AUTONOMOUS CONTROLS ALGORITHM
FOR FORMATION FLYING OF SATELLITES

by

LUIS A. SANTIAGO
B.S. University of Central Florida, 2001

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science in Aerospace Engineering
in the Department of Mechanical Materials and Aerospace Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2006
This document describes the design and analysis of the Navigation, Guidance and Control System for the KnightSat project. The purpose for the project is to test and demonstrate new technologies the Air Force would be interested in for research and development. The primary mission of KnightSat is to show how a constellation of satellites can maintain relative position with each other autonomously using the Microwave Electro Thermal (MET) thruster. The secondary mission is to use multiple satellite imagery to obtain 3 dimensional stereo photographs of observable terrain. Formation flying itself has many possible uses for future applications. Selected missions that require imaging or data collection can be more economically accomplished using smaller multiple satellites. The MET thruster is a very efficient, but low thrust alternative that can provide thrust for a very long time, hence provide the low thrust necessary to maintain the satellites at a constant separation. The challenge is to design a working control algorithm to provide the desired output data to be used to command the MET thrusters. The satellites are to maintain a constant relative distance from each other, and use the least amount of fuel possible. If one satellite runs out of fuel before the other, it would render the constellation less useful or useless. Hence, the satellites must use the same amount of fuel in order to maintain an optimal operational duration on orbit.
To my Lord Jesus Christ

To my son and fiance

To my father Luis F. Santiago

To my mother Anamary Torres

To my grandparents Celso B. Torres and Tati Lugo

may you look down and smile
ACKNOWLEDGMENTS

I would like to thank God for everything I’ve done. Without Him I would be nothing.

I want to thank Dr. Roger W. Johnson for his expertise, knowledge and guidance throughout this thesis. I want to thank my father whose support has been impeccable. I want to thank my mother for instilling in me a sense of drive, discipline and dedication.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xi</td>
</tr>
<tr>
<td>LIST OF ACRONYMS/ABBREVIATIONS/SYMBOLS</td>
<td>xii</td>
</tr>
<tr>
<td>CHAPTER ONE: INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>CHAPTER TWO: LITERATURE REVIEW</td>
<td>4</td>
</tr>
<tr>
<td>CHAPTER THREE: METHODOLOGY</td>
<td>7</td>
</tr>
<tr>
<td>CHAPTER FOUR: MANEUVERS</td>
<td>16</td>
</tr>
<tr>
<td>CHAPTER FIVE: FINDINGS</td>
<td>47</td>
</tr>
<tr>
<td>CHAPTER SIX: CONCLUSION</td>
<td>82</td>
</tr>
<tr>
<td>APPENDIX A: 2 BODY PROBLEM</td>
<td>83</td>
</tr>
<tr>
<td>APPENDIX B: PLANAR ORBITAL STATE EQUATIONS</td>
<td>86</td>
</tr>
<tr>
<td>APPENDIX C: ANGULAR ORBITAL STATE EQUATIONS</td>
<td>89</td>
</tr>
<tr>
<td>APPENDIX D: J2 PERTURBATIONS</td>
<td>96</td>
</tr>
<tr>
<td>APPENDIX E: DRAG PERTURBATIONS</td>
<td>99</td>
</tr>
<tr>
<td>APPENDIX F: SIMULATION MATLAB PROGRAM</td>
<td>101</td>
</tr>
<tr>
<td>LIST OF REFERENCES</td>
<td>131</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1: Knightsat Objective......................................................................................................... 2
Figure 2: 2-D Orbital Parameters...................................................................................................... 8
Figure 3: Position Orbital Parameters............................................................................................ 8
Figure 4: Planar angle $\theta$ ......................................................................................................... 9
Figure 5: Earth's gravitational perturbation effect ........................................................................12
Figure 6: 1st order planar maneuver for a satellite ....................................................................... 17
Figure 7: Argument of perigee change due to ascending node angle maneuver ....................... 18
Figure 8: (a.) In plane leader-follower formation (b.) Out of plane relative circular orbit
       formation............................................................................................................................... 19
Figure 9: Double formation flying maneuver ............................................................................ 20
Figure 10: Out of plane state equations singularities................................................................. 20
Figure 11: Out of plane maneuvering areas.................................................................................. 22
Figure 12: Out of plane maneuvering conditions for (a.) Positive thrust (b.)Negative thrust ...... 22
Figure 13: 1st order formation flying maneuvers ....................................................................... 23
Figure 14: Formation flying double maneuvers example ............................................................. 26
Figure 15: Out of plane relative position of chief satellite from deputy satellite's orbital plane.. 36
Figure 16: (a.)Radial projection of chief satellite on deputy satellite's orbital plane (b.) Desired
           relative distance projection on deputy satellite's orbital plane. ....................................... 36
Figure 17: Desired relative angular difference projection on deputy satellite's orbital plane. ..... 37
Figure 18: Gain changes due to error............................................................................................ 40
Figure 19: Desired relative difference constraint equation........................................................... 41
Figure 20: Newton-Raphson and Runge-Kutta combination maneuvering flow diagram. ....... 46
Figure 21: Orbit of two satellites without perturbations. .......................................................... 48
Figure 22: Orbits of two satellites with J2 and drag perturbations. .......................................... 48
Figure 23: Relative position of chief satellite on deputy satellite ........................................... 49
Figure 24: Relative position due to (a.) Decreasing eccentricity (b.) Increasing eccentricity ...... 50
Figure 25: Relative position of satellite due to true anomaly difference ................................... 51
Figure 26: Relative position of satellite due to inclination difference ....................................... 52
Figure 27: Relative position of satellite due to argument of perigee difference ......................... 53
Figure 28: Relative position of satellite due to ascending node difference ............................... 54
Figure 29: Effects of perturbations with different eccentricities. ............................................. 55
Figure 30: Effects of perturbations after (a.) 2 orbits (b.) 5 orbits and (c.) 10 orbits. ............... 55
Figure 31: Perturbation drift in true anomaly, argument of perigee and planar angle ............... 56
Figure 32: Ascending node drift due to perturbations ............................................................... 56
Figure 33: Perturbation drift on orbits with eccentricities of 0.01, 0.1 and 0.2 on (a.) radial velocity (b.) perpendicular velocity and (c.) out of plane velocity ........................................ 57
Figure 34: Total velocity perturbation drift on orbits with eccentricities of 0.01, 0.1 and 0.2 .... 57
Figure 35: Orbit following simulation results chief satellite. .................................................... 58
Figure 36: Orbit following (a.) Energy error (b.) Radial distance error .................................... 59
Figure 37: Orbit following error for (a.) Inclination (b.) Ascending node ................................ 59
Figure 38: Orbit following thrust, perturbation and acceleration in (a.) Radial direction (b.) Perpendicular direction (c.) Out of plane direction ......................................................... 60
Figure 39: 0.1 eccentricity orbit (a.) Eccentricity (b.) Argument of perigee error ...................... 61
Figure 40: 0.1 eccentricity orbit thrust, perturbation and acceleration in (a.) Radial direction (b.) Perpendicular direction (c.) Out of plane direction. ............................................................. 62

