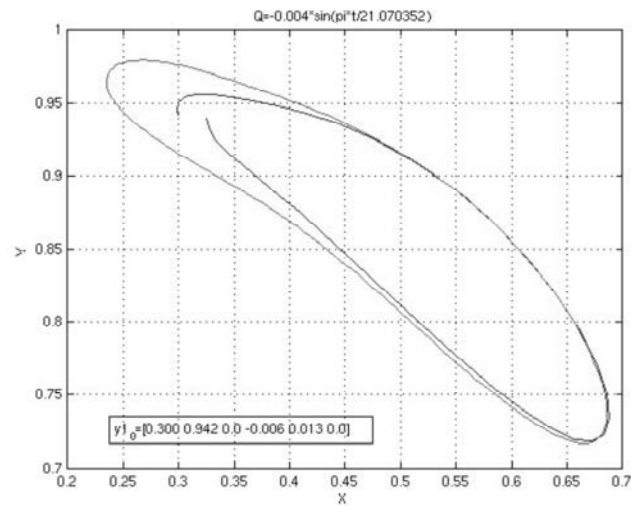


Figure 4.37. At position (0.300, 0.942) a spacecraft traveling along the inner periodic orbit is placed in controlled motion, i.e. when controller no. 1 is turned ON. For this specific case, the values for κ and T were set to 0.004 and 21.070352, respectively. One can see that the spacecraft traverses a trajectory that is tangent to the adjacent, outer periodic orbit.

In Figure 4.36, two periodic orbits are shown along with a selected initial state vector for a spacecraft traveling along the inner periodic orbit, i.e. $[0.300, 0.942, -0.006, 0.013]$. If the spacecraft continues to be driven only by the standard equations of motion it will continue along on the periodic orbit shown. It will be a different case if controller no. 1 is turned ON. After a series of ‘trial and error’ guesses, it was determined that if κ and T were set to 0.004 and 21.070352, the spacecraft would traverse a trajectory that is tangent to the adjacent periodic orbit, which is precisely what was desired (see Figure 4.37).

Now that controller no. 1 has shown to be viable, we now want to phase-lock the four spacecraft shown in Figures 4.33 and 4.34 onto a single orbit, that being the inner most one shown in the two figures. Later we will want to establish a formation on that orbit that resembles a diamond or rhombus (see Figure 4.38). For now, let’s decide on which extremas to use as staging points. After running several test cases, it was determined that the points of the periodic orbits furthest from the L4 equilibrium point would be the best staging points, i.e. transfer trajectory starting points. Figure 4.39 shows the initial condition state vector for each of the four spacecraft.

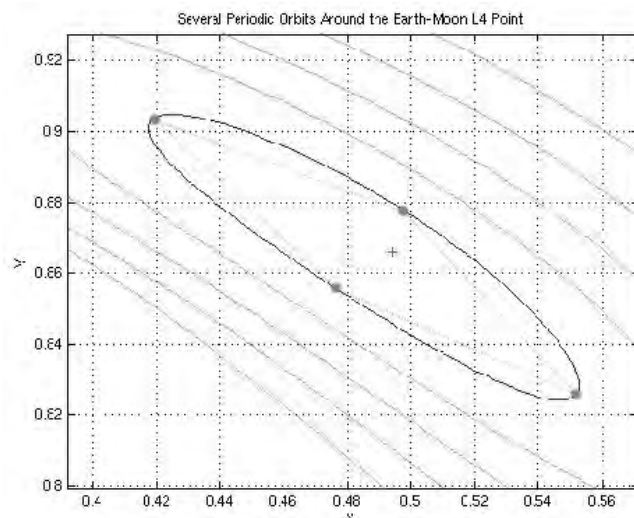


Figure 4.38. After completing the phase-locking step we want each spacecraft to be placed at the corners of a diamond or rhombus, i.e. establish a formation as shown above.

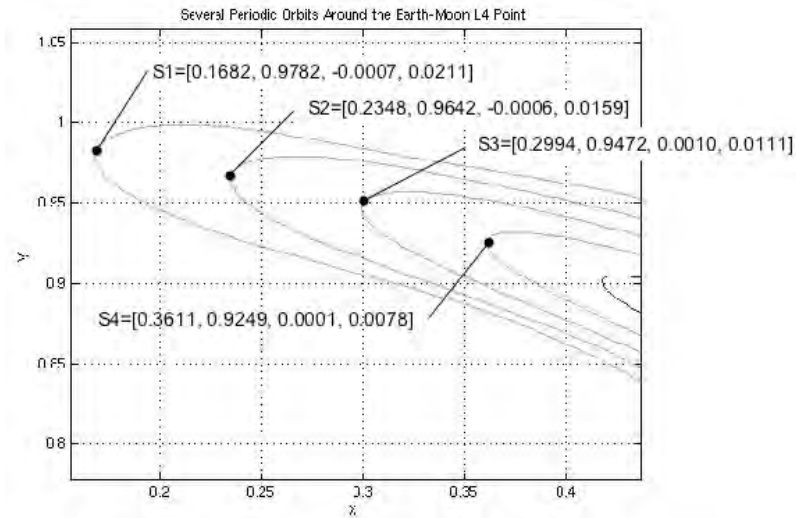


Figure 4.39. Staging points for each of the four spacecraft are shown above. They represent the points at which the periodic orbits are further away from the L4 equilibrium point.

After several iterations, the proper values for each spacecraft transfer trajectory were determined and are shown in Table 4.21 and a plot of each is shown in Figure 4.40. All four transfer trajectory curves are plotted on a single graph in Figure 4.41 to provide one with a relative sense of location in physical space.

Spacecraft No.	κ	α	T
1	0.0096	1	21.070352
2	0.0089	1	21.070352
3	0.0073	1	21.070352
4	0.0004	1	21.070352

Table 4.21. The controller no. 1 parameter values required to allow each spacecraft to leave its initial periodic orbit and be placed on the proper transfer trajectory are shown in the table above.

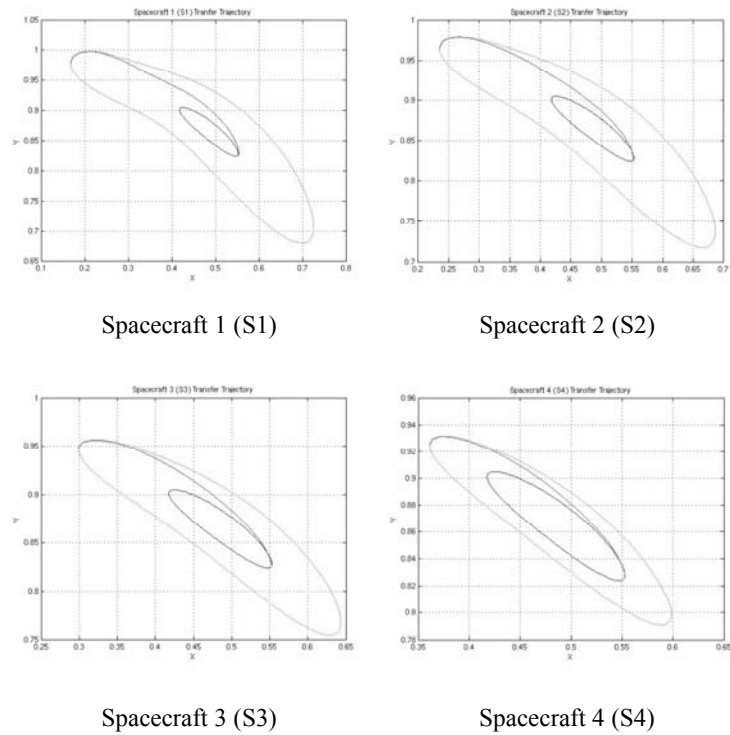


Figure 4.40. The transfer trajectory for each spacecraft is shown here. The starting point for each is where the periodic orbit is furthest away from the L4 equilibrium point.

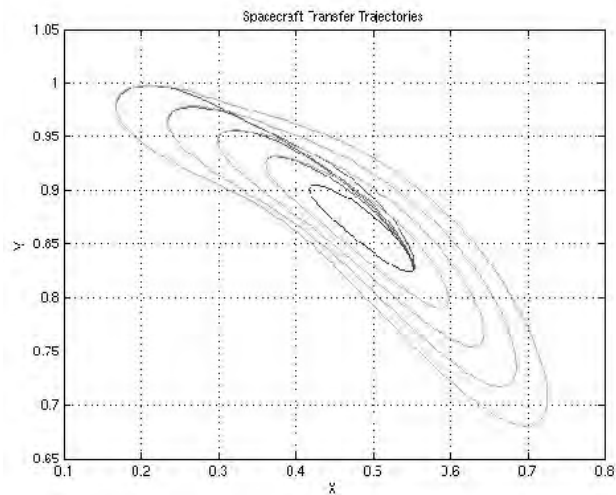


Figure 4.41. The transfer trajectory for each spacecraft is shown on this single plot to provide a sense of relative location in physical space.

Spacecraft No.	Staging Time (Units)	Transfer Time (Units)	Time to Reach Desired Orbit (Units)
1	11.2	10.5	21.7
2	2.2	10.4	12.6
3	14.5	10.0	24.5
4	18.0	9.8	27.8

Table 4.22. The staging time, transfer time, and time to reach the desired orbit are provided for each of the four spacecraft in this table.

The staging time, transfer time, and total time to reach the desired (inner most) periodic orbit are provided in Table 4.22. At $t=27.8$ units of time, all four spacecraft have reached the final destination. It should be noted that since Spacecraft 4 (S4) was the last reach the final orbit it will seed the desired formation, i.e. its actual and desired position are one in the same. The relative locations of each spacecraft with one another as well as the desired positions for formation establishment are shown in Figure 4.42. A simple schematic is shown in Figure 4.43.

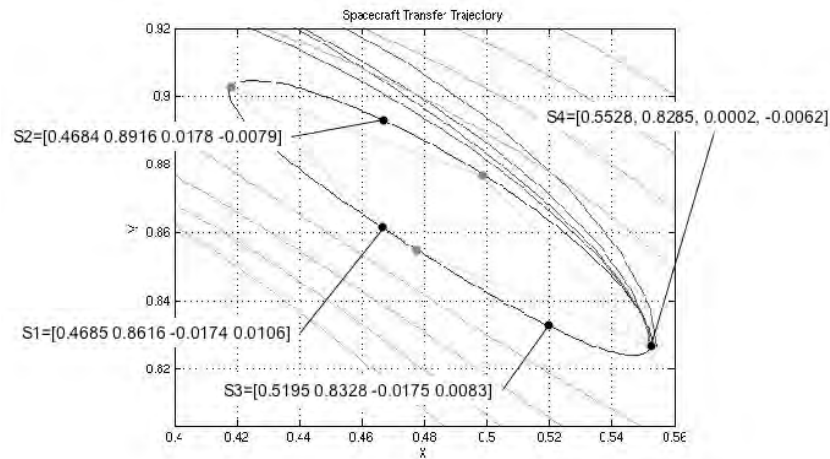


Figure 4.42. The relative positions of each spacecraft with respect to one another (shown in black) and the desired positions for formation establishment (shown in grey shade) are plotted here.

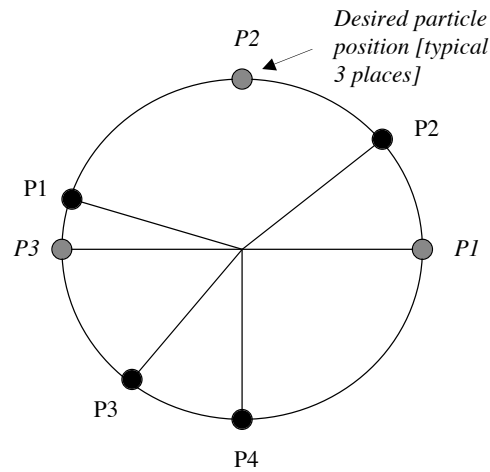


Figure 4.43. This is a simple schematic of Figure 6.10. It more clearly shows that Spacecraft 4 (P4) is already in the desired position for formation establishment. Therefore, the positions of the other three spacecraft will have to be altered with respect to it, e.g. Spacecraft 1 (P1) should follow 1/4-revolution behind P4, but in actuality is almost 3/4-revolution behind.

4.6.3 Formation Establishment

4.6.3.1 Resonant Frequency Approach

The resonant frequency (or orbit) approach based on the procedure and MATLAB computer program developed for the circular, restricted three-vortex problem described in section 3.5.1 is valid, but it is impractical for the planar case of the circular, restricted three-body problem in celestial mechanics. Although the period for each of the five planar orbits is different they are so similar, e.g. the difference in period between the inner most and adjacent orbit is less than 0.2%, that it would take a significant number of orbit rotations (>100) to achieve synchronization for just two spacecraft. To synchronize the entire formation would take greater than 300 revolutions.

4.6.3.2 Controller Method

As in the circular, restricted three-vortex problem in fluid mechanics, a second controller was developed to expedite the formation establishment process. Again, equations 4.22 and 4.23 are the two standard equations of motion for the planar case of the circular, restricted three-body problem in celestial mechanics. The controller was incorporated to each resulting in the following equations:

$$\ddot{x} - \lambda 2\dot{y} - x = -\frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x-1+\mu)}{r_2^3} \quad 4.112$$

$$\ddot{y} + \lambda 2\dot{x} - y = \frac{-(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3} \quad 4.113$$

In both equations, λ is a scale factor. At first, this seemed to produce the desired results, i.e. controlled motion trajectories or orbits that would close on themselves (see Figure 4.44). This would allow a spacecraft to depart from a select extrema on an intermediate orbit with a desired period and return where it started.

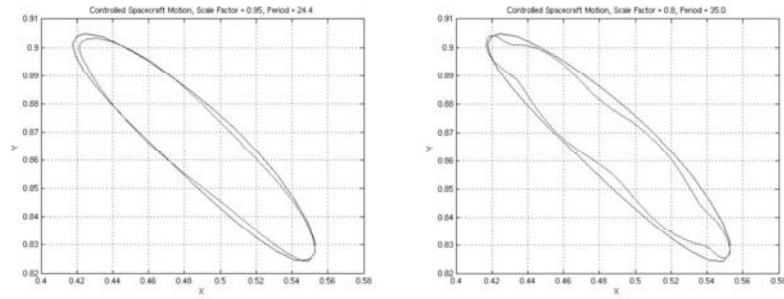


Figure 4.44. These are two examples where the second controller is turned ON. In each plot, the uncontrolled motion periodic orbits envelop the controlled motion trajectories. One can readily see that a spacecraft traveling on the latter would return to its initial starting point.

However, after evaluating several test cases, it was determined that the solution space was not continuous, i.e. there were some desired periods where the trajectories or orbits would not close on

themselves (see Figure 4.45). After much investigation, it was determined that the problem was due to the fact that the same controller value was being used in both equations.

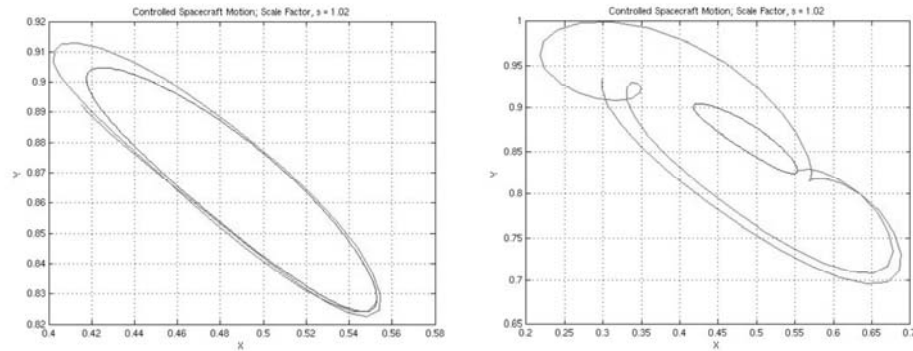


Figure 4.45. These are also two examples where the second controller is turned ON. However, one can clearly see, especially with the second plot, that the controlled motion trajectory does not close on itself.

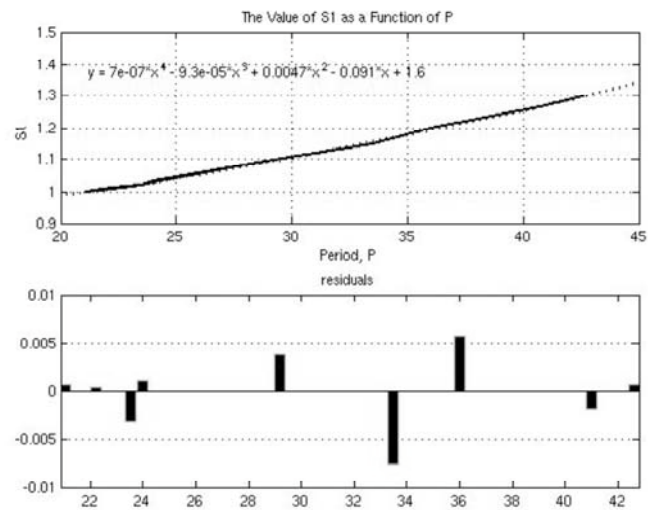


Figure 4.46. This is a curve-fit for the λ_1 (shown as S1 in the plot) in the controlled motion trajectory “period” range of interest.

If λ were to be separated into two different values, λ_1 and λ_2 , it would provide enough flexibility to find a trajectory or orbit that would close on itself with a desired period. Equations 4.112 and 4.113 were slightly altered resulting in the following:

$$\ddot{x} - \lambda_1 2\dot{y} - x = -\frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x-1+\mu)}{r_2^3} \quad 4.112$$

$$\ddot{y} + \lambda_2 2\dot{x} - y = \frac{-(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3} \quad 4.113$$

The values for λ_1 and λ_2 for the controlled motion trajectory “period”, P , range we are interested are provided in Figures 4.46 and 4.47. The values are also provided in tabular form in Table 4.23.

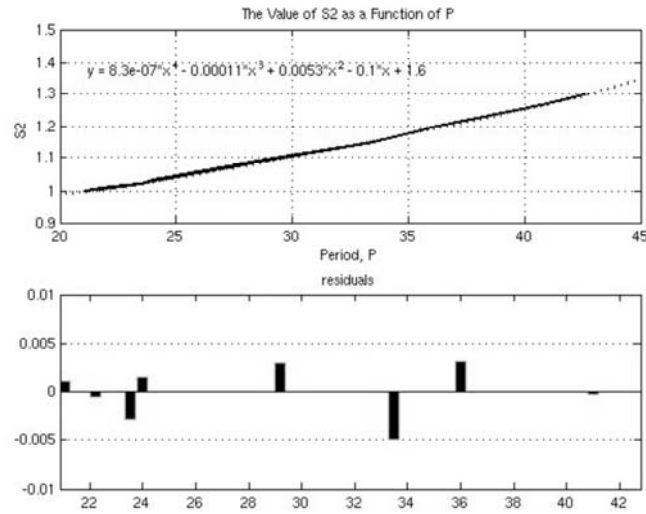


Figure 4.47. This is a curve-fit for the λ_2 (shown as S2 in the plot) in the controlled motion trajectory “period” range of interest.

Following the procedure outlined in the circular, restricted three-vortex problem in fluid mechanics, three spacecraft, S2, S3, and S1, will be placed on an intermediate trajectory or orbit with the necessary period and return to the inner most periodic orbit at precisely the right time and location to establish the desired formation.

“Period”, P’	Value of λ_1	Value of λ_2
21.1	1.0	1.0
22.2	1.01	1.0085
23.5	1.02	1.0195
24.0	1.03	1.0295
25.1	1.0431	1.04
29.2	1.10	1.0975
33.5	1.15	1.1498
36.0	1.2	1.1942
41.0	1.27	1.2696
42.6	1.3	1.2987

Table 4.23. The controller no. 2 parameter values required to create a number of controlled motion trajectories are shown in the table above.

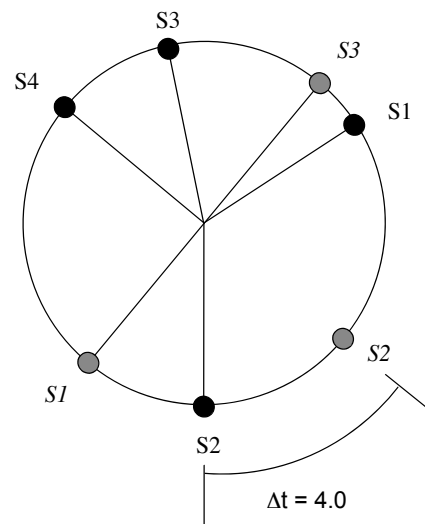


Figure 4.48. This is a simple schematic showing Spacecraft 2 (S2) at the point at which it will be placed on a controlled motion trajectory. It will leave the periodic orbit that it was on and only return at the proper time and location necessary to establish the desired formation.

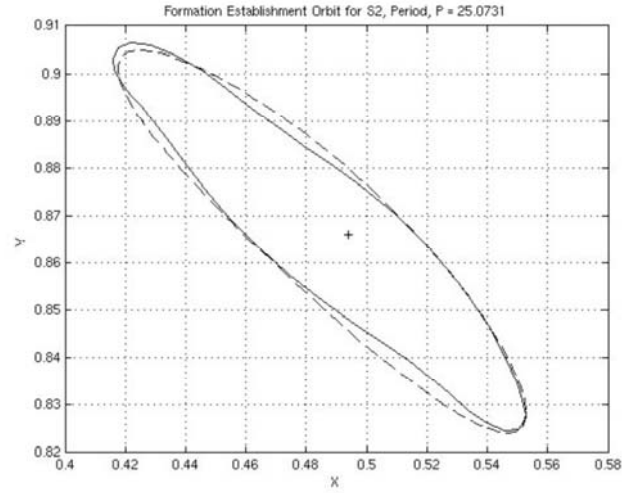


Figure 4.49. The controlled motion trajectory or orbit for Spacecraft 2 (S2) is shown in this plot. The final periodic orbit is shown as a dashed curve.

At $t = 27.8 + 6.5 = 34.3$, Spacecraft 2 (S2) arrives at the identified extrema, i.e. $[0.5528 \ 0.8285 \ 0.0002 \ -0.0062]$ and is placed on an orbit with period = 25.0730 ($21.0730 + 4.0$) by turning on controller no 2. Here, $\lambda_1 = 1.0431$ and $\lambda_2 = 1.04$. The schematic and orbit plot for S2 are shown in Figures 4.48 and 4.49, respectively.

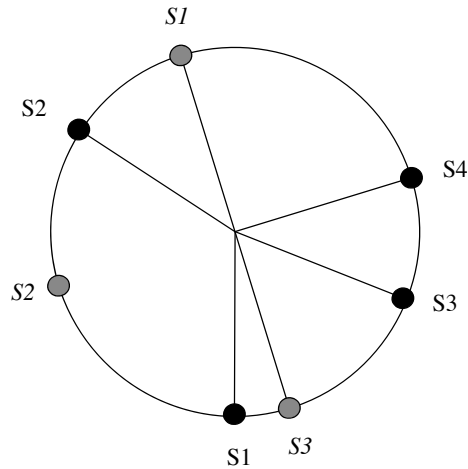


Figure 4.50. This is a simple schematic showing Spacecraft 1 (S1) at the point at which it will be placed on a controlled motion trajectory. It will leave the periodic orbit that it was on and only return at the proper time and location necessary to establish the desired formation.

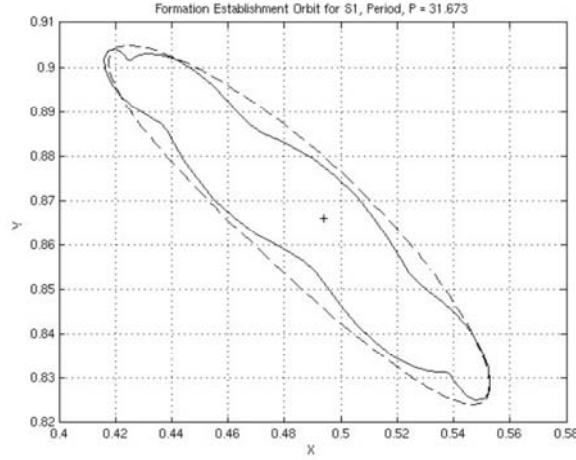


Figure 4.51. The controlled motion trajectory or orbit for Spacecraft 1 (S1) is shown in this plot. The final periodic orbit is shown as a dashed curve.

At $t = 27.8 + 15.5 = 43.3$, Spacecraft 1 (S1) arrives at the identified extrema, i.e. $[0.5528 \ 0.8285 \ 0.0002 \ -0.0062]$ and is placed on an orbit with period = 31.673 ($21.0730 + 10.6$) by turning on controller 2. Here, $\lambda_1 = 1.13$ and $\lambda_2 = 1.1278$. The schematic and orbit plot for S1 are shown in Figures 4.50 and 4.51, respectively.

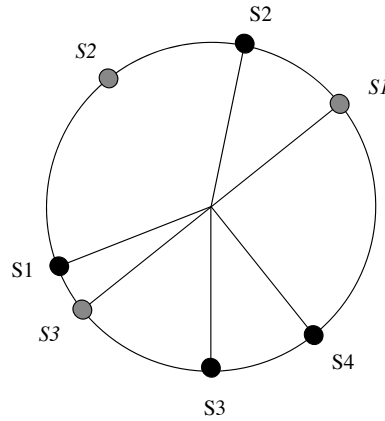


Figure 4.52. This is a simple schematic showing Spacecraft 3 (S3) at the point at which it will be placed on a controlled motion trajectory. It will leave the periodic orbit that it was on and only return at the proper time and location necessary to establish the desired formation.

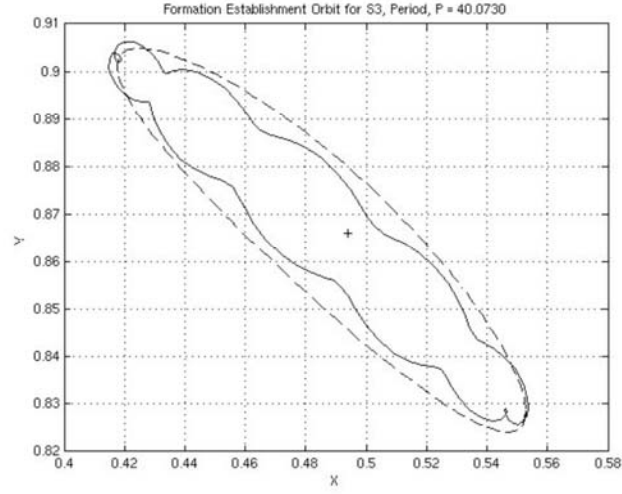


Figure 4.53. The controlled motion trajectory or orbit for Spacecraft 3 (S3) is shown in this plot. The final periodic orbit is shown as a dashed curve.

At $t = 27.8 + 18.0 = 45.8$, Spacecraft 3 (S3) arrives at the identified extrema, i.e. $[0.5528 \ 0.8285 \ 0.0002 \ -0.0062]$ and is placed on an orbit with period = 40.0730 ($21.0730 + 19.0$) by turning on controller 2. Here, $\lambda_1 = 1.255$ and $\lambda_2 = 1.2497$. The schematic and orbit plot for S3 are shown in Figures 4.52 and 4.53, respectively.

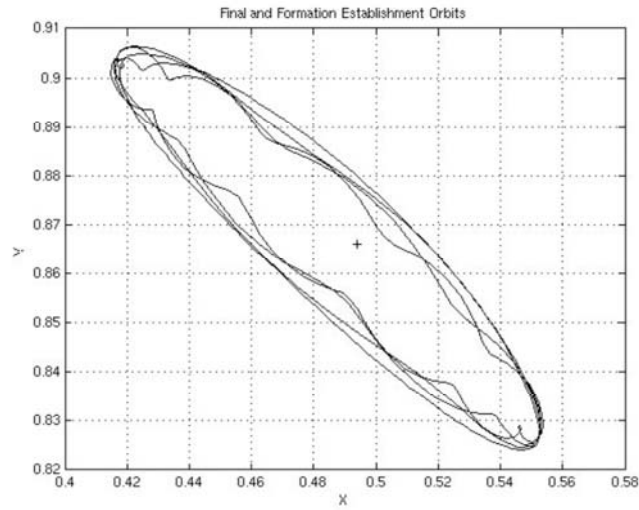


Figure 4.54. Transfer trajectories for Spacecraft 2 (S2), Spacecraft 1 (S1), and Spacecraft 3 (S3) are shown in this plot.

The three transfer trajectories are plotted on a single graph, so as to provide a sense of location in relative space (see Figure 4.54). When each of the three spacecraft have returned to the original periodic orbit they will be placed in the proper location for the formation desired. The simple schematic of this is shown in Figure 4.55.

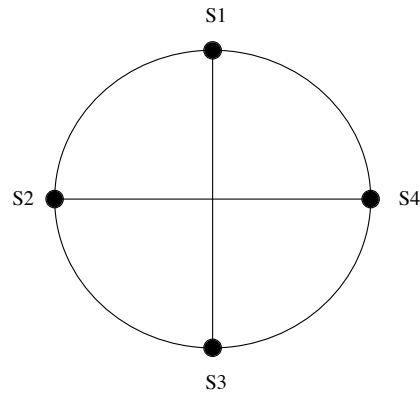


Figure 4.55. In this simple schematic the four spacecraft, S1-S4, are shown to be in the proper relative positions 1/4-revolution apart from one another.

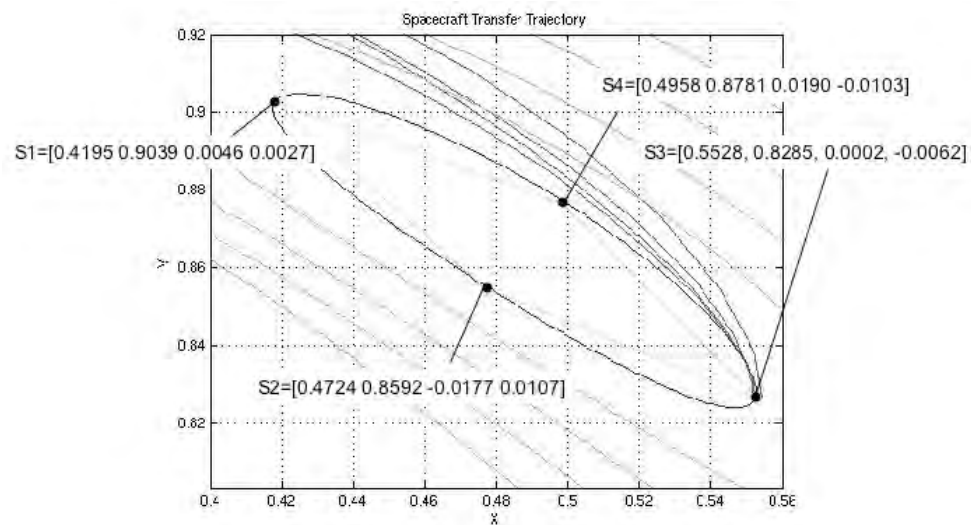


Figure 4.56. As shown in the plot figure above, once Spacecraft 3 (S3) returns to the original periodic orbit the desired formation, i.e. diamond or rhombus, has been established.

State vectors for each spacecraft at the point in time the formation is established are shown in figure 4.56. The total time for phase-locking and formation establishment is 85.8 units of time. From this point on, the spacecraft will be in an uncontrolled motion state and will continue to traverse the periodic orbit.

4.6.4 Example Problem and Solution

As in the case of the circular, restricted three-vortex problem in fluid mechanics, animations were created for phase locking and formation establishment. Figure 4.57 represents the phase-locking animation and Figure 4.58 represents the formation establishment animation.

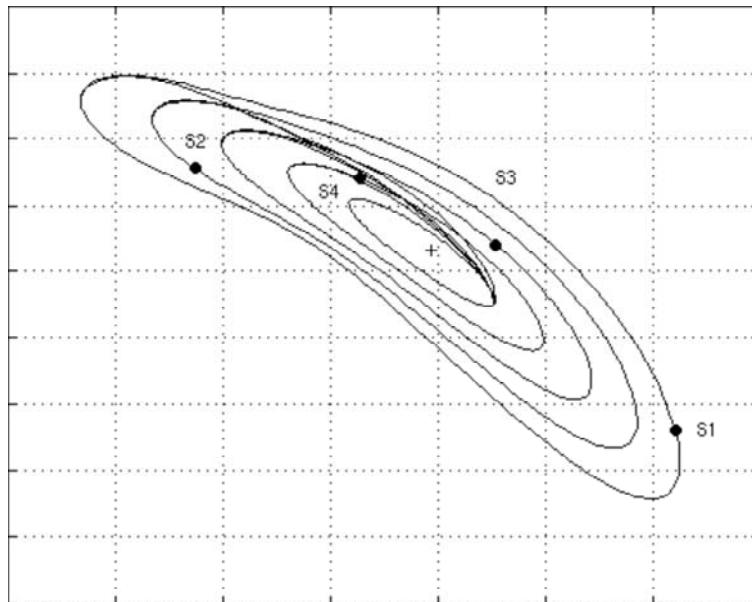


Figure 4.57. This is the first frame of a two hundred-frame animation for phase-locking in the circular, restricted three-body problem in celestial mechanics.

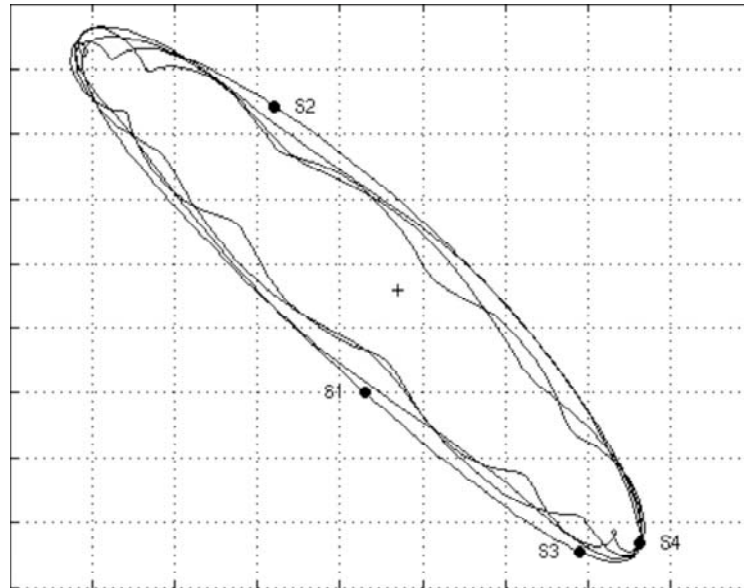


Figure 4.58. This is the first frame of a two hundred-frame animation for formation establishment in the planar case of the circular, restricted three-body problem in celestial mechanics.

4.7 Problem No. 2: Three-Dimensional Case

In the previous chapter, it was determined that phase-locking and formation establishment in the circular, restricted three-vortex problem in fluid mechanics is possible. A controller was used in conjunction with a resonant frequency (or orbit resonance) approach to produce the desired result. This controller was essentially an additional term to the standard equation of motion. It was also shown that the same controller could be used in combination with another, i.e. this time a scale factor was incorporated in the standard equation of motion, to expedite the entire process. It was shown earlier in this chapter that the same methods could be carried forward and used for the two-dimension or planar case of the circular, restricted three-body problem in celestial mechanics. The third and final step is to show that the methods also apply in the three-dimensional case.

4.7.1 Periodic Orbits

The three-dimensional orbits examined in Section 4.5, was used to develop the two controllers needed. Again, it should be noted that contrary to the circular, restricted three-vortex problem in the fluid mechanics, spacecraft motion is clockwise on the XY-plane when viewed from the positive Z-axis. Shown in Figure 4.40, are the five periodic orbits that were used in the study. The four spacecraft travel along the four outer periodic orbits, while the inner most periodic orbit is the desired orbit for phase-locking and formation establishment. In Figure 4.59, initial conditions, selected randomly, for each of the four spacecraft are shown in the plot. The state vectors for each are given in the form $[x,y,z,u,v,w]$, where x , y , and z are the positions along and u , v , and w are the velocities in direction of the X, Y, and Z-axis, respectively. As done in the circular, restricted three-vortex problem in fluid mechanics, animations were created for the circular, restricted three-body problem in celestial mechanics. The first of these for the three-dimensional case is represented in Figure 4.60.

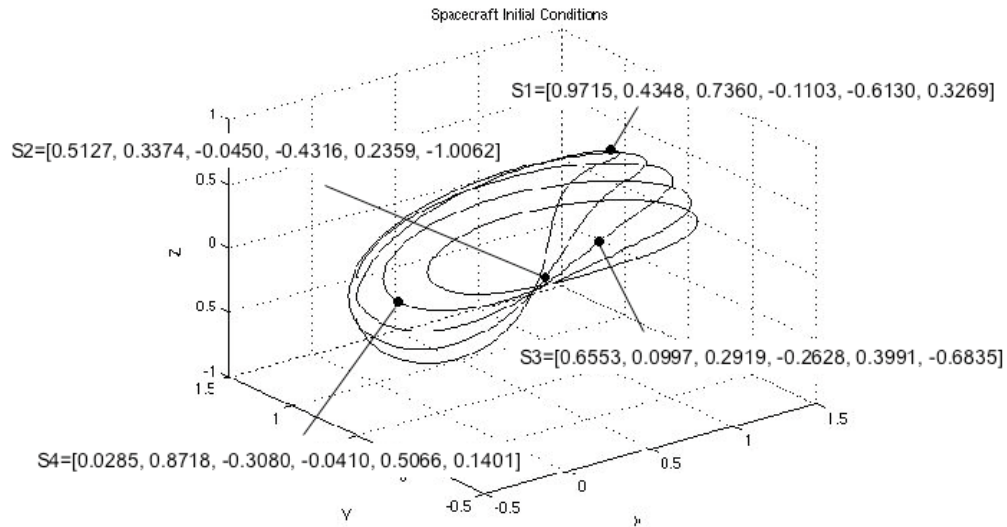


Figure 4.59. The initial condition state vector for each of the four spacecraft is shown in this plot. The terms given in each state vector are, in the order shown, position along the X, Y, and Z-axis and velocity in the X, Y, and Z-axis direction.

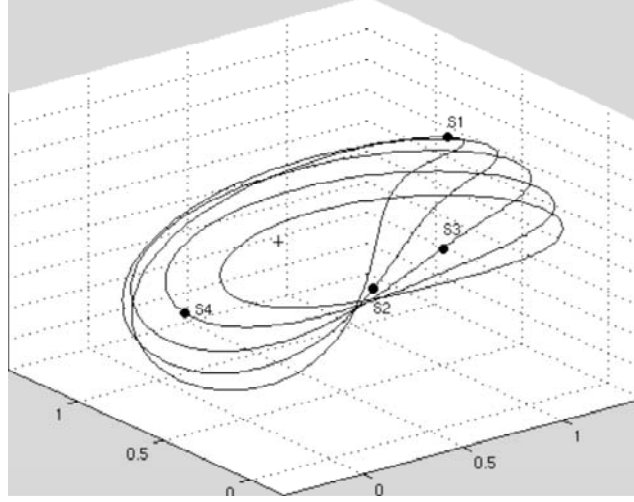


Figure 4.60. This is the first frame of the two hundred frames in the first animation. Since each spacecraft is only driven by the standard equations of motion, i.e. uncontrolled motion state, they will continue to traverse the periodic orbits they are on until a controller is enabled or turned ON.

4.7.2 Phase-Lock Controller

Equations 4.22 through 4.24 are the three standard equations of motion for the three-dimensional case of the circular, restricted three-body problem in celestial mechanics. The controller term was added to each resulting in the following equations:

$$\ddot{x} - 2\dot{y} - x = -\frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x-1+\mu)}{r_2^3} + \kappa_1 \sin\left(\frac{\alpha_1 \pi t}{T_1}\right) \quad 4.114$$

$$\ddot{y} + 2\dot{x} - y = -\frac{(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3} + \kappa_2 \sin\left(\frac{\alpha_2 \pi t}{T_2}\right) \quad 4.115$$

$$\ddot{z} = -\frac{(1-\mu)z}{r_1^3} - \frac{\mu z}{r_2^3} + \kappa_3 \sin\left(\frac{\alpha_3 \pi t}{T_3}\right) \quad 4.116$$

As in the circular, restricted three-vortex problem in fluid mechanics and the two dimensional or planar case in the circular, restricted three-body problem in celestial mechanics, κ is a term scale factor; α is equal to 1/2, 1, or 2; t is the time or the sine function angle multiplier; and T is the base orbit period. The term here as well will be referred to as controller no. 1. Table 4.24 shows the parameter values needed for each spacecraft to leave its original periodic orbit from the extrema, i.e. point on the periodic orbit furthest away from the L4 equilibrium point, and be placed on a transfer trajectory to the desired inner most periodic orbit.

Spacecraft and Equation	κ	α	T	Transfer Time (Units)
Spacecraft 1 (S1)				0.89
X (Equation 6.7)	7.7	1	6.283185	
Y (Equation 6.8)	7.7	1	6.283185	
Z (Equation 6.9)	-1.9	1	6.283185	
Spacecraft 2 (S2)				1.50
X (Equation 6.7)	1.3	1	6.283185	
Y (Equation 6.8)	1.3	1	6.283185	
Z (Equation 6.9)	-1.23	1	6.283185	
Spacecraft 3 (S3)				1.90
X (Equation 6.7)	0.39	1	6.283185	
Y (Equation 6.8)	0.39	1	6.283185	
Z (Equation 6.9)	-0.6	1	6.283185	
Spacecraft 4 (S4)				2.00
X (Equation 6.7)	0.18	1	6.283185	
Y (Equation 6.8)	0.18	1	6.283185	
Z (Equation 6.9)	-0.3	1	6.283185	

Table 4.24. The controller no. 1 parameter values shown in the table will produce the desired transfer trajectory for each spacecraft, i.e. a trajectory that will take it from its initial periodic orbit to the inner most periodic orbit.

Notice that the parameter values are not necessarily the same for all three equations, e.g. For Spacecraft 1 (S1), κ is equal to 7.7 in equations 4.114 and 4.115, but equal to -1.9 in equation 4.116. One of the lessons learned in the two-dimensional or planar case of the circular, restricted three-body problem solved for earlier in this chapter was that the parameter values should not necessarily be the same each equation of motion else flexibility is lost. Also shown in the table are the transfer times for

each spacecraft. It is interesting to note that even though Spacecraft (S1) is on the outer-most periodic orbit, the transfer time is the least among the four. It turns out that that S1 is placed on a more direct path the desired, inner most periodic orbit than the others.

Spacecraft No.	Staging Time (Units)	Transfer Time (Units)	Time to Reach Desired Orbit (Units)
1	3.52	0.89	4.41
2	1.11	1.50	2.61
3	1.50	1.90	3.40
4	6.03	2.00	8.03

Table 4.25. The staging time, transfer time, and total time required for each spacecraft to reach the desired (inner most) periodic orbit are provided in the table above.