Figure 41: Close-up of (a.) Figure 40a (b.) Figure 40b. ........................................................................ 62

Figure 42: Orbit raising maneuver 300km to 1000km .................................................................. 63

Figure 43: Radial velocity of orbit raising maneuver. ................................................................. 63

Figure 44: Orbit raising maneuver (a.) Radial position (b.) Eccentricity. ........................................ 64

Figure 45: Orbit raising maneuver thrust, perturbation and acceleration in (a.) Radial (b.) Perpendicular (c.) Out of Plane direction. ................................................................. 65

Figure 46: Inclination only change maneuver (a.) inclination error (b.) Orbital path. ............. 65

Figure 47: Inclination only change maneuver (a.) Argument of perigee (b.) Ascending node error ................................................................. 66

Figure 48: Inclination only change maneuver thrust, perturbation and acceleration in (a.) Radial (b.) Perpendicular (c.) Out of plane direction ................................................................. 67

Figure 49: Ascending node only change maneuver (a.) relative position from trajectory (b.) Orbital path. .................................................................................................................. 67

Figure 50: Ascending node only change maneuver (a.) Inclination (b.) Ascending node error... 68

Figure 51: Ascending node only change maneuver thrust, perturbation and acceleration in (a.) Radial (b.) Perpendicular (c.) Out of plane direction ................................................................. 69

Figure 52: Singularity effect on relative distance .............................................................................. 70

Figure 53: Dampened singularity effect on relative distance .......................................................... 70

Figure 54: Formation flying radial thrust of (a.) Chief (b.) Deputy ................................................. 71

Figure 55: Formation flying perpendicular thrust of (a.) Chief (b.) Deputy ..................................... 71

Figure 56: Formation flying out of plane thrust of (a.) Chief (b.) Deputy. ........................................ 72
Figure 57: Relative graph of Deputy satellite's motion in IJK coordinates as seen from Chief Satellite. ................................................................................................................................ 73

Figure 58: Relative distance graph of formation flying satellites................................................. 73

Figure 59: Radial, Perpendicular and Out of Plane thrust, perturbation and acceleration for Chief and Deputy Satellites ............................................................................................................ 74

Figure 60: Relative graph of Deputy satellite's motion in IJK coordinates as seen from Chief Satellite. ................................................................................................................................ 75

Figure 61: Relative distance graph of formation flying satellites................................................. 75

Figure 62: Radial, Perpendicular and Out of Plane thrust, perturbation and acceleration for Chief and Deputy Satellites ............................................................................................................ 76

Figure 63: Relative graph of Deputy satellite's motion in IJK coordinates as seen from Chief Satellite. ................................................................................................................................ 77

Figure 64: Relative distance graph of formation flying satellites................................................. 77

Figure 65: Radial, Perpendicular and Out of Plane thrust, perturbation and acceleration for Chief and Deputy Satellites ............................................................................................................ 78

Figure 66: Relative graph of Deputy satellite's motion in IJK coordinates as seen from Chief Satellite. ................................................................................................................................ 79

Figure 67: Relative distance graph of formation flying satellites................................................. 79

Figure 68: Radial, Perpendicular and Out of Plane thrust, perturbation and acceleration for Chief and Deputy Satellites ............................................................................................................ 80

Figure QQQ: Basic 2-Body problem ............................................................................................84

Figure RRR: Angular Momentum Vectors ................................................................................... 90
LIST OF TABLES