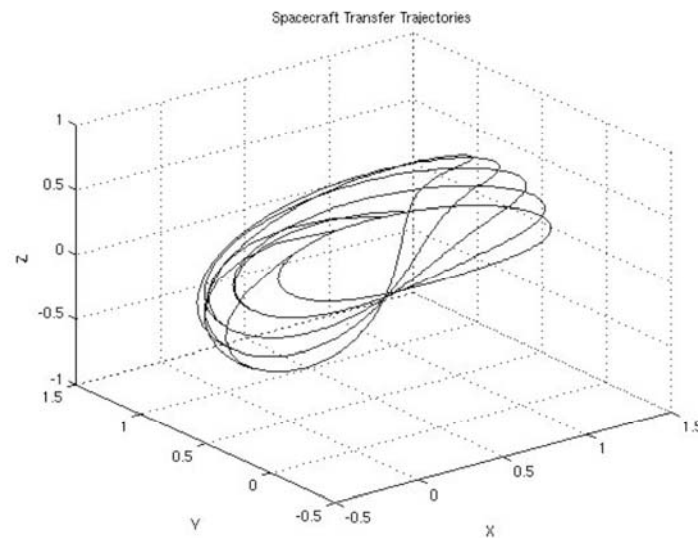


Figure 4.61. Transfer trajectories for each spacecraft are shown in the plot above along with the four original periodic orbits and the inner most periodic orbit where each spacecraft will be phase-locked and placed in the desired formation.

The staging time, transfer time, and total time to reach the desired (inner most) periodic orbit are provided in Table 4.25 (see also Figure 4.61). At $t=8.03$ units of time, all four spacecraft have

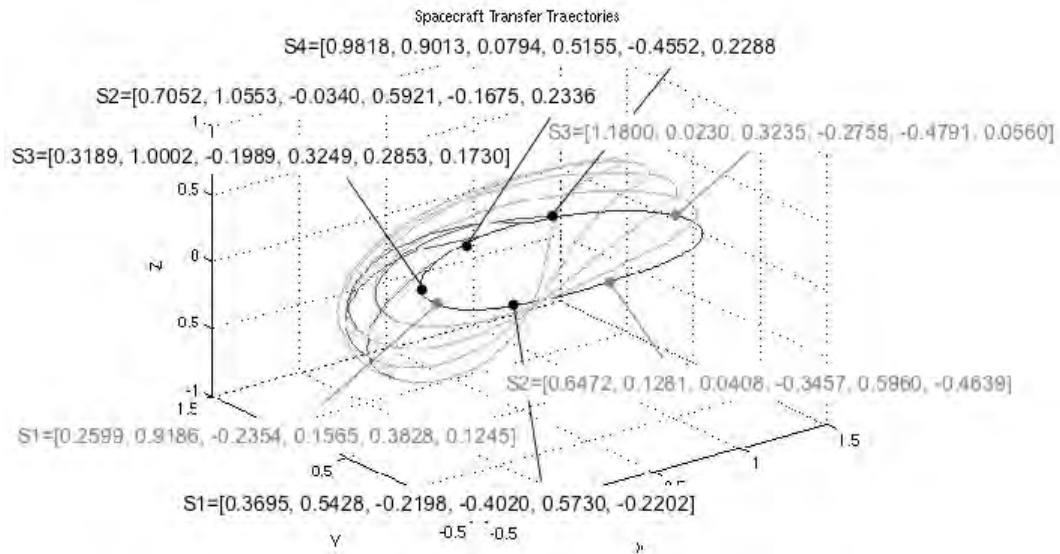


Figure 4.62. The relative positions of each spacecraft with respect to one another (shown in black) and the desired positions for formation establishment (shown in grey shade) are plotted here.

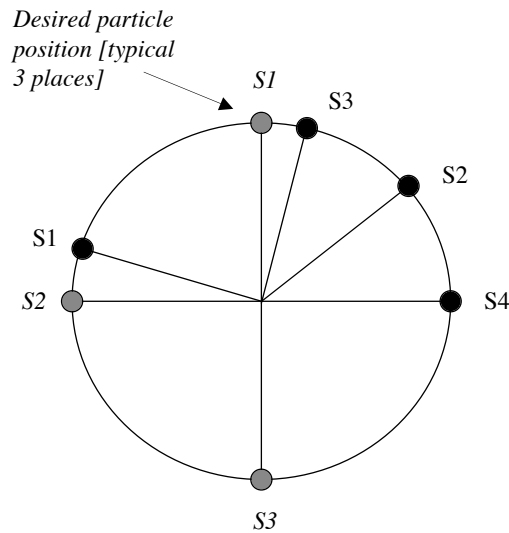


Figure 4.63. This is a simple schematic of Figure 4.62. It more clearly shows that Spacecraft 4 (P4) is already in the desired position for formation establishment. Therefore, the positions of the other three spacecraft will have to be altered with respect to it, i.e. Spacecraft 1 (P1) should follow 1/4-revolution behind P4, but in actuality is almost 1/2-revolution behind.

reached the final destination. It should be noted that since Spacecraft 4 (S4) was the last reach the final orbit it will seed the desired formation, i.e. its actual and desired position are one in the same. The relative locations of each spacecraft with one another as well as the desired positions for formation establishment are shown in Figure 4.62. A simple schematic is shown in Figure 4.63.

4.7.3 Formation Establishment

4.7.3.1 Resonant Frequency Approach

Again, the resonant frequency (or orbit) approach based on the procedure and MATLAB computer program developed for the circular, restricted three-vortex problem described in section 3.5.1 is valid it is also impractical for the three-dimensional case of the circular, restricted three-body problem in celestial mechanics. Although the period for each of the five planar orbits is different they are so similar, e.g. the difference in period between the inner most and adjacent orbit is less than 0.3%, that it would take a significant number of orbit rotations (>100) to achieve synchronization for just two spacecraft. To synchronize the entire formation would take greater than 300 revolutions.

4.7.3.2 Controller Method

As in the circular, restricted three-vortex problem in fluid mechanics and the two-dimensional or planar case of the circular, restricted three-body problem in celestial mechanics a second controller was developed to expedite the formation establishment process. Again, equations 4.22 through 4.24 are the standard equations of motion for the three-dimensional case of the circular, restricted three-body problem in celestial mechanics. The controller was incorporated to each resulting in the following equations:

$$\ddot{x} - \lambda_1 2\dot{y} - x = -\frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x-1+\mu)}{r_2^3} \quad 4.117$$

$$\ddot{y} + \lambda_2 2\dot{x} - y = \frac{-(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3} \quad 4.118$$

$$\ddot{z} = -\frac{\lambda_3(1-\mu)z}{r_1^3} - \frac{\mu z}{r_2^3} \quad 4.119$$

In each equation, λ , i.e. λ_1 , λ_2 , and λ_3 , is a scale factor. The values needed to produce the intermediate trajectories or orbits are provided in Table 4.26.

Spacecraft No.	λ_1	λ_2	λ_3	Orbit Period (Units of Time)
2	-0.0066	-0.0290	0.7000	2.46
3	-0.0790	-0.0180	0.3350	3.17
1	-0.2000	-0.0550	-0.0720	5.49

Table 4.26. The controller no. 2 parameter values shown in the table will produce the desired transfer trajectory for each spacecraft, i.e. a trajectory that will take it from its initial periodic orbit to the inner most periodic orbit.

Following the procedure outlined in the circular, restricted three-vortex problem in fluid mechanics and the two-dimensional case in the circular, restricted three-body problem in celestial mechanics, three spacecraft, S2, S3, and S1, will be placed on an intermediate trajectory or orbit with the necessary period and return to the inner most periodic orbit at precisely the right time and location to establish the desired formation.

At $t = 8.03 + 2.31 = 10.34$, Spacecraft 2 (S2) arrives at the identified extrema, i.e. [1.0568, -0.1081, 0.3263, -0.3906, -0.2344, -0.0456] and is placed on an orbit with period = 2.46. The schematic and orbit plot for S2 are shown in Figures 4.64 and 4.65 respectively.

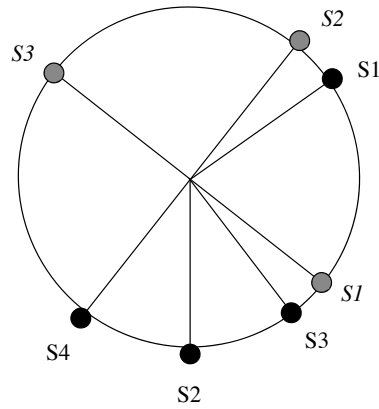


Figure 4.64. This is a simple schematic showing Spacecraft 2 (S2) at the point at which it will be placed on a controlled motion trajectory. It will leave the periodic orbit that it was on and only return at the proper time and location necessary to establish the desired formation.

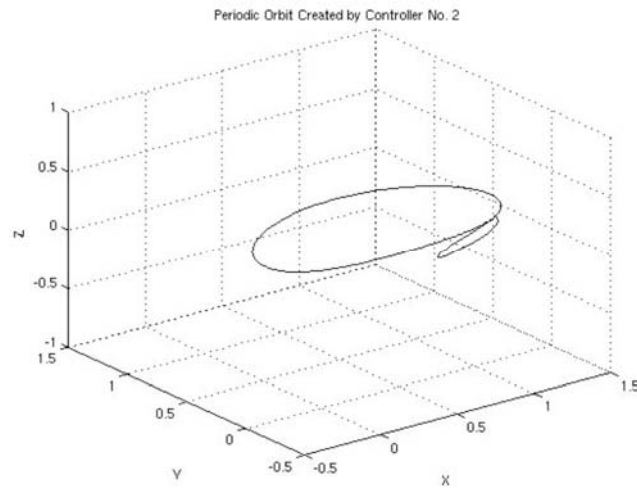


Figure 4.65. The controlled motion trajectory or orbit for Spacecraft 2 (S2) is shown in this plot. The final periodic orbit is shown as the larger orbit in the figure.

At $t = 8.03 + 3.05 = 11.08$, Spacecraft 3 (S3) arrives at the identified extrema, i.e. $[1.0568, -0.1081, 0.3263, -0.3906, -0.2344, -0.0456]$ and is placed on an orbit with period = 3.05. The schematic and orbit plot for S3 are shown in Figures 4.66 and 4.67 respectively.

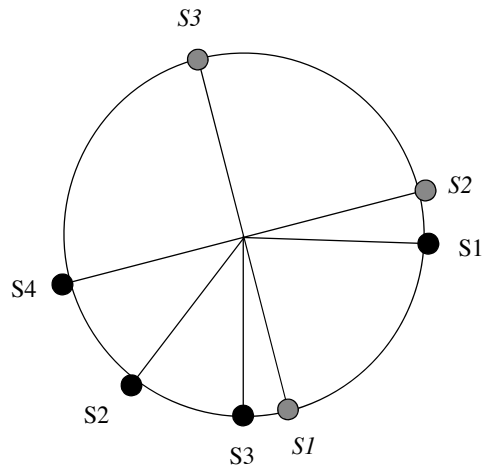


Figure 4.66. This is a simple schematic showing Spacecraft 3 (S3) at the point at which it will be placed on a controlled motion trajectory. It will leave the periodic orbit that it was on and only return at the proper time and location necessary to establish the desired formation.

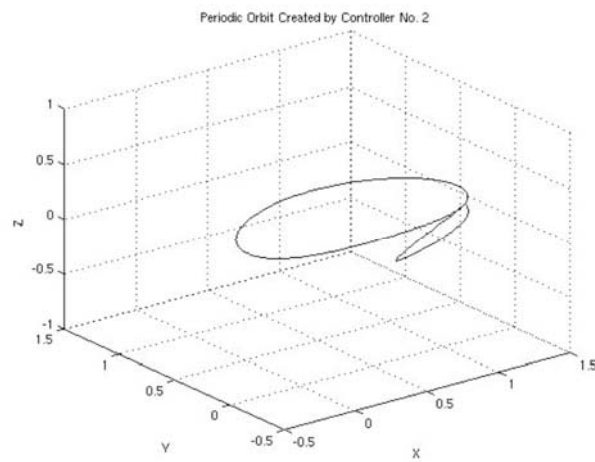


Figure 4.67. The controlled motion trajectory or orbit for Spacecraft 3 (S3) is shown in this plot. The final periodic orbit is shown as the larger orbit in the figure.

At $t = 8.03 + 4.10 = 12.13$, Spacecraft 1 (S1) arrives at the identified extrema, i.e. $[1.0568, -0.1081, 0.3263, -0.3906, -0.2344, -0.0456]$ and is placed on an orbit with period = 4.10. The schematic and orbit plot for S3 are shown in Figures 4.68 and 4.69 respectively.

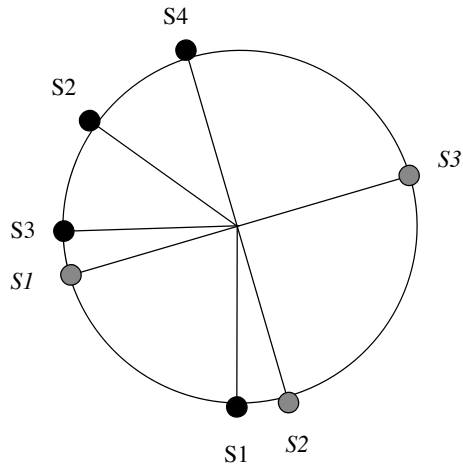


Figure 4.68. This is a simple schematic showing Spacecraft 1 (S1) at the point at which it will be placed on a controlled motion trajectory. It will leave the periodic orbit that it was on and only return at the proper time and location necessary to establish the desired formation.

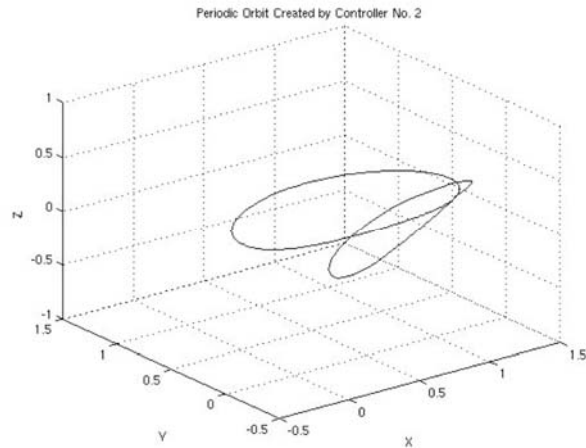


Figure 4.69. The controlled motion trajectory or orbit for Spacecraft 1 (S1) is shown in this plot. The final periodic orbit is shown as the larger orbit in the figure.

The three transfer trajectories are plotted on a single graph, so as to provide a sense of location in relative space (see Figure 4.70). When each of the three spacecraft have returned to the original periodic orbit they will be placed in the proper location for the formation desired. The simple schematic of this is shown in Figure 4.71.

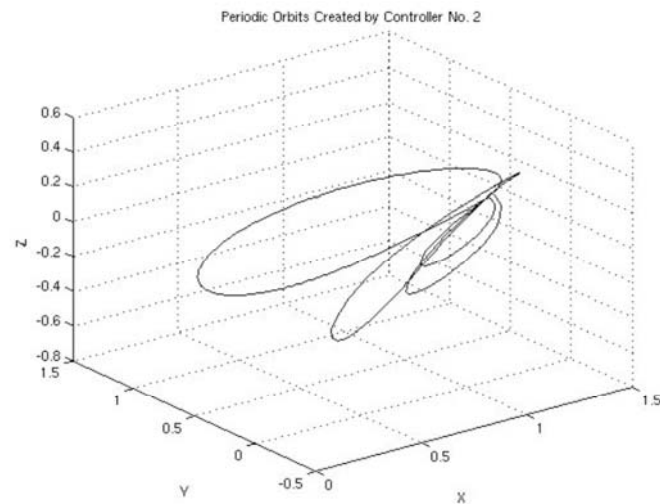


Figure 4.70. Transfer trajectories for Spacecraft 2 (S2), Spacecraft 3 (S3), and Spacecraft 1 (S1) are shown in this plot.

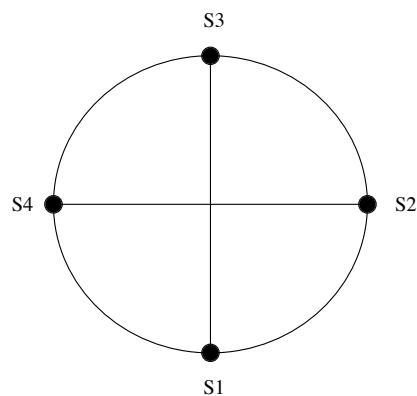


Figure 4.71. In this simple schematic the four spacecraft, S1-S4, are shown to be in the proper relative positions 1/4-revolution apart from one another.

State vectors for each spacecraft at the point in time the formation is established are shown in figure 4.72. The total time for phase-locking and formation establishment is 85.8 units of time. From this point on, the spacecraft will be in an uncontrolled motion state and will continue to traverse the periodic orbit.

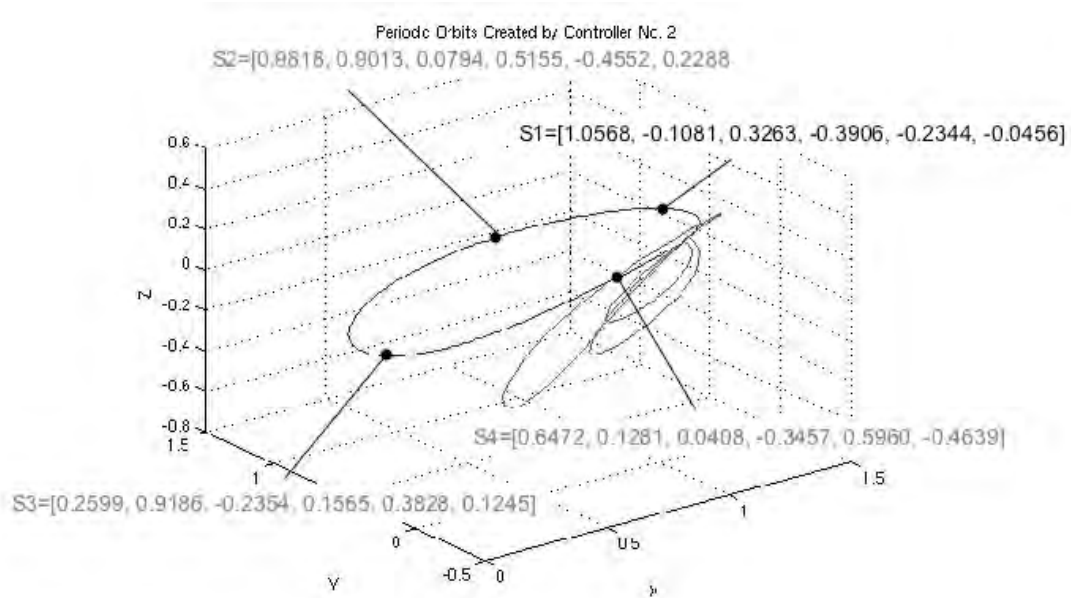


Figure 4.72. Once Spacecraft 1 (S1) returns to the original periodic orbit the desired formation, i.e. temporal separation between each spacecraft are identical.

4.7.4 Example Problem and Solution

As in the case of the circular, restricted three-vortex problem in fluid mechanics, animations were created for phase locking and formation establishment. Figure 4.73 represents the phase-locking animation and Figure 4.74 represents the formation establishment animation.

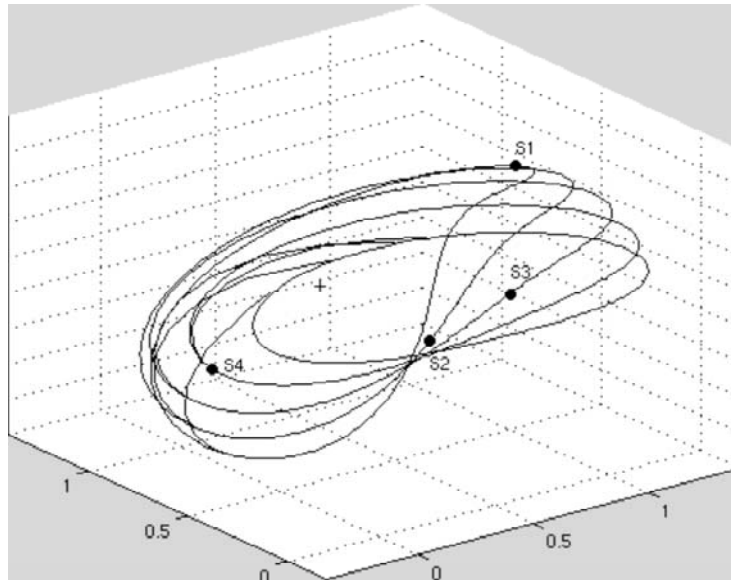


Figure 4.73. This is the first frame of the two hundred frames in the animation for phase-locking in the circular, restricted three-body problem in celestial mechanics.

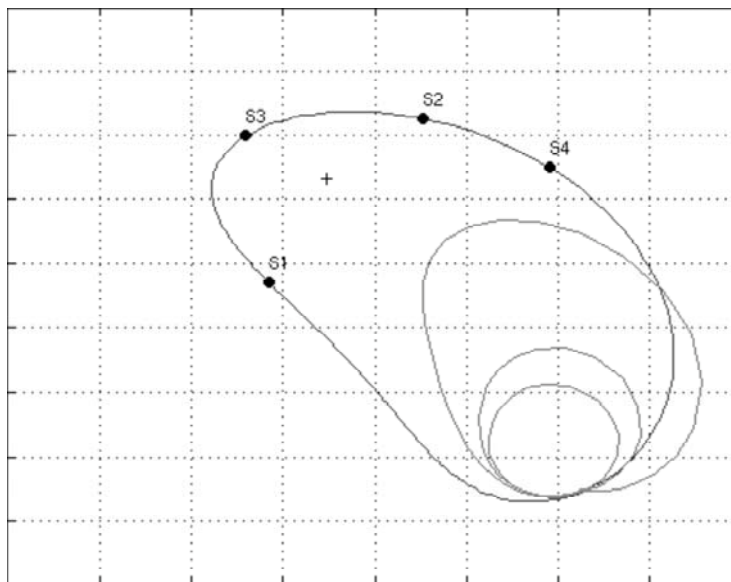


Figure 4.74. This is the first frame of a two hundred frames in the animation for formation establishment in the circular, restricted three-body problem in celestial mechanics.

Chapter 5: Evaluation and Assessment

5.1 Verification & Validation

Verification is an objective evaluation and involves asking the question, “Did we build it correctly?” In other words, does the design meet requirements? In the circular, restricted three-vortex problem in fluid mechanics the high-level requirements were to develop a controller or set of controllers to phase lock and establish a formation of test particles. The first controller, a trigonometric function added to the governing equation of motion, did provide a means for phase locking a number of test particles. It was also shown that when used in a resonant frequency/orbit approach, the controller could be used to create/establish a desired test particle formation as well. However, a second controller - the inclusion of a scale factor to the first term in the governing equation of motion - when used in conjunction with the first, helped to expedite the test particle phase locking and formation establishment process. Although both procedures could be directly carried over to the circular, restricted three-body problem in celestial mechanics as feasible/valid approaches, the two-controller method was the obvious choice. Phase-locking and formation establishment of multiple spacecraft was clearly demonstrated using this process.

Validation is more subjective and involves asking the question, “Did we build the right thing?” In other words, does the product satisfy the customers? If these customers include mission designers, navigation designers and analysts, attitude control engineers, propulsion system engineers, and the scientists who will one day benefit from this work it is hoped that the answer is yes. While

not the complete or optimal solution, it is hoped that this work will form the foundation and serve as an inspiration for a novel, but also practical approach to spacecraft formation flying.

Other verification & validation-related observations: (1) In the circular, restricted three-body problem in fluid mechanics the level curves of the Hamiltonian look identical to those shown in Newton [19], and propagating forward in time a state vector clearly generates orbits that close on themselves, i.e. periodic orbits; (2) the output of the MATLAB script used to locate equilibrium points in both the circular, restricted three-vortex problem and circular, restricted three-body problem compared well to those predicted by the Hill Sphere method; and (3) using an initial value problem approach in a MATLAB computer program produced valid trajectories and periodic orbits.

5.2 Limitations

Although valid for simulating trajectories and stable periodic orbits there is a limitation using an initial value problem approach in a MATLAB computer program. It cannot simulate the periodic orbits generated by AUTO 2000 using the two-point boundary value problem method if the orbits are unstable (see Figure 4.32). AUTO 2000 was created to densely foliate periodic orbits around equilibrium points, however, it would be a significant revision/augmentation effort to create individual periodic orbits of interest with desired start and end states. The circular, restricted three-body problem equations of motion do not include other terms needed for full-force modeling. In actuality, forces such as solar pressure need to be included for more accurate modeling of spacecraft motion. Most celestial body orbits possess some degree of eccentricity, and therefore, are not truly circular. The elliptical restricted three-body problem would produce more accurate results. Finally, the resonant frequency approach in the circular, restricted three-body problem is impractical in that it takes too long to establish a spacecraft formation. The periods of the planar and three-dimensional orbits under examination were so close to one another, e.g. 0.3%, it would take many, e.g. >100 orbits/revolutions to phase lock and establish a formation of even just two spacecraft.

5.3 Lessons Learned

Starting with a simple problem helps to develop the concepts needed to solve more complex problems. Phase locking and establishing a formation of test particles in the circular, restricted three-vortex problem in fluid mechanics provided the structure and basic approach needed to phase lock and establish a formation of spacecraft in the circular, restricted three-body problem in celestial mechanics. However, it must be understood that there are differences that need to be accounted for. Where there was just one equation of motion for the circular, restricted three-vortex problem there were two and three equations in the circular, restricted three-body problem depending on which case was being examined. It was shown that the controllers developed for a single equation couldn't be directly carried over to the two or three equation case without some modification, e.g. the actual controller parameters for each of the two or three equations should not be constrained to be identical. Otherwise, it may not be possible to create the desired transfer and/or intermediate trajectory/orbit.

Although a single controller and the resonant frequency approach were demonstrated to be viable and practical in the circular, restricted three-vortex problem, it was impractical for use in the circular, restricted three-body problem. It was not realized until late in the investigation that this was the case. Fortunately, a second controller had been developed that served to expedite the formation establishment process for the circular, restricted three-vortex problem. In the example problem, the total time required to phase lock and establish a desired formation using two controllers was an order of magnitude less than that using a single controller and the resonant frequency approach, i.e. 2.18 v. 21.74 units of time. Use of the two controllers was carried over to the circular, restricted three-body problem as really the only practical approach. In solving complex problems one might be tempted to declare success prematurely, but cautious/guarded optimism should prevail. Hopefully, given available time and resources it is best to clearly demonstrate a robust solution or develop alternatives to demonstrate resiliency.

5.4 Potential Scientific Applications

Creation of dynamically natural formations or multi-spacecraft platforms will enable the ‘loiter, synchronize/coordinate, and observe’ approach for future engineering and scientific missions where flexibility is a top-level requirement and key to mission success. Instruments on these spacecraft can be those needed for remote sensing observations, e.g. infrared measurements, or those needed for *in situ* field and particles measurements, e.g. magnetometer readings.

5.5 Recommendations for Future Work

The feasibility of dynamically natural spacecraft formations has been demonstrated in this body of work. To bring the concept closer to practical application there are a number of additional steps that can be taken. Firstly, the two controllers developed in this body of work can be optimized to minimize the time necessary to phase lock and establish a formation of multiple spacecraft. Secondly, a two point boundary value problem approach in MATLAB or AUTO 2000 can be used so that the velocity components as well as the position components at the point at which each controller is turned OFF, i.e. the phase locking and formation establishment end states, match those at the desired/final periodic orbit entry point. This would eliminate the need for an impulsive maneuver at the conclusion of the phase locking and/or formation establishment stage. Thirdly, a low energy propulsion system can be matched for each or for both controllers. Finally, the equations for both the uncontrolled and controlled motion cases can be augmented to account for and allow for the compensation of external forces, e.g. solar wind.

Chapter 6: Conclusions

The circular, restricted three-vortex problem in fluid mechanics was successfully used as a proof-of-concept model and for development of two controllers. The first controller, a trigonometric term added to the standard equation of motion, was used for phase locking of test particles originally traveling along individual periodic orbits in the complex coordinate frame. The controller was used in a resonant frequency approach that allowed for the creation of a desired geometric formation. A second controller, the incorporation of a scale factor in the first term in the standard equation of motion, was also developed. When used in conjunction with the first controller, it served to expedite the formation establishment process. This resulted in an order of magnitude decrease in the time required in the sample problem.

The two controllers developed in the circular, restricted three-vortex problem in fluid mechanics were carried over to the planar case of the circular, restricted three-body problem in celestial mechanics. It was shown that the first controller could be used for phase locking of spacecraft originally traveling along individual periodic orbits. The controller could also be used in a resonant frequency approach that allows for the creation of a desired geometric formation. However, use of the second controller in conjunction with the first did expedite the formation establishment process. It was clearly shown that the value of the scale factor could not be constrained to be the same for both equations of motion. In the sample problem the values needed to be different in order for the spacecraft trajectories/orbits to close.

The controllers used in the planar case of the circular, restricted three-body problem in celestial mechanics were also valid for the three-dimensional case. However, while examining the

original spacecraft periodic orbits it was noticed many of the three-dimensional periodic orbits created by the AUTO 2000 software tool were unstable. Spacecraft operating in unstable periodic orbits would have to occasionally perform trajectory correction maneuvers to stay on course. However, the mere creation of these orbits is testimony to the power of the tool and the two-point boundary value problem method used.

With the advent of solar electric propulsion and other low thrust actuator systems, it appears feasible for the controllers developed as part of this thesis to actually be used. Creation of dynamically natural formations or multi-spacecraft platforms will enable the ‘loiter, synchronize/coordinate, and observe’ approach for future engineering and scientific missions where flexibility is a top-level requirement and key to mission success.

Glossary

AUTO 2000	A continuation and bifurcation analysis software package
Barycenter	The center of mass of a multi-body system
Chaos	Complex dynamical behavior characterized by a general lack of periodicity and a great sensitivity to initial conditions
Constellation	A collection or group of spacecraft or satellites in close proximity where each one serves a common purpose or goal and some mechanism is employed for centralized or distributed coordination
Controller	A mechanism for affecting change in a dynamical system
Entropy	A measure of disorder, disorganization, or degradation in a system
Equilibrium	A state of balance, i.e. no net change
Formation	A collection or group of spacecraft or satellites in close proximity where each one serves a common purpose or goal and some mechanism is employed for centralized or distributed coordination. This term is sometimes used interchangeably with “constellation”.
Formation Flying	Spacecraft or man-made satellites operating together to meet the intent of a formation
General Formation Flying	Spacecraft formation flying with relatively large error tolerances
Hamiltonian (system)	A system where energy may change form, but the total energy is constant over time
Interferometer	Two or more telescopes working in unison where the effective diameter of the interferometer - an instrument that measures wavefront through interference of light waves - is equal to the distance between the two furthest telescopes
Keplerian	Pertaining to the motion described by Kepler’s laws
Libration Point	In the context of this work is synonymous with equilibrium point
MATLAB	A programming language and software application
Precision Formation Flying	Spacecraft formation flying with relatively small error tolerances
Staging Time	The time it takes for a test particle or spacecraft to move from it’s initial periodic orbit to the desired periodic orbit via a controller-enabled transfer trajectory
Vortex	A rotary circulation with an associated strength

Bibliography

- [1] Acebron, Juan A., et. al., "The Kuramoto Model: A Simple Paradigm for Synchronization Phenomena", *Reviews of Modern Physics*, vol. 77, Issue 1, pp. 137-185, 2005.
- [2] Analytical Graphics, Inc., Technical Notes on Rotating Libration Points, <http://www.agi.com/resources/help/stk613/helpSystem/extfile/gator/eq-rlp.htm>
- [3] Atkins, Ella and Yannick Penneçot, "Autonomous Satellite Formation Assembly and Reconfiguration with Gravity Fields", IEEE, Paper #296, 2001.
- [4] Basilio, Ralph R., Eastwood Im, Mark J. Rokey, and Deborah G. Vane, "A spaceborne microwave radar system for looking inside clouds", *Proceedings of the SPIE (International Society for Optical Engineering) Europe Remote Sensing Conference*, Volume No. 6361, September 2006.
- [5] Boain, Ronald J., "The CloudSat Mission: A Virtual Platform", *Proceedings of the 13th AAS/AIAA Space Flight Mechanics Meeting*, Puerto Rico, 09-13 February, 2003.
- [6] Barrow-Green, June, *Poincaré and the Three-Body Problem*, American Mathematical Society, 1997.
- [7] Clohessy, W. H. and R. S. Wilshire, "Terminal Guidance Systems for Satellite Rendezvous", *Journal of Aerospace Sciences*, 653-658, September 1960.
- [8] Danby, John M. A., *Fundamental of Celestial Mechanics*, Willmann-Bell, Inc., 1992.
- [9] The ESA (European Space Agency) and NASA (National Aeronautics and Space Administration) LISA (Laser Interferometer Space Antenna) web site, <http://lisa.jpl.nasa.gov/WHATIS/intro.html>.
- [10] Eisner, Thomas, *For Love of Insects*, Bleknap Press, Cambridge, MA, 2005.
- [11] Gurfil, Pini, and N. Jeremy Kasdin, "Stability and control of spacecraft formation flying in the trajectories of the restricted three-body problem", *Acta Astronautica*, Elsevier Science Ltd., 2003.
- [12] Hilborn, R. C., *Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers*, Oxford University Press, Second Edition, 2000.
- [13] Hill, George William, "Researches into the Lunar Theory", *American Journal of Mathematics*, 5-26, 1878.
- [14] Kaplan, Marshall, *Modern Spacecraft Dynamics and Control*, John Wiley and Sons, 1976, page 17.
- [15] Koon, W. S., M. W. Lo, J. E. Marsden, and S. D. Ross, "Dynamical Systems, the Three-Body Problem and Space Mission Design", International Conference on Differential Equations, Editors: B. Fiedler, K. Groger, and J. Sprekels, *World Scientific*, 2000, pp. 1167-1181.

- [16] Leonard, Naomi, et. al., Adaptive Sampling Using Feedback Control of an Autonomous Underwater Glider Fleet, *Proc. 13th International Symposium on Unmanned, Untethered Submersible Technology*, 2003.
- [17] Marchand, B. G. and K. C. Howell, "Formation Flight Near L1 and L2 in the Sun-Earth/Moon Ephemeris System Including Radiation Pressure", AAS 03-596.
- [18] Meyer, K. R., *Periodic Solutions of the N-Body Problem*, Springer, 1999.
- [19] Newton, P. K., *The N-Vortex Problem: Analytical Techniques*, Springer-Verlag, New York, 2001.
- [20] Paffenroth, R., "Continuation of Periodic Orbits Around Lagrange Points and AUTO 2000: The Three-Body Problem and Space Mission Design", Caltech presentation charts, 19 Feb 02.
- [21] Paffenroth, R. C., Doedel, E. J., and Dichmann, D. J., Continuation of Periodic Orbits Around Lagrange Points and AUTO2000, AAS paper 01-303, *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, 2001.
- [22] Paffenroth, Randy and Eusebius Doedel, "The AUTO2000 Command Line User Interface", *Proceedings of the Ninth...*, <http://python9.org/p9-cdrom/02/index.htm>.
- [23] Reynolds, Craig W., "Flocks, Herds, and Schools: A Distributed Behavioral Model", *Computer Graphics*, 21(4), July 1987, pp. 25-34.
- [24] Robertson, Andrew, Gokhan Inalhan, and Jonathan P. How, "Spacecraft Formation Flying Control Design for the Orion Mission", AIAA-99-4266, 1999.
- [25] Sanchez, David A., *Ordinary Differential Equations and Stability Theory: An Introduction*, Dover Publications, Inc, New York, 1979.
- [26] Strogatz, Steven, *Sync: How Order Emerges from Chaos in the Universe, Nature, and Daily Life*, Hyperion Books, New York, New York, 2003.
- [27] Sourceforge AUTO2000 website: <http://sourceforge.net/projects/auto2000>
- [28] Tollefson, Mark V., "Relative Orbit Design Tool", IEEE, 2001.
- [29] Valtonen, Mauri, and Hann Karttunen, *The Three-Body Problem*, Cambridge University Press, 2006.
- [30] Verhulst, Ferdinand, *Nonlinear Differential Equations and Dynamical Systems*, Second Edition, Springer-Verlag, Berlin, Heidelberg, New York, 2000.
- [31] Zhang, F. and P. S. Krishnaprasad, "Formation Dynamics Under a Class of Control Laws", *Proceedings of the 2002 American Control Conference*, pp. 1678-1685, 2002.

Appendices

Appendix A: MATLAB and AUTO 2000 Computer Tools

During the research stage programming languages, e.g. C/C++, and several software tools, e.g. Mathematica[®] and FreeFlyer[®], were investigated for applicability. A decision was made to utilize MATLAB[®] given the fundamental attributes of the programming language and software application. For example, there is the ability of the software application to numerically-solve Ordinary Differential Equations (ODEs) and Initial-Value Problems (IVPs). The software application also includes a compiler. Therefore, programs developed in MATLAB can be converted to a stand-alone application without having to first convert the code to another programming language like C/C++. MATLAB can also be used to produce plots and create animations. Although MATLAB has the ability to solve Boundary Value Problems (BVPs) as well, the AUTO 2000 software package was chosen to generate closed/periodic orbits, since much of the capability already existed. MATLAB and AUTO 2000 as well as some of the salient fundamental concepts associated with each are described in more detail below.

Mathworks Matrix Laboratory (a.k.a. MATLAB)

MATLAB is a concatenation of essentially the first syllables in “matrix” and “laboratory”. It is the name given to an interactive programming language and the commercial-off-the-shelf software package offered by MathWorks, Inc. The programming language is intuitive and mathematical relationships are expressed in familiar notation. However, this belies how powerful this computation and visualization tool can be. The basic data element is the array or matrix, allowing for even the most complex mathematical relationships to be solved in a relatively short amount of time. MATLAB works with Windows, Mac, UNIX (e.g. Sun), and UNIX-type (e.g. Linux) operating systems. Although MATLAB can be used for “batch jobs” or “background processing”, the system is primarily used in interactive mode through single commands or “.m” functions, scripts, or programs

containing multiple commands. To acquaint the reader with the format and syntax of MATLAB a simple example of a script is shown below, and the resulting plot is shown in Figure A.1.

```
% This script produces a two-dimensional plot representing
% a microwave radar system operating in pulsed (specifically,
% chirp) rather than continuous wave mode

x = 0:0.05:5;

y = sin(x.^2);

plot(x,y); xlabel('x'), ylabel('y'), title('Chirp Mode')
```

One important note is that MATLAB is an interpretative rather than compiled language. Unlike C/C++, header and source code files do not have to be “compiled” in order to create object files or executable routines. The elimination of this step increases the efficiency of the operator/programmer in debugging or modifying programs.

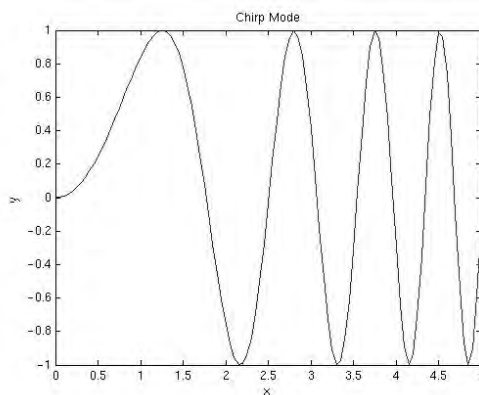


Figure A.1. This is a MATLAB plot of a microwave radar system operating in pulsed (specifically, chirp) rather than continuous wave mode.

As stated earlier, MATLAB can quickly solved ODEs and IVPs. Specifically, MATLAB uses the Runge-Kutta method to solve a system of first-order differential equations. Given a set of initial conditions; the derivative function at the start, mid, and end-points of an integral; and the

unknown function at a previous point this technique produces fairly accurate solutions. However, rather than utilize the existing 2nd-3rd or 4th-5th order Runge-Kutta library functions, a 7th-8th order function file obtained from Martin W. Lo was utilized.

Most of the time the MATLAB programs used to support this Ph.D. thesis were executed in Version 7.0.1.24704 (Release 14), Service Pack 1, released on 13 September 2004 running on a SunOS, Release 5.8 with CDE (Common Desktop Environment) 1.4.8, X11 Version 6.4.1. The workstation platform is a Sun Microsystems Ultra 60. In instances where physical or remote access to this computer was not possible, MATLAB programs were executed in MATLAB 7 (Release 14) running on the Mac Operating System X, v10.3.9.

AUTO 2000 Continuation and Bifurcation Analysis Tool

AUTO 2000 is a software package that is used for dynamical system analysis in a number of areas (e.g. fluid dynamics, cardiac electrophysiology, and the n-body problem). Simple algebraic problems and ODEs can be solved using AUTO 2000. Specifically, continuation and bifurcation analysis techniques are used to compute parameter-dependent families of solutions. Written in 1980, it has since been ported to C and an interface, based on the Python programming language, has been added. The software runs on Linux and Portable Operating System (POSIX)-compatible operating systems such as UNIX and BSD, the version of UNIX developed at the University of California, Berkeley. The AUTO 2000 software package solves equations of the form:

$$F(x) = 0, F : R^{n+1} \rightarrow R^n, n \in W \quad \text{A.1}$$

Basically, the dynamical system must be conserved, i.e. a Hamiltonian System, and possess one or more fixed-points. Here the system has one more unknowns than equations and is, therefore, considered underdetermined. Since n is an element of the manifold, solution sets lie on an “ n ”-dimensional manifold in “ $(n+1)$ ”-dimensional space. For example, if we are interested in a three-

dimensional system, the solution sets would lie on a two-dimensional manifold. Paffenroth [A2] states that through the use of a number of continuation techniques, a user can vary one component of the solution to develop a new solution. A basic introduction to continuation methods and a brief description of the pseudo-arclength method used are described in the following paragraphs. Continuation, a.k.a. homotopy, methods are numerical techniques for computing solution manifolds/branches. Qualitatively, this method provides a connection between an easy problem and a hard problem that is actually of interest. The solution to the simple problem is gradually transformed to the solution of the difficult problem by tracing a path. Computing a piece of the solution manifold near one solution usually through a predictor-corrector procedure, then selecting another solution from this set, and repeating the process accomplishes this. There generally exists a solution branch, i.e., a one-dimensional family of points, which passes through a solution, x_0 . To compute another nearby point, x_1 , on this branch, an additional parameter called the continuation parameter, λ , can be introduced. Therefore,

$$H(x, \lambda) = F(x) - (1 - \lambda)F(x_0) \quad \text{A.2}$$

where x_0 is a given point in \mathcal{R}^n . The problem $H(x, \lambda) = 0$ is then solved for values of λ between 0 and 1. When $\lambda = 0$, $x = x_0$, and when $\lambda = 1$, $H(x, 1) = F(x)$. The latter indicates that the solution of $H(x, 1) = 0$ coincides with the solution for $F(x) = 0$. In the natural parameter continuation approach one would merely introduce a small change, d , make an initial guess, and use an iteration scheme, e.g. Newton-Raphson Method, to search for a unique solution. However, one of the drawbacks of this approach is if the guess is not “sufficiently close”, the solution may not converge. Tangent continuation is similar to the natural parameter continuation method, except that a higher-order initial guess is used. This higher-order initial guess usually allows for quicker convergence. However, there is a limitation to this approach. At a “fold”, where the solution curve bends backwards, the continuation parameter cannot be used for parameterization. Pseudo-arclength continuation can be used. If one uses the arclength of the curve as the continuation parameter, the situation described

immediately above is avoided. It is summarized by Paffenroth et al. [A3]. A better guess for x_1 is defined as $x_1^\# = x_0 + \dot{x}_0 \Delta s$, where \dot{x}_0 is the unit tangent to the solution curve at x_0 and Δs is the step size. The step size approximates the arclength along the solution curve. The new solution, x_1 , is constrained to lie on a hyperplane perpendicular to the unit tangent vector, \dot{x}_0 . This allows for the value of the continuation parameter, λ , to vary. It will be shown later that AUTO 2000, one of the two computer software tools used in this thesis, utilizes this approach, and therefore, provides a robust method for solving complex problems. A simplified sketch of the three continuation methods described is shown in Figure A.2.