Table 1: Program’s density output versus the 1976 U.S. Standard Atmosphere.comparison ..... 47
LIST OF ACRONYMS/ABBREVIATIONS/SYMBOLS

r  Radial distance from satellite to center of Earth
V  Velocity of satellite
V_r  Velocity of satellite in radial direction
V_s  Velocity of satellite in plane, perpendicular to radial direction
V_w  Velocity of satellite perpendicular to orbital plane
\mu  Earth’s Gravitational Parameter, 3.986 \times 10^5 \text{ km}^3/\text{s}^2
G  Universal Gravitational Constant
\upsilon  True anomaly
\omega  Argument of perigee
i  Inclination angle
\Omega  Longitude of ascending node
e  Eccentricity
M_{\text{earth}}  Earth’s mass
M_{\text{sat}}  Satellite’s mass
R_{\text{earth}}  Earth’s radius
r_{\text{earthsat}}  Distance Earth to satellite
r_{\text{satearth}}  Distance satellite to Earth
F_g  Gravitational force
P_{\text{earth}}  Perturbations acting on the Earth
P_{\text{sat}}  Perturbations acting on the satellite
P  Total Perturbations
\( P_R \) Perturbations acting on the satellite in the radial direction

\( P_S \) Perturbations acting on the satellite perpendicular to radial direction in orbital plane

\( P_W \) Perturbations acting on the satellite perpendicular to the orbital plane

\( R \) or \( R_{J2} \) J2 perturbations in the radial direction

\( S \) or \( S_{J2} \) J2 perturbations perpendicular to radial direction in orbital plane

\( W \) or \( W_{J2} \) J2 perturbations perpendicular to the orbital plane

\( u \) Thrust control input

\( u_R \) Thrust control input in the radial direction

\( u_S \) Thrust control input perpendicular to radial direction in orbital plane

\( u_W \) Thrust control input perpendicular to the orbital plane

\( h \) Angular momentum

\( H \) Matrix of orbital moments

\( h_x \) Orbital moment in inertial x direction

\( h_y \) Orbital moment in inertial y direction

\( h_z \) Orbital moment in inertial z direction

\( C_d \) Drag coefficient of satellite

\( A \) Cross-sectional face area perpendicular to velocity direction

\( \rho \) Atmospheric density

\( \rho(k) \) Reference density used for estimating atmospheric density

\( r(k) \) Reference radial distance used for estimating atmospheric density

\( \text{Drag} \) Drag acceleration

\( \text{Drag}_R \) Drag Acceleration in the radial direction
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Drag}_S$</td>
<td>Drag Acceleration perpendicular to radial direction in orbital plane</td>
</tr>
<tr>
<td>$\text{Drag}_W$</td>
<td>Drag Acceleration perpendicular to the orbital plane</td>
</tr>
<tr>
<td>$\varphi_{\text{lat}}$</td>
<td>Latitude angle of satellite</td>
</tr>
<tr>
<td>$J$</td>
<td>Performance index</td>
</tr>
<tr>
<td>$J_{\text{max}}$</td>
<td>Desired performance index</td>
</tr>
<tr>
<td>$t$</td>
<td>Time</td>
</tr>
<tr>
<td>$w_t$</td>
<td>Calculation weight used as independent time constant in Taylor Series</td>
</tr>
<tr>
<td>$w_{t_0}$</td>
<td>Calculation weight of planar angle equation</td>
</tr>
<tr>
<td>$w_r$</td>
<td>Calculation weight of radial position equation</td>
</tr>
<tr>
<td>$w_{E}$</td>
<td>Calculation weight of energy equation</td>
</tr>
<tr>
<td>$w_{i}$</td>
<td>Calculation weight of inclination angle equation</td>
</tr>
<tr>
<td>$w_{\Omega}$</td>
<td>Calculation weight of ascending node angle equation</td>
</tr>
<tr>
<td>$w_{t_{\text{max}}}$</td>
<td>Calculation weight at convergence point equation</td>
</tr>
<tr>
<td>$K$</td>
<td>Gain</td>
</tr>
<tr>
<td>$K_{\theta}$</td>
<td>Gain of planar angle equation</td>
</tr>
<tr>
<td>$K_r$</td>
<td>Gain of radial position equation</td>
</tr>
<tr>
<td>$K_{E}$</td>
<td>Gain of energy equation</td>
</tr>
<tr>
<td>$K_i$</td>
<td>Gain of inclination angle equation</td>
</tr>
<tr>
<td>$K_{\Omega}$</td>
<td>Gain of ascending node angle equation</td>
</tr>
<tr>
<td>$K_{\text{max}}$</td>
<td>Gain at convergence point equation</td>
</tr>
<tr>
<td>$x$</td>
<td>Arbitrary state</td>
</tr>
<tr>
<td>$x_{\text{des}}$</td>
<td>Desired reference state</td>
</tr>
<tr>
<td>$X$</td>
<td>Matrix of states</td>
</tr>
</tbody>
</table>
f Partial derivative of state equations with respect to the unknown variables.

F Jacobian Matrix

ΔX Error of state

ΔX_{\text{max}} Error of state to activate maximum gain $K_{\text{max}}$

ΔR_{1\text{on}2} Radial position of chief satellite when transformed to the deputy RSW frame

ΔS_{1\text{on}2} Perpendicular position of chief satellite when transformed to the deputy RSW frame

ΔW_{1\text{on}2} Out of plane position of chief satellite when transformed to the deputy RSW frame

Δd_{\text{des}} Desired relative distance between deputy and chief

Δd_{\text{des\,on\,plane}} Desired relative distance on deputy plane

C_{2\text{JK},\text{R2S2W2}} Coordinate conversion matrix from inertial to deputy RSW frame

C_{1\text{JK},\text{R1S1W1}} Coordinate conversion matrix from inertial to chief RSW frame

C_{12\text{R1S1W1},\text{R2S2W2}} Coordinate conversion matrix from chief to deputy RSW frame

error Convergence error inside Newton-Raphson loop

r_{\text{deputy}} Radial distance of deputy satellite from Earth’s center

r_{\text{chief}} Radial distance of chief satellite from Earth’s center

r_{\text{chief\,on\,plane}} Projection of radial distance of chief satellite on deputy plane

α Camera angle from local vertical for each satellite
RSW  Body centered orbital coordinate frame where the R axis runs along the radial position vector, S is perpendicular to the R axis on the orbital plane, and W is orthogonal to the orbital plane pointing up.

IJK  Earth centered orbital coordinate frame where the I axis points towards the vernal equinox, the J axis is perpendicular to the I axis pointing east in the equatorial plane and the K axis points through the North pole and is orthogonal to the equatorial plane.

MET  Microwave Electro Thermal.

DMD  Distance Measuring Device.

GPS  Ground Positioning System
CHAPTER ONE: INTRODUCTION

Since the early 1960’s, space reconnaissance has played a vital role in homeland security, and intelligence gathering. Until now data gathering from space has been limited to just one satellite. Lately, the use of multiple satellites to gather information has been a topic of consideration within the defense community. Instead of sending one large satellite, a set of multiple satellites can act as one entity and gather the needed information. This reduces cost and structural wear on the satellite. The information gathered will be more accurate, depending on the number of satellites in the constellation.

The Air Force has been interested in such a technology and has granted the University of Central Florida funds for further development. These funds will be used to develop a constellation of two satellites flying in formation to take stereo images of ground terrain. This has become known as the KnightSat project. Its primary objectives are:

- Demonstrate maintenance of a formation of two satellites.
- Demonstrate ability to coordinate formation flight maneuvers.
- Perform proper attitude control maneuvers to perform secondary mission.

Its secondary objectives are

- Construct and fabricate a three dimensional ground imagining system using the data from two satellites in formation that are imaging the same area.
- Collect and process individual images from each satellite in the ground station to produce a three dimensional topological ground map.
- Perform analysis of the three dimensional images to recognize structures and objects within the image.
The two satellites are to be deployed at the same time in Low Earth Orbit. They are to perform detumbling maneuvers and come to a stable orientation. One satellite is to maneuver itself away until it reaches a desired relative distance. Once this occurs, the constellation will orient itself to accurately point at a ground target and take pictures of the terrain as shown in Figure 1.

![Diagram of satellite constellation](image)

**Figure 1: Knightsat Objective**

Each satellite’s onboard computers will be programmed to autonomously maneuver themselves to maintain their desired attitude and distance from each other. Their positions are determined by distance measuring devices (DMD) and GPS. The DMDs determine their relative distance from each other, and GPS determines their position with respect to the Earth and verifies the distance measured between the satellites. This data is analyzed and averaged to determine a more...
accurate prediction of their actual position. Meanwhile, the satellites are to maintain a constant distance apart while flying relatively close to the same orbital plane.

**Statement of Work**

My work in the KnightSat project is to develop the algorithms that will be used to determine the thrust required by the MET to maintain the formation parameters of each satellite in its desired orbital state. By manipulating the Keplerian Equations of motion, six state equations are derived to simulate the orbital motion of the satellites in 3 dimensions. From these orbital equations, the algorithms are derived. Several methods of control are developed to demonstrate the capabilities of autonomous control algorithm to maintain an optimal separation and orientation of the formation.
CHAPTER TWO: LITERATURE REVIEW

David A. Vallado et al [1] writes a very useful “handyman’s” book to Orbital Dynamics. It covers, in detail, the principles of Orbiting bodies and their resulting equations. Below are the equations of motion in the orbital plane.

\[
\frac{dr}{dt} = V_R \\
\frac{dV_R}{dt} = \frac{V_S^2}{r} - \frac{\mu}{r^2} + P_R + U_R \\
\frac{dV_S}{dt} = \frac{V_R \cdot V_S}{r} + P_S + U_S \\
\frac{dv}{dt} = \frac{V_S}{r}
\]

I reached an understanding of how the Earth’s distorted shape perturbs the satellite’s motion, and other perturbation forces, through chapters 8 and 9. Vallado’s book only gave the equations in the inertial frame, which required a long coordinate conversion matrix to transfer it to the body centered RSW frame.