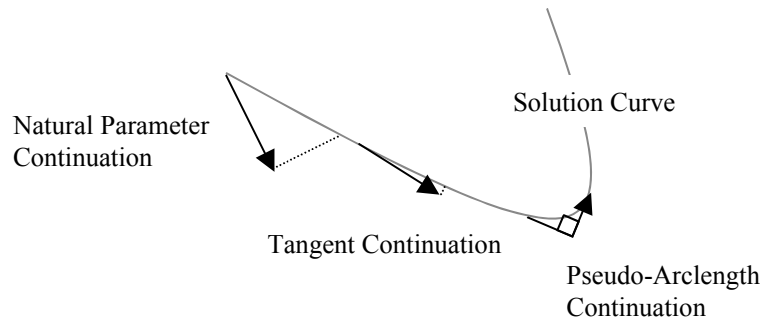


Figure A.2. This is a simple schematic of three different continuation or parameterization methods.

The characteristics associated with a fixed point depend on the various parameters that describe it. A bifurcation is a qualitative change as one of the control parameters is smoothly varied. Take for example a dripping faucet with the water pressure being the control parameter. At a relatively low pressure each drop follows the previous at a fixed period of time, T . The pressure is then increased to a point where the drops come in pairs and each pair follows the previous every $2T$. This transition is called a period doubling effect. If the pressure is increased further the drops will eventually fall in a random manner signaling a transition to chaos. A bifurcation diagram is sometimes used to illustrate these transitions. It should be noted that a certain control parameter value could also cause a fixed point to suddenly shift from an attractor to a repellor, or vice-versa.

Paffenroth et al. [A3] provides an example that is more germane to the thesis. However, it first must be mentioned that there is a theorem that is applicable to Hamiltonian systems with a nondegenerate first integral, e.g. circular, restricted three-body problem in celestial mechanics. Meyer [A1] describes the Cylinder Theorem as follows: An elementary periodic orbit of a system with an integral I lies in a smooth cylinder of periodic solutions parameterized by I . Therefore, this implies that a solution branch without a parameter exists. Paffenroth et al. [A3] asks that the following simple conservative system be considered

$$\begin{aligned}x_1' &= x_2 \\x_2' &= -x_1(1 - x_1)\end{aligned}\tag{A.3}$$

where the first integral is $F = \frac{1}{2}x_2^2 + \frac{1}{2}x_1^2 - \frac{1}{3}x_1^3$. The set of equations in A.3 possesses a nested branch of periodic orbits that are analogous to the level curves of the Hamiltonian. However, equation A.3 needs to be rephrased to include a continuation parameter and associated term as follows

$$\begin{aligned}x_1' &= x_2 \\x_2' &= -x_1(1 - x_1) + \lambda x_2\end{aligned}\tag{A.4}$$

Equation A.4 is equal to equation A.3 if $\lambda = 0$. Paffenroth et al. [A3] require that there be no periodic orbits for $\lambda \neq 0$ else the solution is not valid. They add a term, specifically a damping term, which destroys all periodic orbits to satisfy this constraint. The bifurcation diagram is shown in Figure A.3. The vertical axis is a measure of the solution and can be any number of attributes. Here it is just the

L2 norm, i.e., $|x| = \sqrt{\sum_{k=1}^n |x_k|^2}$. Each point on the bifurcation diagram represents a periodic solution. It

is obvious that one cannot use the natural or tangent continuation approach with λ as the parameter given that λ is constrained to equal zero. Only the pseudo-arclength continuation method can be employed, since the arclength is allowed to vary while λ can be fixed at a value of zero.

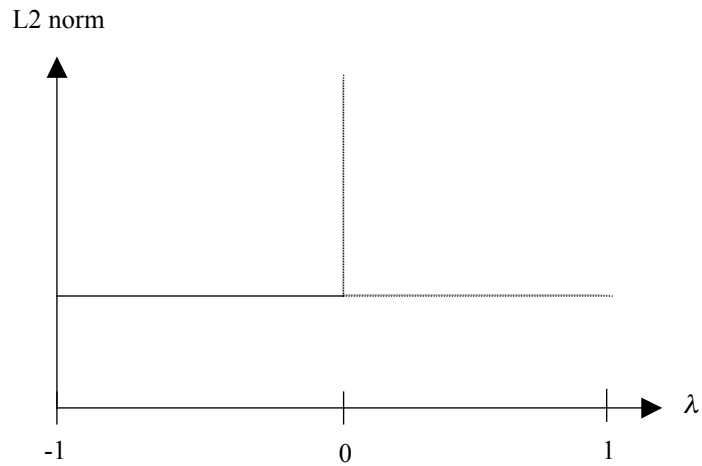


Figure A.3. This is the bifurcation diagram for the system shown in equation 2.27. [Credit: Paffenroth, R. C., Doedel, E. J., and Dichmann, D. J., Continuation of Periodic Orbits Around Lagrange Points and AUTO2000, AAS paper 01-303, *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, 2001]

The current version of AUTO 2000 is 0.9.7, and was created on 27 June 2002. The software and release notes are publicly available. Paffenroth and Doedel [A4] provide some written information on the software, and the software itself can be downloaded from the Sourceforge web site [A5]. The AUTO 2000 scripts used to support this thesis were executed in a Linux Operating System environment.

References:

- [A1] Meyer, K. R., *Periodic Solutions of the N-Body Problem*, Springer, 1999.
- [A2] Paffenroth, R., “Continuation of Periodic Orbits Around Lagrange Points and AUTO 2000: The Three-Body Problem and Space Mission Design”, Caltech presentation charts, 19 Feb 02.
- [A3] Paffenroth, R. C., Doedel, E. J., and Dichmann, D. J., Continuation of Periodic Orbits Around Lagrange Points and AUTO2000, AAS paper 01-303, *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, 2001.
- [A4] Paffenroth, Randy and Eusebius Doedel, “The AUTO2000 Command Line User Interface”, *Proceedings of the Ninth...*, <http://python9.org/p9-cdrom/02/index.htm>.
- [A5] Sourceforge AUTO2000 website: <http://sourceforge.net/projects/auto2000>

Appendix B: MATLAB Scripts, Function Files, and Programs

Contents

File Name	Description	Page No.
eigenvalues.m	Finds the eigenvalues of the circular, restricted three-body problem standard equations of motion to determine the stability of the selected equilibrium point	150
find_libration_points.m	Script for finding the libration point coordinates of the circular, restricted three-body problem	151
main_script_v5_1.m	Script for the orbit resonance approach for phase-locking and formation establishment in the circular, restricted three-vortex problem	153
ode78.m	7 th -8 th Order Runge-Kutta ODE solver function file	175
three_body_script_v5_1.m	General script for the circular, restricted three-body problem	177
three_body_v4_1.m	Function file called by three_body_script_v5_1.m	181
three_vortex.m	Function file called by main_script_v5_1.m	182
two_body_init_v5.m	Script for the general two-body problem	183
two_body_func.m	Function file called by two_body_init_v5.m	186

eigenvalues.m

```

% =====
% = program: eigenvalues.m                                =
% =                                                       =
% = MathWorks MATLAB script for the determining the stability =
% = of the equilibrium/libration points in the circular,    =
% = restricted three-body problem by finding the eigenvalues =
% = of a characteristic (fourth-order differential) equation =
% =                                                       =
% = Written by Ralph R. Basilio, Ph.D. Candidate            =
% = Version: 3.1                                           =
% = Date: 14 November 2006                                =
% =                                                       =
% = AME 794 Dissertation - Advisor: Professor Paul K. Newton =
% = Aerospace and Mechanical Engineering Department         =
% = Viterbi School of Engineering                         =
% = University of Southern California                     =
% =                                                       =
% =====

% =====
% Definitions
% =====
% m1 = mass of the greater of the two primary bodies
% m2 = mass of the lesser of the two primary bodies
% mu = m2/(m1+m2)
% Example 1: For the earth-moon system, mu = 0.012150
% Example 2: For the Saturn-titan system, mu = 0.000238
%
% x1 = X-axis position of the greater primary
% x2 = X-axis position of the lesser primary

% =====
% Inputs
% =====
% Define the value of mu
mu = 0.000238;

% Select which equilibrium point to analyze,
% e.g. EP = 1 is the co-linear equilibrium point located
% between the two primary bodies along the line/axis jointing
% the two.
EP = 4;

% =====
% Other Inputs
% =====
% Obtain the position of the equilibrium point of interest
% from the output of the find_libration_points.m script

x = 0.4998;
y = 0.8660;

% =====
% Calculation Section
% =====

x1 = -mu;
x2 = 1-mu;

```

```

r1 = 1.0;
r2 = 1.0;

if (EP <= 3)
    r1 = x+mu;
    r2 = x+mu-1;
elseif (EP >=4)
    r1 = 1.0;
    r2 = 1.0;
end

A = 1;
B = -(1-mu)/r1^3;
C = -mu/r2^3;
D = 3*(1-mu)*(x-x1).^2/r1^5;
E = 3*mu*(x-x2).^2/r2^5;
Uxx = A + B + C + D + E;

F = 1;
G = -(1-mu)/r1^3;
H = -mu/r2^3;
I = 3*(1-mu)*y^2/r1^5;
J = 3*mu*y^2/r2^5;
Uyy = F + G + H + I + J;

K = 3*(1-mu)*(x-x1)*y/r1^5;
L = 3*mu*(x-x2)*y/r2^5;
Uxy = K + L;

polynomial = [1 0 (4-Uxx-Uyy) 0 Uxx*Uyy-Uxy^2];
eigen = roots (polynomial)

% =====
% Notes on stability
% =====
% If any of the eigenvalues have imaginary parts, then the
% solution orbits around the equilibrium point and can be
% considered stable.
%
% If any of the eigenvalues have a real part that is less than
% or equal to zero the solution is stable.
%
% If any of the eigenvalues have a real part that is greater
% than zero the solution is unstable.
% =====

```

find_libration_points.m

```

% =====
% =
% = program: find_libration_points.m
% =
% = MathWorks MATLAB script for finding the locations,
% = i.e. coordinates, of the equilibrium/libration points in
% = the circular, restricted three-body problem
% =
% = Written by Ralph R. Basilio, Ph.D. Candidate
% = Version: 2.0
% = Date: 23 November 2006
% =

```

```

% = AME 794 Dissertation - Advisor: Professor Paul K. Newton      =
% = Aerospace and Mechanical Engineering Department              =
% = Viterbi School of Engineering                               =
% = University of Southern California                           =
% =                                                              =
% =====

% =====
% Definitions
% =====
% m1 = mass of the greater of the two primary bodies
% m2 = mass of the lesser of the two primary bodies
% mu = m2/(m1+m2)
% Example 1: For the earth-moon system,    mu = 0.012150
% Example 2: For the Saturn-titan system, mu = 0.000238

% =====
% Input: Define the value of mu
% =====

    mu = 0.012150;

% =====
% Calculation Section
% =====

L1_polynomial = [1 -1*(3-mu) 3-2*mu    -mu    2*mu    -mu];
L2_polynomial = [1 -1*(3-mu) 3-2*mu    -mu    -2*mu    -mu];
L3_polynomial = [1    (2+mu) 1+2*mu -(1-mu) -2*(1-mu) -1*(1-mu)];

L1_roots = roots(L1_polynomial);
L2_roots = roots(L2_polynomial);
L3_roots = roots(L3_polynomial);

for i = 1:5
    if isreal(L1_roots(i))
        L1_dist_from_m2 = L1_roots(i);
    end
    if isreal(L2_roots(i))
        L2_dist_from_m2 = L2_roots(i);
    end
    if isreal(L3_roots(i))
        L3_dist_from_m2 = L3_roots(i);
    end
end

L1_x_position = (1-mu) - L1_dist_from_m2;
L1_y_position = 0.0;
L2_x_position = (1-mu) + L2_dist_from_m2;
L2_y_position = 0.0;
L3_x_position = -mu - L3_dist_from_m2;
L3_y_position = 0.0;

L4_x_position = 0.5-mu;
L4_y_position = 0.5*sqrt(3);
L5_x_position = 0.5-mu;
L5_y_position = -0.5*sqrt(3);

L1_x_position
L1_y_position
L2_x_position

```

L2_y_position
 L3_x_position
 L3_y_position
 L4_x_position
 L4_y_position
 L5_x_position
 L5_y_position

main_scriptv5_1.m

```
% =====
% =
% = script: main_script_v5_1
% =
% = MathWorks Matlab Script for the Restricted Three-Vortex Problem =
% =
% = This script takes a set of randomly distributed particles and =
% = phase-locks them at a desired energy-level (i.e. Hamiltonian) =
% = and orients them with respect to one another to create a =
% = dynamically-natural formation (i.e. geometric shape). =
% =
% = Written by: Ralph R. Basilio, Ph.D. Candidate =
% = Version: 5.1 =
% = Date: 17 October 2005 =
% =
% = AME 794 Dissertation - Advisor: Professor Paul K. Newton =
% = Aerospace and Mechanical Engineering Department =
% = Viterbi School of Engineering =
% = University of Southern California =
% =
% = This script uses a seventh & eighth-order Runge-Kutta-Fehlberg =
% = integration method to produce an accurate solution. =
% =
% =====

% =====
% Background
% =====

% The restricted three-vortex problem is synonymous with the
% restricted three-body problem in celestial mechanics. In this case
% there are two vortices that are called the primaries. The vortex
% strengths are  $\text{cap\_gamma}_1 \geq \text{cap\_gamma}_2 > 0$ . A third vortex of
% negligible strength,  $\text{cap\_gamma}_3 = 0$ , is then introduced into the
% realm. The motion of the third vortex is affected by the
% primaries, but does not influence the motion of the primaries.
% The governing equation of motion given assumes a rotating
% coordinate frame, such that the primaries are at rest. We will
% call the third vortex (or any other vortex of negligible strength)
% a "particle".

% A controller term is added to the governing equation of motion.
% When the particle moves under the influence of the controller it
% will be said to be in a "controlled state", and when is it
% influenced only by the standard form of the equation it will be
% said to be in an "uncontrolled state".

% =====
% User Interface Section (Interactive Mode)
% =====
```

```

% -----
% Banner
% -----

disp(' ');
disp('=====');
disp('= This is a Mathworks Matlab(tm) program for phase-locking      =');
disp('= and establishing a relative formation of four (4) test        =');
disp('= particles in the Circular, Restricted Three-Vortex Problem,    =');
disp('= specifically in the vicinity of the second primary vortex.     =');
disp(' ');
disp('= This program was written by Ralph R. Basilio, under the       =');
disp('= direction of Professor Paul K. Newton, Fall Semester 2005,     =');
disp('= University of Southern California.                               =');
disp('=====');
disp(' ');

% -----
% Particle Initial Conditions
% -----

disp('Let''s start with some general questions. ');
disp(' ');
disp('1. Use the default set of particle initial conditions? ');
user_defined = input('Please type in "1" for "yes" or "0" for "no": ');

disp(' ');
if (user_defined == 0),
    disp('That''s fine. We''ll use the Matlab random number generator. ');
    disp(' ');
    junk = input('(Please hit the return key again) ');
    disp(' ');
end

% -----
% Energy-Levels (i.e. Hamiltonian Values)
% -----

disp('2. Use the default value for the desired energy-level? ');
default_H = input('Please type in a "1" for "yes" or "0" for "no": ');
disp(' ');

if (default_H == 0),
    disp('Energy-levels (i.e. Hamiltonian values) range from a low of
about ');
    disp('-2.4 near the second primary vortex to a high of -0.7 further ');
    disp('away. ');
    disp(' ');
    H_Desired_user = input('What do you want the new energy-level to be?
');
    disp(' ');
end

% -----
% Formation (i.e. Geometric Shape)
% -----

disp('I assume that you wish the particle formation to resemble a
rhombus ');
disp('(or diamond shape). 3. Is this correct? ');

```



```

rhombus = input('Please type in "1" for "yes" or "0" for "no": ');
disp(' ');
if (rhombus == 0),
    disp('Sorry, you''re out of luck...at least for now.')
    disp(' ');
    junk = input('(Please hit the return key again)');
    disp(' ');
end

% -----
% Formation Error Tolerance
% -----

disp('4. Use the default value for the formation entry error tolerance?');
entry_tol_ques = input ('Please type in "1" for "yes" or "0" for "no" : ');
disp(' ');

if (entry_tol_ques == 0),
    entry_tol_user = input('Please type in a number from 0.01 to 0.50 : ');
    disp(' ');
end

% -----
% Plots
% -----

disp('5. Do you want to produce plots of the pertinent information/data?');
plots = input('Please type in "1" for "yes" or "0" for "no" : ');

% -----
% Animations
% -----

disp(' ');
disp('6. Do you want to create some of the more interesting animations?');
animation_check = input('Please type in "1" for "yes" or "0" for no : ');
disp(' ');
if (animation_check == 1),
    disp('Are you absolutely sure you want to create animations? This. ');
    disp('will take several minutes. ');
    animations = input('Please type in "1" for "yes" or "0" for "no" : ');
end
if (animation_check == 0),
    animations = animation_check;
end

disp(' ');

% =====
% Set global parameters
% =====

global cap_gamma_1 cap_gamma_2 D xi1 xi2 kappa

```

```

% =====
% User Input Section (Batch Jobs)
% =====

% Define initial and final times

t0 = 0;
tf = 002.5000;

% Define tolerance level

tol = 1.0e-9;

% Define lambda [Note to Ralph: Define variable]

lambda = 0.5;

% Define strength of the first primary, vortex no. 1

cap_gamma_1 = pi;

% Define the distance between the two primaries

D = 1.0;

% Default particle initial conditions

xi_0(1) = 0.6898 - 0.2494i; % Particle No. 1
xi_0(2) = 0.7181 - 0.1812i; % Particle No. 2
xi_0(3) = 0.5844 - 0.2194i; % Particle No. 3
xi_0(4) = 0.5179 + 0.1906i; % Particle No. 4

% xi_0 = 0.20 + 0.00i; % Test particle (scratch pad)

% Default desired energy-level (i.e. Hamiltonian)

H_Desired = -1.0911;

% =====
% Matlab Calculation Section
% =====

% -----
% Set the new energy-level to the user-defined value
% -----

if (default_H == 0),
    H_Desired = H_Desired_user;
end

% -----
% Variables needed to create a line for "H_Desired"
% -----

X = [0 6];
Y = [H_Desired H_Desired];

% -----
% Calculate the strength of the second primary, vortex no. 2
% -----

```

```

cap_gamma_2 = 2*pi*lambda;

% -----
% Locate the two primary vortices
% -----

xi1 = -lambda;
xi2 = 1 - lambda;

% -----
% Use random number generator for particle initial conditions
% -----

% if (user_defined == 0),
%     imag_string = 'i';
%     imag_str = sscanf(imag_string,'%c');
%     xi_0(1) = rand*0.2 + 0.1 ...
%             + (rand*0.4*str2num(imag_str) - 0.2i); % Particle No. 1
%     xi_0(2) = rand*0.2 + 0.1 ...
%             + (rand*0.4*str2num(imag_str) - 0.2i); % Particle No. 2
%     xi_0(3) = rand*0.2 + 0.1 ...
%             + (rand*0.4*str2num(imag_str) - 0.2i); % Particle No. 3
%     xi_0(4) = rand*0.2 + 0.1 ...
%             + (rand*0.4*str2num(imag_str) - 0.2i); % Particle No. 4
% end

if (user_defined == 0),
    imag_string = 'i';
    imag_str = sscanf(imag_string,'%c');
    A1 = rand;
    B1 = rand;
    A2 = rand;
    B2 = rand;
    A3 = rand;
    B3 = rand;
    A4 = rand;
    B4 = rand;
    xi_0(1) = (A1*0.15+0.2)+(B1*0.4*str2num(imag_str)-0.2i);
    xi_0(2) = (A2*0.15+0.2)+(B2*0.4*str2num(imag_str)-0.2i);
    xi_0(3) = (A3*0.15+0.2)+(B3*0.4*str2num(imag_str)-0.2i);
    xi_0(4) = (A4*0.15+0.2)+(B4*0.4*str2num(imag_str)-0.2i);
end

% -----
% Calculate energy-levels (i.e. Hamiltonians) for each particle
% -----

for i=1:4
    H1 = (-1/2)*(real(xi_0(i)).^2+imag(xi_0(i)).^2);
    H2 = (1-lambda)*log(sqrt((real(xi_0(i))+lambda).^2+imag(xi_0(i)).^2));
    H3 = lambda*log(sqrt((real(xi_0(i))+lambda-1).^2+imag(xi_0(i)).^2));
    H(i) = H1+H2+H3;
end

% -----
% Produce the initial periodic orbits for each particle. Since the Matlab
% conditional statements do not handle the imaginary part of complex
% numbers, we have to create a new array called "Z2" to facilitate period
% determination. Also, to avoid ambiguity in determining the period,
% because of negative values in Z2 (or Z1 for that matter), 1.0 was added

```

```

% all values.
% -----

[t1,xi_1] = ode78(@three_vortex, t0, tf, xi_0(1), tol);
[t2,xi_2] = ode78(@three_vortex, t0, tf, xi_0(2), tol);
[t3,xi_3] = ode78(@three_vortex, t0, tf, xi_0(3), tol);
[t4,xi_4] = ode78(@three_vortex, t0, tf, xi_0(4), tol);

% -----
% Determine the orbit periods for each particle. Additionally,
% find "time to stage" (i.e. the time it takes for the particle to
% travel from it's initial position to the "top dead center" of its
% initial periodic orbit). Also, identify the orbit "min"s and
% "max"s (e.g. maximum real component of the complex number). Finally,
% calculate the orbit energy-levels for each orbit data point.
% -----

% -----
% P1 Orbit Period
% -----

t1_new = t0 : 0.01 : tf;
xi_1_new = interp1(t1,xi_1,t1_new,'spline');
Z1_1 = real(xi_1_new)+1.0;
Z2_1 = imag(xi_1_new)+1.0;
for i = 4:250
    if ((Z1_1(i) >= Z1_1(1)*0.99) && (Z1_1(i) <= Z1_1(1)*1.01) && ...
        (Z2_1(i) >= Z2_1(1)*0.99) && (Z2_1(i) <= Z2_1(1)*1.01)),
        P1_period = t1_new(i);
        disp('The initial orbit period for Particle No. 1 is: '),
        disp(t1_new(i)),
        disp(i),
        break
    end
end

% -----
% P1 Staging Time
% -----

for j = 1 : i
    Z3_1(j) = Z2_1(j);
end
for j = 1 : i
    if (Z3_1(j) == max(Z3_1)),
        P1_stage_t = t1_new(j);
        disp('The P1 staging time is:'),
        disp(P1_stage_t),
        break
    end
end

% -----
% P1 Orbit "Min"s and "Max"s
% -----

for j = 1 : i,
    xi_1_new_real(j) = real(xi_1_new(j));
    xi_1_new_imag(j) = imag(xi_1_new(j));
end

```

```

for j = 1 : i,
    xi_1_new_real_max = max(xi_1_new_real);
    xi_1_new_real_min = min(xi_1_new_real);
    xi_1_new_imag_max = max(xi_1_new_imag);
    xi_1_new_imag_min = min(xi_1_new_imag);
end

% -----
% Calculate P1 Orbit Energy-Levels (i.e. Hamiltonians)
% -----

for j = 1 : numel(t1_new)
    H1 = (-1/2)*(real(xi_1_new(j)).^2+imag(xi_1_new(j)).^2);
    H2 = (1-lambda)*log(sqrt((real(xi_1_new(j))+lambda).^2 ...
        +imag(xi_1_new(j)).^2));
    H3 = lambda*log(sqrt((real(xi_1_new(j))+lambda-1).^2 ...
        +imag(xi_1_new(j)).^2));
    xi_1_new_H(j) = H1+H2+H3;
end

% -----
% P2 Orbit Period
% -----

t2_new = t0 : 0.01 : tf;
xi_2_new = interp1(t2,xi_2,t2_new,'spline');
Z1_2 = real(xi_2_new)+1.0;
Z2_2 = imag(xi_2_new)+1.0;
for i = 4:250
    if ((Z1_2(i) >= Z1_2(1)*0.99) && (Z1_2(i) <= Z1_2(1)*1.01) && ...
        (Z2_2(i) >= Z2_2(1)*0.99) && (Z2_2(i) <= Z2_2(1)*1.01)),
        P2_period = t2_new(i);
        disp('The initial orbit period for Particle No. 2 is: '),
        disp(t2_new(i)),
        disp(i),
        break
    end
end

% -----
% P2 Staging Time
% -----

for j = 1 : i
    Z3_2(j) = Z2_2(j);
end
for j = 1 : i
    if (Z3_2(j) == max(Z3_2)),
        P2_stage_t = t2_new(j);
        disp('The P2 staging time is:'),
        disp(P2_stage_t),
        break
    end
end

% -----
% P2 Orbit "Min"s and "Max"s
% -----

for j = 1 : i,
    xi_2_new_real(j) = real(xi_2_new(j));

```

```

        xi_2_new_imag(j) = imag(xi_2_new(j));
    end

    for j = 1 : i,
        xi_2_new_real_max = max(xi_2_new_real);
        xi_2_new_real_min = min(xi_2_new_real);
        xi_2_new_imag_max = max(xi_2_new_imag);
        xi_2_new_imag_min = min(xi_2_new_imag);
    end

% -----
% Calculate P2 Orbit Energy-Levels (i.e. Hamiltonians)
% -----

    for j = 1 : numel(t2_new)
        H1 = (-1/2)*(real(xi_2_new(j)).^2+imag(xi_2_new(j)).^2);
        H2 = (1-lambda)*log(sqrt((real(xi_2_new(j))+lambda).^2 ...
            +imag(xi_2_new(j)).^2));
        H3 = lambda*log(sqrt((real(xi_2_new(j))+lambda-1).^2 ...
            +imag(xi_2_new(j)).^2));
        xi_2_new_H(j) = H1+H2+H3;
    end

% -----
% P3 Orbit Period
% -----

    t3_new = t0 : 0.01 : tf;
    xi_3_new = interp1(t3,xi_3,t3_new,'spline');
    Z1_3 = real(xi_3_new)+1.0;
    Z2_3 = imag(xi_3_new)+1.0;
    for i = 4:250
        if ((Z1_3(i) >= Z1_3(1)*0.99) && (Z1_3(i) <= Z1_3(1)*1.01) && ...
            (Z2_3(i) >= Z2_3(1)*0.99) && (Z2_3(i) <= Z2_3(1)*1.01)),
            P3_period = t3_new(i);
            disp('The initial orbit period for Particle No. 3 is: '),
            disp(t3_new(i)),
            disp(i),
            break
        end
    end

% -----
% P3 Staging Time
% -----

    for j = 1 : i
        Z3_3(j) = Z2_3(j);
    end
    for j = 1 : i
        if (Z3_3(j) == max(Z3_3)),
            P3_stage_t = t3_new(j);
            disp('The P3 staging time is:'),
            disp(P3_stage_t),
            break
        end
    end

    for j = 1 : i
        Z3_3(j) = Z2_3(j);
    end
end

```

```

for j = 1 : i
    if (Z3_3(j) == max(Z3_3)),
        P3_stage_t = t3_new(j);
        disp('The P3 staging time is: '),
        disp(P3_stage_t),
        break
    end
end

% -----
% P3 Orbit "Min"s and "Max"s
% -----

for j = 1 : i,
    xi_3_new_real(j) = real(xi_3_new(j));
    xi_3_new_imag(j) = imag(xi_3_new(j));
end

for j = 1 : i,
    xi_3_new_real_max = max(xi_3_new_real);
    xi_3_new_real_min = min(xi_3_new_real);
    xi_3_new_imag_max = max(xi_3_new_imag);
    xi_3_new_imag_min = min(xi_3_new_imag);
end

% -----
% Calculate P3 Orbit Energy-Levels (i.e. Hamiltonians)
% -----

for j = 1 : numel(t3_new)
    H1 = (-1/2)*(real(xi_3_new(j)).^2+imag(xi_3_new(j)).^2);
    H2 = (1-lambda)*log(sqrt((real(xi_3_new(j))+lambda).^2 ...
        +imag(xi_3_new(j)).^2));
    H3 = lambda*log(sqrt((real(xi_3_new(j))+lambda-1).^2 ...
        +imag(xi_3_new(j)).^2));
    xi_3_new_H(j) = H1+H2+H3;
end

% -----
% P4 Orbit Period
% -----

t4_new = t0 : 0.01 : tf;
xi_4_new = interp1(t4,xi_4,t4_new,'spline');
Z1_4 = real(xi_4_new)+1.0;
Z2_4 = imag(xi_4_new)+1.0;
for i = 4:250
    if ((Z1_4(i) >= Z1_4(1)*0.99) && (Z1_4(i) <= Z1_4(1)*1.01) && ...
        (Z2_4(i) >= Z2_4(1)*0.99) && (Z2_4(i) <= Z2_4(1)*1.01)),
        P4_period = t4_new(i);
        disp('The initial orbit period for Particle No. 4 is: '),
        disp(t4_new(i)),
        disp(i),
        break
    end
end

% -----
% P4 Staging Time
% -----

```

```

for j = 1 : i
    Z3_4(j) = Z2_4(j);
end

for j = 1 : i
    if (Z3_4(j) == max(Z3_4)),
        P4_stage_t = t4_new(j);
        disp('The P4 staging time is: '),
        disp(P4_stage_t),
        break
    end
end

% -----
% P4 Orbit "Min"s and "Max"s
% -----

for j = 1 : i,
    xi_4_new_real(j) = real(xi_4_new(j));
    xi_4_new_imag(j) = imag(xi_4_new(j));
end

for j = 1 : i,
    xi_4_new_real_max = max(xi_4_new_real);
    xi_4_new_real_min = min(xi_4_new_real);
    xi_4_new_imag_max = max(xi_4_new_imag);
    xi_4_new_imag_min = min(xi_4_new_imag);
end

% -----
% Calculate P4 Orbit Energy-Levels (i.e. Hamiltonians)
% -----

for j = 1 : numel(t4_new)
    H1 = (-1/2)*(real(xi_4_new(j)).^2+imag(xi_4_new(j)).^2);
    H2 = (1-lambda)*log(sqrt((real(xi_4_new(j))+lambda).^2 ...
        +imag(xi_4_new(j)).^2));
    H3 = lambda*log(sqrt((real(xi_4_new(j))+lambda-1).^2 ...
        +imag(xi_4_new(j)).^2));
    xi_4_new_H(j) = H1+H2+H3;
end

% -----
% Determine the characteristic distance for the periodic orbit
% associated with the desired energy-level (i.e. Hamiltonian).
% Rather than using Matlab to solve a rather complicated
% underdetermined linear system, a 5th-order polynomial curve
% fit of the data produced the necessary mathematical
% relationship. The "R squared" value for the expression below
% is 0.996.
% -----

d_Desired_1 = 0.659*H_Desired^5 + 5.4193*H_Desired^4 +17.392*H_Desired^3;
d_Desired_2 = 27.307*H_Desired^2 + 21.183*H_Desired + 6.6785;
d_Desired = d_Desired_1 + d_Desired_2;

% -----
% Produce the periodic orbit associated with the desired energy-
% level (i.e. Hamiltonian).
% -----

```



```

xi_0(5) = (0.50-d_Desired) + 0.0i;
[tDesired,xi_Desired] = ode78(@three_vortex, t0, tf, xi_0(5), tol);

% -----
% Find period of the new, desired orbit
% -----

t_Desired_new = t0 : 0.01 : tf;
xi_Desired_new = interp1(tDesired,xi_Desired,t_Desired_new,'spline');
Z1 = real(xi_Desired_new)+1.0;
Z2 = imag(xi_Desired_new)+1.0;
for i = 4:200
    if ((Z1(i) >= Z1(1)*0.99) && (Z1(i) <= Z1(1)*1.01) && ...
        (Z2(i) >= Z2(1)*0.99) && (Z2(i) <= Z2(1)*1.01)),
        desired_period = t_Desired_new(i);
        disp('The period of the desired orbit is: '),
        disp(t_Desired_new(i)),
        disp(i),
        break
    end
end

% -----
% Desired Orbit "Min"s and "Max"s
% -----

for j = 1 : i,
    xi_Desired_new_real(j) = real(xi_Desired_new(j));
    xi_Desired_new_imag(j) = imag(xi_Desired_new(j));
end

for j = 1 : i,
    xi_Desired_new_real_max = max(xi_Desired_new_real);
    xi_Desired_new_real_min = min(xi_Desired_new_real);
    xi_Desired_new_imag_max = max(xi_Desired_new_imag);
    xi_Desired_new_imag_min = min(xi_Desired_new_imag);
end

% -----
% Calculate Desired Orbit Energy-Levels (i.e. Hamiltonians)
% -----

for j = 1 : numel(t_Desired_new)
    xi_Desired_new_H(j) = H_Desired;
end

% -----
% Determine "time on transfer trajectory" for each particle. For
% now, we're going to set these values. Later, we'll call another
% script or function file or have a mathematical relationship
% included in this program (current in work).
% -----

% -----
% Default values
% -----

P1_trans_t = 0.40;
P2_trans_t = 0.34;
P3_trans_t = 0.26;

```

```

P4_trans_t = 0.20;

% -----
% Create transfer trajectory and calculate "time on transfer
% trajectory" for each of the four particle initial period orbits.
% -----

imag_string = 'i';
imag_str = sscanf(imag_string,'%c');

% -----
% P1 Transfer Trajectory
% -----

for j = 1 : 40
    kappa = 0.05*j;
    [tt_time_1, xi_1_tt] = ode78(@controller,0,0.5,...
                                0.5+str2num(imag_str)*xi_1_new_imag_max,tol);
    t_new = 0.0 : 0.005 : 0.5;
    xi_1_tt_possible = interp1(tt_time_1,xi_1_tt,t_new,'spline');
    if (min(imag(xi_1_tt_possible)) <= xi_Desired_new_imag_min*0.98) &&
...
        (min(imag(xi_1_tt_possible)) >= xi_Desired_new_imag_min*1.02),
        disp('STOP - The tt 1 has been found'),
        break
    end
end

for k = 1 : 100
    if (real(xi_1_tt_possible(k)) <= 0.5),
        xi_1_tt_final(k) = xi_1_tt_possible(k);
    end
end

P1_trans_t = numel(xi_1_tt_final)*0.01;

% -----
% P1 Transfer Trajectory Energy-Levels
% -----

for j = 1 : numel(xi_1_tt_final)
    H1 = (-1/2)*(real(xi_1_tt_final(j)).^2+imag(xi_1_tt_final(j)).^2);
    H2 = (1-lambda)*log(sqrt((real(xi_1_tt_final(j))+lambda).^2 ...
        +imag(xi_1_tt_final(j)).^2));
    H3 = lambda*log(sqrt((real(xi_1_tt_final(j))+lambda-1).^2 ...
        +imag(xi_1_tt_final(j)).^2));
    xi_1_tt_final_H(j) = H1+H2+H3;
end

% -----
% P2 Transfer Trajectory
% -----

for j = 1 : 40
    kappa = 0.05*j;
    [tt_time_2, xi_2_tt] = ode78(@controller,0,0.5,...
                                0.5+str2num(imag_str)*xi_2_new_imag_max,tol);
    t_new = 0.0 : 0.005 : 0.5;
    xi_2_tt_possible = interp1(tt_time_2,xi_2_tt,t_new,'spline');
    if (min(imag(xi_2_tt_possible)) <= xi_Desired_new_imag_min*0.98) &&
...

```

```

        (min(imag(xi_2_tt_possible)) >= xi_Desired_new_imag_min*1.02),
        disp('STOP - The tt 2 has been found'),
        break
    end
end

for k = 1 : 100
    if (real(xi_2_tt_possible(k)) <= 0.5),
        xi_2_tt_final(k) = xi_2_tt_possible(k);
    end
end

P2_trans_t = numel(xi_2_tt_final)*0.01;

% -----
% P2 Transfer Trajectory Energy-Levels
% -----

for j = 1 : numel(xi_2_tt_final)
    H1 = (-1/2)*(real(xi_2_tt_final(j)).^2+imag(xi_2_tt_final(j)).^2);
    H2 = (1-lambda)*log(sqrt((real(xi_2_tt_final(j))+lambda).^2 ...
        +imag(xi_2_tt_final(j)).^2));
    H3 = lambda*log(sqrt((real(xi_2_tt_final(j))+lambda-1).^2 ...
        +imag(xi_2_tt_final(j)).^2));
    xi_2_tt_final_H(j) = H1+H2+H3;
end

% -----
% P3 Transfer Trajectory
% -----

for j = 1 : 40
    kappa = 0.05*j;

    [tt_time_3, xi_3_tt] = ode78(@controller,0,0.5,...
        0.5+str2num(imag_str)*xi_3_new_imag_max,tol);
    t_new = 0.0 : 0.005 : 0.5;
    xi_3_tt_possible = interp1(tt_time_3,xi_3_tt,t_new,'spline');
    if (min(imag(xi_3_tt_possible)) <= xi_Desired_new_imag_min*0.97) &&
...
        (min(imag(xi_3_tt_possible)) >= xi_Desired_new_imag_min*1.04),
        disp('STOP - The tt 3 has been found'),
        break
    end
end

for k = 1 : 100
    if (real(xi_3_tt_possible(k)) <= 0.5),
        xi_3_tt_final(k) = xi_3_tt_possible(k);
    end
end

P3_trans_t = numel(xi_3_tt_final)*0.01;

% -----
% P3 Transfer Trajectory Energy-Levels
% -----

for j = 1 : numel(xi_3_tt_final)
    H1 = (-1/2)*(real(xi_3_tt_final(j)).^2+imag(xi_3_tt_final(j)).^2);
    H2 = (1-lambda)*log(sqrt((real(xi_3_tt_final(j))+lambda).^2 ...

```

```

        +imag(xi_3_tt_final(j)).^2));
    H3 = lambda*log(sqrt((real(xi_3_tt_final(j))+lambda-1).^2 ...
        +imag(xi_3_tt_final(j)).^2));
    xi_3_tt_final_H(j) = H1+H2+H3;
end

% -----
% P4 Transfer Trajectory
% -----

for j = 1 : 40
    kappa = 0.05*j;
    [tt_time_4,xi_4_tt] = ode78(@controller,0,0.5, ...
        0.5+str2num(imag_str)*xi_4_new_imag_max,tol);
    t_new = 0.0 : 0.005 : 0.5;
    xi_4_tt_possible = interp1(tt_time_4, xi_4_tt, t_new,'spline');
    if (min(imag(xi_4_tt_possible)) <= xi_Desired_new_imag_min*0.97) &&
...
        (min(imag(xi_4_tt_possible)) >= xi_Desired_new_imag_min*1.03),
        disp('STOP - The tt 4 has been found'),
        break
    end
end

for k = 1 : 100
    if (real(xi_4_tt_possible(k)) <= 0.5),
        xi_4_tt_final(k) = xi_4_tt_possible(k);
    end
end

P4_trans_t = numel(xi_4_tt_final)*0.01;

disp('hello');

% -----
% P4 Transfer Trajectory Energy-Levels
% -----

for j = 1 : numel(xi_4_tt_final)
    H1 = (-1/2)*(real(xi_4_tt_final(j)).^2+imag(xi_4_tt_final(j)).^2);
    H2 = (1-lambda)*log(sqrt((real(xi_4_tt_final(j))+lambda).^2 ...
        +imag(xi_4_tt_final(j)).^2));
    H3 = lambda*log(sqrt((real(xi_4_tt_final(j))+lambda-1).^2 ...
        +imag(xi_4_tt_final(j)).^2));
    xi_4_tt_final_H(j) = H1+H2+H3;
end

% -----
% PHASE-LOCKING AND FORMATION ESTABLISHMENT SECTION
% -----

% Define formation entry tolerance (e.g. +/- 0.01 units of time)

entry_tol = 0.01;

if (entry_tol_ques == 0),
    entry_tol = entry_tol_user;
end

% -----
% P1 Entry into the Formation

```

```

% -----

P1_total_t = P1_stage_t + P1_trans_t;

revs1 = int8(P1_total_t/desired_period);

% -----
% P2 Entry into the Formation
% -----

P2_entry_t = P1_total_t + 0.25*desired_period;

P2_total_t_maybe = P2_stage_t + P2_trans_t;

% if (P2_total_t_maybe <= P2_entry_t),
while (P2_total_t_maybe <= P2_entry_t),
    P2_total_t_maybe = P2_total_t_maybe + P2_period;
end

timer(1) = desired_period;
TIMER(1) = P2_period;

for n = 2 : 150
    timer(n) = timer(n-1) + desired_period;
end
for N = 2 : 50
    TIMER(N) = TIMER(N-1) + P2_period;
end

for n = 1 : 150
    for N = 1 : 50
        if ((TIMER(N) >= timer(n)-entry_tol) && ...
            (TIMER(N) <= timer(n)+entry_tol)),
            disp('Found it!'),
            disp(['Time since P1 entered formation: ' timer(n)]),
            disp(timer(n)),
            disp(TIMER(N)),
            revs2 = timer(n)/desired_period;
            disp('Number of full inner orbit revolutions is :'),
            disp(revs2),
            REVS2 = TIMER(N)/P2_period;
            disp('Number of full P2 orbit revolutions is :'),
            disp(REVS2),
            break
        end
        if ((TIMER(N) >= timer(n)-entry_tol) && ...
            (TIMER(N) <= timer(n)+entry_tol)),
            break,
        end,
    end
    end
    if ((TIMER(N) >= timer(n)-entry_tol) && ...
        (TIMER(N) <= timer(n)+entry_tol)),
        break,
    end
end
end

P2_total_t = P2_total_t_maybe + timer(n);

disp('P2_total_t ='),
disp(P2_total_t),

```

```

% -----
% P3 Entry into the Formation
% -----

P3_total_t_maybe = P3_stage_t + P3_trans_t;

P3_entry_t = P2_total_t + 0.25*desired_period;

while (P3_total_t_maybe <= P3_entry_t),
    P3_total_t_maybe = P3_total_t_maybe + P3_period;
end

timer(1) = desired_period;
TIMER(1) = P3_period;

for n = 2 : 150
    timer(n) = timer(n-1) + desired_period;
end
for N = 2 : 50
    TIMER(N) = TIMER(N-1) + P3_period;
end

for n = 1 : 150
    for N = 1 : 50
        if ((TIMER(N) >= timer(n)-entry_tol) && ...
            (TIMER(N) <= timer(n)+entry_tol)),
            disp('Found it!'),
            disp('Time since P2 particle entered formation: '),
            disp(timer(n)),
            disp(TIMER(N)),
            revs3 = timer(n)/desired_period;
            disp('Number of full inner orbit revolutions is :'),
            disp(revs3),
            REVS3 = TIMER(N)/P3_period;
            disp('Number of full P3 orbit revolutions is :'),
            disp(REVS3),
            break
        end
        if ((TIMER(N) >= timer(n)-entry_tol) && ...
            (TIMER(N) <= timer(n)+entry_tol)),
            break,
        end,
    end
    if ((TIMER(N) >= timer(n)-entry_tol) && ...
        (TIMER(N) <= timer(n)+entry_tol)),
        break,
    end
end

P3_total_t = P3_total_t_maybe + timer(n);

disp('P3_total_t ='),
disp(P3_total_t),

% -----
% P4 Entry into the Formation
% -----

P4_total_t_maybe = P4_stage_t + P4_trans_t;

P4_entry_t = P3_total_t + 0.25*desired_period;

```

```

while (P4_total_t_maybe <= P4_entry_t),
    P4_total_t_maybe = P4_total_t_maybe + P4_period;
end

timer(1) = desired_period;
TIMER(1) = P4_period;

for n = 2 : 150
    timer(n) = timer(n-1) + desired_period;
end
for N = 2 : 50
    TIMER(N) = TIMER(N-1) + P4_period;
end

for n = 1 : 150
    for N = 1 : 50
        if ((TIMER(N) >= timer(n)-entry_tol) && ...
            (TIMER(N) <= timer(n)+entry_tol)),
            disp('Found it!'),
            disp('Time since P3 entered formation: '),
            disp(timer(n)),
            disp(TIMER(N)),
            revs4 = timer(n)/desired_period;
            disp('Number of full inner orbit revolutions is :'),
            disp(revs4),
            REVS4 = TIMER(N)/P4_period;
            disp('Number of full P4 orbit revolutions is :'),
            disp(REVS4),
            break
        end
        if ((TIMER(N) >= timer(n)-entry_tol) && ...
            (TIMER(N) <= timer(n)+entry_tol)),
            break,
        end,
    end
    if ((TIMER(N) >= timer(n)-entry_tol) && ...
        (TIMER(N) <= timer(n)+entry_tol)),
        break,
    end
end

P4_total_t = P4_total_t_maybe + timer(n);

% =====
% THE FORMATION RESULTS
% =====

revs_all = revs1 + revs2 + revs3 + revs4;

for n = 1 : 100*desired_period+1
    ring(n) = xi_Desired_new(n);
end

% Always late

P1_position_A = xi_Desired_new(int8(1));
P2_position_A =
xi_Desired_new(int8(1+100*desired_period*3/4+100*entry_tol));
P3_position_A =
xi_Desired_new(int8(1+100*desired_period*1/2+100*entry_tol));

```

```

    P4_position_A =
xi_Desired_new(int8(1+100*desired_period*1/4+100*entry_tol));

% Always early

    P1_position_B = xi_Desired_new(int8(1));
    P2_position_B = xi_Desired_new(int8(1+100*desired_period*3/4-
100*entry_tol));
    P3_position_B = xi_Desired_new(int8(1+100*desired_period*1/2-
100*entry_tol));
    P4_position_B = xi_Desired_new(int8(1+100*desired_period*1/4-
100*entry_tol));

% =====
% Output Section
% =====

    disp(sprintf('Desired Orbit Energy = %1.4f', H_Desired));
    disp(sprintf('Desired Orbit Period = %1.2f',desired_period));
    disp(' ');
    disp('Initial Orbit Energy:');
    disp(sprintf('P1 = %1.4f; P2 = %1.4f; P3 = %1.4f; P4 = %1.4f', ...
        H(1),H(2),H(3),H(4)));
    disp(' ');
    disp('Initial Orbit Periods:');
    disp(sprintf('P1 = %1.2f; P2 = %1.2f; P3 = %1.2f; P4 = %1.2f', ...
        P1_period,P2_period,P3_period,P4_period));
    disp(' ');
    disp('Staging Time:');
    disp(sprintf('P1 = %1.2f; P2 = %1.2f; P3 = %1.2f; P4 = %1.2f', ...
        P1_stage_t,P2_stage_t,P3_stage_t,P4_stage_t));
    disp(' ');
    disp('Transfer Trajectory Time:');
    disp(sprintf('P1 = %1.2f; P2 = %1.2f; P3 = %1.2f; P4 = %1.2f', ...
        P1_trans_t,P2_trans_t,P3_trans_t,P4_trans_t));
    disp(' ');
    disp('Number of full revs for P1 formation entry:');
    disp(sprintf('P1 orbit = %d',0));
    disp(sprintf('Desired orbit = %1.0f',revs1));
    disp(' ');
    disp('Number of full revs for P2 formation entry:');
    disp(sprintf('P2 orbit = %1.0f',REVS2));
    disp(sprintf('Desired orbit = %1.0f',revs2));
    disp(' ');
    disp('Number of full revs for P3 formation entry:');
    disp(sprintf('P3 orbit = %1.0f',REVS3));
    disp(sprintf('Desired orbit = %1.0f',revs3));
    disp(' ');
    disp('Number of full revs for P4 formation entry:');
    disp(sprintf('P4 orbit = %1.0f',REVS4));
    disp(sprintf('Desired orbit = %1.0f',revs4));
    disp(' ');
    disp('Total Time Required to Enter Formation:');
    disp(sprintf('P1 = %1.2f; P2 = %1.2f; P3 = %1.2f; P4 = %1.2f', ...
        P1_total_t,P2_total_t,P3_total_t,P4_total_t));
    disp(' ');

% =====
% Plot Section
% =====

```



```

if (plots == 1),

% Plot initial energy-levels (i.e. Hamiltonians) for each particle

figure, plot(H,'o','MarkerSize',5,'MarkerFaceColor','k');
title('Initial Particle Energy-Levels (i.e. Hamiltonians)'),
xlabel('Particle No. '),
ylabel('Value of Hamiltonian, H'),
xlim([0 6]),
ylim([-1.2 -0.7]), % Use for default value of new H
% ylim([-2.5 -0.7]),
text(1.1,H(1),['Particle No. 1, H = ',num2str(H(1))]),
text(2.1,H(2),['Particle No. 2, H = ',num2str(H(2))]),
text(3.1,H(3),['Particle No. 3, H = ',num2str(H(3))]),
text(4.1,H(4),['Particle No. 4, H = ',num2str(H(4))]),
line(X,Y),
text(2.5,H_Desired+0.01,['Desired H: ',num2str(H_Desired)]),
grid on

% Plot the four initial and the single desired periodic orbits in the
% rotating, Cartesian coordinate frame

figure, plot(real(xi_1),imag(xi_1),'g',real(xi_2),imag(xi_2),'g',...
real(xi_3),imag(xi_3),'g',real(xi_4),imag(xi_4),'g',...
real(xi_Desired),imag(xi_Desired),'b',...
real(xi_0(1)),imag(xi_0(1)),'-ko',...
real(xi_0(2)),imag(xi_0(2)),'-ko',...
real(xi_0(3)),imag(xi_0(3)),'-ko',...
real(xi_0(4)),imag(xi_0(4)),'-ko','MarkerFaceColor','k'),
title('Initial Conditions and the Desired Periodic Orbit'),
xlabel('Real Axis'),
ylabel('Imaginary Axis'),
xlim([0 1]),
ylim([-0.5 0.5]),
grid on,
text(0.03+real(xi_0(1)),imag(xi_0(1)),['P1 = ',num2str(xi_0(1))]),
text(0.03+real(xi_0(2)),imag(xi_0(2)),['P2 = ',num2str(xi_0(2))]),
text(0.03+real(xi_0(3)),imag(xi_0(3)),['P3 = ',num2str(xi_0(3))]),
text(0.03+real(xi_0(4)),imag(xi_0(4)),['P4 = ',num2str(xi_0(4))]),
text(0.03,0.47,'Orbit Periods:'),
text(0.03,0.44,['P1 = ',num2str(P1_period)]),
text(0.03,0.41,['P2 = ',num2str(P2_period)]),
text(0.03,0.38,['P3 = ',num2str(P3_period)]),
text(0.03,0.35,['P4 = ',num2str(P4_period)]),
text(0.03,0.32,['Desired orbit = ',num2str(desired_period)]),
text(0.03,-0.30,'Energy-Levels:'),
text(0.03,-0.33,['P1 = ',num2str(H(1))]),
text(0.03,-0.36,['P2 = ',num2str(H(2))]),
text(0.03,-0.39,['P3 = ',num2str(H(3))]),
text(0.03,-0.42,['P4 = ',num2str(H(4))]),
text(0.03,-0.45,['Desired orbit = ',num2str(H_Desired)]),

% Plot staging and transfer trajectory times for each particle

figure, plot(1,P1_stage_t,'-ko',2,P2_stage_t,'-ko',...
3,P3_stage_t,'-ko',4,P4_stage_t,'-ko',...
1,P1_stage_t+P1_trans_t,'-ko',...

```

```

        2,P2_stage_t+P2_trans_t,'-ko',...
        3,P3_stage_t+P3_trans_t,'-ko',...
        4,P4_stage_t+P4_trans_t,'-ko',...
        'MarkerFaceColor','k'),
    title('Staging and Transfer Trajectory Times for Each Particle'),
    xlabel('Particle No.'),
    ylabel('Total Time, t'),
    xlim([0 5]),
    grid on,
    text(1.2,P1_stage_t,['P1 staging time = ',num2str(P1_stage_t)]),
    text(1.2,P1_stage_t+P1_trans_t,...
        ['P1 transfer tra] time = ',num2str(P1_trans_t)]),
    text(2.2,P2_stage_t,['P2 staging time = ',num2str(P2_stage_t)]),
    text(2.2,P2_stage_t+P2_trans_t,...
        ['P2 transfer tra] time = ',num2str(P2_trans_t)]),
    text(3.2,P3_stage_t,['P3 staging time = ',num2str(P3_stage_t)]),
    text(3.2,P3_stage_t+P3_trans_t,...
        ['P3 transfer tra] time = ',num2str(P3_trans_t)]),
    text(4.2,P4_stage_t,['P4 staging time = ',num2str(P4_stage_t)]),
    text(4.2,P4_stage_t+P4_trans_t,...
        ['P4 transfer tra] time = ',num2str(P4_trans_t)]),

% Plot formation entry times for each of the four particles

figure, plot(P1_total_t,1,'-ko',P2_total_t,2,'-ko',...
    P3_total_t,3,'-ko',P4_total_t,4,'-ko',...
    'MarkerFaceColor','k'),
    title('Formation Entry Times for Each of the Four Particles'),
    xlabel('Time, t'),
    ylabel('Particle No.'),
    xlim([0 P4_total_t+2.0]),
    ylim([0 5]),
    grid on,
    text(0.5,4.8,['Formation Tolerance = +/-',num2str(entry_tol)]),
    text(0.5,4.6,['A = Number of full desired orbit revolutions']),
    text(0.5,4.4,['B = Number of full initial orbit revolutions']),
    text(0.5,4.2,['Sum(A) = ',num2str(revs_all)]),
    text(0.5+P1_total_t,1,['P1 = ',num2str(P1_total_t)]),
    text(0.5+P1_total_t,1-0.2,['A = ',num2str(revs1)]),
    text(0.5+P1_total_t,1-0.4,['B = ',num2str(0)]),
    text(0.5+P2_total_t,2,['P2 = ',num2str(P2_total_t)]),
    text(0.5+P2_total_t,2-0.2,['A = ',num2str(revs2)]),
    text(0.5+P2_total_t,2-0.4,['B = ',num2str(REVS2)]),
    text(0.5+P3_total_t,3,['P3 = ',num2str(P3_total_t)]),
    text(0.5+P3_total_t,3-0.2,['A = ',num2str(revs3)]),
    text(0.5+P3_total_t,3-0.4,['B = ',num2str(REVS3)]),
    text(0.5+P4_total_t,4,['P4 = ',num2str(P4_total_t)]),
    text(0.5+P4_total_t,4-0.2,['A = ',num2str(revs4)]),
    text(0.5+P4_total_t,4-0.4,['B = ',num2str(REVS4)]),

% Plot resultant particle formation

figure, plot(real(ring),imag(ring),'b',...
    real(ring(1)),imag(ring(1)),'-ko',...
    real(ring(1+int8(100*desired_period/4))), ...
    imag(ring(1+int8(100*desired_period/4))),'-ko',...
    real(ring(1+int8(100*desired_period/2))), ...
    imag(ring(1+int8(100*desired_period/2))),'-ko',...
    real(ring(1+int8(100*desired_period*3/4))), ...
    imag(ring(1+int8(100*desired_period*3/4))),'-ko', ...
    real(P4_position_A),imag(P4_position_A),'d', ...

```

```

        real(P4_position_B),imag(P4_position_B),'d', ...
        real(P3_position_A),imag(P3_position_A),'d', ...
        real(P3_position_B),imag(P3_position_B),'d', ...
        real(P2_position_A),imag(P2_position_A),'d', ...
        real(P2_position_B),imag(P2_position_B),'d', ...
        real(P1_position_A),imag(P1_position_A),'d', ...
        real(P1_position_B),imag(P1_position_B),'d', ...
        'MarkerFaceColor','k'),
    title('Resultant Particle Formation at New Energy-Level'),
    xlabel('Real Axis'),
    ylabel('Imaginary Axis'),
    xlim([0 1]),
    ylim([-0.5 0.5]),
    grid on,
    text(max(real(xi_Desired))+0.03,0,'P3'),
    text(0.53,max(imag(xi_Desired)), 'P2'),
    text(min(real(xi_Desired))+0.03,0,'P1'),
    text(0.53,min(imag(xi_Desired)), 'P4'),
    text(0.05,0.45,'THIS PLOT STILL NEEDS WORK.')

% Plot resultant particle formation

figure,plot3(real(xi_1_new),imag(xi_1_new),xi_1_new_H,'g', ...
    real(xi_2_new),imag(xi_2_new),xi_2_new_H,'g', ...
    real(xi_3_new),imag(xi_3_new),xi_3_new_H,'g', ...
    real(xi_4_new),imag(xi_4_new),xi_4_new_H,'g', ...

real(xi_Desired_new),imag(xi_Desired_new),xi_Desired_new_H,'b', ...
    real(xi_0(1)),imag(xi_0(1)),H(1),'-ko', ...
    real(xi_0(2)),imag(xi_0(2)),H(2),'-ko', ...
    real(xi_0(3)),imag(xi_0(3)),H(3),'-ko', ...
    real(xi_0(4)),imag(xi_0(4)),H(4),'-ko', ...
    real(xi_1_tt_final),imag(xi_1_tt_final),xi_1_tt_final_H,'r',
...
    real(xi_2_tt_final),imag(xi_2_tt_final),xi_2_tt_final_H,'r',
...
    real(xi_3_tt_final),imag(xi_3_tt_final),xi_3_tt_final_H,'r',
...
    real(xi_4_tt_final),imag(xi_4_tt_final),xi_4_tt_final_H,'r',
...

    real(ring(1)),imag(ring(1)),xi_Desired_new_H,'-ko',...
    real(ring(1+int8(100*desired_period/4))), ...
    imag(ring(1+int8(100*desired_period/4))), ...
    xi_Desired_new_H,'-ko',...
    real(ring(1+int8(100*desired_period/2))), ...
    imag(ring(1+int8(100*desired_period/2))), ...
    xi_Desired_new_H,'-ko',...
    real(ring(1+int8(100*desired_period*3/4))), ...
    imag(ring(1+int8(100*desired_period*3/4))), ...
    xi_Desired_new_H,'-ko', ...
    'MarkerFaceColor','k'),
    title('Orbit, Trajectory, and Energy Plot'),
    xlabel('Real Axis'),
    ylabel('Imaginary Axis'),
    zlabel('Energy-Level'),
    xlim([0 1]),
    ylim([-0.5 0.5]),
    zlim([-1.2 -0.5]),
    grid on,

```

```

end % This the "end" for the "if statement" re:plots

% =====
% Animation Section
% =====

if (animations == 1),

    figure, plot(real(xi_1),imag(xi_1),'g',real(xi_2),imag(xi_2),'g',...
        real(xi_3),imag(xi_3),'g',real(xi_4),imag(xi_4),'g',...
        real(xi_Desired),imag(xi_Desired),'b',...
        real(xi_0(1)),imag(xi_0(1)),'-ko',...
        real(xi_0(2)),imag(xi_0(2)),'-ko',...
        real(xi_0(3)),imag(xi_0(3)),'-ko',...
        real(xi_0(4)),imag(xi_0(4)),'-
ko','MarkerFaceColor','k'),
        title('Uncontrolled Particle Motion'),
        xlabel('Real Axis'),
        ylabel('Imaginary Axis'),
        xlim([0 1]),
        ylim([-0.5 0.5]),
        grid on,
        text(0.03+real(xi_0(1)),imag(xi_0(1)),'P1'),
        text(0.03+real(xi_0(2)),imag(xi_0(2)),'P2'),
        text(0.03+real(xi_0(3)),imag(xi_0(3)),'P3'),
        text(0.03+real(xi_0(4)),imag(xi_0(4)),'P4'),
        text(0.03,0.47,'Orbit Periods:'),
        text(0.03,0.44,['P1 = ',num2str(P1_period)]),
        text(0.03,0.41,['P2 = ',num2str(P2_period)]),
        text(0.03,0.38,['P3 = ',num2str(P3_period)]),
        text(0.03,0.35,['P4 = ',num2str(P4_period)]),
        text(0.03,0.32,['Desired orbit = ',num2str(desired_period)]),
        text(0.03,-0.30,'Energy-Levels:'),
        text(0.03,-0.33,['P1 = ',num2str(H(1))]),
        text(0.03,-0.36,['P2 = ',num2str(H(2))]),
        text(0.03,-0.39,['P3 = ',num2str(H(3))]),
        text(0.03,-0.42,['P4 = ',num2str(H(4))]),
        text(0.03,-0.45,['Desired orbit = ',num2str(H_Desired)]),
        hold on
    for z = 1 : 52
%       for z = 1 : 2
        plot(real(xi_1_new(z)),imag(xi_1_new(z)),'o',...
            real(xi_2_new(z)),imag(xi_2_new(z)),'o',...
            real(xi_3_new(z)),imag(xi_3_new(z)),'o',...
            real(xi_4_new(z)),imag(xi_4_new(z)),'o',...
            'EraseMode','none','MarkerSize',5),
            hold on
        F(z) = getframe;
    end

    figure, plot(real(xi_3),imag(xi_3),'g',...
        real(xi_3_tt_final),imag(xi_3_tt_final),'r',...
        real(xi_Desired),imag(xi_Desired),'b',...
        real(xi_0(3)),imag(xi_0(3)),'-ko',...
        'MarkerFaceColor','k'),
        title('Controlled Particle Motion - Particle No. 3'),
        xlabel('Real Axis'),
        ylabel('Imaginary Axis'),
        xlim([0 1]),
        ylim([-0.5 0.5]),
        grid on,

```

```

text(0.03+real(xi_0(3)),imag(xi_0(3)),'P3'),
text(0.03,0.47,'Orbit Periods:'),
text(0.03,0.44,['P1 = ',num2str(P1_period)]),
text(0.03,0.41,['P2 = ',num2str(P2_period)]),
text(0.03,0.38,['P3 = ',num2str(P3_period)]),
text(0.03,0.35,['P4 = ',num2str(P4_period)]),
text(0.03,0.32,['Desired orbit = ',num2str(desired_period)]),
text(0.03,-0.30,'Energy-Levels:'),
text(0.03,-0.33,['P1 = ',num2str(H(1))]),
text(0.03,-0.36,['P2 = ',num2str(H(2))]),
text(0.03,-0.39,['P3 = ',num2str(H(3))]),
text(0.03,-0.42,['P4 = ',num2str(H(4))]),
text(0.03,-0.45,['Desired orbit = ',num2str(H_Desired)]),
hold on
for z = 1 : 100*P3_stage_t
plot(real(xi_3_new(z)),imag(xi_3_new(z)),'o',...
'EraseMode','none','MarkerSize',5),
hold on
G(z) = getframe;
end
for z = 1+(100*P3_stage_t) : (100*P3_stage_t)+(P3_trans_t*100)
plot(real(xi_3_tt_final(z-100*P3_stage_t)),...
imag(xi_3_tt_final(z-100*P3_stage_t)),'o',...
'EraseMode','none','MarkerSize',5),
hold on
G(z) = getframe;
end
end

zbest = (100*P3_stage_t)+(P3_trans_t*100);

for z = 1+zbest : zbest+int8(desired_period*100*3/4)
plot(real(xi_Desired_new(z-zbest+int8(desired_period*100/4))),...
imag(xi_Desired_new(z-
zbest+int8(desired_period*100/4))), 'o',...
'EraseMode','none','MarkerSize',5),
hold on
G(z) = getframe;
end

end % This is the "end" for the "if statement" re:animations

% ===== end =====

```

ode78.m

```

function [tout, yout] = ode78(F, t0, tfinal, y0, tol, trace)
% The Fehlberg coefficients:
% From Matlab website, 1996

alpha = [ 2./27. 1/9 1/6 5/12 .5 5/6 1/6 2/3 1/3 1 0 1 ]';
beta = [ [ 2/27 0 0 0 0 0 0 0 0 0 0 0 ]
[ 1/36 1/12 0 0 0 0 0 0 0 0 0 0 ]
[ 1/24 0 1/8 0 0 0 0 0 0 0 0 0 ]
[ 5/12 0 -25/16 25/16 0 0 0 0 0 0 0 0 ]
[ .05 0 0 .25 .2 0 0 0 0 0 0 0 ]

```

```

[ -25/108  0  0  125/108  -65/27  125/54  0  0  0  0  0  0  0  0 ]
[ 31/300  0  0  0  61/225  -2/9  13/900  0  0  0  0  0  0  0 ]
[ 2  0  0  -53/6  704/45  -107/9  67/90  3  0  0  0  0  0  0 ]
[ -91/108  0  0  23/108  -976/135  311/54  -19/60  17/6  -1/12  0  0  0  0 ]
]
[2383/4100 0 0 -341/164 4496/1025 -301/82 2133/4100 45/82 45/164 18/41 0 0
0]
[ 3/205  0  0  0  0  -6/41  -3/205  -3/41  3/41  6/41  0  0  0 ]
[-1777/4100 0 0 -341/164 4496/1025 -289/82 2193/4100 ...
51/82 33/164 12/41 0 1 0]...
]';
chi = [ 0 0 0 0 0 34/105 9/35 9/35 9/280 9/280 0 41/840 41/840]';
psi = [1 0 0 0 0 0 0 0 0 0 0 1 -1 -1]';
pow = 1/8;
if nargin < 6, trace = 0; end
if nargin < 5, tol = 1.e-6; end

% Initialization
t = t0;
hmax = (tfinal - t)/2.5;
hmin = (tfinal - t)/800000000; % tweek THIS
h = (tfinal - t)/100;
y = y0(:);
f = y*zeros(1,13);
tout = t;
yout = y.';
tau = tol * max(norm(y, 'inf'), 1);

if trace
% clc, t, h, y
    clc, t, y
end
% The main loop
while (t < tfinal) & (h >= hmin)
    if t + h > tfinal, h = tfinal - t; end

    % Compute the slopes
    f(:,1) = feval(F,t,y);
    for j = 1: 12
        f(:,j+1) = feval(F, t+alpha(j)*h, y+h*f*beta(:,j));
    end

    % Truncation error term
    gammal = h*41/840*f*psi;

    % Estimate the error and the acceptable error
    delta = norm(gammal,'inf');
    tau = tol*max(norm(y,'inf'),1.0);

    % Update the solution only if the error is acceptable
    if delta <= tau
        t = t + h;
        y = y + h*f*chi;
        tout = [tout; t];
        yout = [yout; y.'];
    end
    if trace
%
        home, t, h, y
        home, t, y
    end
end

```

```

    % Update the step size
    if delta ~= 0.0
        h = min(hmax, 0.8*h*(tau/delta)^pow);
    end
end;

if (t < tfinal)
    disp('SINGULARITY LIKELY.')
    t
end

```

three_body_script_v5_1.m

```

% =====
% =
% = script: three_body_script_v5.1
% =
% = MathWorks Matlab Script for Circular, Restricted Three-Body
% = Problem (CR3BP)
% =
% = Multiple (Formation Flying) Spacecraft
% = Matlab Movie and AVI (Audio Visual Interleaved) Functions
% = Utilized for Viewing Spacecraft Formation Over Time
% =
% = Written by Ralph R. Basilio, Ph.D. Student
% = Version: 5.1
% = Date: 06 March 2005 (Original: 23 July 2004)
% =
% = AME 790 Research - Advisor: Professor Paul K. Newton
% = Aerospace and Mechanical Engineering Department
% = Viterbi School of Engineering
% = University of Southern California
% =
% = This script uses a seventh & eighth-order Runge-Kutta-Fehlberg
% = integration method to produce an accurate solution.
% =
% =====

% Set initial time and final time
tspan = [ 0 6.3 ];

t0 = 0;
tf = 21.070352;

tol = 1.0e-9;

% Define spacecraft 1 initial conditions (i.e. column vector g0)
% y1_0(1): Position vector, X coordinate
% y1_0(2): Position vector, Y coordinate
% y1_0(3): Position vector, Z coordinate
% y1_0(4): Velocity vector, X direction
% y1_0(5): Velocity vector, Y direction
% y1_0(6): Velocity vector, Z direction
y1_0 = [ 0.61523162 0.86349029 0.0 0.08274045 -0.06439854 0.0]';

% Define spacecraft 2 initial conditions (i.e. column vector g0)
% y2_0(1): Position vector, X coordinate
% y2_0(2): Position vector, Y coordinate
% y2_0(3): Position vector, Z coordinate

```

```

% y2_0(4): Velocity vector, X direction
% y2_0(5): Velocity vector, Y direction
% y2_0(6): Velocity vector, Z direction
y2_0 = [ 0.55585566 -0.8004144 0.0 0.02769284 0.01245845 0.0]';

% Define spacecraft 3 initial conditions (i.e. column vector g0)
% y3_0(1): Position vector, X coordinate
% y3_0(2): Position vector, Y coordinate
% y3_0(3): Position vector, Z coordinate
% y3_0(4): Velocity vector, X direction
% y3_0(5): Velocity vector, Y direction
% y3_0(6): Velocity vector, Z direction
y3_0 = [ 0.0 0.0 0.0 0.0 0.0 0.0]';

% Define Options
% options = odeset('RelTol', 1e-5, 'AbsTol', 1e-4, 'OutputFcn', @odephas2);

% Invoke Matlab integrator (i.e. ode45)
[t,y1] = ode78(@three_body_v4_1, t0, tf, y1_0, tol);
[t,y2] = ode78(@three_body_v4_2, t0, tf, y2_0, tol);
[t,y3] = ode78(@three_body_v4_3, t0, tf, y3_0, tol);

% Plot spacecraft 1 position vector components (remove comment symbol, "%")
% figure; plot(y1(:,1),y1(:,2));
% title('Restricted Three-Body Problem - Spacecraft 1');
% ylabel('y(t)');
% xlabel('x(t)');

% Plot spacecraft 2 position vector components (remove comment symbol, "%")
% figure; plot(y2(:,1),y2(:,2));
% title('Restricted Three-Body Problem - Spacecraft 2');
% ylabel('y(t)');
% xlabel('x(t)');

% Plot spacecraft 3 position vector components (remove comment symbol, "%")
% figure; plot(y3(:,1),y3(:,2));
% title('Restricted Three-Body Problem - Spacecraft 3');
% ylabel('y(t)');
% xlabel('x(t)');

% Plot formation - 2D View
% figure,
plot(y1(:,1),y1(:,2),'r',y2(:,1),y2(:,2),'b',y3(:,1),y3(:,2),'g');
% grid on;
% title('Restricted Three-Body Problem - 2D View of Formation');
% xlabel('x(t)');
% ylabel('y(t)');
% zlabel('z(t)');

% Determine position vector magnitude, velocity vector magnitude (speed),
% and angular rate

L4x = 0.48785;
L4y = 0.8660;
L5x = 0.48785;
L5y = -0.8660;

pos_mag_1 = sqrt((y1(:,1)-L4x).^2+(y1(:,2)-L4y).^2);
pos_mag_2 = sqrt((y2(:,1)-L5x).^2+(y2(:,2)-L5y).^2);
vel_mag_1 = sqrt(y1(:,4).^2+y1(:,5).^2);

```



```

vel_mag_2 = sqrt(y2(:,4).^2+y2(:,5).^2);
omega_1    = vel_mag_1(:,1).*(2*pi*pos_mag_1(:,1)).^(-1);
omega_2    = vel_mag_2(:,1).*(2*pi*pos_mag_2(:,1)).^(-1);

% Plot formation - 2D view
figure, plot(y1(:,1),y1(:,2),'-r+',...
             y2(:,1),y2(:,2),'-gx',...
             -0.012150,0,'-bo',...
             1-0.012150,0,'-bo',...
             0.48785,0.8660,'-ko',...
             0.48785,-0.8660,'-ko',...
             'MarkerFaceColor','b'),
            grid on,
            title('CR3BP - Two-Dimensional View of Spacecraft Formation'),
            legend('Spacecraft 1','Spacecraft 2',2),
            text(0.012150,-0.1,'Earth'), text(1-0.012150,-0.1,'Moon'),
            text(0.48785,0.7,'L4'), text(0.48785,-0.7,'L5'),
            axis square, xlim([-1.1 1.1]), ylim([-1.1 1.1]),
            xlabel('x(t)'), ylabel('y(t)');

% Plot position magnitude

figure, plot(pos_mag_1,'-r+'), grid on,
            title('Spacecraft 1 - Distance from L4'),
            xlim([0 45]),ylim([0 0.35]),
            xlabel('Data Point No. '), ylabel('Normalized Distance');

figure, plot(pos_mag_2,'-gx'), grid on,
            title('Spacecraft 2 - Distance from L5'),
            xlim([0 45]),ylim([0 0.35]),
            xlabel('Data Point No. '), ylabel('Normalized Distance');

% Plot angular rate versus distance

figure, plot(omega_1,pos_mag_1,'-r+'), grid on,
            title('Spacecraft 1 - Angular Rate vs Distance from L4'),
            xlim([0 0.25]),ylim([0 0.35]),
            xlabel('Angular Rate'),ylabel('Distance from L4');

figure, plot(omega_2,pos_mag_2,'-gx'), grid on,
            title('Spacecraft 2 - Angular Rate vs Distance from L5'),
            xlim([0 0.25]),ylim([0 0.35]),
            xlabel('Angular Rate'),ylabel('Distance from L5');

% Plot formation - 3D View
figure, plot3(y1(:,1),y1(:,2),y1(:,3),'-r+',...
              y2(:,1),y2(:,2),y2(:,3),'-bx',...
              y3(:,1),y3(:,2),y3(:,3),'-g*'),
              'MarkerEdgeColor','r',...
              'MarkerFaceColor',[1 0 0],...
              'MarkerSize',2,...
              y2(:,1),y2(:,2),y2(:,3),'b', y3(:,1),y3(:,2),y3(:,3),'g');
            grid on,
            title('Restricted Three-Body Problem - 3D View of Formation'),
            legend('spacecraft 1 (c)','spacecraft 2 (l)','spacecraft 3 (r)'),
            xlabel('x(t)'), ylabel('y(t)'), zlabel('z(t)');

figure, plot3(y1(:,1),y1(:,2),y1(:,3))
for i=1:21

```

```

plot3(y1(i*6,1),y1(i*6,2),y1(i*6,3),'r+',...
      y2(i*6,1),y2(i*6,2),y2(i*6,3),'bx',...
      y3(i*6,1),y3(i*6,2),y3(i*6,3),'g*'),
grid on,
axis([-1.5 1.5 -1 1 -0.2 0.2]),
title('Restricted Three-Body Problem'),
legend('spacecraft 1 (c)','spacecraft 2 (l)','spacecraft 3 (r)'),
xlabel('x(t)'), ylabel('y(t)'), zlabel('z(t)');

% Identify pip (point in plane)
pip(i*6,1) = (1/3)*(y1(i*6,1)+y2(i*6,1)+y3(i*6,1));
pip(i*6,2) = (1/3)*(y1(i*6,2)+y2(i*6,2)+y3(i*6,2));
pip(i*6,3) = (1/3)*(y1(i*6,3)+y2(i*6,3)+y3(i*6,3));

% Define Vector A (from pip to spacecraft 2)
A(i*6,1) = pip(i*6,1)-y2(i*6,1);
A(i*6,2) = pip(i*6,2)-y2(i*6,2);
A(i*6,3) = pip(i*6,3)-y2(i*6,3);

% Define Vector B (from pip to spacecraft 3)
B(i*6,1) = pip(i*6,1)-y3(i*6,1);
B(i*6,2) = pip(i*6,2)-y3(i*6,2);
B(i*6,3) = pip(i*6,3)-y3(i*6,3);

% Define Vector C (vector normal to spacecraft formation plane)
C(i*6,1) = A(i*6,2)*B(i*6,3)-A(i*6,3)*B(i*6,2);
C(i*6,2) = -A(i*6,1)*B(i*6,3)-A(i*6,3)*B(i*6,1);
C(i*6,3) = A(i*6,1)*B(i*6,2)-A(i*6,2)*B(i*6,1);

C_length = sqrt(C(i*6,1).^2+C(i*6,2).^2+C(i*6,3).^2);

C_unit(i*6,1) = C(i*6,1)/C_length;
C_unit(i*6,2) = C(i*6,2)/C_length;
C_unit(i*6,3) = C(i*6,3)/C_length;

obs(i*6,1) = 10*C_unit(i*6,1);
obs(i*6,2) = 10*C_unit(i*6,2);
obs(i*6,3) = 10*C_unit(i*6,3);

% view([0 -10 0]) % View XZ plane
% view([-10 0 0]) % View YZ plane
% view([0 0 -10]) % View XY plane
% view([-10 -10 10]) % Three-dimensional view
view([obs(i*6,1) obs(i*6,2) obs(i*6,3)]) % Moving observer

% view_x = 0.0; % DO= 0.0, XZ= 0.0, YZ=-3.0, XY= 0.0
% view_y = -3.0; % DO=-3.0, XZ=-3.0, YZ= 0.0, XY= 0.0
% view_z = 0.0; % DO= 1.0, XZ= 0.0, YZ= 0.0, XY= 3.0
% campos = ([cpx, cpy, cpz])
% view = ([view_x,view_y,view_z])
% ctx = 0.0;
% cty = 0.0;
% ctz = 0.0;
% camtarget = ([ctx, cty, ctz])
% camlookat(y1);
F(i) = getframe(gcf);
% figure, plot3(y1(i*6,1),y1(i*6,2),y1(i*6,3),'-r+'),
% plot3(y1(i*6,1),y1(i*6,2),y1(i*6,3),'-r+'),
% grid on,
% axis([-1.5 1.5 -1.5 1.5 -1.5 1.5]),
% legend('spacecraft 1 (c)','spacecraft 2 (l)','spacecraft 3 (r)'),

```

```
%    xlabel('x(t)'), ylabel('y(t)'), zlabel('z(t)')
%    drawnow
%    h=gcf
end
```

```
movie(F,1)
```

three_body_v4_1.m

```
% =====
% =
% = function: three_body_v4_1
% =
% = MathWorks Matlab Function for the Circular, Restricted
% = Three-Body Problem (CR3BP)
% =
% = Written by Ralph R. Basilio, Ph.D. Student
% = Version: 4.0
% = Date: 06 May 2004
% =
% = AME 790 Research - Advisor: Professor Paul K. Newton
% = Aerospace and Mechanical Engineering Department
% = Viterbi School of Engineering
% = University of Southern California
% =
% =====
%
% -----
% Create a function containing the governing equations of motion
% for mass 3 (the third body of infinitesimally small mass):
%
% x_dot_dot = 2*y_dot*x-(1-mu)*(x+mu)/r1^3-mu*(x-(1-mu))/r2^3
% y_dot_dot = -2*x_dot*y-(1-mu)*y/r1^3-mu*y/r2^3
% z_dot_dot = -(1-mu)*z/r1^3-mu*z/r2^3
% -----
% The second order equations above can be re-written as a system
% of first order differential equations (state vector):
% y_dot_sub1_1=y(4)
% y_dot_sub1_2=y(5)
% y_dot_sub1_3=y(6)
% y_dot_sub2_1=2*y(5)+y(1)-(1-mu)*(y(1)+mu)/r1^3-mu*(y(1)-(1-mu))/r2^3
% y_dot_sub2_2=-2*y(4)+y(2)-(1-mu)*y(2)/r1^3-mu*y(2)/r2^3
% y_dot_sub2_3=-(1-mu)*y(3)/r1^3-mu*y(3)/r2^3
% -----
% Define functions
function y1_dot = three_body_v4_1(t,y1)

% Mass of first object, mass1 = earth (kg)
% Mass of second object, mass2 = moon (kg)

% Define mu, normalized mass of second object
% mu = mass_2/(mass_1+mass_2)
mu = 7.1688e22/(5.974e24+7.1688e22);

% Determine magnitude of position vector from mass 1, r1
r1 = sqrt ((y1(1) + mu )^2 + y1(2)^2);

% Determine magnitude of position vector from mass 2, r2
```

```

r2 = sqrt ((y1(1) - (1-mu))^2 + y1(2)^2);

y1_dot = [
                                                    y1(4)
                                                    y1(5)
                                                    y1(6)
2*y1(5) + y1(1) - (1-mu)*(y1(1)+mu)/r1^3 - mu*(y1(1)-(1-mu))/r2^3
-2*y1(4) + y1(2) - (1-mu)*    y1(2)/r1^3 - mu*y1(2)    /r2^3
              - (1-mu)*    y1(3)/r1^3 - mu*y1(3)    /r2^3];

```

three_vortex.m

```

% =====
% =
% = function: three_vortex
% =
% = MathWorks Matlab Function for the Restricted Three-Vortex
% = Problem
% =
% = Written by Ralph R. Basilio, Ph.D. Candidate
% = Version: 1.0
% = Date: 06 July 2005
% =
% = AME 790 Research - Advisor: Professor Paul K. Newton
% = Aerospace and Mechanical Engineering Department
% = Viterbi School of Engineering
% = University of Southern California
% =
% =====

% =====
% Define function
% =====

function xi_dot = three_vortex(t,xi)

% =====
% Define global parameters
% =====

global cap_gamma_1 cap_gamma_2 D x11 xi2

% Determine the orbit frequency of both primary vortices

omega = (cap_gamma_1 + cap_gamma_2)/(2*pi*D^2);

% =====
% Equations section
% =====

% A = 2.37*-i*omega*xi;
% A = 1.0*-i*omega*xi;
% B = (i*cap_gamma_1)/(2*pi)*(xi-x11)/abs(xi-x11).^2;
% C = (i*cap_gamma_2)/(2*pi)*(xi-xi2)/abs(xi-xi2).^2;

% Standard equation of motion
xi_dot = A + B + C;

% Equation of motion with an anti-damping term

```

```
% With time dependency
% xi_dot = A + B + C + 3.000*t;

% Without time dependency
% xi_dot = A + B + C + 0.29;

% With periodic/sinusoidal dependency
% xi_dot = A + B + C + 0.011*sin((t/0.85)*(1.00*pi));

% ===== end =====
```

two_body_init_v5.m

```
% =====
% =
% = two_body_init_v5.m
% =
% = MathWorks Matlab Script for Two-Body Orbit Propagation
% =
% = Written by Ralph R. Basilio, Ph.D. Student
% = Version: 5.0
% = Date: 02 July 2004
% =
% = AME 790 Research - Advisor: Professor Paul K. Newton
% = Aerospace and Mechanical Engineering Department
% = Viterbi School of Engineering
% = University of Southern California
% =
% = This script calls a fourth and fifth-order Runge-Kutta-Fehlberg
% = integration function to produce an accurate solution.
% =
% =====
%
% Initial Conditions:
%
% Define gravitational constant (km^3/kg-sec^2) and masses (kg)
%
global m1 m2 G d
G = 6.6720e-20;
m1 = 5.974e24 ; % Mass of the Earth
% m1 = 7.1688e22; % Mass of the Moon;
% m2 = 5.974e24 ; % Mass of the Earth
m2 = 7.1688e22; % Mass of the Moon
% m2 = 1000.0 ; % Mass of a small spacecraft
d = 384467.0 ; % Separation distance between masses (km)
%
% x0(1) through x0(3): mass 1, init pos (x,y,z) - km
% x0(4) through x0(6): mass 2, init pos (x,y,z) - km
% x0(7) through x0(9): mass 1, init vel (x,y,z) - km/sec
% x0(10) through x0(12): mass 2, init vel (x,y,z) - km/sec
%
% Note: To prevent center-of-mass migration, set the initial velocity
% of the other body in a direction opposite and magnitude inversely
% proportional to the mass ratio.
%
% Note: Center-of-mass defined to be coordinate frame origin
%
% x0(1) = -d/((m1/m2)+1);
```

```

x0(1) = -d/(m1/m2);
x0(2) = 0.0;
x0(3) = 0.0;
x0(4) = d+x0(1);
x0(5) = 0.0;
x0(6) = 0.0;
x0(7) = 0.0;
x0(8) = -0.9468/(m1/m2);
x0(9) = 0.0;
x0(10) = 0.0;
x0(11) = 0.9468;
x0(12) = 0.0;

% x0=[0 0 0 384467 0 0 0.0 -0.9468/83.33 0.0 0.0 0.9468 0.0]';
% x0=[-1 0 0 1 0 0 0.0 -1.0 0.0 0.0 2.0 0.0]';
% x0=[0 0 0 384467 0 0 0.0 -0.9468 0.0 0.0 0.9468 0.0]';
%
% =====
%
% Define time span in sec:
%
start = 0;
increment = 10000;
stop = 25920000;
tspan = [start : increment : stop];
%
% =====
%
% Set options:
%
options = odeset('RelTol', 1e-5, 'AbsTol', 1e-4, 'OutputFcn',
@odephas2,...
'OutputSel', [4 5]);
options = odeset('RelTol', 1e-5, 'AbsTol', 1e-8);
%
% =====
%
% Call ODE (Ordinary Differential Equation) solver:
%
[t,x] = ode45('two_body_func', tspan, x0, options);
%
% =====
%
% Calculate the magnitude of the center-of-mass velocity (speed)
%
cm_vel(:,1) = (x(:,7)*m1 + x(:,10)*m2) / (m1+m2);
cm_vel(:,2) = (x(:,8)*m1 + x(:,11)*m2) / (m1+m2);
cm_vel(:,3) = (x(:,9)*m1 + x(:,12)*m2) / (m1+m2);
% cm_vel_mag = norm (cm_vel);
cm_vel_mag = sqrt(cm_vel(:,1).^2+cm_vel(:,2).^2+cm_vel(:,3).^2);
%
% =====
%
% Determine total energy, E (kg-km^2/sec^2)
%
r1 = x(:,1)-x(:,4);
r2 = x(:,2)-x(:,5);
r3 = x(:,3)-x(:,6);
r_mag = (r1.^2+r2.^2+r3.^2).^0.5;
v1 = x(:,7)-x(:,10);
v2 = x(:,8)-x(:,11);