The equations in section 4.7 of Howard D. Curtis et al [2] were used for perturbation effects in the RSW frame.

\[
F_{J2}^{RSW} = \begin{pmatrix}
R \\
S \\
W
\end{pmatrix} = \begin{pmatrix}
\frac{-3 \mu \cdot R_{\text{earth}}}{2 \cdot r^4} \left[ 1 - 3 \left( \sin(i) \cdot \sin(\omega + v) \right)^2 \right] \\
\frac{-3 \mu \cdot R_{\text{earth}}^2 \cdot J_2}{r^4} \left( \sin(i) \cdot \sin^2(\omega + v) \cdot \cos(\omega + v) \right) \\
\frac{-3 \mu \cdot R_{\text{earth}}^2 \cdot J_2}{r^4} \left( \sin(i) \cdot \sin(\omega + v) \cdot \cos(i) \right)
\end{pmatrix}
\]

The rate of change of orbital angles in the presence of perturbations were derived in detail in the same Chapter. These are obtained from the rates of change of the angular momentum vector as will be described in Chapter 3. Below are the orbital angle state equations used for simulation and control.

\[
\frac{d\omega}{dt} = \sqrt{\frac{\mu}{\left(1 + e \cdot \cos(v)\right)}} \left[ \frac{-\cos(v)}{e} \left( p_R + u_R \right) + \frac{\sin(v) \cdot (2 + e \cdot \cos(v))}{e \cdot (1 + e \cdot \cos(v))} \left( p_S + u_S \right) + \frac{\sin(\omega + v)}{\tan(i) \cdot (1 + e \cdot \cos(v))} \left( p_W + u_W \right) \right]
\]

\[
\frac{d\Omega}{dt} = \sqrt{\frac{\mu}{\left(1 + e \cdot \cos(v)\right)}} \left[ \frac{\sin(\omega + v)}{\sin(i) \cdot (1 + e \cdot \cos(v))} \left( p_W + u_W \right) \right]
\]

\[
\frac{di}{dt} = \sqrt{\frac{\mu}{\left(1 + e \cdot \cos(v)\right)}} \cos(\omega + v) \left( p_W + u_W \right)
\]

For different controls theory and criterion, Donald E. Kirk’s Optimal Controls Theory et al [4] book was a guide for determining what kind of controls algorithm to develop. His description of performance indexes was the basis for determining the derivation of the desired thrust output. As shown in Chapter 4, the second order Taylor Series is used as the performance index to develop the controls equations.


\[
V_S = r \cdot \frac{d^2 \theta}{dt^2} + 2 \cdot \frac{dr}{dt} \cdot \frac{d\theta}{dt}
\]

This equation is used in the algorithm to control the planar angle.

The text book et al [6] was provided by my advisor, Dr. Roger W. Johnson at the University of Central Florida. This was the basis for simulation and control of the algorithms in the plane of orbit.
The state equations in the text \[6\] are very similar to the ones used from Vallado \[1\]. The derivations in the text \[6\] were used to determine their derivation of the equations (see appendix). This was an excellent guide book for simulation, guidance and controls as it shows good examples for application.

Terrence J. Akai’s book et al \[7\] analyzes different iteration methods for solving linear and non-linear systems of equations. In Chapter 7.1.3 he describes the Classical Runge-Kutta Fourth Order Iteration Method, which is shown below.

\[
k_1 = f(x_i, \eta_i)
\]

\[
k_2 = f\left(x_i + \frac{h}{2}, \eta_i + \frac{h \cdot k_1}{2}\right)
\]

\[
k_3 = f\left(x_i + \frac{h}{2}, \eta_i + \frac{h \cdot k_2}{2}\right)
\]

\[
k_4 = f(x_i + h, \eta_i + h \cdot k_3)
\]

\[
\eta_{i+1} = \eta_i + \left(\frac{h}{6}\right)[k_1 + 2(k_2 + k_3) + k_4]
\]

This was the iteration method used inside a conditional “while” loop to simulate the equations of motion, and their orbital response.

In Chapter 3.2 Akai describes the Newton-Raphson Non-Linear System of Equations Solver, shown below.

\[
X_{n+1} = X_n - \left(\frac{dF(X_{i+1}, U, t)}{dX}\right)^{-1} \cdot F(X_{i+1}, U, t)
\]

This method is presented in Chapter 4 in conjunction with the Classical Runge-Kutta Method, to develop a controls algorithm for determining the desired thrust output.
CHAPTER THREE: METHODOLOGY

The goal of formation flying is to maintain a desired relative state between orbiting satellites. If there is a difference between the actual and desired relative states then maneuvers need to be performed. Using thrusters, the satellites can compensate for these errors between them. In order to control the satellites a full, intuitive understanding of their orbital motion is needed. Then, the maneuvers can be visualized and simulated using iterative methods. Using gravitational equations, and applying them to orbital parameters, we can simulate the satellite’s motion.

To initialize the simulations at least six orbital parameters and their resultant time rate of change need to be described. All other parameters can be obtained from these values. To start off, it is sometimes good to do a 2 dimensional orbital analysis in order to see the orbit of a satellite more clearly. To describe orbital motion you need four orbital parameters. The radial position $r$, the perpendicular velocity $V_R$, the radial velocity $V_S$ and the true anomaly $\nu$, are the orbital parameters chosen. These parameters are easy to visualize/manipulate to determine the reaction of the orbital elements as shown in Figure 2. This makes it easier to debug the equations when running simulations. The initialization parameters can differ from the simulation parameters. The true anomaly $\nu$, and the radial position $r$ provide starting coordinates, and the eccentricity $e$, initializes the trajectory.