```

```

% v3 = x(:,9)-x(:,12);
% v_mag = (v1.^2+v2.^2+v3.^2).^0.5;
% KE = (1/2)*m2*v_mag.^2;
% PE = -G*m1*m2*r_mag.^-1;
% E = KE+PE;

% Energy of mass 1 (per unit/total mass) about the barycenter
r1(:,1) = x(:,1);
r1(:,2) = x(:,2);
r1(:,3) = x(:,3);
r1_mag = sqrt(r1(:,1).^2+r1(:,2).^2+r1(:,3).^2);
v1(:,1) = x(:,7);
v1(:,2) = x(:,8);
v1(:,3) = x(:,9);
v1_mag = sqrt(v1(:,1).^2+v1(:,2).^2+v1(:,3).^2);
KE1 = (1/2)*v1_mag.^2;
PE1 = G*m2*r1_mag.^-1;
E1 = (KE1-PE1);

% Energy of mass 2 (per unit/total mass) about the barycenter
r2(:,1) = x(:,4);
r2(:,2) = x(:,5);
r2(:,3) = x(:,6);
r2_mag = sqrt(r2(:,1).^2+r2(:,2).^2+r2(:,3).^2);
v2(:,1) = x(:,10);
v2(:,2) = x(:,11);
v2(:,3) = x(:,12);
v2_mag = sqrt(v2(:,1).^2+v2(:,2).^2+v2(:,3).^2);
KE2 = (1/2)*v2_mag.^2;
PE2 = G*m1*r2_mag.^-1;
E2 = (KE2-PE2);

% Total Energy
E = E2-E1;

% Energy about mass 1 and about mass 2
dist = sqrt((x(:,1)-x(:,4)).^2+(x(:,2)-x(:,5)).^2+(x(:,3)-x(:,6)).^2);
speed = sqrt((x(:,7)-x(:,10)).^2+(x(:,8)-x(:,11)).^2+(x(:,9)-
x(:,12)).^2);
% E_m1 = speed^2/2-G/dist;
% E_m2 = speed^2/2-G/dist;
%

%
% =====
%
% Generate plots [remove comment symbol(s), %, as appropriate]
%
% Plot mass 1 position vector components
% figure; plot(t,x(:,1),'r',t,x(:,2),'g',t,x(:,3),'b')
% title('Mass 1 Position Vector Components Versus Time')
% ylabel('Position (km)'), xlabel('Time (sec)'), legend('x','y','z');
%
% Plot mass 1 velocity vector components
% figure; plot(t,x(:,7),'r',t,x(:,8),'g',t,x(:,9),'b')
% title('Mass 1 Velocity Vector Components Versus Time')
% ylabel('Velocity (km/sec)'), xlabel('Time (sec)'), legend('x','y','z');
%
% Plot mass 2 position vector components

```

```

% figure; plot(t,x(:,4),'r',t,x(:,5),'g',t,x(:,6),'b')
% title('Mass 2 Position Vector Components Versus Time')
% ylabel('Position (km)'), xlabel('Time (sec)'), legend('x','y','z');
%
% Plot mass 2 velocity vector components
% figure; plot(t,x(:,10),'r',t,x(:,11),'g',t,x(:,12),'b')
% title('Mass 2 Velocity Vector Components Versus Time')
% ylabel('Velocity (km/sec)'), xlabel('Time (sec)'), legend('x','y','z');
%
% Two-dimension orbit plot (mass 1)
% figure, plot(x(:,1),x(:,2),'b'), title('Two-Dimension Orbit Plot (Mass
1)')
% ylabel('y'), xlabel('x'), axis equal;
%
% Two-dimension orbit plot (mass 2)
% figure, plot(x(:,4),x(:,5),'r'), title('Two-Dimension Orbit Plot (Mass
2)')
% ylabel('y'), xlabel('x'), axis equal;
%
% Two-dimension orbit plot (mass 1 and mass 2)
figure, plot(x(:,1),x(:,2),'b',x(:,4),x(:,5),'r'),
title('Two-Dimension Orbit Plot (Mass 1 and Mass 2)')
ylabel('y'), xlabel('x'), legend('Mass 1','Mass 2'), axis equal;
%
% Three-dimension orbit plot (mass 1 and mass 2)
% figure, plot3(x(:,1),x(:,2),x(:,3),'b',x(:,4),x(:,5),x(:,6),'r'), grid on
% title('Three-Dimension Orbit Plot (Mass 1 and Mass 2)')
% ylabel('y'), xlabel('x'), zlabel('z')
% legend('Mass 1','Mass 2'), axis equal;
%
% Plot center-of-mass velocity
figure, plot(t,cm_vel_mag)
title('Center-Of-Mass Velocity (Speed) Versus Time')
ylabel('Velocity (km/sec)'), xlabel('Time (sec)');
%
% Plot energy
% figure, plot(t,E1,'r',t,E2,'b',t,E,'g')
% figure, plot(t,E1,'r',t,E2,'b')
figure, plot(t,E1,'r',t,E2,'b',t,E,'g')
title('Mechanical Energy of the Two-Body System')
ylabel('Energy (km^2/sec^2)'), xlabel('Time (sec)')
legend('E1','E2','E');

```

two_body_func.m

```

% two_body_func.m

function x_dot = two_body_func(t,x)
global m1 m2 G
r1 = x(1:3);
r2 = x(4:6);
v1 = x(7:9);
v2 = x(10:12);
r = norm(x(1:3)-x(4:6));

x_dot = [
                                x(7) ;
                                x(8) ;
                                x(9) ;
                                x(10) ;

```



```

x(11) ;
x(12) ;
(G*m2/r^3)*(x(4)-x(1)) ;
(G*m2/r^3)*(x(5)-x(2)) ;
(G*m2/r^3)*(x(6)-x(3)) ;
(G*m1/r^3)*(x(1)-x(4)) ;
(G*m1/r^3)*(x(2)-x(5)) ;
(G*m1/r^3)*(x(3)-x(6)) ] ;

```

Appendix C: AUTO 2000 Program Files

Contents

File Name	Description	Page No.
c.3d	Parameter definitions and values	189
3d.c	Equations	189
compute_lagrange_points_0.5.auto	Script for computing Lagrange Points	191
compute_periodic_orbits.xauto	Script for solving two-point BVP	192

The files and scripts used for simulating the Earth-moon system are provided in this appendix. Important note: These files and scripts were originally generated by Randy Paffenroth, formerly Staff Scientist, Applied and Computational Mathematics Department, California Institute of Technology and were used/modified for this particular project.

c.3d

6 0 0 0	NDIM,IPS,IRS,ILP
4 1 10 15 16	NICP,(ICP(I),I=1,NICP)
50 4 3 2 1 0 0 0	NTST,NCOL,IAD,ISP,ISW,IPLT,NBC,NINT
2000 -1. 1e10 0 1e10	NMX,RL0,RL1,A0,A1
2000 0 2 8 5 3 0	NPR,MXBF,IID,ITMX,ITNW,NWTN,JAC
1e-9 1e-9 1e-4	EPSL,EPSU,EPSS
1e-3 1e-4 1e-2 1	DS,DSMIN,DSMAX,IADS
1	NTHL,((I,THL(I)),I=1,NTHL)
10 0	
0	NTHU,((I,THU(I)),I=1,NTHU)
0	NUZR,((I,UZR(I)),I=1,NUZR)

3d.c

```

/* ----- */
/* ----- */
/*          nb :      The restricted 3-body problem          */
/* ----- */
/* ----- */
#include "auto_f2c.h"
/* ----- */
/* ----- */
int func (integer ndim, const double *u, const integer *icp,
          const double *par, integer ijac, double *f, double *dfdu, double *dfdp)
{
    double mu,p;
    double x, y, z;
    double xp, yp, zp;
    double rone, rone2, rone3;
    double rtwo, rtwo2, rtwo3;
    double Cx, Cy, Cz, Cxp, Cyp, Czp;

    x = u[0];
    y = u[1];
    z = u[2];
    xp = u[3];
    yp = u[4];
    zp = u[5];

    mu = par[1];
    p = par[2];

    rone = sqrt( (x+mu)*(x+mu) + y*y + z*z );
    rone2 = rone*rone;
    rone3 = rone2*rone;

    rtwo = sqrt( (x-1+mu)*(x-1+mu) + y*y + z*z );
    rtwo2 = rtwo*rtwo;
    rtwo3 = rtwo2*rtwo;

    Cx = x - (1-mu)*(x+mu)/rone3 - mu*(x-1+mu)/rtwo3;
    Cy = y - (1-mu)*y/rone3 - mu*y/rtwo3;
    Cz = - (1-mu)*z/rone3 - mu*z/rtwo3;

```

```

Cxp = -2*xp;
Cyp = -2*yp;
Czp = -2*zp;

f[0] = xp;
f[1] = yp;
f[2] = zp;
#ifdef NEGATIVE_U
f[3] = 2*yp - x + (1-mu)*(x+mu)/rone3 + mu*(x-1+mu)/rtwo3;
f[4] = -2*xp - y + (1-mu)*y/rone3 + mu*y/rtwo3;
f[5] = (1-mu)*z/rone3 + mu*z/rtwo3;
#else
f[3] = 2*yp + x - (1-mu)*(x+mu)/rone3 - mu*(x-1+mu)/rtwo3;
f[4] = -2*xp + y - (1-mu)*y/rone3 - mu*y/rtwo3;
f[5] = - (1-mu)*z/rone3 - mu*z/rtwo3;
#endif

#ifdef ALL_UNFOLDING
f[0] += p*Cx;
f[1] += p*Cy;
f[2] += p*Cz;
f[3] += p*Cxp;
f[4] += p*Cyp;
f[5] += p*Czp;
#else
f[3] += p*Cxp;
f[4] += p*Cyp;
f[5] += p*Czp;
#endif

return 0;
}
/* ----- */
/* ----- */
int stpnt (integer ndim, double t, double *u, double *par)
{
double mu;

mu = 0.0;
par[1] = mu;
par[2] = 0.;

u[0] = 0.14107;
u[1] = 0.99;
u[2] = 0.0;
u[3] = 0.0;
u[4] = 0.0;
u[5] = 0.0;

return 0;
}
/* ----- */
/* ----- */
int pvls (integer ndim, const double *u, double *par)
{
integer tmp;
extern double getp();
double x, y, z;

```

```

double xp, yp, zp;
double mu;
double rone, rtwo;

mu = par[1];

x = getp("BV0", 1, u);
y = getp("BV0", 2, u);
z = getp("BV0", 3, u);

xp = getp("BV0", 4, u);
yp = getp("BV0", 5, u);
zp = getp("BV0", 6, u);

rone = sqrt( (x+mu)*(x+mu) + y*y + z*z );
rtwo = sqrt( (x-1+mu)*(x-1+mu) + y*y + z*z );
par[15]=x*x+y*y+2*(1-mu)/rone+2*mu/rtwo-xp*xp-yp*yp-zp*zp;
par[16]=y;

return 0;
}
/* ----- */
/* ----- */
int bcnd () { return 0; }
/* ----- */
/* ----- */
int icnd () { return 0; }
/* ----- */
/* ----- */
int fopt() { return 0; }
/* ----- */
/* ----- */

```

compute_lagrange_points_0.5.auto

```

# This script computes the initial circle of solutions for mu=0
# as well as the bifurcating branches which give us the
# Lagrange points.

# Load 3d.c and c.3d into the AUTO CLUI
load('3d')

# Add a stopping condition so we only compute the loop once

# We tell AUTO to stop when parameter 16 is 0.991, parameter 1 is -0.1,
# or parameter 1 is 1.1. If parameter1 is 0.5 we just report
# a point.
cc('UZR',[[ -16,0.991],
            [-1,-0.1],
            [1,0.5],
            [-1,1.1]])

# We also want to compute branches for the first 3 bifurcation
# points.
cc('MXBF',-3)

# IPS=0 tells AUTO to compute a family of equilibria.

```

```

cc('IPS',0)

# Compute the circle.
run()

# Extract the 5 Lagrange points for each of the branches
# which we will use in later calculations.

# This command parses the solution file fort.8 and returns
# a Python object which contains all of the data in the
# file in an easy to use format.
data=sl()
i=0

# For every solution in the fort.8 file...
for x in data:
    # If the solution is a user defined point...
    if x["Type name"] == "UZ":
        # We look at the value of one of the components
        # to determine which Lagrange point it is.

        # The solution is a Python dictionary. One of the
        # elements of the dictionary is an array called "data"
        # which contains the values of the solution. For example,
        # x["data"][0]["t"] is the 't' value of the first point
        # of the solution. x["data"][0]["u"] is an array of which
        # contains the value of the solution at t=0.
        if x["data"][0]["u"][1] > 0.01:
            # When we determine which Lagrange point we have we save it.
            x.writeFilename("s.14")
        elif x["data"][0]["u"][1] < -0.01:
            x.writeFilename("s.15")
        elif x["data"][0]["u"][0] > 0.01:
            x.writeFilename("s.12")
        elif x["data"][0]["u"][0] < -0.01:
            x.writeFilename("s.13")
        else:
            x.writeFilename("s.11")

```

compute_periodic_family.xauto

```

# This is an example of an 'expert' AUTO CLUI script.
# This scripts takes the Lagrange points computed
# by the compute_lagrange_points_0.5.auto and
# computes the periodic orbits emanating from them.

# In expert scripts we need to explicitly import the
# AUTOclui library
from AUTOclui import *
# There isn't a AUTO CLUI command for diagnostic
# file parsing yet, but in a script such as
# this we can just as easily import the parsing
# class directly.
import parseD

# We also import a few general Python utility

```

```

# libraries.
import sys
import string
import math

# We have divided the functionality if this
# script into two functions, so that the
# same ideas may be more easily used in
# other contexts.
def compute_periodic_family(starting_point,mu,compute_bifur_flag="no",npr=20):
    # We load the 3d.c, the starting point file, and
    # c.3d into the AUTO CLUI
    load(c='3d',s=starting_point,e='3d')
    # And we parse the starting solution. This
    # is mainly to determine what label the
    # file contains.
    starting_solution=sl(starting_point)

    # We setup the calculation by setting the
    # starting solution to be the appropriate label.
    cc('IRS',starting_solution[0]["Label"])
    # And setting the problem type. In this case
    # we want to compute a family of equilibria.
    cc('IPS',1)

    # Our initial calculation it to go from 0.5
    # to the desired mu value, so we put in a
    # stopping condition for the mu value we want.
    cc('UZR',[[-1,mu]])
    # Since we are starting at mu=0.5 we want to
    # go down if the desired value is less, and
    # go up if the desired value is more.
    if mu < 0.5:
        cc('DS',-pr('DS'))
    run()

    # We save this solution
    sv('hopf_bifurcation')
    # And get a parsed version as well.
    hopf_bifurcation = sl('hopf_bifurcation')

    # This will eventually become an AUTO2000 internal
    # NOTE: the interface to the parseD object is
    # under development and may change significantly
    # in the final version
    parseD_object=parseD.parseD('fort.9')

    # We print out the eigenvalues of the Jacobian.
    print parseD_object[-1]["Eigenvalues"]

    # In this loop we look for all eigenvalues
    # with "zero" (i.e. sufficiently small) real part.
    # We begin by defining an array in which the periods
    # of the Hopf bifurcations will be stored.
    periods = []
    # The parseD_object is basically a Python list,
    # so we use standard Python syntax to iterate
    # over it.

```

```

for eigenvalue in parseD_object[-1]["Eigenvalues"]:
    if math.fabs(eigenvalue[0]) < 1e-8:
        # If the real part is sufficiently small
        # we get the imaginary part
        imag = math.fabs(eigenvalue[1])
        print "imaginary part: ", imag
        print "period      : ", 2*math.pi/imag
        # and compute the period. If it is not in our
        # list of periods (i.e. it is not a complex conjugate
        # to one we have already computed) we add it.
        if 2*math.pi/imag not in periods:
            periods.append(2*math.pi/imag)

# Now we have an array which contains the initial periods of all of the
# periodic orbits emanating from the Hopf bifurcation.
# We iterate over them and calculate each family.
for period in periods:
    # Since we have a parsed version of the initial solution
    # it is easy to modify it to include the period
    # we want. In AUTO, the 11th parameter is always
    # the period.
    hopf_bifurcation[-1]["p"][10] = period
    # Now, when this point was computed we had Hopf
    # bifurcation detection turned off (since all
    # points were Hopf bifurcations). So, we manually
    # mark the point as a Hopf bifurcation.
    hopf_bifurcation[-1]["Type number"] = 3
    hopf_bifurcation[-1]["Type name"] = "HB"

# We load in the above modified solution and the constants file.
# NOTE: There are several ways to set the solution file.
# It can be a filename, an open file descriptor, or a
# Python object of the parseS class.
load(c='3d', s=hopf_bifurcation)
# We set the problem type, in this case we want to
# compute a family of periodic orbits.
cc('IPS', 2)
# We turn off torus bifurcation detection, since there are
# lots of torus bifurcations.
cc('ISP', 3)
# We want additional solutions to be saved, so we set NPR to
# a smaller value.
cc('NPR', npr)
# We want the period, the y value at t=0, and the Jacobi constant to
# be printed out, so we add the appropriate parameters,
cc('ICP', [2, 10, 15, 16])
# We set the IRS to be the label of the desired starting point.
cc('IRS', hopf_bifurcation[-1]["Label"])

# And we run the calculation.
run()
# Finally, we save the solution.
sv('%s_mu_%f_period_%f'%(starting_point, mu, period))

# Now, if there were any bifurcation points detected we want
# to compute the branches emanating from them as well.
# Since this is a very common task, we have put that functionality
# into a subroutine.

```



```

    if compute_bifur_flag == "yes":
        compute_bifur('3d', '%s_mu_%f_period_%f'%(starting_point,mu,period),npr)

# This subroutine takes a problem name and a solution file, and for
# every bifurcation point in the solution file it attempts to
# compute a bifurcating branch.
def compute_bifur(problem,solution_file,npr=20):
    # Load the problem file and constants file
    ld(problem)
    # and the solution file.
    ld(s=solution_file)
    # Set the problem type
    cc('IPS',2)
    # Turn off torus bifurcation detection
    cc('ISP',3)
    # Increase the amount of data output
    cc('NPR',npr)
    # We want the period, the y value at t=0, and the Jacobi constant to
    # be printed out, we add the appropriate parameters,
    cc('ICP',[2,10,15,16])

    # We parse the solution file to get the labels of the
    # solutions.
    data = sl(solution_file)

    # The solution object is basically a Python list,
    # so we use standard Python syntax to iterate
    # over it.
    for solution in data:
        # For every solution we test to see if it is a bifurcation point
        if solution["Type name"] == "BP":
            # And if it is we use it as a starting point for a new calculation
            ch("IRS", solution["Label"])
            # This is the syntax for telling AUTO to switch branches at the bifurcation
            ch("ISW", -1)
            # Compute forward
            run()
            # Save the branch
            sv(solution_file+"_"+"`"+solution["Label"]`)
            # Compute backward by making the initial step-size negative
            ch("DS",-pr("DS"))
            run()
            # Save the branch
            sv(solution_file+"_"+"`"+solution["Label"]`)

# This is the Python syntax for making a script runnable
if __name__ == '__main__':
    # We want to have the option of computing the bifurcating
    # branches or not, so we use the Python getopt
    # routines to process command line options.
    import getopt
    # This line process the command line options and
    # looks for a -b option
    opts_list,args=getopt.getopt(sys.argv[1:], "bn:")
    # We take the list of options generated by
    # getopt command and turn it into a dictionary.
    # This is not strictly necessary, but it makes

```

```
# it easier to use.
opts={}
for x in opts_list:
    opts[x[0]]=x[1]

# If you use the -b option then we want to compute the bifurcating
# branches.
if opts.has_key("-b"):
    compute_bifur_flag="yes"
else:
    compute_bifur_flag="no"

npr = 20
if opts.has_key("-n"):
    npr = string.atoi(opts["-n"])

# The first argument is the name of the file in
# which you find the starting point
starting_point = args[0]
# The second argument is the desired mu value.
mu = string.atof(args[1])
compute_periodic_family(starting_point,mu,compute_bifur_flag,npr)
```

Appendix D: Catalogue of Periodic Orbits Around the Earth-Moon L4 Point

AME 794b Dissertation

A CATALOGUE OF PERIODIC ORBITS AROUND THE EARTH-MOON L4 LIBRATION POINT

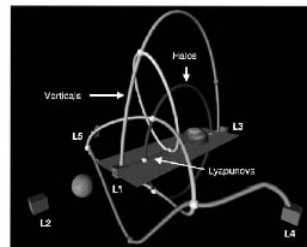
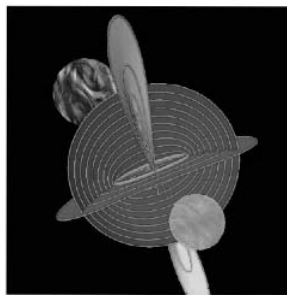
Ralph R. Basilio
01 April 2006

01 Apr 06

AME794b - Dissertation

Ralph R. Basilio - 1

Periodic Orbit Types



The periodic orbits: Lyapunov (planar), halo, and vertical orbits

[Figures are from Paffenroth, R., "Continuation of Periodic Orbits Around Lagrange Points and AUTO2000: The Three-Body Problem and Space Mission Design", Caltech presentation charts, 19 Feb 02.]

01 Apr 06

AME794b - Dissertation

Ralph R. Basilio - 2

Auto 2000 Periodic Orbit Files for Earth-Moon L4

```

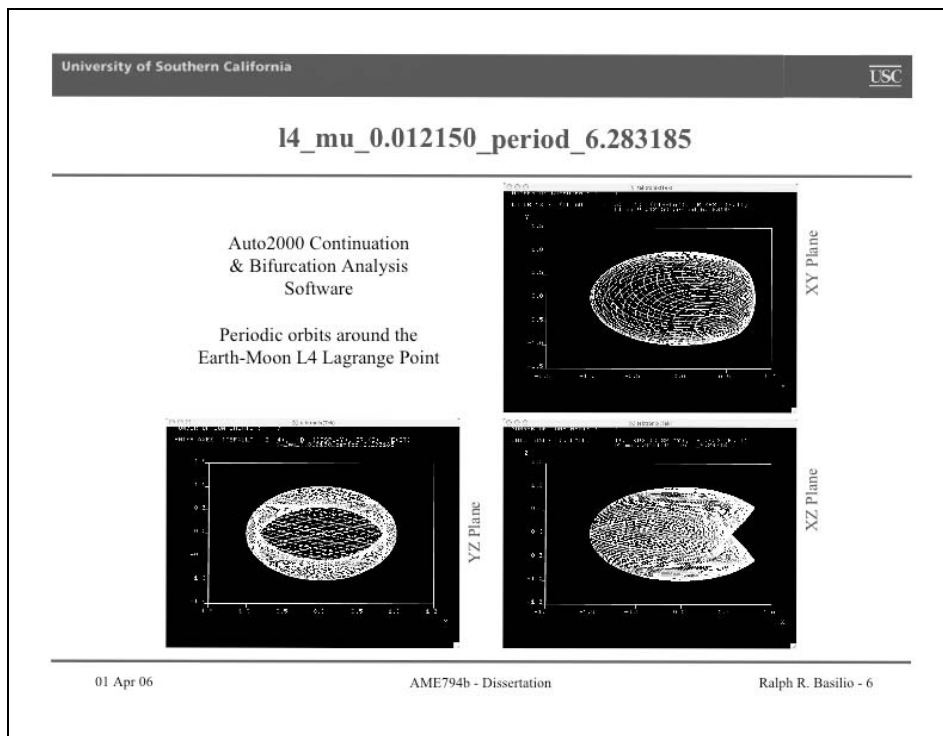
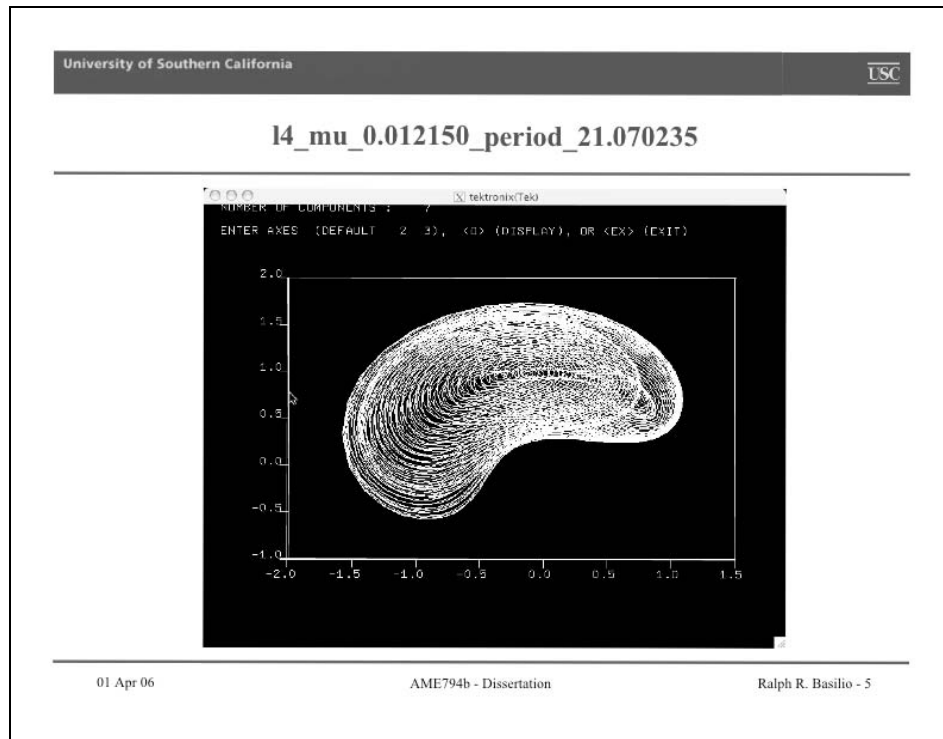
auto fort.17 fort.18
node21% cd auto/R9LPH
node21% ls b.14*
b.14_mu_0.012150_period_21.070352 b.14_mu_0.012150_period_6.283185 - 53
b.14_mu_0.012150_period_21.070352" b.14_mu_0.012150_period_6.283185 - 64
b.14_mu_0.012150_period_6.283185 b.14_mu_0.012150_period_6.283185 - 76
b.14_mu_0.012150_period_6.283185 + 101 b.14_mu_0.012150_period_6.283185 - 83
b.14_mu_0.012150_period_6.283185 + 112 b.14_mu_0.012150_period_6.283185 - 89
b.14_mu_0.012150_period_6.283185 + 16 b.14_mu_0.012150_period_6.283185 -
b.14_mu_0.012150_period_6.283185 + 27 b.14_mu_0.012150_period_6.582675
b.14_mu_0.012150_period_6.283185 + 39 b.14_mu_0.012150_period_6.582675 + 53
b.14_mu_0.012150_period_6.283185 + 46 b.14_mu_0.012150_period_6.582675 + 55
b.14_mu_0.012150_period_6.283185 + 53 b.14_mu_0.012150_period_6.582675 + 61
b.14_mu_0.012150_period_6.283185 + 64 b.14_mu_0.012150_period_6.582675 + 70
b.14_mu_0.012150_period_6.283185 + 76 b.14_mu_0.012150_period_6.582675 + 79
b.14_mu_0.012150_period_6.283185 + 83 b.14_mu_0.012150_period_6.582675 + 84
b.14_mu_0.012150_period_6.283185 + 89 b.14_mu_0.012150_period_6.582675 - 53
b.14_mu_0.012150_period_6.283185 - 101 b.14_mu_0.012150_period_6.582675 - 55
b.14_mu_0.012150_period_6.283185 - 112 b.14_mu_0.012150_period_6.582675 - 61
b.14_mu_0.012150_period_6.283185 - 16 b.14_mu_0.012150_period_6.582675 - 70
b.14_mu_0.012150_period_6.283185 - 27 b.14_mu_0.012150_period_6.582675 - 79
b.14_mu_0.012150_period_6.283185 - 39 b.14_mu_0.012150_period_6.582675 - 84
b.14_mu_0.012150_period_6.283185 - 46 b.14_mu_0.012150_period_6.582675"
node21%

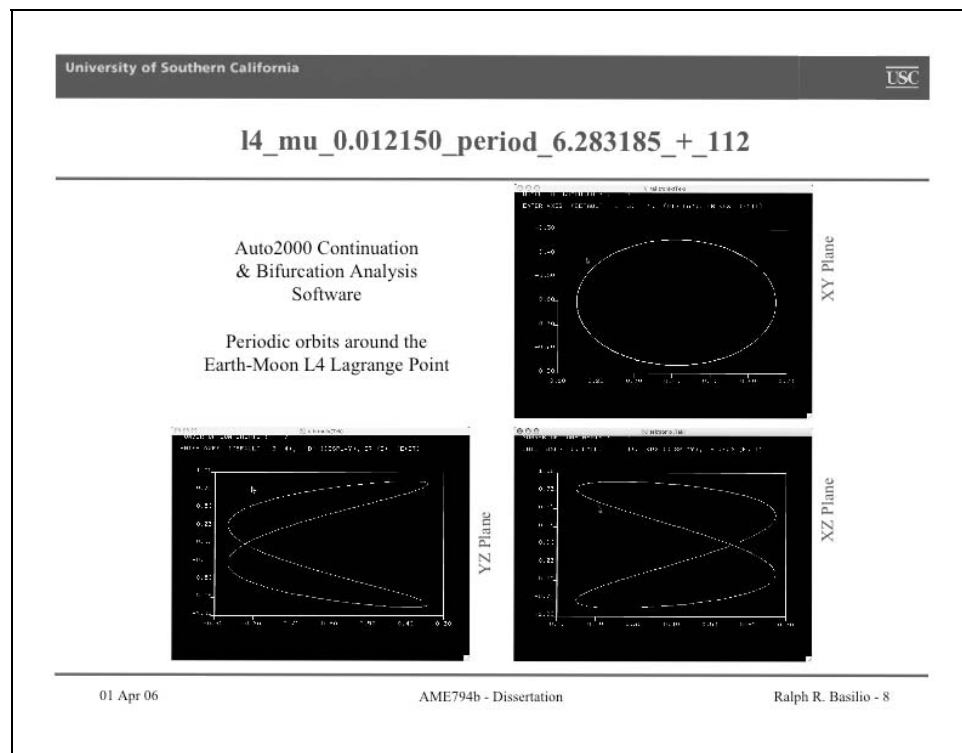
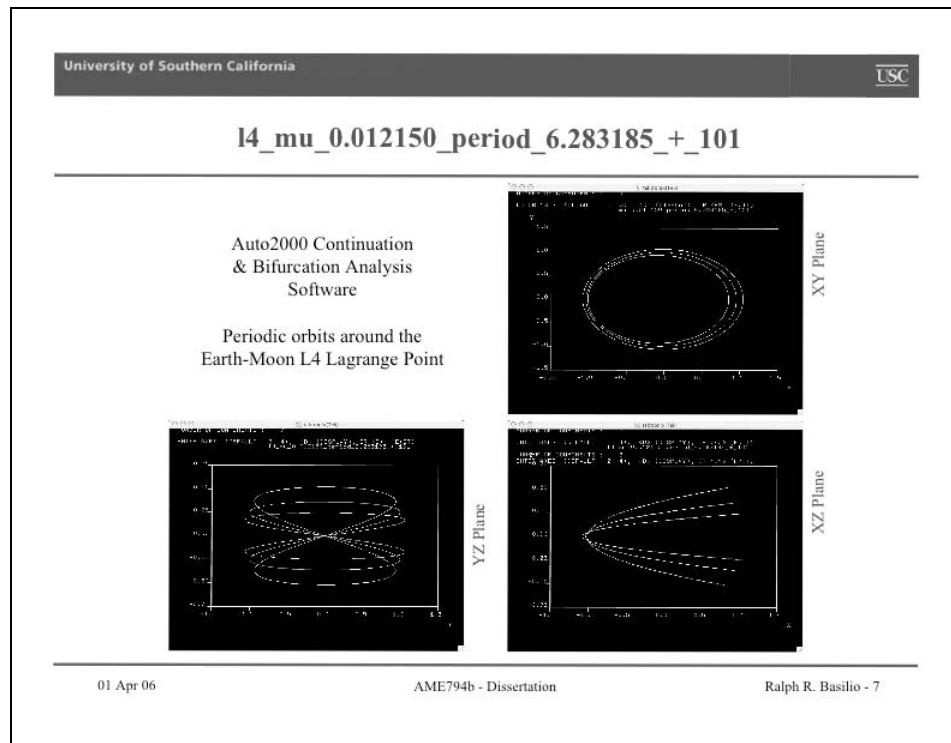
```

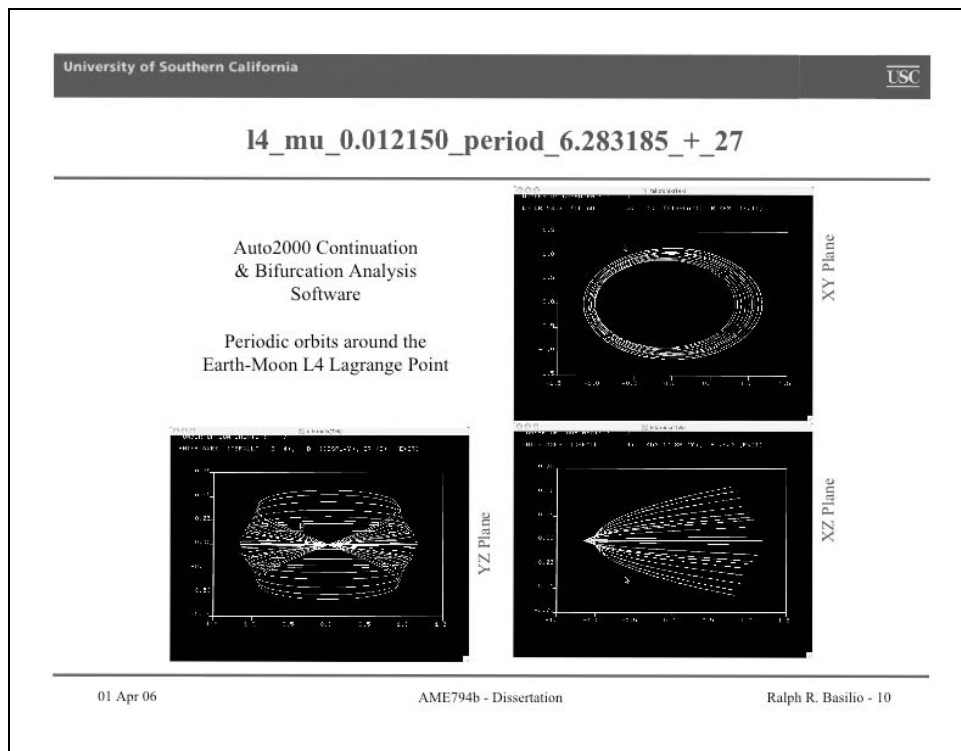
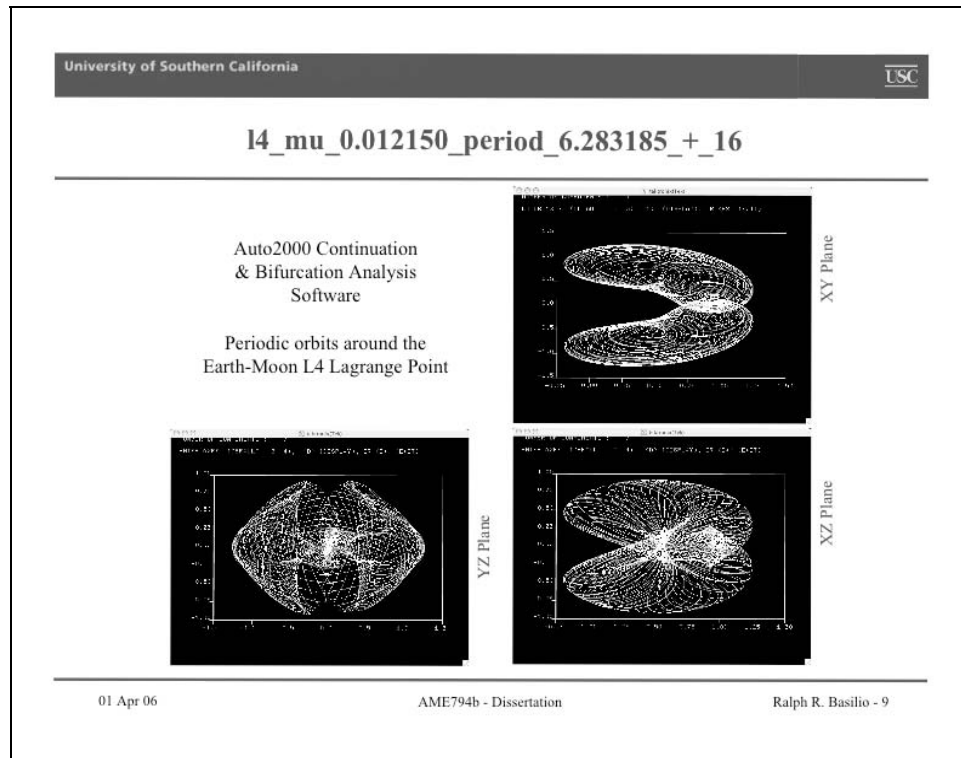
Catalogue of Periodic Orbits Around Earth-Moon L4

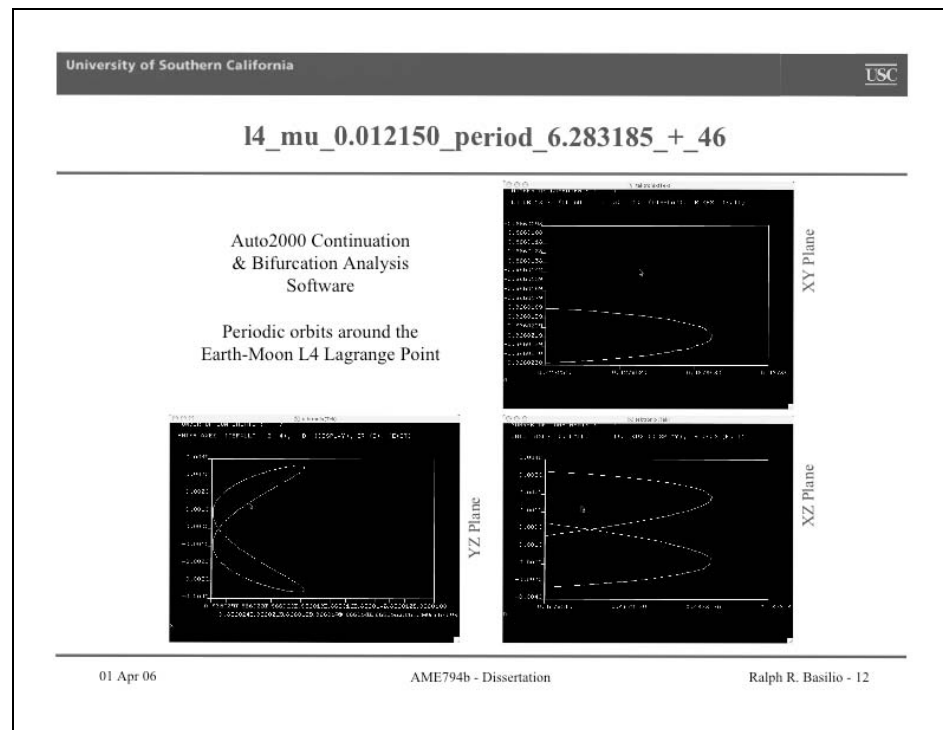
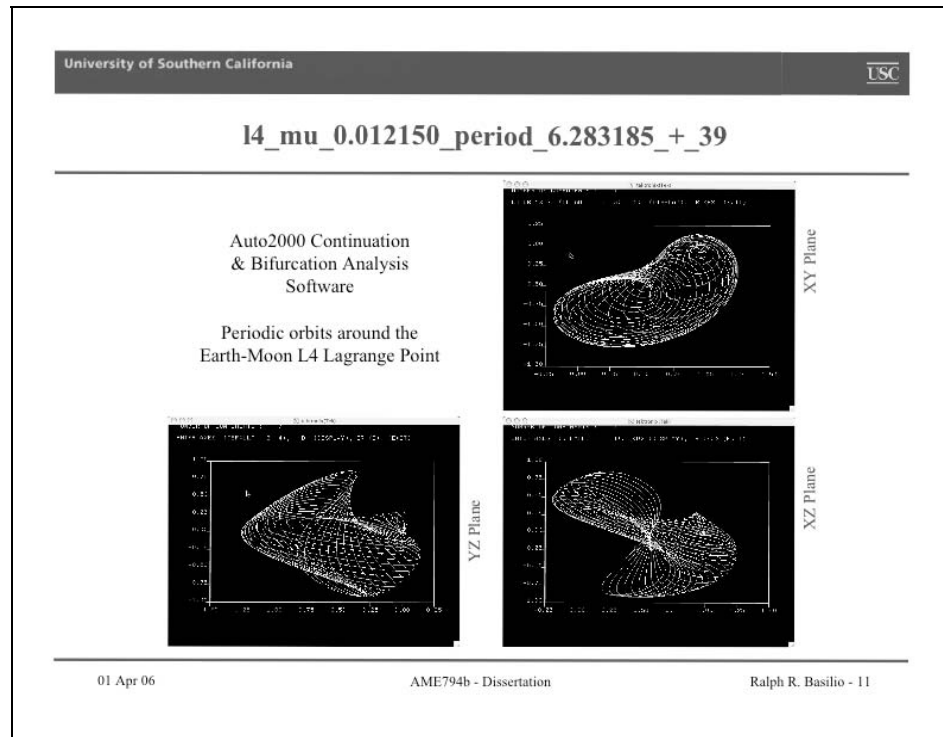
PERIODIC ORBIT FAMILY	PAGE NO.
Base Period = 21.070352	
Basic	5
Base Period = 6.283135	
Basic	6
+101	7
+112	8
+16	9
+27	10
+39	11
+46	12
+53	13
+64	14
+76	15
-101	16
-27	17

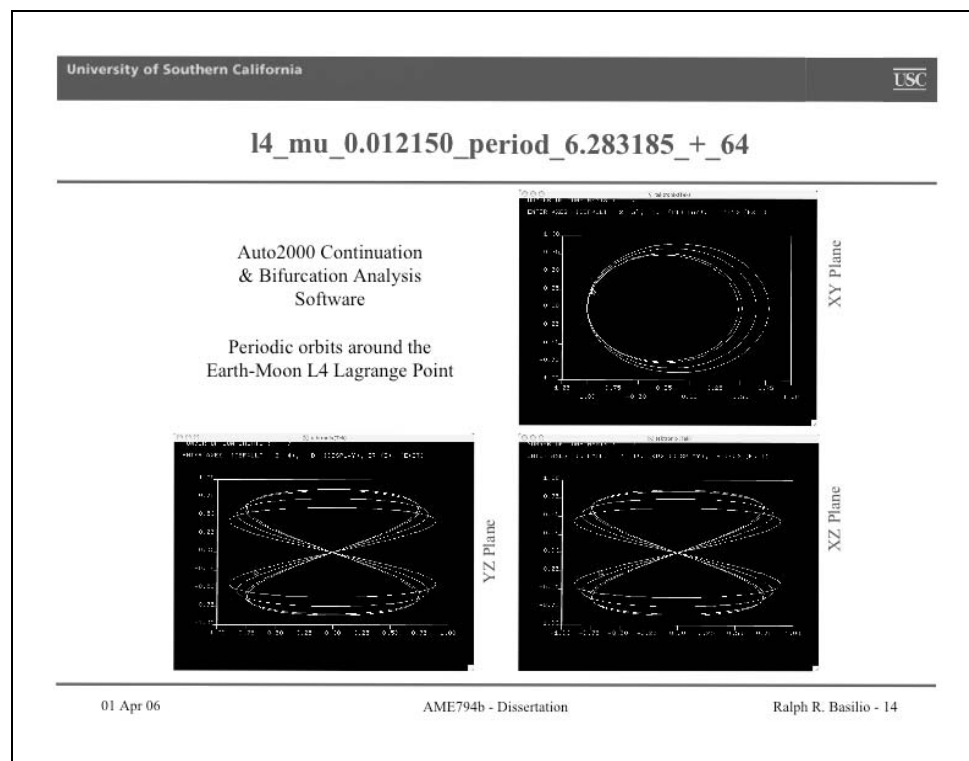
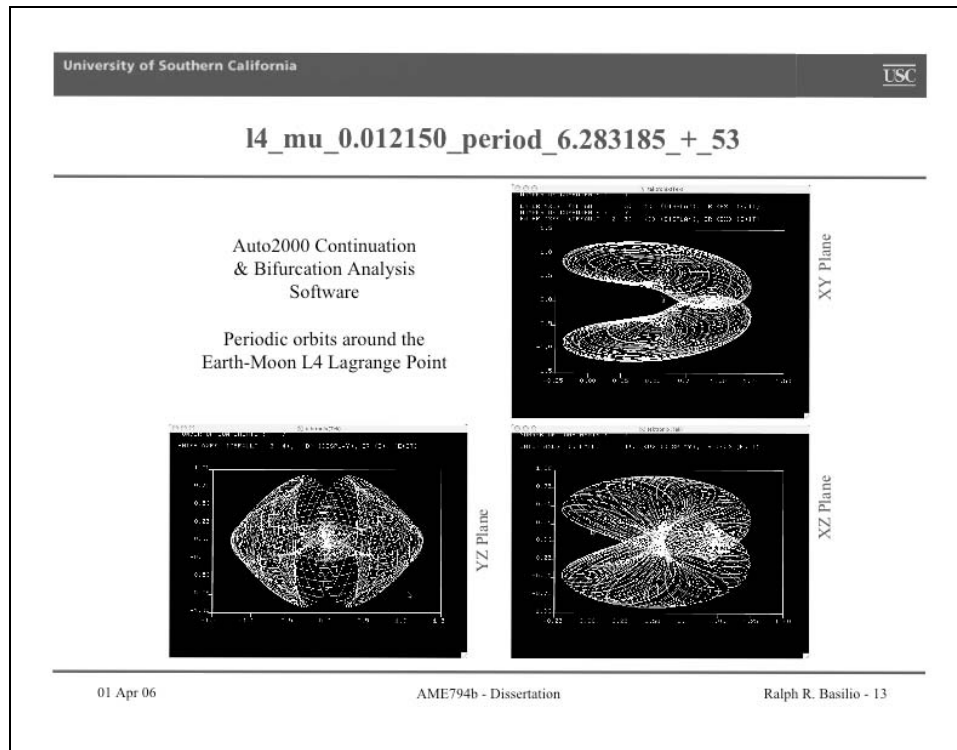
PERIODIC ORBIT FAMILY	PAGE NO.
-64	18
-89	19
Others (e.g. +83)	20
Base Period = 6.582675	
Basic	21
+53	22
+55	23
+61	24
+70	25
+79	26
+84	27
Others (e.g. -53)	28

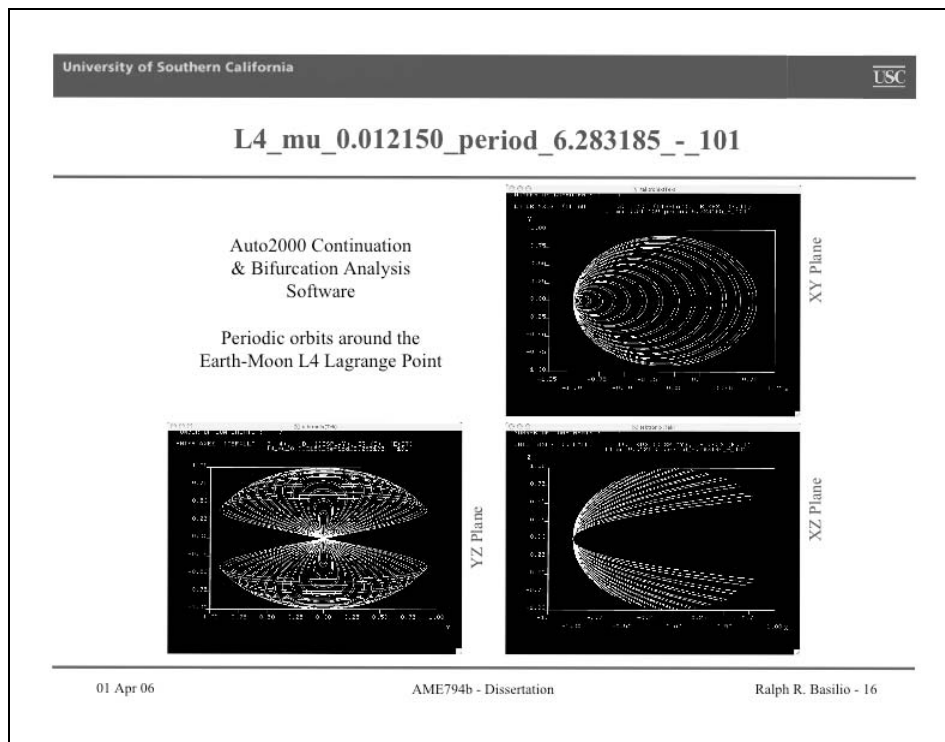
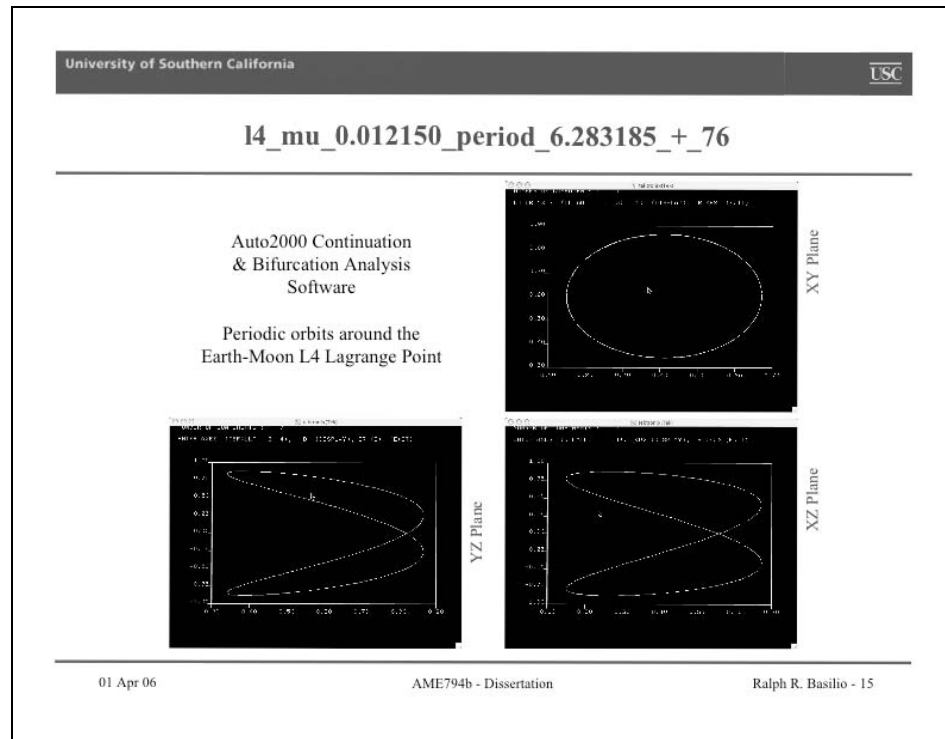


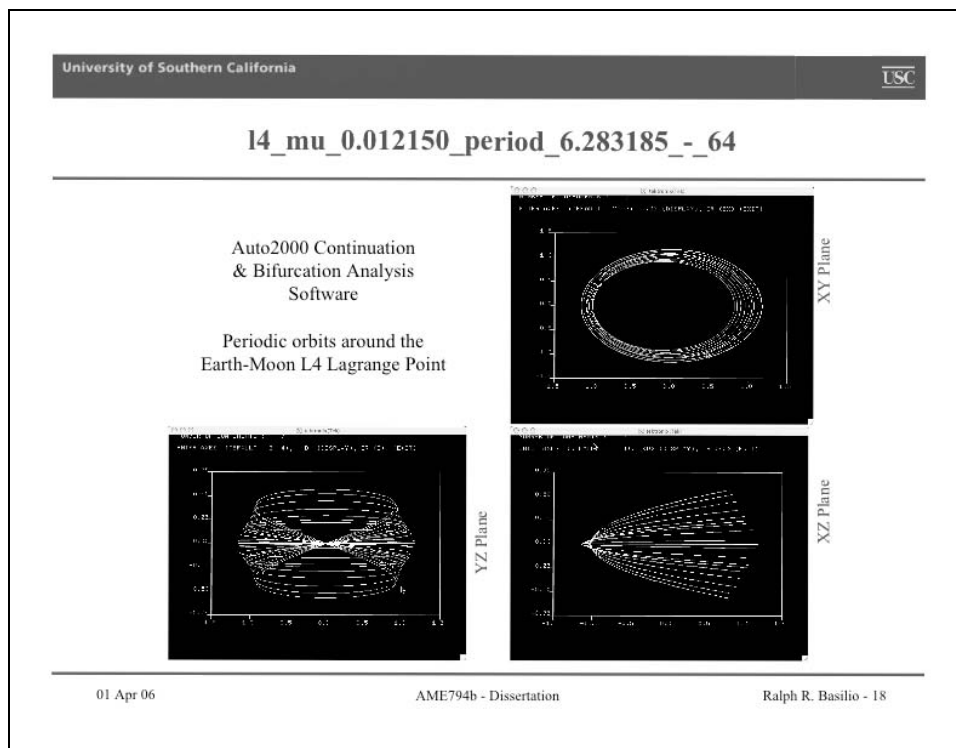
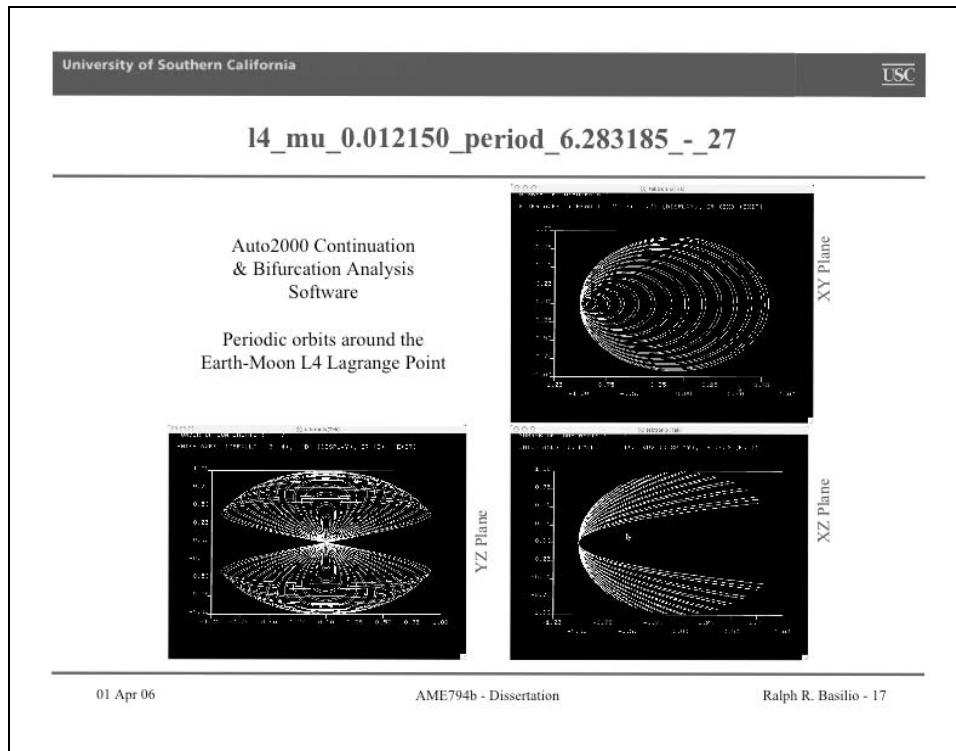


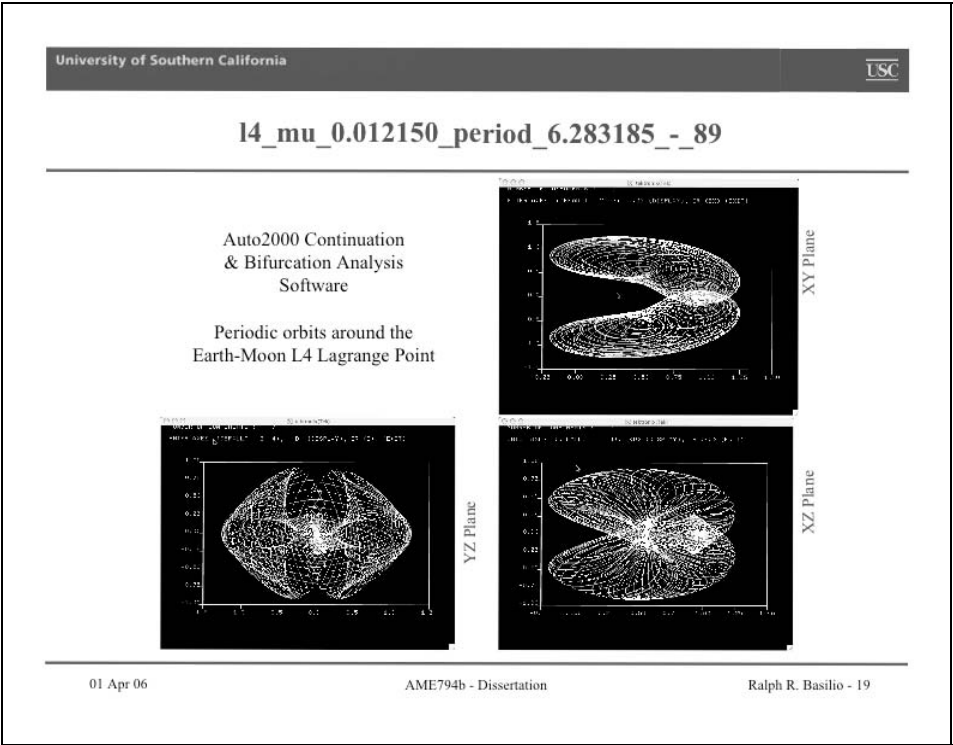












University of Southern California

USC

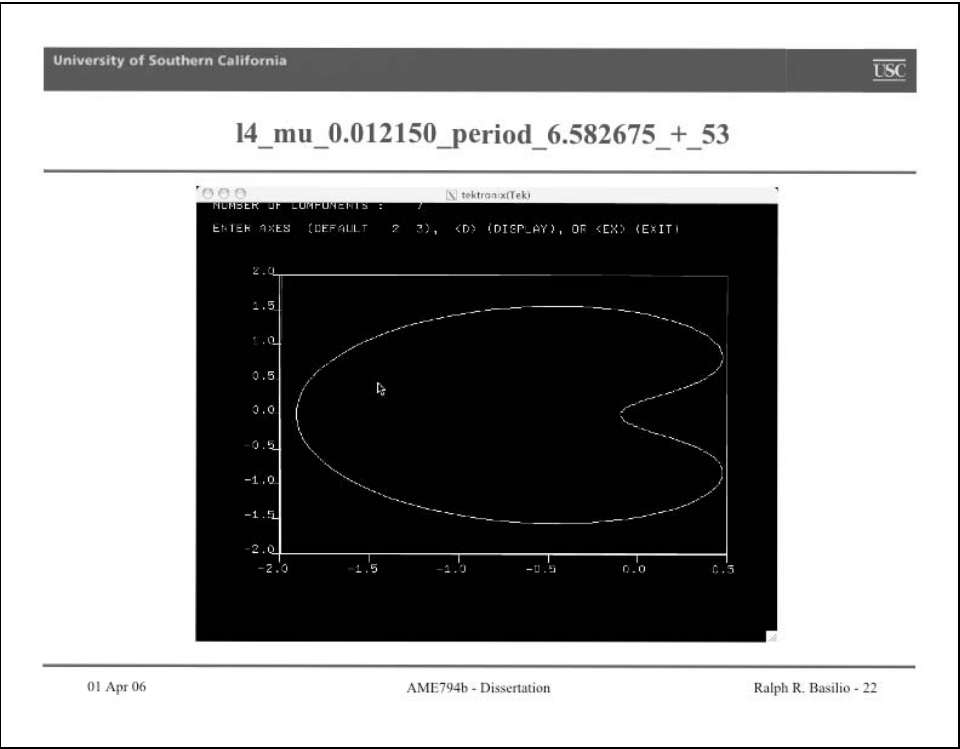
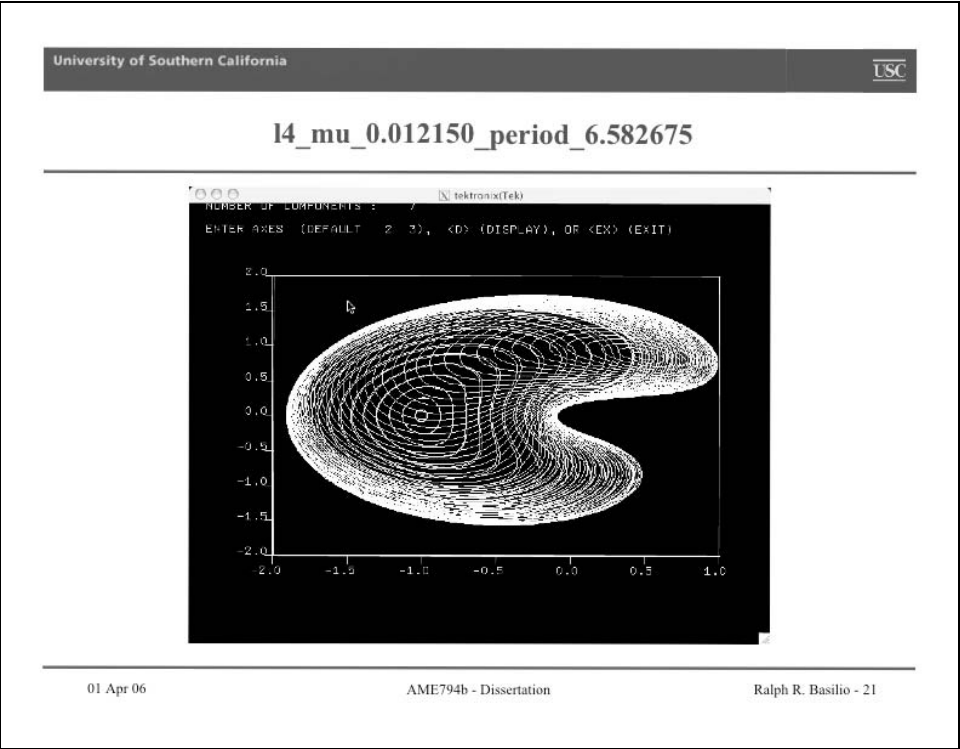
Other 14_mu_0.012150_period_6.283185 Orbits

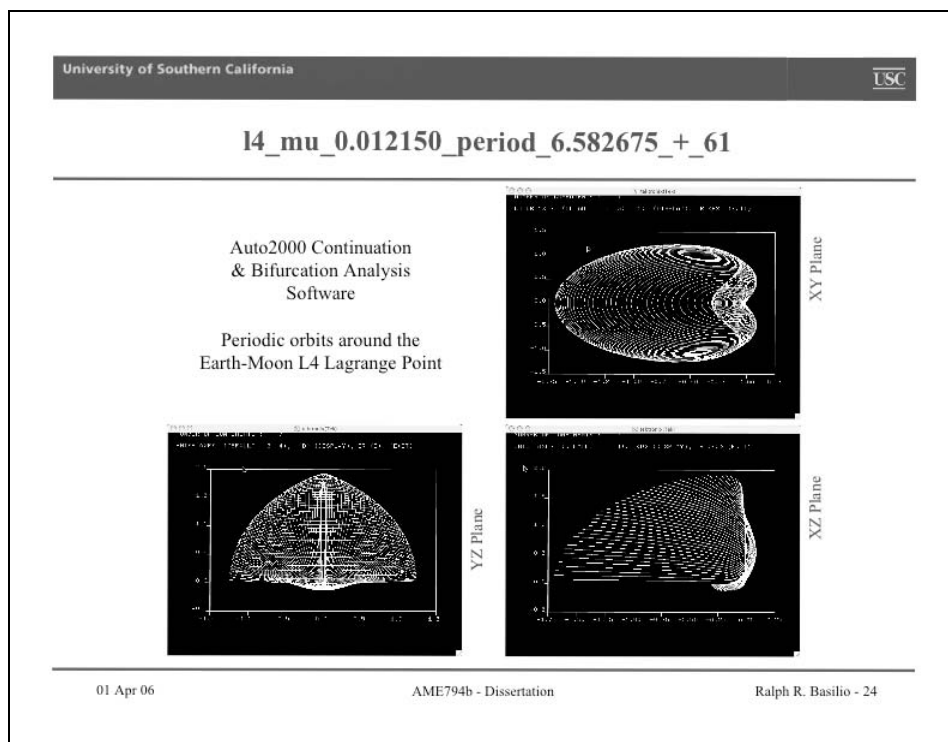
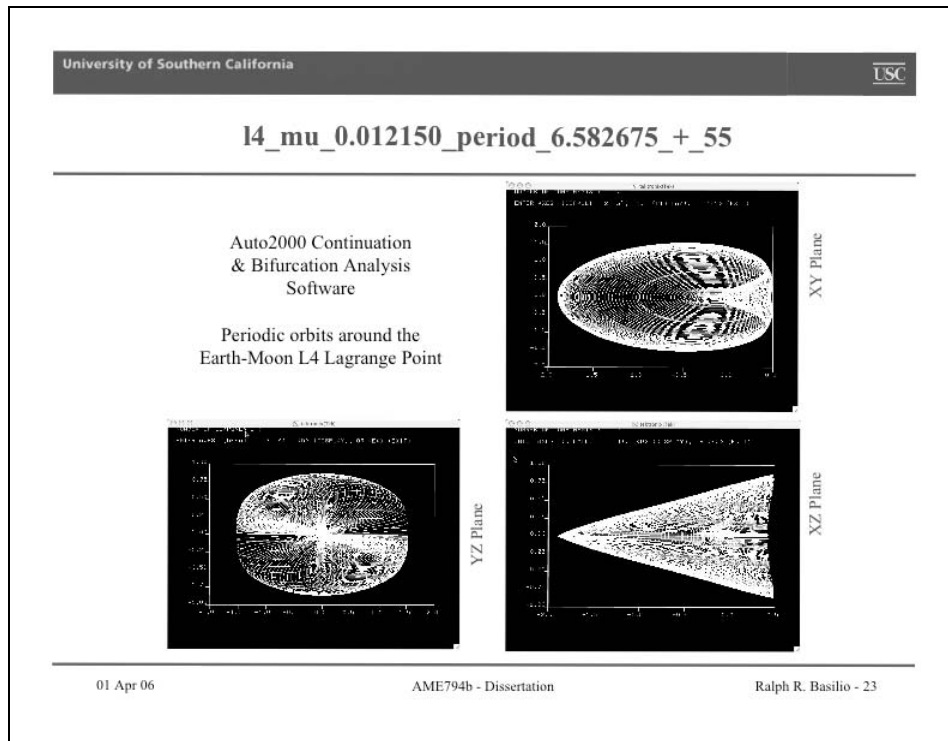
PERIODIC ORBIT FAMILY	COMMENT
+83	Undefined
+89	Undefined
-112	Symmetry with +112
-16	Symmetry with +16
-39	Symmetry with +39
-46	Symmetry with +46
-53	Symmetry with +53

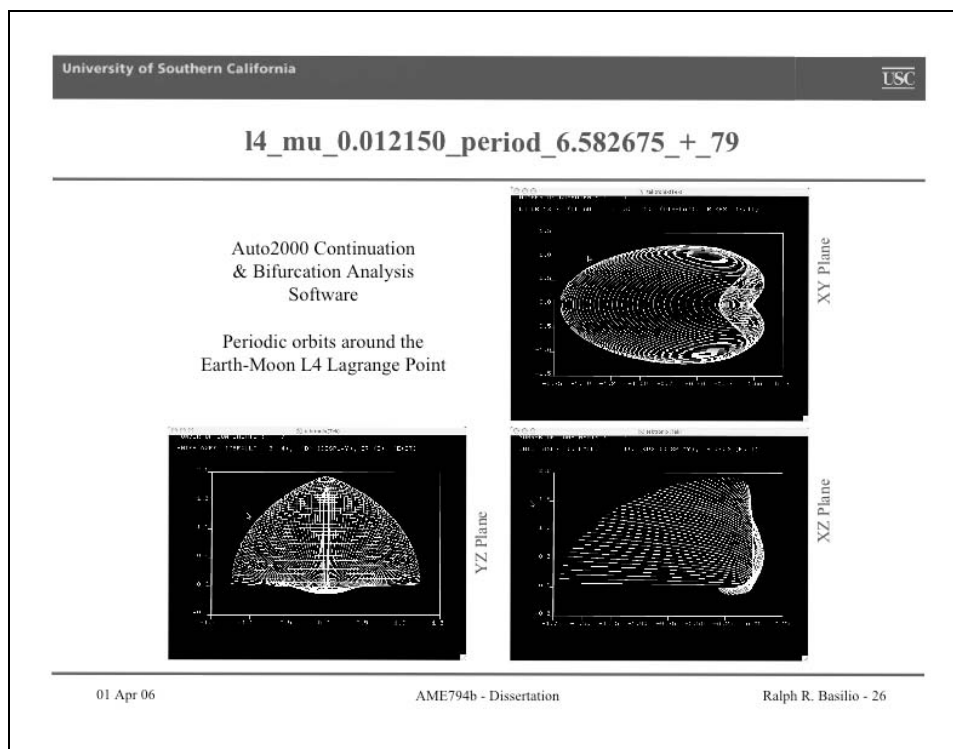
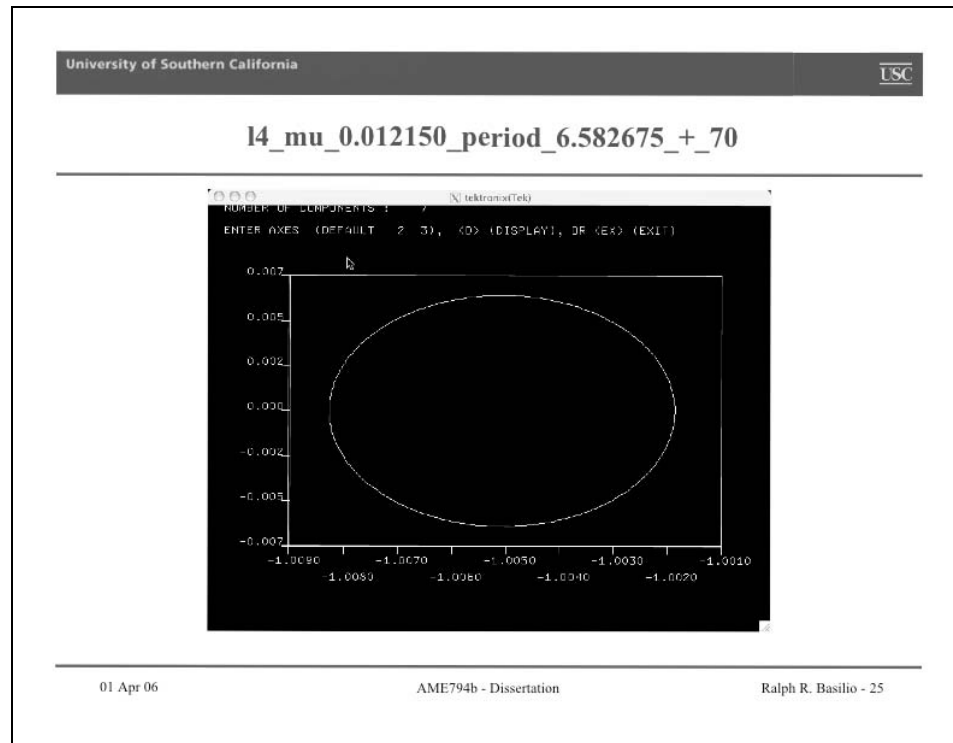
01 Apr 06

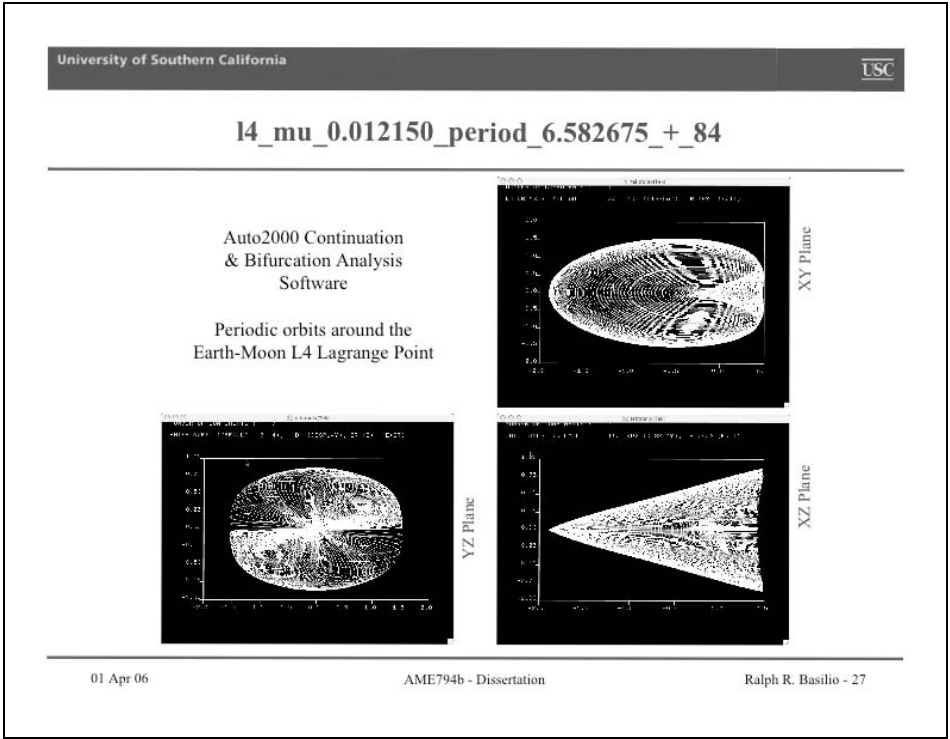
AME794b - Dissertation

Ralph R. Basilio - 20









Appendix E: Floquet Theory and the Monodromy Matrix

Although slightly more complex, the stability of periodic orbits can be found in a manner similar to that used for equilibrium points where constant coefficient linear systems and the eigenvalues of the Jacobian matrix are used to deduce information on stability. The method described here is called Floquet theory in honor of Gaston Floquet, a famous nineteenth and twentieth century French mathematician. Consider the following equation

$$\dot{x} = A(t)x \quad \text{E.1}$$

where $-\infty < t < +\infty$ and $A(t)$ is a continuous T periodic $n \times n$ matrix. If $\Phi(t)$ is a fundamental matrix then $\Phi(t+T)$ is also a fundamental matrix defined as

$$\Phi(t+T) = \Phi(t)C \quad \text{E.2}$$

where $C = \Phi^{-1}(0)\Phi(T)$. Therefore, equation E.2 can also be written as

$$\Phi(t+T) = \Phi(t)\Phi^{-1}(0)\Phi(T) \quad \text{E.3}$$

Furthermore, equation E.3 can be written as the product of two matrices,

$$\Phi(T) = P(T)e^{BT} \quad \text{E.4}$$

Here $P(t)$ is T periodic and B is an $n \times n$ matrix. The proof that $\Phi(t+T) = \Phi(t)$ is provided by Verhulst [E2] and is given below

Proof

Set $\tau = t + T$. Then

$$\dot{x} = A(t)x = A(\tau - T)x = A(\tau)x \quad \text{E.5}$$

$\Phi(\tau)$ along with $\Phi(t)$ and $\Phi(t+T)$ are all fundamental matrices that are linearly dependent.

Therefore, there exists a nonsingular matrix C such that

$$\Phi(t+T) = \Phi(t)C \quad \text{E.6}$$

This is just equation E.2. There is also a constant matrix B such that $C = e^{BT}$. Now from equation E.4,

$$\Phi(t)e^{-Bt} = P(t) \quad \text{E.7}$$

Then

$$P(t+T) = \Phi(t+T)e^{-B(t+T)}$$

$$P(t+T) = \Phi(t)Ce^{-B(t+T)}$$

$$P(t+T) = \Phi(t)e^{-Bt}$$

$$P(t+T) = P(t)$$

The non-singular matrix, C , is sometimes referred to as the monodromy matrix of equation E.1. The monodromy matrix is simply the linearization of the period T mapping evaluated at the fixed point. The eigenvalues, λ , of the matrix, C , are called the characteristic multipliers or sometimes the Floquet multipliers. Sanchez [E1] states that the eigenvalues are regarded as characteristic multipliers since they are independent of the choice of the fundamental matrix. Suppose $\Omega(t)$ is another fundamental matrix. Then

$$\Omega(t) = \Phi(t)A \quad \text{E.8}$$

where A is a constant matrix. Given equation E.2, the following is a result

$$\Omega(t+T) = \Phi(t+T)A = \Phi(t)CA = \Omega(t)A^{-1}CA \quad \text{E.9}$$

He states that from linear algebra, the characteristic roots, i.e. eigenvalues, of C and $A^{-1}CA$ are identical. Because the system of differential equations is energy conserving, the period T mapping is preserving. Therefore, the determinant of the matrix, C , is unimodular, i.e. equal to 1, and the product of the eigenvalues must also be equal to 1. In a three-dimensional physical space, there are a total of six eigenvalues. Two are equal to 1, two are complex numbers with both real and imaginary components, and two others (also complex numbers) are reciprocals of one another. If the Floquet multipliers, i.e. the modulus of each complex number, are all less than or equal to unity the system is said to be stable. If the Floquet multipliers, i.e. the modulus of each complex number, are each greater than unity the system is said to be unstable. If one of the Floquet multipliers, i.e. the modulus of the complex number, is less than unity, but the other (or others) are greater than unity the system possesses a saddle node, but still unstable. Important note: Since the stability of a periodic orbit is that same along the entire trajectory, the eigenvalues of all monodromy matrices are also the same.

Example E.1.

Determine the stability of a system of ordinary differential equations with the following eigenvalues:

Multiplier 0:	1.0000000+0.000000i
Multiplier 1:	-0.6874934-0.726191i
Multiplier 2:	-0.6874934+0.726191i
Multiplier 3:	0.7072087-0.707005i
Multiplier 4:	0.7072087+0.707005i
Multiplier 5:	1.0000000+0.000000i

Solution: Multiplier 0 and 5 are the two eigenvalues equal to 1. Multipliers 1 and 2 are two complex numbers with both real and imaginary components. The product of the two numbers is equal to 1, i.e. for $a + bi$ and $c + di$, the product is $(ac - bd) + (ad + bc)i$. Multipliers 3 and 4 are reciprocals of one another, i.e. for a complex number, $a + bi$, the reciprocal is $\frac{a}{(a^2 + b^2)} - \frac{b}{(a^2 + b^2)}i$. Since none of the multipliers has a modulus greater than unity, the system is stable.

Example E.2.

Determine the stability of a system of ordinary differential equations with the following eigenvalues:

Multiplier 0:	1.0000000+0.000000i
Multiplier 1:	0.2062284-0.978504i
Multiplier 2:	0.2062284-0.978504i
Multiplier 3:	1.0000000+0.000000i
Multiplier 4:	1.0000134+0.000000i
Multiplier 5:	0.9998627+0.000000i

Solution: Multiplier 0 and 3 are the two eigenvalues equal to 1. Multipliers 1 and 2 are two complex numbers with both real and imaginary components. The product of the two numbers is equal to 1, i.e. for $a + bi$ and $c + di$, the product is $(ac - bd) + (ad + bc)i$. Multipliers 4 and 5 are reciprocals of one another, i.e. for a complex number, $a + bi$, the reciprocal is $\frac{a}{(a^2 + b^2)} - \frac{b}{(a^2 + b^2)}i$. Since the modulus of Multiplier 4 is greater than unity, the system is unstable.

The complex number, ω_i , is called the characteristic exponent or sometimes the Floquet exponent if the relationship below is true.

$$\lambda = e^{\omega T} \quad \text{E.10}$$

Finally the real components of the Floquet exponents, $\text{Re}(\omega_i)$, are called the Lyapunov exponents.

References:

- [E1] Sanchez, David A., *Ordinary Differential Equations and Stability Theory: An Introduction*, Dover Publications, Inc, New York, 1979.
- [E2] Verhulst, Ferdinand, *Nonlinear Differential Equations and Dynamical Systems*, Second Edition, Springer-Verlag, Berlin, Heidelberg, New York, 2000.

Appendix F: 2005 SIAM Dynamical Systems Conference Presentation Charts

The charts provided in this appendix were presented at the SIAM (Society for Industrial and Applied Mathematics) Conference on Application of Dynamical Systems, 22-26 May 2005, Snowbird, Utah.