To describe the orbital position in 3 dimensions, the three angles that describe their position of the orbit are added. These are the longitudinal node angle $\Omega$, the argument of perigee $\omega$, and the inclination angle $i$. To describe the angular position of the orbit, these elements can also be substituted by angular moment vectors. Figure 3 shows the 3 dimensional angles of the orbit.
Figure 2: 2-D Orbital Parameters

Figure 3: Position Orbital Parameters
The Planar motion of the satellite describes the position of the satellite in the plane of orbit. Planar motion differs from orbital motion in using the ascending node as the x-axis, and includes the argument of perigee. The angular position of the satellite is the summation of the true anomaly and the argument of perigee, or $\theta$, as shown in Figure 4.

![Figure 4: Planar angle $\theta$](image)

Combining these things together, the satellite’s orbital position can be obtained and its motion in time can be described. To initialize the orbital simulations, an initial radial position, angular position, and eccentricity are provided as independent variables. The radial distance and planar angle provide an initial position, while the eccentricity is used as a guideline for future desired states. From these the satellite’s initial 4 orbital states can be determined. Using the argument of perigee, inclination angle and ascending longitudinal node, the satellite’s parameters in 3 dimensions can be converted using the coordinate conversion matrix below et al [1].
\[
\begin{pmatrix}
\cos(\omega + v)\cos(\Omega) - \sin(\omega + v)\sin(\Omega)\cos(i) & \cos(\omega + v)\sin(\Omega) + \sin(\omega + v)\cos(\Omega)\cos(i) & \sin(\omega + v)\sin(i) \\
-\sin(\omega + v)\cos(\Omega) - \cos(\omega + v)\sin(\Omega)\cos(i) & -\sin(\omega + v)\sin(\Omega) + \cos(\omega + v)\cos(\Omega)\cos(i) & \cos(\omega + v)\sin(i) \\
\sin(\Omega)\sin(i) & -\cos(\Omega)\sin(i) & \cos(i)
\end{pmatrix}
\]

There are many possible combinations of state equations that can be used. The equations are derived from the basic 2 body problem et al [6]. The derivations are described in Appendix A. The final equation of orbital motion is.

\[
\vec{r}'' + \mu \frac{\vec{r}}{r^3} = \vec{P} + \vec{u}
\]

From this relation is where the state equations of control in the plane of orbit are obtained. The goal is to obtain the time rate of change of the 6 orbital elements of the satellite. These are the radial distance \( r \), the radial velocity \( V_R \), the perpendicular velocity \( V_S \), the planar angular position \( \theta \), the ascending node \( \Omega \), and the inclination angle \( i \). The derivations of the 4 state equations of the orbital parameters in the plane of orbit are demonstrated in Appendix B. The resultant equations are.

\[
\frac{dr}{dt} = V_R
\]
\[
\frac{dV_R}{dt} = \frac{V_S^2}{r} - \frac{\mu}{r^2} + \vec{P}_R + \vec{U}_R
\]
\[
\frac{dV_S}{dt} = -\frac{V_R \cdot V_S}{r} + \vec{P}_S + \vec{U}_S
\]
\[
\frac{d\nu}{dt} = \frac{V_S}{r}
\]

The ascending node and inclination angles can be obtained using the angular momentum and angular momentum time rate of change vectors et al [3]. The argument of perigee is more complicated to obtain since it is not directly obtained geometrically from the angular momentum vector. The derivations of the inclination, ascending node and argument of perigee state equations are demonstrated in Appendix C.
The resultant equations are.

\[
\frac{d \omega}{dt} = \frac{\cos(\omega + v)\cdot(P_W + U_W)}{V_S} \quad \text{or} \quad \frac{d \omega}{dt} = \frac{\frac{r}{\sqrt{\mu(1 + e\cos(v))}} \cdot \cos(\omega + v)\cdot(P_W + U_W)}{}
\]

\[
\frac{d \Omega}{dt} = \left[\frac{\sin(\omega + v)}{V_S \sin(i)}\right] \cdot \left(P_W + U_W\right) \quad \text{or} \quad \frac{d \Omega}{dt} = \frac{\sqrt{\frac{r(1 + e\cos(v))}{\mu}} \cdot \frac{\sin(\omega + v)}{\sin(i)(1 + e\cos(v))}\cdot(P_W + U_W)}{}
\]

\[
\frac{d \omega}{dt} = \frac{\sqrt{\frac{r(1 + e\cos(v))}{\mu}} \cdot \left[\frac{-\cos(v)}{e} \cdot (P_R + U_R) + \frac{\sin(v)(2 + e\cos(v))}{e(1 + e\cos(v))}\cdot(P_S + U_S) + \frac{\sin(\omega + v)}{\tan(i)(1 + e\cos(v))}\cdot(P_W + U_W)\right]}{}
\]

These angular rates of change are only affected by perturbation or thrust forces acting on the satellite. In order for the simulations to be accurate, some natural perturbations have to be accounted for. The greatest perturbation is due to the oblateness of the Earth, otherwise known as the J2 term. This perturbation is due to the difference between the Earth’s center of gravity and geometric center of mass due to the distance between the poles being longer than the equatorial diameter. Since the gravitational force vector acting on the satellite points slightly differently than its geometric center, it causes its orbit to drift. This is due to the satellite’s reaction to the angular change in angular momentum due to the oblated Earth. Figure 5 shows an illustration demonstrating this effect visually.
Without this perturbation, the satellite’s angular velocity vector would remain in the same position. There are other terms accounting for differences in the Earth’s mass distribution, but have far less of an effect than the J2 term, the closest one being the J3 term with a magnitude of $10^{-3}$. The perturbation is accounted for in the state equations as an extra acceleration term et al [1]. The derivations of the perturbation accelerations in the RSW frame are derived in Appendix D. Below are the resultant equations.
Yet another perturbation accounted for is the atmospheric drag. When orbiting around the Earth, there is a thin atmosphere which slows a satellite’s velocity. The atmospheric density decreases as the satellite’s orbit increases. Below is the equation for the force of drag exerted on the satellite et al [1], [3].

\[ \text{Drag} = \frac{C_d \rho \cdot A \cdot V^2}{2m} \]

Where \( C_d \) is the ballistic drag coefficient determined by the shape of the satellite, \( \rho \) is the density of the environment, \( A \) is the cross-sectional area of the space craft aligned along the velocity vector, and \( m \) is the mass of the satellite. The density decrease with respect to height needs to be taken into account in the drag calculations. For this reason an interpolation model of equations from actual data gathered from the 1976 U.S. Standard Atmosphere was developed to calculate the surrounding density from the radial distance. Since air density changes at varying rates, depending on the section of the atmosphere, different interpolation equations are used dependant on height. By using equation

\[ \rho = \rho(k) \cdot e^{-a \left(\frac{r(k) - r(k)}{\pi(k) - \text{R}_{\text{earth}}}\right)^c} \]

where \( \rho(k) \) is a reference density taken from actual data, \( r(k) \) is the reference radial distance, and \( a \) and \( c \) are interpolation constants dependant on height. A separate program was written specifically for the task of calculating the surrounding density. Different values of \( a \), \( c \) and \( r(k) \) are chosen, depending on what section of the atmosphere the satellite is in. Starting from a height of 100 km, the equations
used are divided into 100 km segments. The program automatically stops if the satellite gets closer than 100 km from the surface of the Earth. The coefficients remain the same for heights greater than 800 km above the surface. The coefficients were determined using an excel file to change their values by hand. The results of the equation were compared to actual density data from the U.S. Standard 1976 Atmospheric Model, until their difference was within a reasonable parameter. You can see the comparison in Chapter 5.

Like the J2 perturbation, the drag effects can also be considered as an acceleration acting in the opposite direction of velocity. In order to apply it to the state equations, the radial, perpendicular and out of plane components need to be accounted for. The derivations of these are demonstrated in Appendix E. The resultant drag perturbations in the RSW frame are.

\[
\begin{align*}
\text{Drag}_R &= \frac{C_d \rho \cdot A}{2m} \sqrt{V_R^2 + V_S^2 + V_W^2} \cdot V_R \\
\text{Drag}_S &= \frac{C_d \rho \cdot A}{2m} \sqrt{V_R^2 + V_S^2 + V_W^2} \cdot V_S \\
\text{Drag}_W &= \frac{C_d \rho \cdot A}{2m} \sqrt{V_R^2 + V_S^2 + V_W^2} \cdot V_W
\end{align*}
\]

To finalize the effects of the perturbations in the equations, simply sum their components et al [1]

\[
\begin{align*}
P_R &= R_{J2} + \text{Drag}_R \\
P_S &= S_{J2} + \text{Drag}_S \\
P_W &= W_{J2} + \text{Drag}_W
\end{align*}
\]

And the following equations are obtained for the perturbation.

\[
\begin{align*}
P_R &= \frac{-3 \mu M_{\text{Earth}}^2 J_2}{2r^4} \left[ 1 - 3 \left( \sin(i) \sin(\omega + v) \right) \right] + \frac{C_d \rho \cdot A}{2m} \sqrt{V_R^2 + V_S^2 + V_W^2} \cdot V_R \\
P_S &= \frac{-3 \mu M_{\text{Earth}}^2 J_2}{r^4} \left( \sin(i) \sin(\omega + v) \cdot \cos(i) \right) + \frac{C_d \rho \cdot A}{2m} \sqrt{V_R^2 + V_S^2 + V_W^2} \cdot V_S \\
P_W &= \frac{-3 \mu M_{\text{Earth}}^2 J_2}{r^4} \left( \sin(i) \sin(\omega + v) \cdot \cos(i) \right) + \frac{C_d \rho \cdot A}{2m} \sqrt{V_R^2 + V_S^2 + V_W^2} \cdot V_W
\end{align*}
\]
These are the perturbations in the radial, perpendicular and out of orbital plane direction.

There are other perturbations that affect the satellite, but are not taken into account since they don’t make enough of a difference for such a short mission.