SIAM Conference on Applications of Dynamical Systems
Snowbird, Utah, 22-26 May 2005

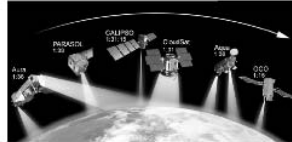
VIRTUAL RIGID BODIES IN THE CIRCULAR,
RESTRICTED THREE-BODY PROBLEM: DYNAMICALLY
NATURAL SPACECRAFT FORMATIONS

Prepared By
Ralph R. Basilio and Paul K. Newton

Presented By
Ralph R. Basilio

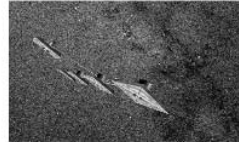
- Purpose of study
- Circular, restricted three-body problem
- Periodic orbits
- The challenge of phase-locking

- Formation flying involves operating multiple spacecraft in a pre-determined geometrical configuration that yields both individual and system benefits (e.g. creating a large diameter telescope or interferometer in space)



A-Train Constellation

<http://csc.gallaudet.edu/soarhigh/A-TrainExplain.html>



Terrestrial Planet Finder (TPF-I)

http://planetquest.jpl.nasa.gov/TPF/tpf_architectures.cfm

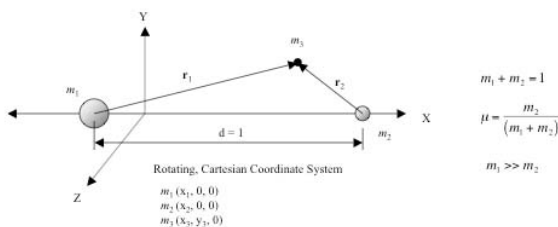
- Environmental forces in near-Earth orbit generally prohibit precision formation flying. Missions are primarily limited to "loose" formation flying.
- Even away from the Earth, spacecraft control is generally required to maintain a precision formation and can be complex and/or operationally costly
- Let's look at a different approach to spacecraft formation flying
 - Loiter > organize > observe
 - Enables scientific missions where there is the flexibility to define or change mission objectives during operations

22 May 2005

CP9 Sun PM - 2

RRB - 3

- A special case of the general three-body problem: determine the motion of a third object, m_3 (of negligible mass), under the gravitational influence of two primaries, m_1 and m_2 , in circular orbits about the system barycenter (origin)



- There are a number of ways to tackle this problem, but we use a simple Newtonian approach. The general expression for the force acting on m_3 is

$$\mathbf{F}_3 = \mathbf{F}_1 - \mathbf{F}_2 - m_3(2\mathbf{w} \times \dot{\mathbf{r}}) - m_3[\mathbf{w} \times (\mathbf{w} \times \mathbf{r})]$$

- Accounting for coriolis and centrifugal forces allows us to state the m_3 equations of motion in the rotating, Cartesian coordinate system

22 May 2005

CP9 Sun PM - 2

RRB - 4

- The m_3 second-order ODEs (Ordinary Differential Equations) are:

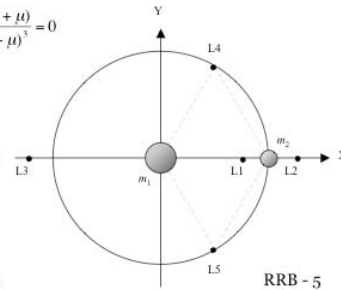
$$(1) \quad \ddot{x} - 2\dot{y} - x = -\frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x-1+\mu)}{r_2^3} \quad (2) \quad \ddot{y} + 2\dot{x} - y = \frac{-(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3} \quad (3) \quad \ddot{z} = -\frac{(1-\mu)z}{r_1^3} - \frac{\mu z}{r_2^3}$$

[We use the proven Runge-Kutta technique for numerically-integrating the ODEs]

- In the rotating coordinate frame, there are several fixed-points where the velocity and acceleration terms are zero, i.e. $\dot{x} = \dot{y} = \dot{z} = 0$ and $\ddot{x} = \ddot{y} = \ddot{z} = 0$
- From eq. 1 and 2 above, $r_1 = r_2 = 1$, forming two legs of an equilateral triangle
- Noticing that $y=0$ is a solution to eq. 2, and substituting for r_1 and r_2 , eq. 1 becomes

$$f(x) = x - \frac{(1-\mu)(x+\mu)}{(x+\mu)^3} - \frac{\mu(x-1+\mu)}{(x-1+\mu)^3} = 0$$

- The three real roots of the quintic equation above must be solved for numerically
- The sketch on the right shows the approximate locations of all five fixed-points
- Linear stability theory can be used to show that L_1 , L_2 , and L_3 are unstable and L_4 & L_5 are stable fixed-points, respectively



22 May 2005

CP9 Sun PM - 2

RRB - 5

- To expedite the process of finding periodic orbits about fixed-points, the continuation & bifurcation analysis tool, AUTO2000, can be employed
- AUTO2000 allows one to solve simple algebraic problems and ODEs
- In general AUTO2000 solves equations of the form: $F(x) = 0, F: \mathbf{R}^{n+1} \rightarrow \mathbf{R}^n, n \in \mathbb{N}$
- The computation is phrased as a two-point boundary value problem to (1) normalize the periodicity to "1" and (2) solve for the unknown period, T
- The system is then discretized, so that Newton's method can be used to find the solution
- An "unfolding" parameter, λ , is introduced
- Therefore, the equations of motion (as a system of first-order ODEs) look like:

$$(1) \quad \dot{x} = Tv_x + \lambda E_x \quad (2) \quad \dot{y} = Tv_y + \lambda E_y \quad (3) \quad \dot{z} = Tv_z + \lambda E_z$$

$$(4) \quad \dot{v}_x = T[2v_y + x - \frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x-1+\mu)}{r_2^3}] + \lambda E_{v_x} \quad (5) \quad \dot{v}_y = T[-2v_x + y - \frac{(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3}] + \lambda E_{v_y}$$

$$(6) \quad \dot{v}_z = T[-\frac{(1-\mu)z}{r_1^3} - \frac{\mu z}{r_2^3}] + \lambda E_{v_z}$$

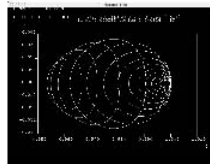
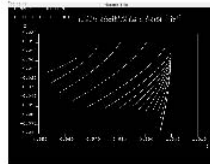
$$\text{where, } E = -\frac{1}{2}C = \frac{1}{2}(v_x^2 + v_y^2 + v_z^2) - U(x, y, z) - \frac{1}{2}\mu(1-\mu)$$

22 May 2005

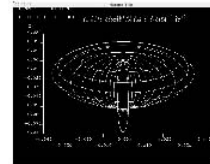
CP9 Sun PM - 2

RRB - 6

- “Halo” orbits
 - Example AUTO2000 output (three-dimensional orbit family case)

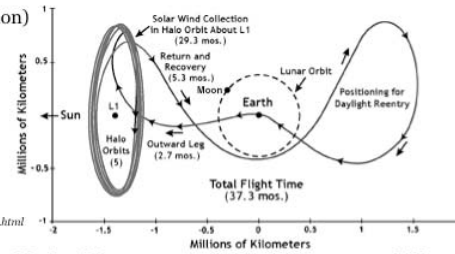
XY (m_1 and m_2 Orbit Plane)

XZ Plane



YZ Plane

- Application (e.g. Genesis Mission)



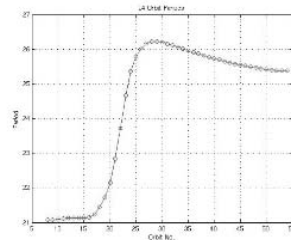
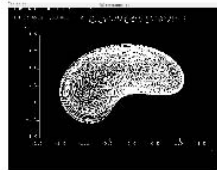
<http://genesismission.jpl.nasa.gov/mission/LOI.html>

22 May 2005

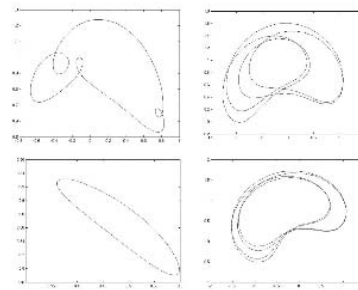
CP9 Sun PM - 2

RRB - 7

- One periodic orbit family (planar case) about the Earth-moon L_4 fixed-point



Period for each L_4 orbit. Primary orbit period = $2.1070352E+01$.



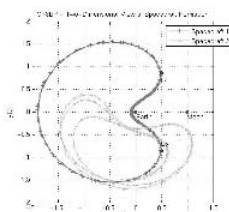
Orbit traces. Clockwise from lower left: (a) $2.10944E+01$, (b) $2.57786E+01$, (c) $2.57317E+01$, and (d) $2.54126E+01$.

22 May 2005

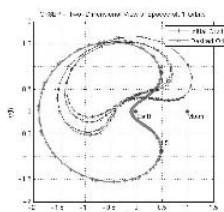
CP9 Sun PM - 2

RRB - 8

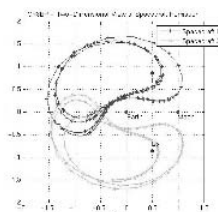
- Consider two spacecraft traveling in two different periodic orbits with different orbit periods [1]
- Find a periodic orbit with the same orbit period as the other that the first spacecraft can be maneuvered into [2]
- Determine the delta V magnitude, direction, and execution time to phase-lock the two spacecraft [3]



[1]



[2]



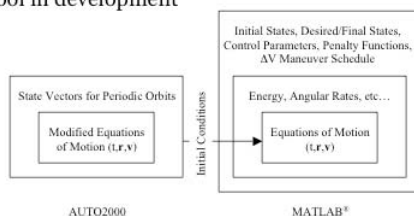
[3]

22 May 2005

CP9 Sun PM - 2

RRB - 9

- The computer tool in development



- AUTO2000 will be used to create a 'library' of periodic orbits
- Matlab
 - The initial state vector for each spacecraft will be defined along with the desired/final states
 - Control parameters such as orbit period or distance from the respective fixed-points will then be identified
 - Penalty functions (e.g. minimum propellant consumption) will be specified
 - Finally, a delta V maneuver schedule will be produced

22 May 2005


CP9 Sun PM - 2

RRB - 10

*We hope to be able to show you the finished
product at a future SIAM conference*

Appendix G: 2006 SIAM PDE Conference Presentation Charts


The charts provided in this appendix were presented at the SIAM (Society for Industrial and Applied Mathematics) Conference on Analysis of Partial Differential Equations, 10-12 July 2006, Boston, Massachusetts.

University of Southern California


SIAM Conference on Analysis of Partial Differential Equations
10-12 July 2006, Boston, Massachusetts


Session: CP2 - Stochastic and Multiscale Processes

**PARTICLE MOTION IN THE
RESTRICTED, THREE-VORTEX PROBLEM**



Ralph R. Basilio and Paul K. Newton
11:10 - 11:25 am EDT, Monday, 10 July 2006
Emerson Conference Room, Boston Park Plaza Hotel

10 July 2006
SIAM Conference on Analysis of PDEs
Ralph R. Basilio - 1

University of Southern California


Agenda

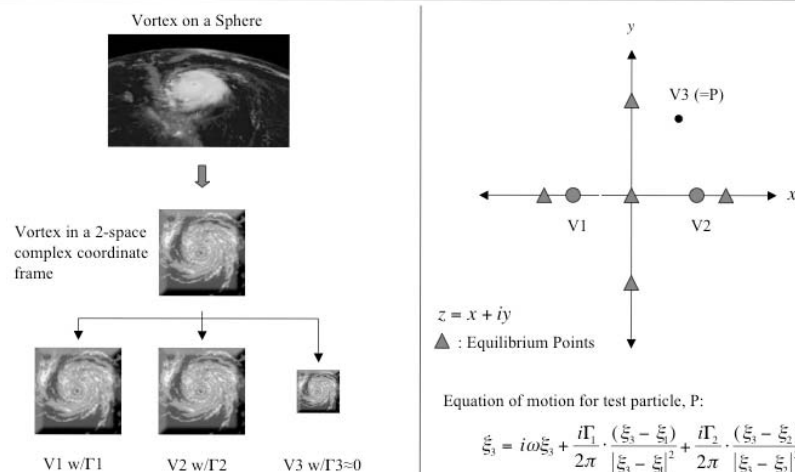
- Background and General Problem Statement
- Setting Up the General Problem
- Level Curves of the Hamiltonian
- Particle Study Set
- Controller 1: Phase-Locking
- Controller 2: Formation Establishment
- Example Problem
- Additional Information

10 July 2006
SIAM Conference on Analysis of PDEs
Ralph R. Basilio - 2

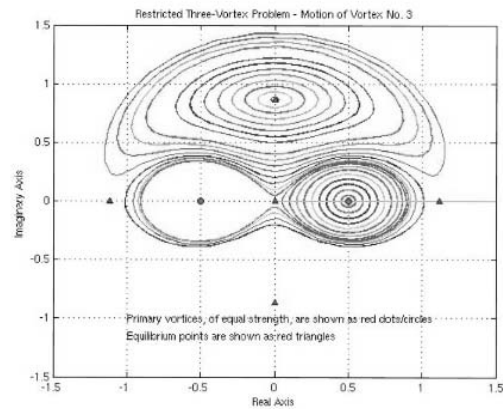
Background and General Problem Statement

- The general **three-body problem** involves investigating the motion of three mutually-attracting bodies
- The **circular restricted, three-body problem** is a specialized case where the third body is of negligible mass and does not influence the motion of the other two bodies
- As part of an investigation to phase-lock and establish dynamically-natural spacecraft formations, the **circular restricted, three-vortex problem** was studied as a proof of concept
- Through the use of **two separate controllers** four test particles traveling in different periodic orbits were
 - transferred to a single, new periodic orbit, and then
 - placed in a desired formation

Setting Up the General Problem



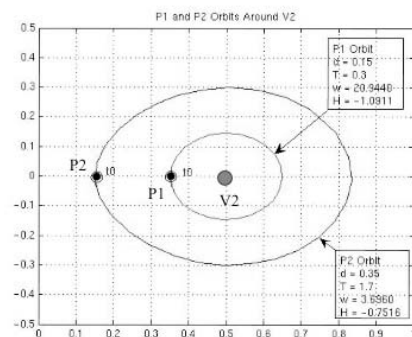
Level Curves of the Hamiltonian



$$H(u,v) = -\frac{1}{2}(u^2 + v^2) + (1-\lambda)\log(\sqrt{(u+\lambda)^2 + v^2}) + \lambda\log(\sqrt{(u+\lambda-1)^2 + v^2})$$

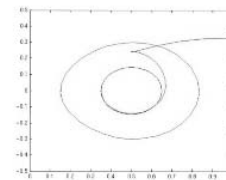
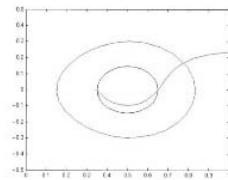
Particle Study Set

Attribute	P1 Orbit	P2 Orbit
Characteristic Dist, d	0.15	0.35
Orbit Period, T	0.3	1.7
Angular Rate, τ (or ω)	20.9440	3.6960
Hamiltonian, H	-1.0911	-0.7516



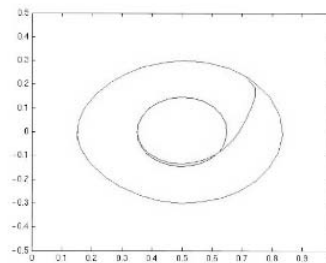
Controller 1: Phase-Locking

- Question 1: How can the standard equation of motion be altered to facilitate phase-locking?
- Three types of **additive** terms were examined
 - Time-independent (see lower left plot)
 - Time-dependent (see lower right plot)
 - Time-dependent trigonometric function (next page)



$$\dot{\xi}_3 = i\omega\xi_3 + \frac{i\Gamma_1}{2\pi} \cdot \frac{(\xi_3 - \xi_1)}{|\xi_3 - \xi_1|^2} + \frac{i\Gamma_2}{2\pi} \cdot \frac{(\xi_3 - \xi_2)}{|\xi_3 - \xi_2|^2} + \kappa$$

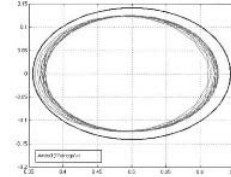
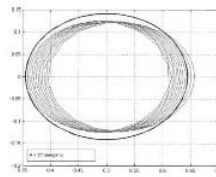
Controller 1: Phase-Locking (*continued*)



$$\dot{\xi}_3 = i\omega\xi_3 + \frac{i\Gamma_1}{2\pi} \cdot \frac{(\xi_3 - \xi_1)}{|\xi_3 - \xi_1|^2} + \frac{i\Gamma_2}{2\pi} \cdot \frac{(\xi_3 - \xi_2)}{|\xi_3 - \xi_2|^2} + \kappa \sin(\alpha\pi(t/T))$$

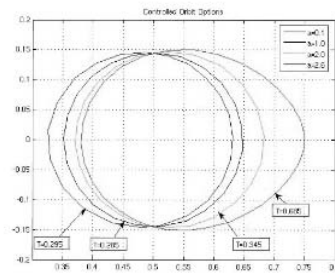
Controller 2: Formation Establishment

- Question 2: How can the standard equation of motion be altered to facilitate formation establishment?
- Three types of **first-term factors** were examined
 - Time-independent (next page)
 - Time-dependent (see lower left plot)
 - Time-dependent trigonometric function (see lower right plot)



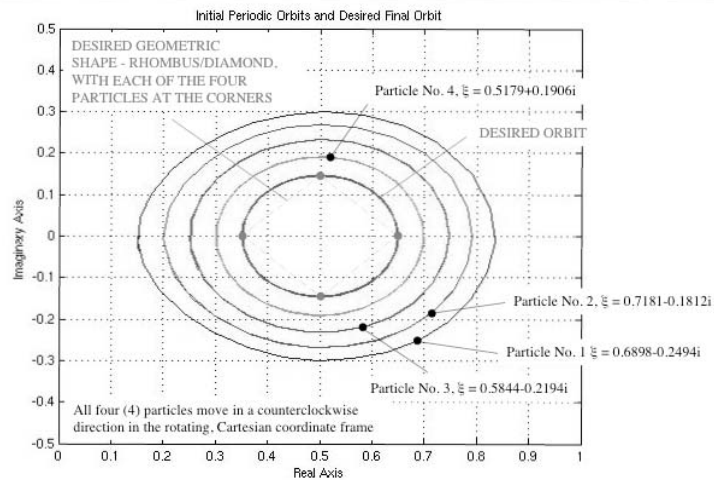
$$\dot{\xi}_3 = i\omega\xi_3 + \frac{i\Gamma_1}{2\pi} \frac{(\xi_3 - \xi_1)}{|\xi_3 - \xi_1|^2} + \frac{i\Gamma_2}{2\pi} \frac{(\xi_3 - \xi_2)}{|\xi_3 - \xi_2|^2} \quad \dot{\xi}_3 = \sin(t)i\omega\xi_3 + \frac{i\Gamma_1}{2\pi} \frac{(\xi_3 - \xi_1)}{|\xi_3 - \xi_1|^2} + \frac{i\Gamma_2}{2\pi} \frac{(\xi_3 - \xi_2)}{|\xi_3 - \xi_2|^2}$$

Controller 2: Formation Establishment (*continued*)



$$\dot{\xi}_3 = a i \omega \xi_3 + \frac{i\Gamma_1}{2\pi} \frac{(\xi_3 - \xi_1)}{|\xi_3 - \xi_1|^2} + \frac{i\Gamma_2}{2\pi} \frac{(\xi_3 - \xi_2)}{|\xi_3 - \xi_2|^2}$$

Example Problem

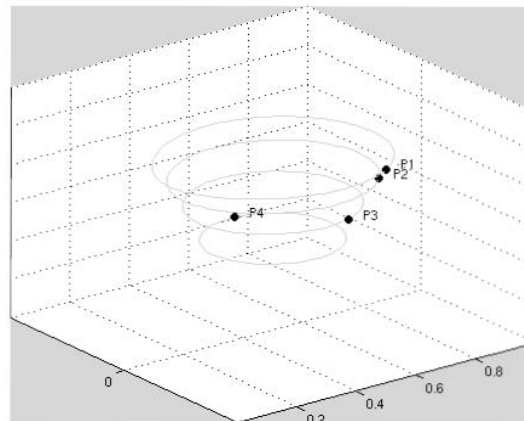


10 July 2006

SIAM Conference on Analysis of PDEs

Ralph R. Basilio - 11

Uncontrolled Particle Motion

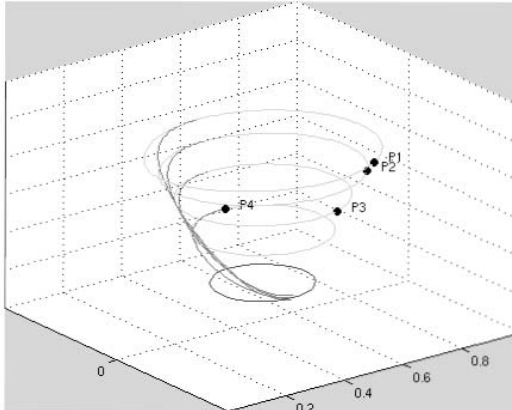


10 July 2006

SIAM Conference on Analysis of PDEs

Ralph R. Basilio - 12

Controlled Particle Motion Using Controller 1

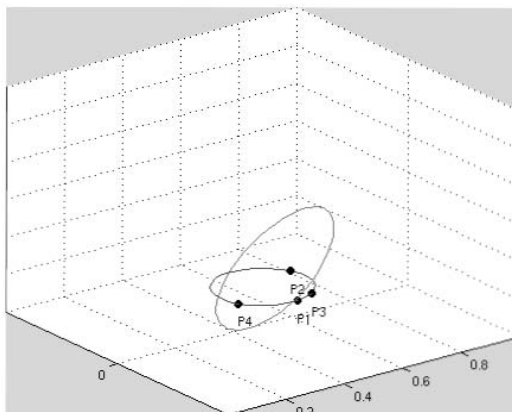


10 July 2006

SIAM Conference on Analysis of PDEs

Ralph R. Basilio - 13

Controlled Particle Motion Using Controller 2



10 July 2006

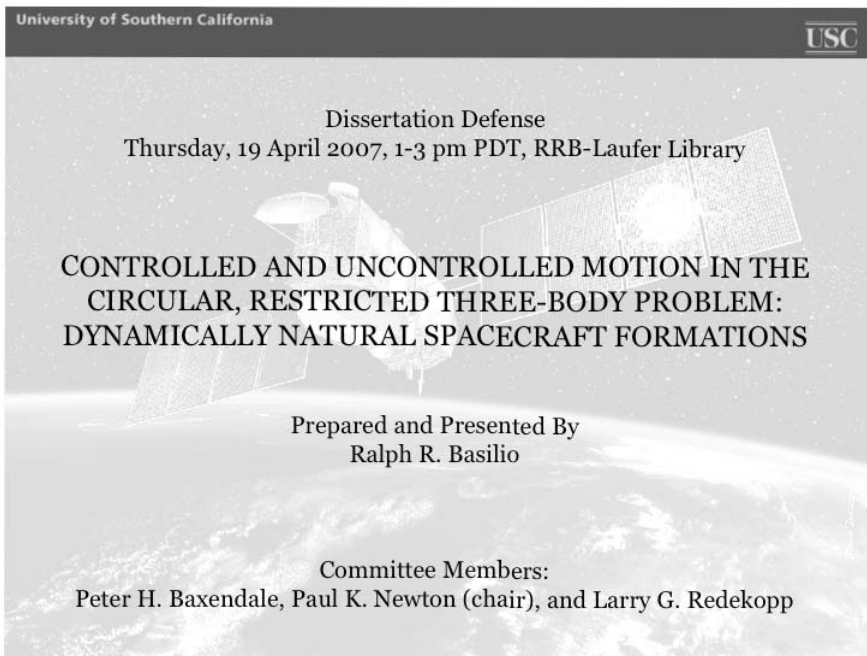
SIAM Conference on Analysis of PDEs

Ralph R. Basilio - 14

Additional Information

- Controller 1 can actually be used for both phase-locking and formation establishment using a **orbit resonant frequency** approach. In fact, a Matlab program was written that takes user defined or randomly-generated particle initial conditions to determine the total time to create a formation.
- The main motivation for Controller 2 was the desire to **reduce the total time** for phase-locking and formation establishment. For the example problem,
 - Orbit resonant frequency approach (Controller 1 only): $t = 21.74$
 - Controller 1 and Controller 2: $t = 2.18$
- *At the SIAM Conference on the Application of Dynamical Systems, 28 May - 01 Jun 07, Snowbird, Utah, we plan on demonstrating the use of similar techniques to establish **dynamically-natural spacecraft formations***

Appendix H: Dissertation Defense Presentation Charts



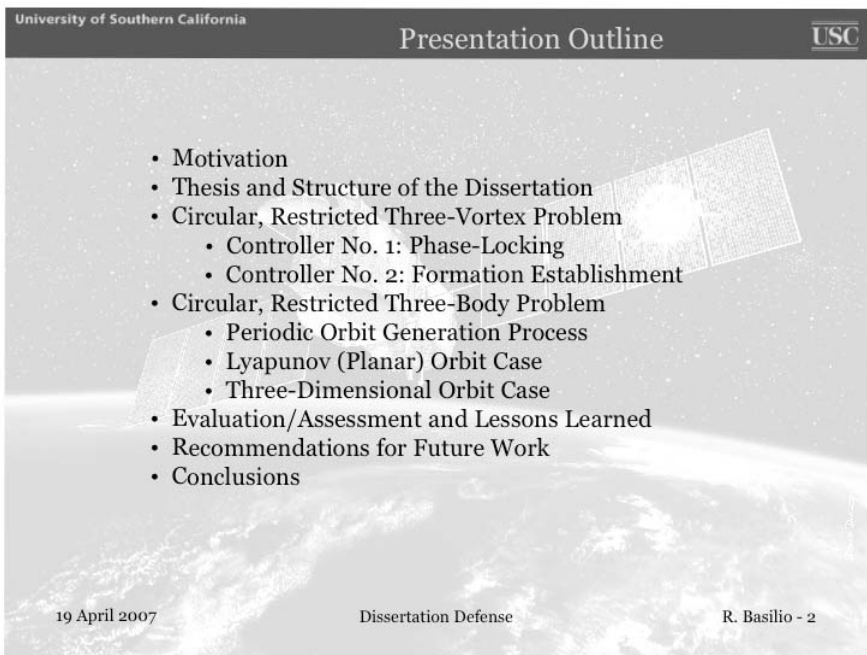
University of Southern California USC

Dissertation Defense
Thursday, 19 April 2007, 1-3 pm PDT, RRB-Laufer Library

**CONTROLLED AND UNCONTROLLED MOTION IN THE
CIRCULAR, RESTRICTED THREE-BODY PROBLEM:
DYNAMICALLY NATURAL SPACECRAFT FORMATIONS**

Prepared and Presented By
Ralph R. Basilio

Committee Members:
Peter H. Baxendale, Paul K. Newton (chair), and Larry G. Redekopp



University of Southern California USC

Presentation Outline

- Motivation
- Thesis and Structure of the Dissertation
- Circular, Restricted Three-Vortex Problem
 - Controller No. 1: Phase-Locking
 - Controller No. 2: Formation Establishment
- Circular, Restricted Three-Body Problem
 - Periodic Orbit Generation Process
 - Lyapunov (Planar) Orbit Case
 - Three-Dimensional Orbit Case
- Evaluation/Assessment and Lessons Learned
- Recommendations for Future Work
- Conclusions

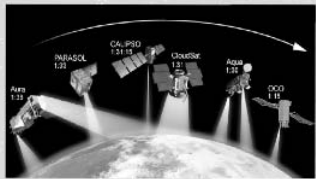
19 April 2007 Dissertation Defense R. Basilio - 2

University of Southern California

Motivation

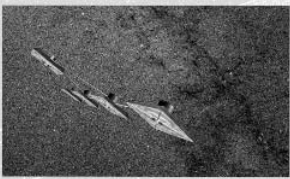
USC

- Formation flying involves operating multiple spacecraft in a pre-determined geometrical configuration that yields both individual and system benefits



A-Train Constellation

<http://csc.gallaudet.edu/soarhigh/A-TrainExplain.html>



Terrestrial Planet Finder (TPF-I)

http://planetquest.jpl.nasa.gov/TPF/tpf_architectures.cfm

- Spacecraft formations are generally constrained to operate in Keplerian orbits, i.e. circular or elliptical orbits, parabolic paths, or hyperbolic trajectories
- Continuation and bifurcation techniques can be used to densely foliate periodic orbits in the circular, restricted three-body problem (CR3BP)
- Genesis is one of NASA's first to use the CR3BP as a foundation, but for a single-spacecraft mission
- Can a novel concept for formation flying be developed around the CR3BP?

19 April 2007 Dissertation Defense R. Basilio - 3

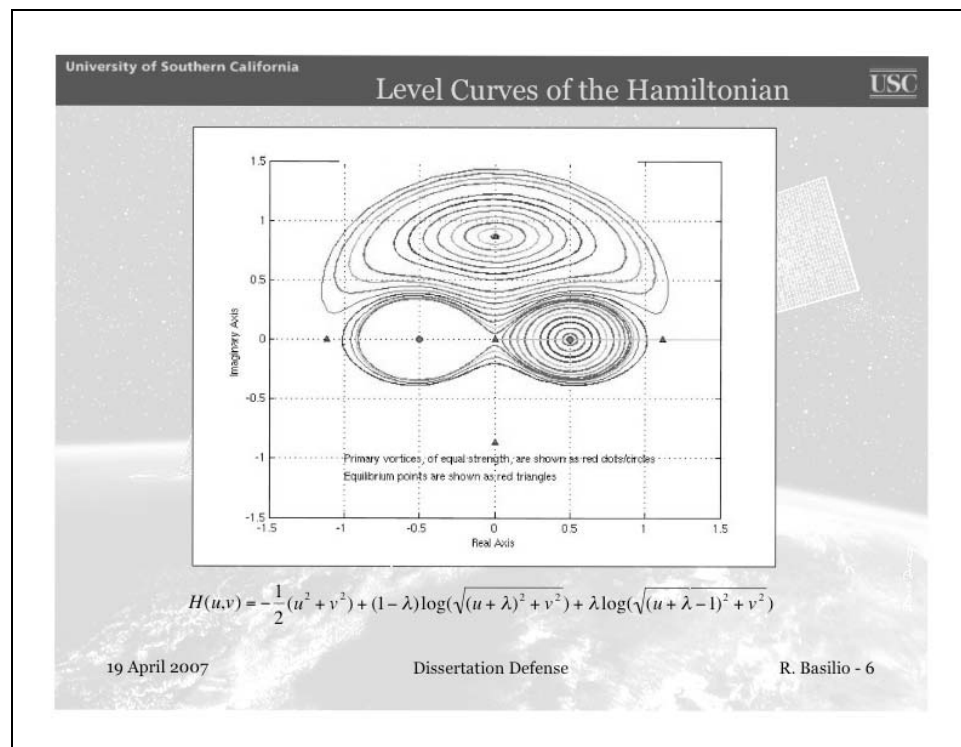
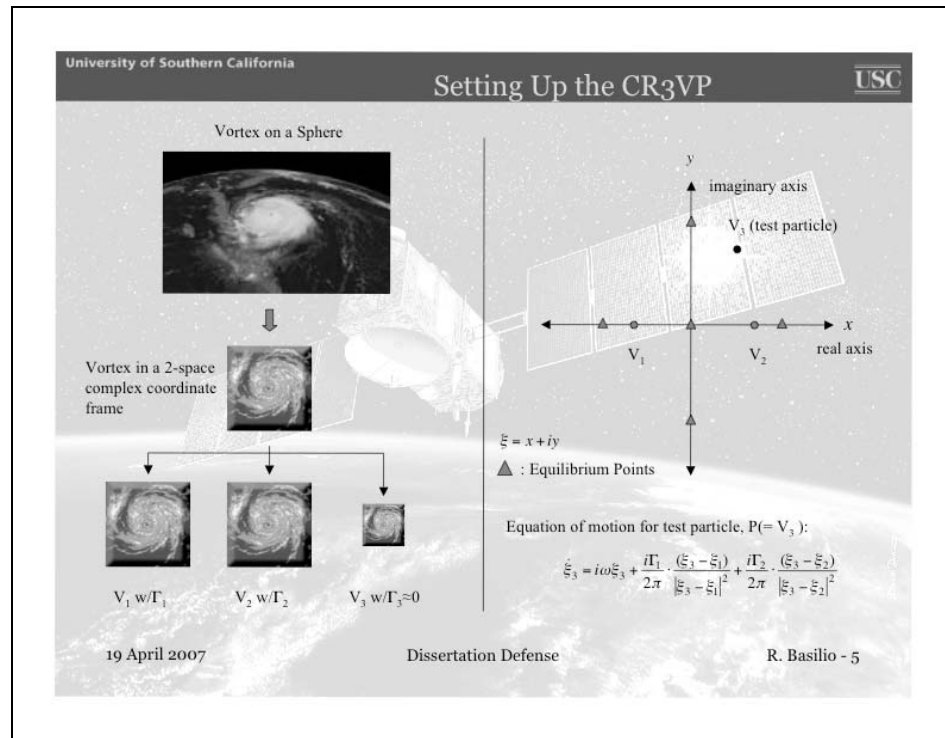
University of Southern California

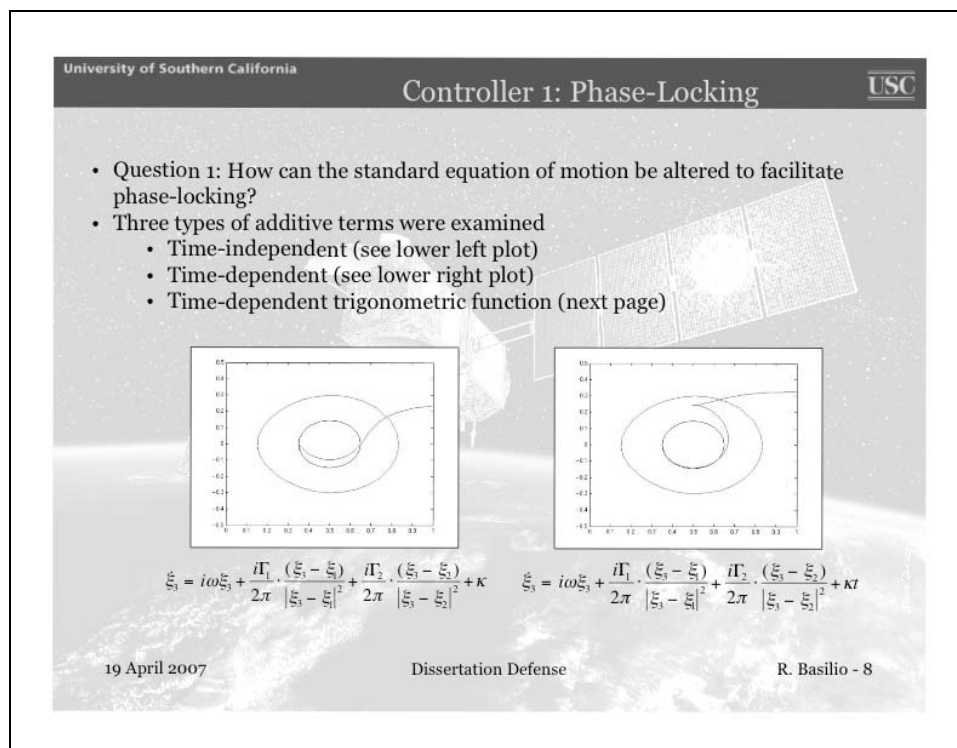
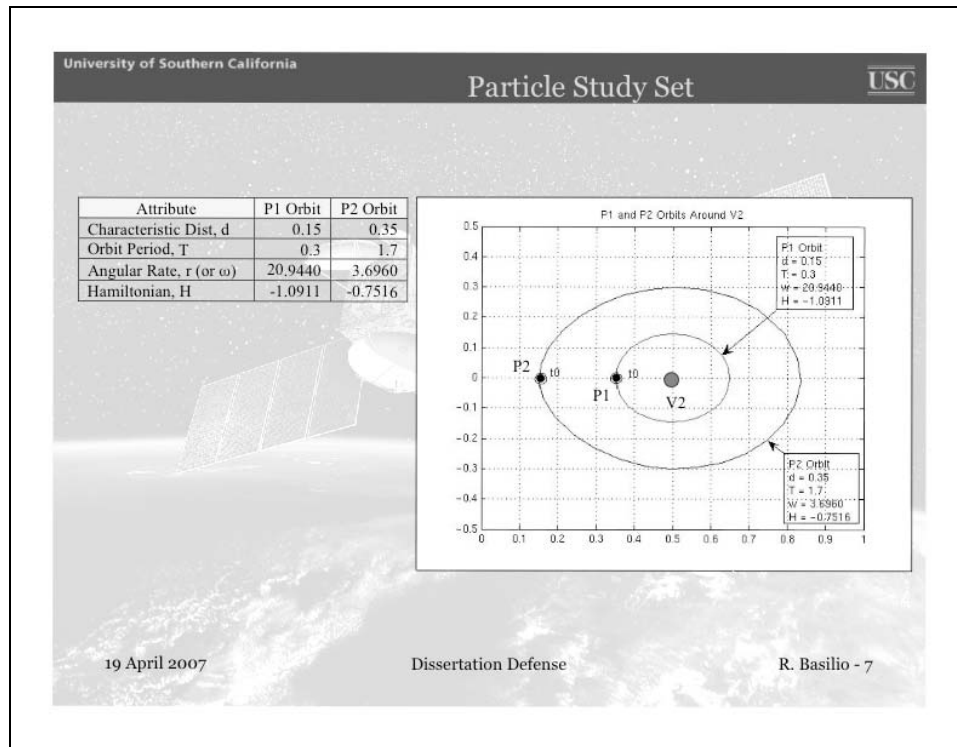
Thesis and Structure of the Dissertation

USC

- Thesis: Spacecraft traveling in uncontrolled motion along separate and distinct periodic orbits in the CR3BP
 - Can be placed in controlled motion, i.e. a controller is turned ON and OFF appropriately, to have them phase-locked on a single periodic orbit
 - Can also be placed in a desired formation using the controller in a resonant frequency/orbit approach, however, a second controller can be used in conjunction with the first to expedite the entire process
- Structure of the Dissertation
 - MATLAB computer simulations of Keplerian (two-body) orbits proved useful for subsequent modeling and analysis
 - MATLAB computer simulations of test particle motion in the circular, restricted three-vortex problem (CR3VP) in fluid mechanics served as a proof-of-concept
 - The AUTO 2000 continuation and bifurcation software tool was used to generate periodic orbits in the CR3BP
 - MATLAB was used to create computer simulations of controlled and uncontrolled spacecraft motion in the CR3BP for two specific cases
 - Lyapunov orbits (orbits in the plane of the primaries)
 - Three-dimensional orbits

19 April 2007 Dissertation Defense R. Basilio - 4

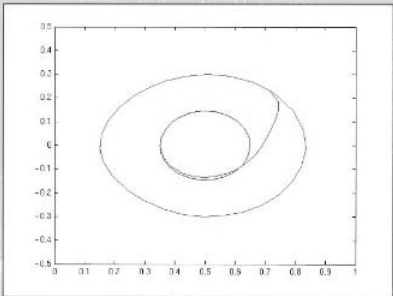




University of Southern California

Controller 1: Phase-Locking (cont.)

USC



$$\dot{\xi}_3 = i\omega\xi_3 + \frac{i\Gamma_1}{2\pi} \frac{(\xi_3 - \xi_1)}{|\xi_3 - \xi_1|^2} + \frac{i\Gamma_2}{2\pi} \frac{(\xi_3 - \xi_2)}{|\xi_3 - \xi_2|^2} + \kappa \sin(\alpha\pi(t/T))$$

19 April 2007

Dissertation Defense

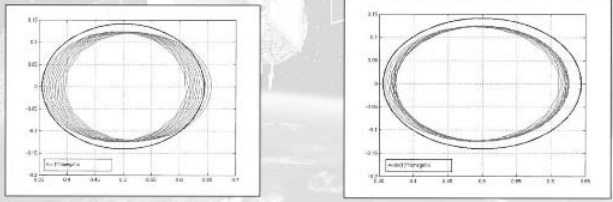
R. Basilio - 9

University of Southern California

Controller 2: Formation Establishment

USC

- Question 2: How can the standard equation of motion be altered to facilitate formation establishment?
- Three types of first-term factors were examined
 - Time-independent (next page)
 - Time-dependent (see lower left plot)
 - Time-dependent trigonometric function (see lower right plot)



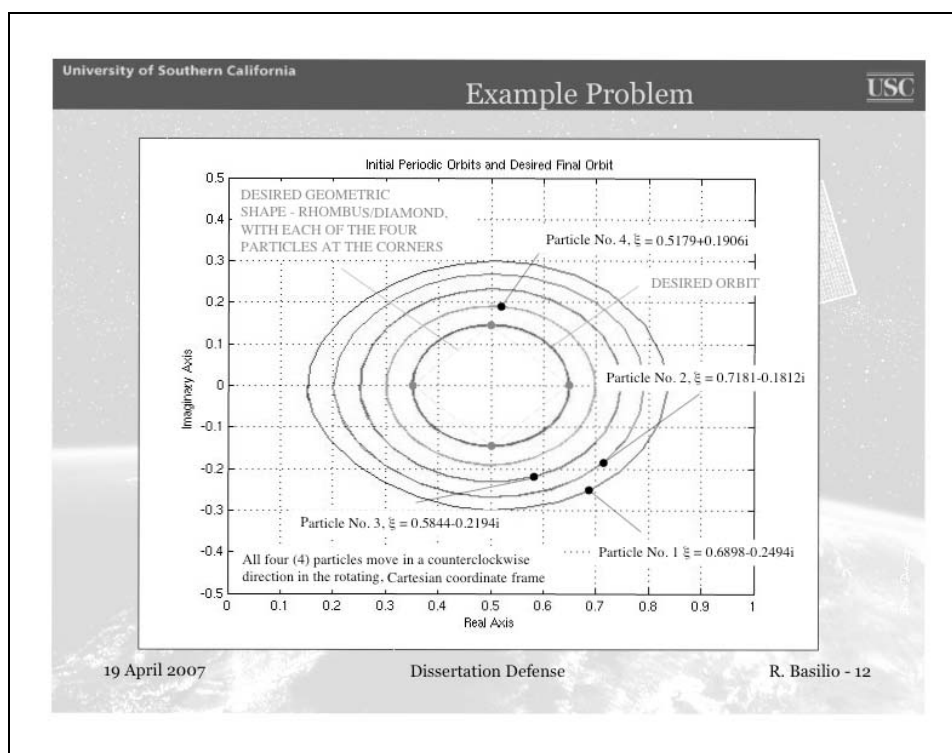
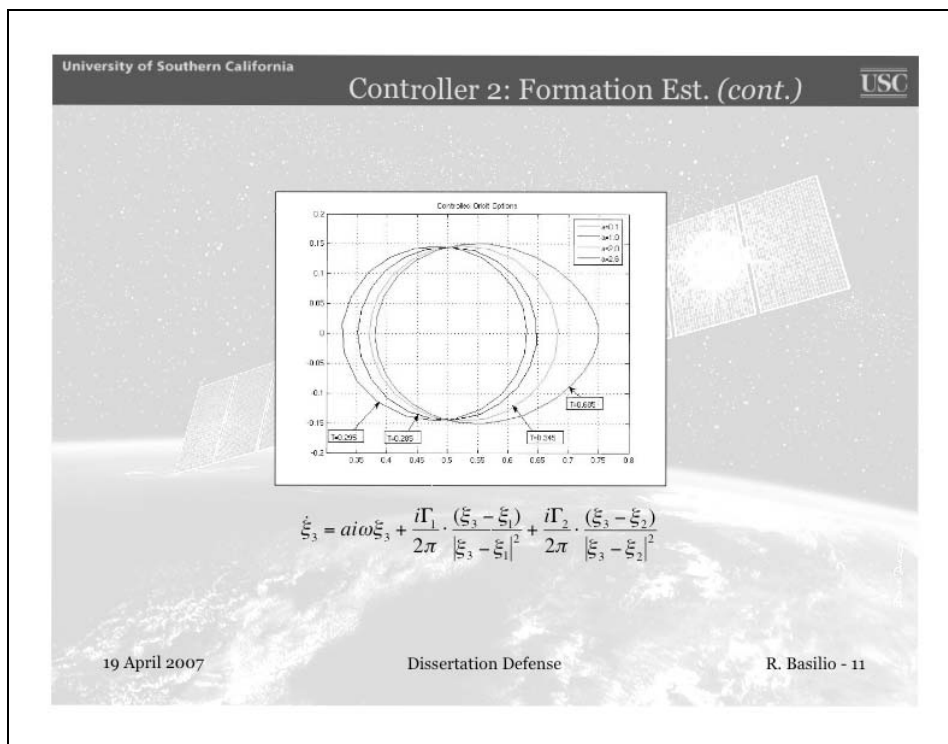
$$\dot{\xi}_3 = ti\omega\xi_3 + \frac{i\Gamma_1}{2\pi} \frac{(\xi_3 - \xi_1)}{|\xi_3 - \xi_1|^2} + \frac{i\Gamma_2}{2\pi} \frac{(\xi_3 - \xi_2)}{|\xi_3 - \xi_2|^2}$$

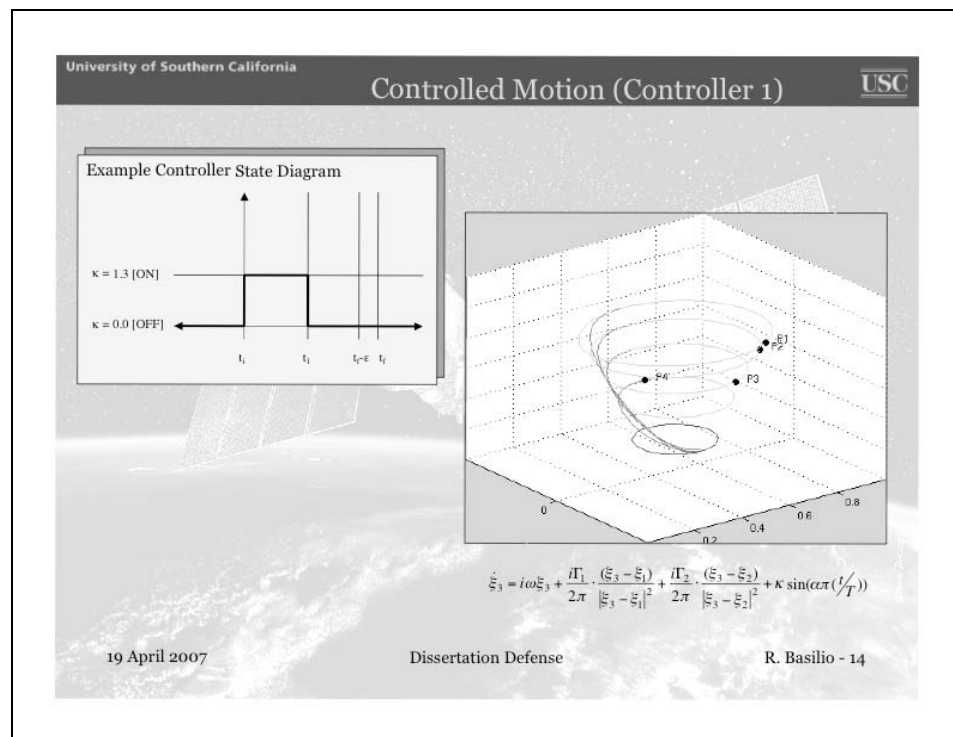
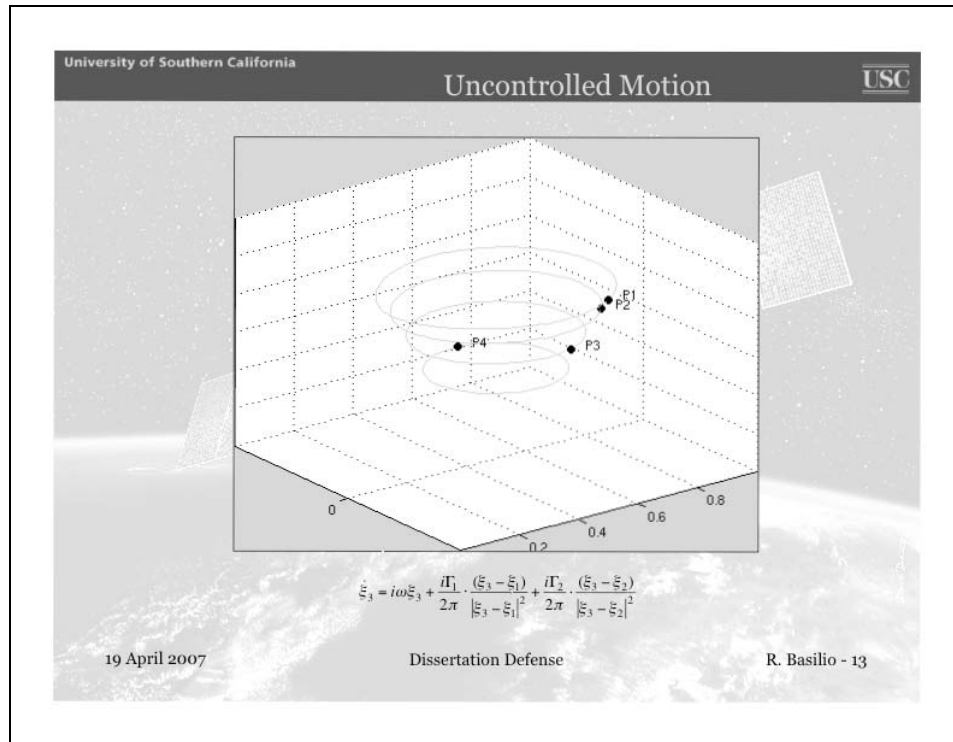
$$\dot{\xi}_3 = \sin(t)i\omega\xi_3 + \frac{i\Gamma_1}{2\pi} \frac{(\xi_3 - \xi_1)}{|\xi_3 - \xi_1|^2} + \frac{i\Gamma_2}{2\pi} \frac{(\xi_3 - \xi_2)}{|\xi_3 - \xi_2|^2}$$

19 April 2007

Dissertation Defense

R. Basilio - 10

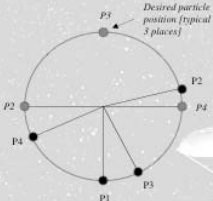


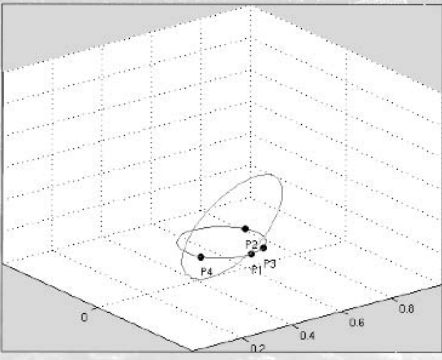


University of Southern California

Controlled Motion (Controller 2)

USC





$t = 1.48 + 0.05 = 1.53$

Place P4 on orbit where period = $0.28 + 0.16 = 0.44$
(Controller no. 2 scale factor = 2.37)
P4 in formation = $1.53 + 0.44 = 1.97$

$t = 1.48 + 0.20 = 1.68$

Place P2 on orbit where period = $0.28 + 0.15 = 0.43$
(Controller no. 2 scale factor = 2.37)
P2 in formation = $1.68 + 0.43 = 2.11$

$t = 1.48 + 0.26 = 1.74$

Place P3 on orbit where period = $0.28 + 0.16 = 0.44$
(Controller no. 2 scale factor = 2.37)
P3 in formation = $1.74 + 0.44 = 2.18$

$$\ddot{\mathbf{r}}_i = a\omega\ddot{\mathbf{r}}_s + \frac{i\Gamma_i \cdot (\frac{\mathbf{r}_i}{r_i} - \frac{\mathbf{r}_j}{r_j})}{2\pi \frac{|\mathbf{r}_i - \mathbf{r}_j|^2}{|\mathbf{r}_i|^2}} + \frac{i\Gamma_j \cdot (\frac{\mathbf{r}_j}{r_j} - \frac{\mathbf{r}_i}{r_i})}{2\pi \frac{|\mathbf{r}_i - \mathbf{r}_j|^2}{|\mathbf{r}_j|^2}}$$

19 April 2007

Dissertation Defense

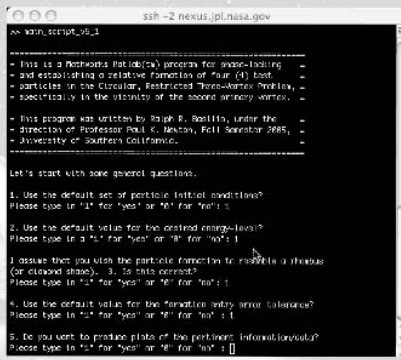
R. Basilio - 15

University of Southern California

Results of CR3VP Study

USC

- Controller 1 can actually be used for both phase-locking and formation establishment using a orbit resonant frequency approach. In fact, a MATLAB program was written that takes user defined or randomly-generated particle initial conditions to determine the total time to create a formation.
- The main motivation for Controller 2 was the desire to reduce the total time for phase-locking and formation establishment. For the example problem,
 - Orbit resonant frequency approach (Controller 1 only): $t = 21.74$
 - Controller 1 and Controller 2: $t = 2.18$



19 April 2007

Dissertation Defense

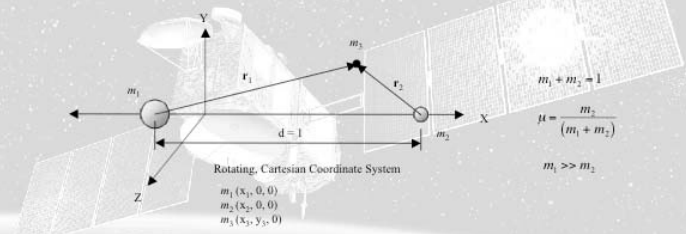
R. Basilio - 16

University of Southern California

Setting Up the CR₃BP

USC

- A special case of the general three-body problem: determine the motion of a third object, m_3 (of negligible mass), under the gravitational influence of two primaries, m_1 and m_2 , in circular orbits about the system barycenter (origin)



Rotating, Cartesian Coordinate System
 $m_1(x_1, 0, 0)$
 $m_2(x_2, 0, 0)$
 $m_3(x_3, y_3, 0)$

- There are a number of ways to tackle this problem, but we use a simple Newtonian approach. The general expression for the force acting on m_3 is

$$\mathbf{F}_3 = \mathbf{F}_1 - \mathbf{F}_2 - m_3(2\mathbf{w} \times \dot{\mathbf{r}}) - m_3[\mathbf{w} \times (\mathbf{w} \times \mathbf{r})]$$

- Accounting for coriolis and centrifugal forces allows us to state the m_3 equations of motion in the rotating, Cartesian coordinate system

19 April 2007 Dissertation Defense R. Basilio - 17

University of Southern California

Setting Up the CR₃BP (cont.)

USC

- The m_3 second-order ODEs (Ordinary Differential Equations) are:

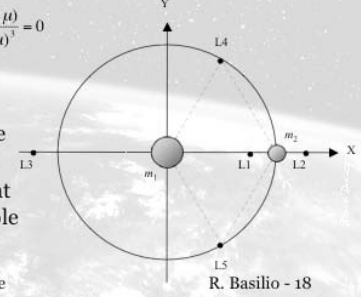
$$(1) \quad \ddot{x} - 2\dot{y} - x = -\frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x-1+\mu)}{r_2^3} \quad (2) \quad \ddot{y} + 2\dot{x} - y = \frac{-(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3} \quad (3) \quad \ddot{z} = -\frac{(1-\mu)z}{r_1^3} - \frac{\mu z}{r_2^3}$$

[We use the proven Runge-Kutta technique for numerically-integrating the ODEs]

- In the rotating coordinate frame, there are several fixed-points where the velocity and acceleration terms are zero, i.e. $\dot{x} = \dot{y} = \dot{z} = 0$ and $\ddot{x} = \ddot{y} = \ddot{z} = 0$
- From eq. 1 and 2 above, $r_1 = r_2 = 1$, forming two legs of an equilateral triangle
- Noticing that $y=0$ is a solution to eq. 2, and substituting for r_1 and r_2 , eq. 1 becomes

$$f(x) = x - \frac{(1-\mu)(x+\mu)}{(x+\mu)^3} - \frac{\mu(x-1+\mu)}{(x-1+\mu)^3} = 0$$

- The three real roots of the quintic equation above must be solved for numerically
- The sketch on the right shows the approximate locations of all five fixed-points
- Linear stability theory can be used to show that L_1 , L_2 , and L_3 are unstable and L_4 & L_5 are stable fixed-points, respectively



19 April 2007 Dissertation Defense R. Basilio - 18

University of Southern California

AUTO 2000

USC

- To expedite the process of finding periodic orbits about fixed-points, the continuation & bifurcation analysis tool, AUTO 2000, was employed
- The tool allows one to solve simple algebraic problems and ODEs
- In general AUTO 2000 solves equations of the form: $F(x) = 0, F: \mathbf{R}^{n+1} \rightarrow \mathbf{R}^n, n \in W$
- The computation is phrased as a two-point Boundary Value Problem (BVP) to (1) normalize the periodicity to "1" and (2) solve for the unknown period, T
- The system is then discretized, so that Newton's method can be used to find the solution
- An "unfolding" parameter, λ , is introduced
- Therefore, the equations of motion (as a system of first-order ODEs) look like:

$$\begin{aligned}
 (1) \quad \dot{x} &= TV_x + \lambda E_x & (2) \quad \dot{y} &= TV_y + \lambda E_y & (3) \quad \dot{z} &= TV_z + \lambda E_z \\
 (4) \quad \dot{v}_x &= T[2v_x + x - \frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x-1+\mu)}{r_2^3}] + \lambda E_{v_x} & (5) \quad \dot{v}_y &= T[-2v_y + y - \frac{(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3}] + \lambda E_{v_y} \\
 (6) \quad \dot{v}_z &= T[-\frac{(1-\mu)z}{r_1^3} - \frac{\mu z}{r_2^3}] + \lambda E_{v_z}
 \end{aligned}$$

where, $E = \frac{1}{2}(v_x^2 + v_y^2 + v_z^2) - \frac{1}{2}(x^2 + y^2) - \frac{1}{2}\mu(1-\mu)$

19 April 2007

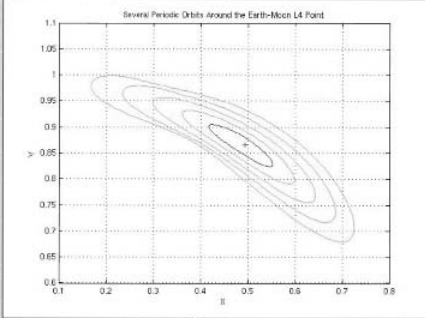
Dissertation Defense

R. Basilio - 19

University of Southern California

AUTO 2000 Lyapunov Orbits

USC



- A catalogue of periodic orbits around the earth-moon L_4 point were created
- Five of the planar orbits were selected for study (*see figure*)
- Floquet multipliers indicate these orbits are stable (*multipliers for the outer orbit are shown in the table*)

- Using an Initial Value Problem (IVP) approach in MATLAB to propagate an arbitrary state vector associated with an AUTO 2000-generated stable periodic orbit produces a similar orbit

Multiplier No.	Real Component	Imaginary Component
0	1.000000	0.000000
1	-0.6059415	0.795509
2	-0.6059415	-0.795509
3	1.000000	0.000000
4	0.3035971	0.952801
5	0.3035971	-0.952801

19 April 2007

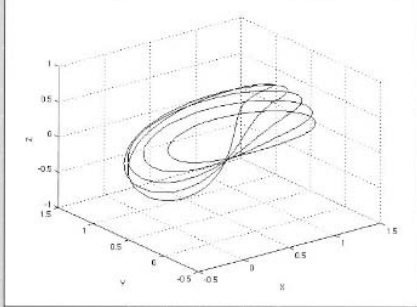
Dissertation Defense

R. Basilio - 20

University of Southern California

AUTO 2000 Three-Dimensional Orbits

USC



- Five of the three-dimensional (3D) orbits from the catalogue were selected for study (*see figure*)
- Floquet multipliers indicate these orbits are unstable (*multipliers for the outer orbit are shown in the table*)

Multiplier No.	Real Component	Imaginary Component
0	1.000000	0.000000
1	0.2900044	0.957025
2	0.2900044	-0.957025
3	1.000000	0.000000
4	0.8964660	0.000000
5	1.115491	0.000000

- Using an Initial Value Problem (IVP) approach in MATLAB to propagate an arbitrary state vector associated with an AUTO 2000-generated unstable periodic orbit does not produce a similar orbit

19 April 2007 Dissertation Defense R. Basilio - 21

University of Southern California

The Controllers

USC

- Controller No. 1 - The revised set of second-order ODEs are:

$$(1) \quad \ddot{x} - 2\dot{y} - x = -\frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x-1+\mu)}{r_2^3} + \kappa \sin\left(\frac{c\pi t}{T}\right) \quad (3) \quad \ddot{z} = -\frac{(1-\mu)z}{r_1^3} - \frac{\mu z}{r_2^3} + \kappa \sin\left(\frac{c\pi t}{T}\right)$$

$$(2) \quad \ddot{y} + 2\dot{x} - y = -\frac{-(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3} + \kappa \sin\left(\frac{c\pi t}{T}\right)$$

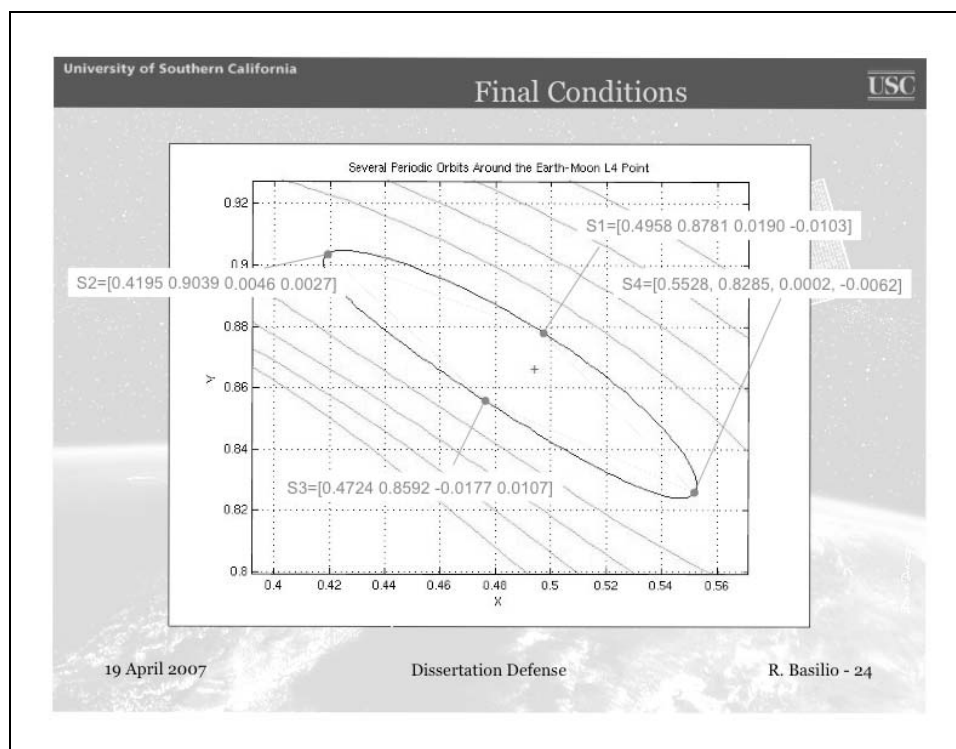
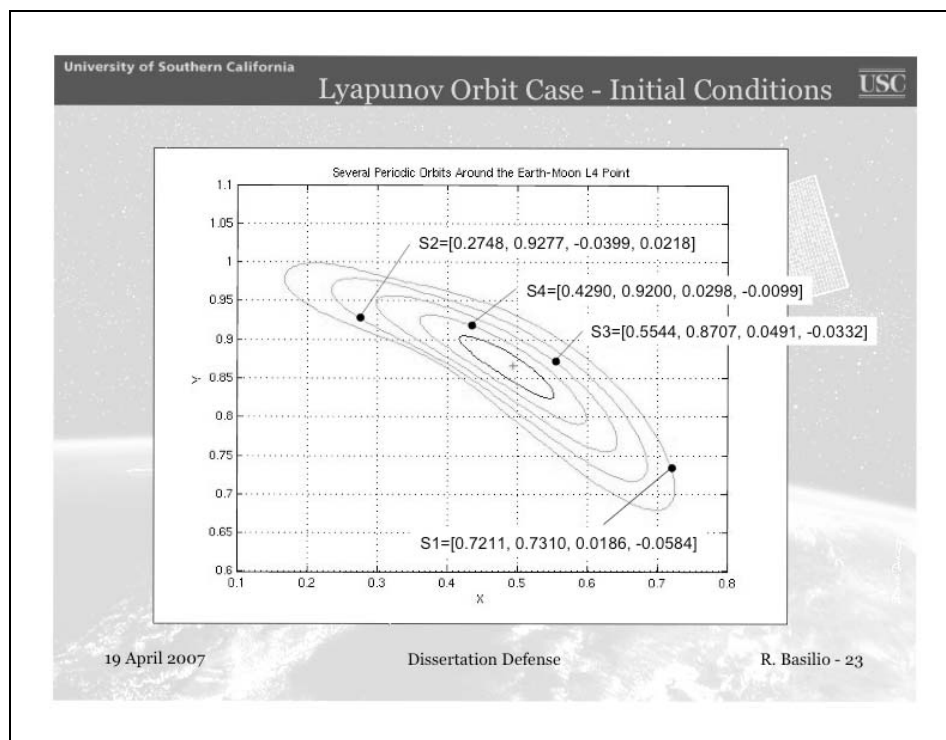
- Controller No. 2 - The revised set of second-order ODEs are:

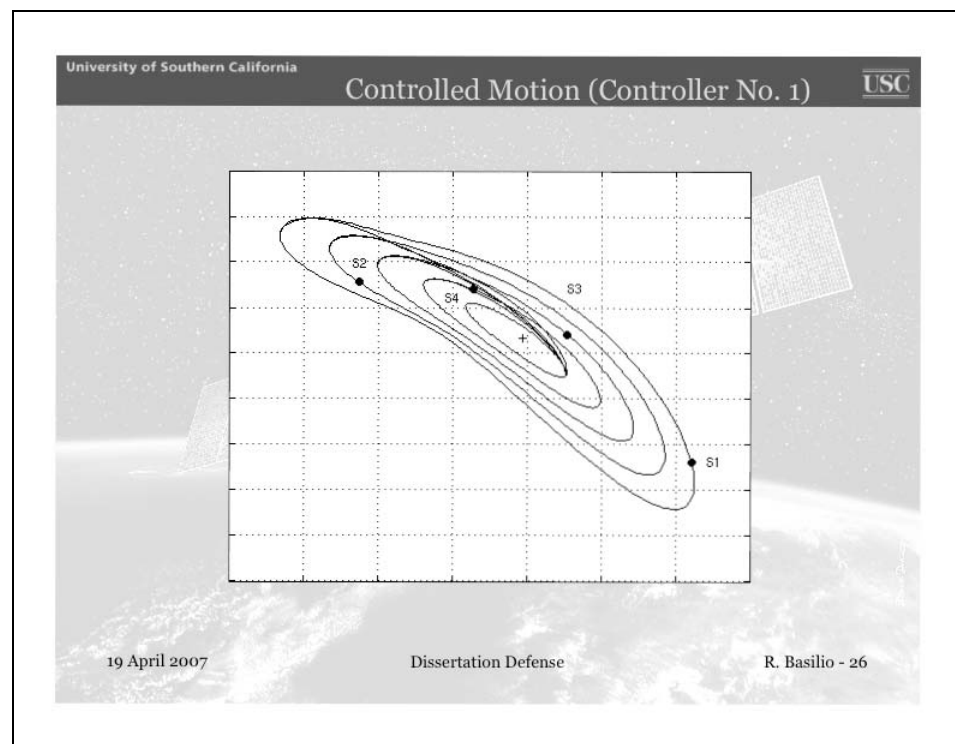
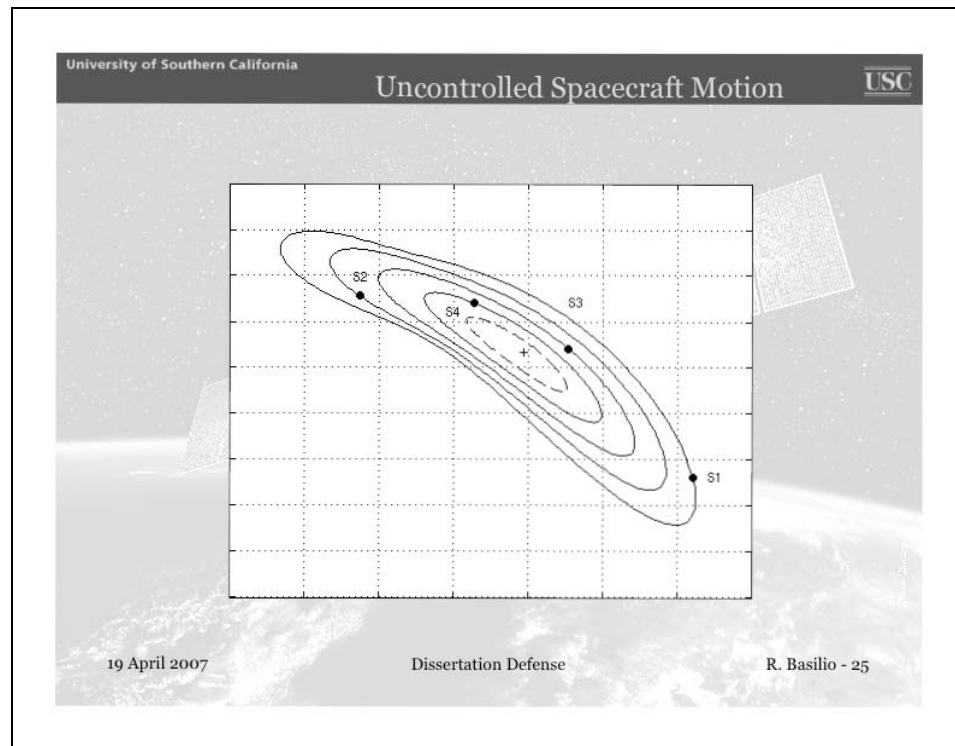
$$(1) \quad \ddot{x} + s_1 2\dot{y} - x = -\frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x-1+\mu)}{r_2^3} \quad (3) \quad \ddot{z} = -\frac{s_1(1-\mu)z}{r_1^3} - \frac{\mu z}{r_2^3}$$

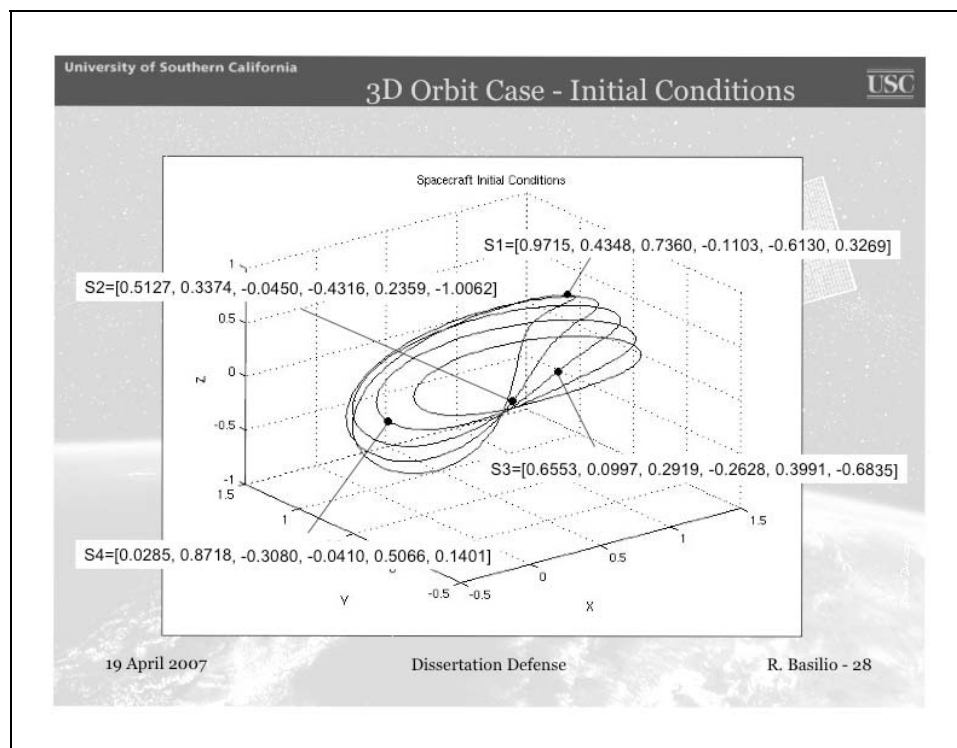
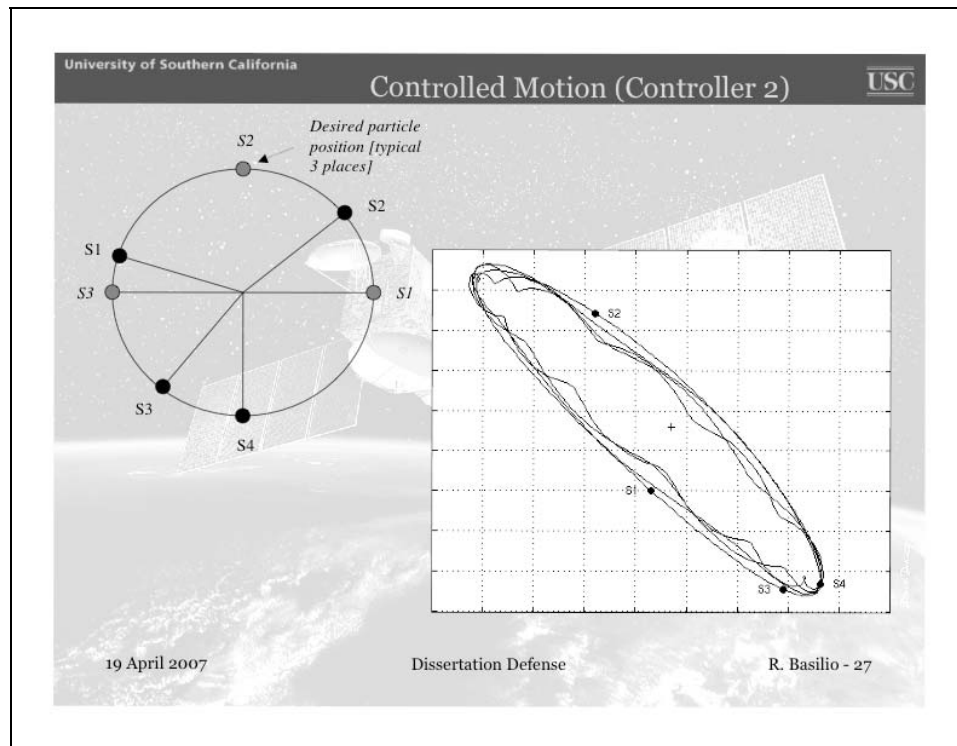
$$(2) \quad \ddot{y} + s_2 2\dot{x} - y = -\frac{-(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3}$$

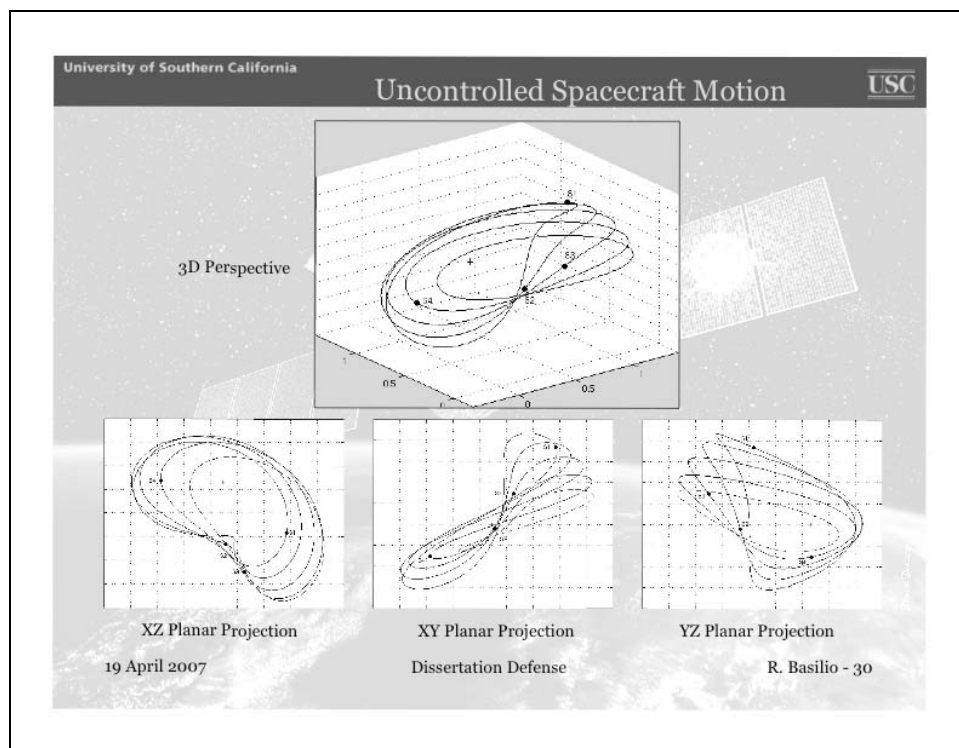
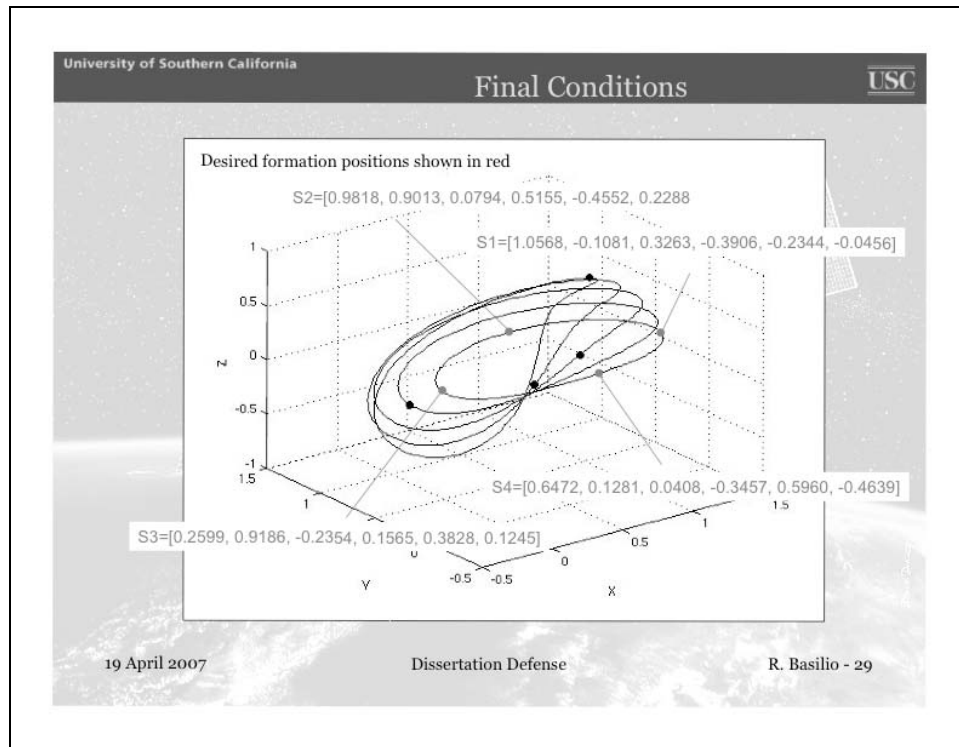
- Equation (3) in each case above is not applicable for Lyapunov orbits
- Orbit periods are so similar, e.g. 0.3% difference between adjacent orbits, that Controller No. 2 is required to create formations
- Although the positions will match, an impulse is needed at the conclusion of the controlled motion state to correct velocity term(s)

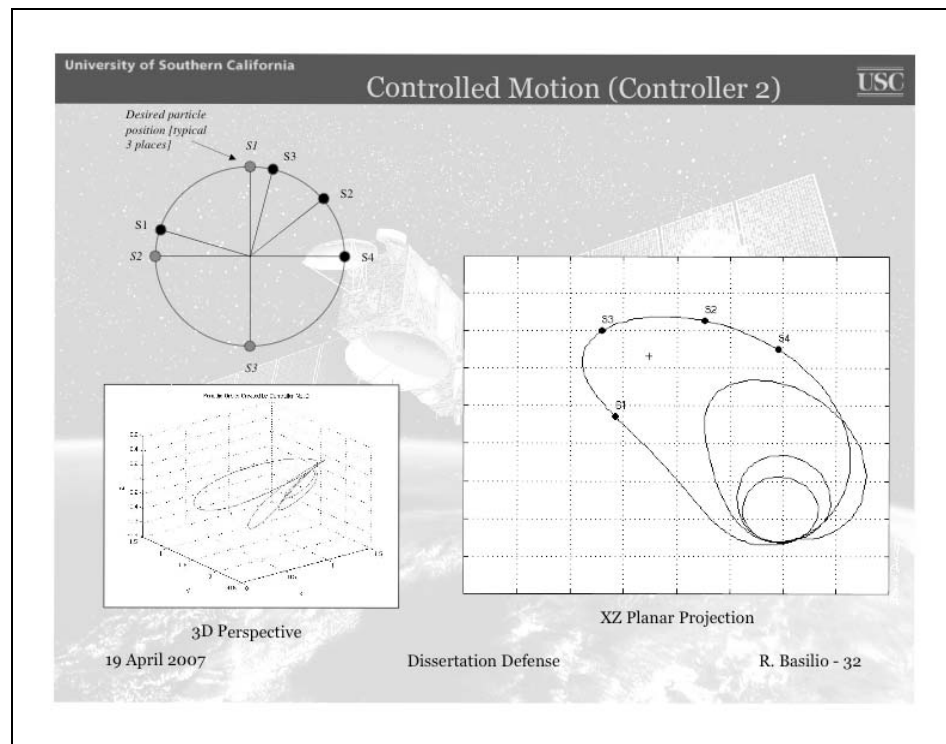
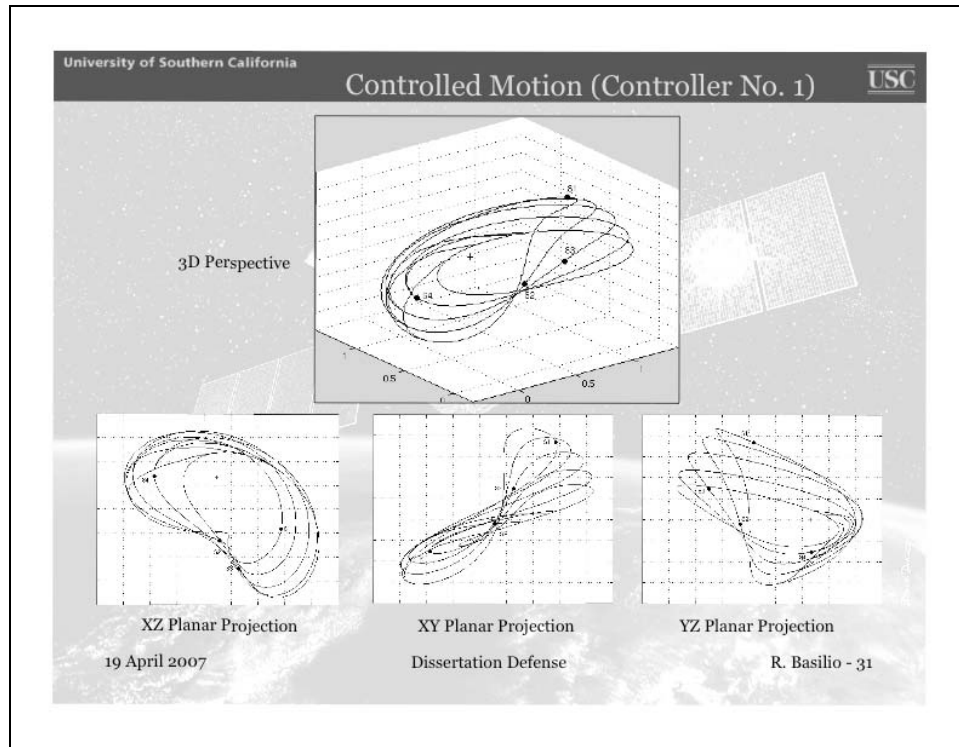
19 April 2007 Dissertation Defense R. Basilio - 22











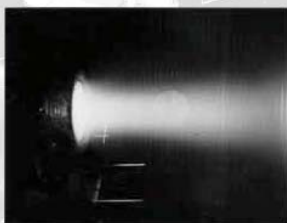
University of Southern California
Evaluation/Assessment...
USC

- In the CR3VP
 - The trigonometric function of Controller No. 1 is a smooth function and creates a smooth transfer trajectory
 - Controller No. 1 can be used in a resonant frequency/orbit approach to create a desired formation
 - Controller No. 2 expedites the formation establishment process
- In the CR3BP
 - The trigonometric function of Controller No. 1 is a smooth function and creates a relatively smooth transfer trajectory
 - Controller No. 1 can also be used in a resonant frequency/orbit approach to create a desired formation, but is not as practical given that the periods of the orbits examined are so similar to one another
 - Controller No. 2 truly expedites the formation establishment process
 - Values of the controller terms (for both Controller No. 1 and No. 2) must be independent for each applicable equation to create the necessary trajectories
 - The MATLAB IVP approach can simulate AUTO 2000 BVP results as long as the periodic orbits are stable
 - An impulse is needed to correct any velocity term(s) at the conclusion of each controlled mode state

19 April 2007
Dissertation Defense
R. Basilio - 33

University of Southern California
Recommendations for Future Work
USC

- Use an elliptical, restricted three-body problem (ER3BP) approach, since most natural and artificial satellite orbits have some degree of eccentricity, and therefore, are not truly circular
- Incorporate disturbance force and torque terms, e.g. solar pressure, for more accurate, full-force modeling of controlled and uncontrolled motion cases
- Tune controllers, so as to eliminate the need for impulsive maneuvers
- Match each controller with a low thrust actuator, e.g. solar electric propulsion (*see figure below*), to verify practicality
- Augment AUTO 2000 to create specific, user-defined periodic orbits



<http://nmp.jpl.nasa.gov/ds1/tech/sep.html>

19 April 2007
Dissertation Defense
R. Basilio - 34

Using the circular, restricted three-vortex problem in fluid mechanics as a proof-of-concept, it was shown that controllers (forcing functions) in the circular, restricted three-body problem can be used to phase lock multiple spacecraft onto a single periodic orbit and also to establish a desired formation

- Near-Term Plans
 - A joint JPL-USC proposal on "Spacecraft Formation Flying in the Circular, Restricted Three-Body Problem" will be submitted to the JPL Call for Innovative Space Mission and Instrument Concepts in May
 - This work will be described in the 2007 SIAM Conference on Applications of Dynamical Systems poster session, 28 May - 01 June, 2007, Snowbird, Utah
 - A draft abstract/paper/article on the circular, restricted three-vortex problem completed as part of the effort will be submitted to a conference or journal for consideration in August