All the components necessary to simulate the satellite’s orbital motion are in place. In actuality, there are 7 orbital parameters to describe a satellite’s orbit. The argument of perigee and true anomaly combine into $\theta$, making it 6 orbital parameters. The state equations will give you the rates of change of the orbital parameters per time.

$$\frac{d\mathbf{r}}{dt} = \mathbf{V}_R$$

$$\frac{d\mathbf{V}_R}{dt} = \frac{\mathbf{V}_S^2}{r} - \frac{\mu}{r^2} + \mathbf{P}_R + \mathbf{U}_R$$

$$\frac{d\mathbf{V}_S}{dt} = \frac{\mathbf{V}_R \cdot \mathbf{V}_S}{r} + \mathbf{P}_S + \mathbf{U}_S$$

$$\frac{d\nu}{dt} = \frac{\mathbf{V}_S}{r} - \frac{\mathbf{P}_S}{r} + \mathbf{U}_S$$

$$\frac{d\omega}{dt} = \frac{\mathbf{P}_W + \mathbf{U}_W}{\mu} \cdot \frac{\sin(\omega + v)}{\sin(i) \cdot (1 + e \cdot \cos(v))}$$

$$\frac{d\Omega}{dt} = \frac{\mathbf{P}_W + \mathbf{U}_W}{\mu} \cdot \frac{\cos(\omega + v)}{\left(1 + e \cdot \cos(v)\right)}$$

$$\frac{di}{dt} = \frac{\mathbf{P}_W + \mathbf{U}_W}{\mu \cdot (1 + e \cdot \cos(v))} \cdot \cos(i)$$

Since the satellite’s time derivatives change depending on orbital factors, a non-linear iteration process needs to be used to accurately simulate the satellite’s motion. A Fourth Order Runge Kutta Iteration process et al [7] accurately simulates the orbital motions. The result of these simulations are demonstrated in Chapter 5.
CHAPTER FOUR: MANEUVERS

The Earth’s oblateness and drag are natural perturbations that slightly change the orbit in time. However, some artificial perturbations can be applied to change the orbit to a desired state. These artificial perturbations, or the thrust component $u$, are applied by the thrusters to modify the satellite’s orbit. In the state equations, the thrust component is always added to the perturbations in their respective direction. To control the satellite in translation, three different thrusts are applied in the radial, perpendicular and out of plane direction. Orbital maneuvers can be classified as planar and non-coplanar. Planar maneuvers manipulate the first four state equations by thrusting in the radial and perpendicular direction. Non-coplanar maneuvers manipulate the inclination and ascending node by thrusting in the out of plane direction. Out of plane maneuvers don’t affect the planar orbit, but they do affect the position of the orbit within the orbital plane. Since the satellite’s planar position is affected by the argument of perigee, an out of plane maneuver affects the position of the perigee. The ascending node and inclination angle are only affected by out of plane thrusting. Planar and non-coplanar maneuvers are very different in their approach and must be dealt with as separate entities.

Planar maneuvers occur solely in the radial and perpendicular direction. The state equations are used to solve for the necessary $\Delta V$’s to place the satellites in their perspective planar orbital position. Figure 6 shows an example of a simple first order approximation of planar orbital maneuvering.
Figure 6: 1st order planar maneuver for a satellite

\( \frac{r_{\text{des}} - r}{\Delta t} = V_{R\text{des}} \)

\( V_{R\text{des}} - V_R - a_{gR} = \Delta V_R \)

\( \frac{r_{\text{des}}(\theta_{\text{des}} - \theta)}{\Delta t} = V_{S\text{des}} \)

\( V_{S\text{des}} - V_S - a_{gS} = \Delta V_S \)

\( V_{R\text{des}}, V_{S\text{des}}, r_{\text{des}} \) and \( \theta_{\text{des}} \) are the desired radial velocity, perpendicular velocity, radial position and angular position at the end of the maneuver. The \( a_{gR} \) and \( a_{gS} \) are the gravitational and perturbation velocity changes in the radial and perpendicular direction. Solving for \( \Delta V_S \) and \( \Delta V_R \) will result in a first order estimation of the change in the radial and perpendicular velocities at a given time, \( \Delta t \) to reach the desired orbital position and state. This is not a very efficient way of solving for the velocity change, but can give a good quick estimate for the necessary velocity change required for the maneuver if the error is small. The smaller the time step per maneuver, the smaller the error in the total maneuvering. Each successive time step updates the actual values, making adjustments to the \( \Delta V \)’s to close in on the final desired states. It usually involves making two burns, one to speed the satellite up and the other to slow it down. Since applying a thrust in the perpendicular direction either increases or decreases the velocity, it also changes the energy and orbital parameters of the orbit. To
compensate for this effect, a thrust in the radial direction must be applied to maintain the satellite from being raised or lowered. The energy equation is.

\[ E = \frac{V_R^2 + V_S^2}{2} - \frac{\mu}{r} \]

In orbital maneuvering the true angular position of each satellite is the summation of the argument of perigee and true anomaly. Since the rate of change of the argument of perigee is affected by the ascending node rate of change, the out of plane component of thrust affects the planar angular position of the satellites. This is due to the drift in the ascending node due to the Earth’s oblateness perturbation. This decreases the argument of perigee as shown below.

Figure 7: Argument of perigee change due to ascending node angle maneuver

The perpendicular and radial components of thrust cause a change in the perigee position hence the angular positions of the satellites are affected by all components of the thrust. There are several ways
to make these maneuvers in compliance with their orbital motion for formation flying. One way has a chief satellite maintaining a trajectory, and its purpose is to superimpose the perturbations to remain in the desired orbit. A deputy satellite would follow the chief satellite at a constrained desired distance. The deputy satellite can either remain in the same orbit or go around in and out of phase. Figure 8 depicts the constellation.

![Same Orbit](image1)

![Different Orbits](image2)

Figure 8: (a.) In plane leader-follower formation (b.) Out of plane relative circular orbit formation

Since corrective maneuvers are constantly performed for in plane and out of plane thrusting, the relative orbit of both satellites will look more like a mixture between the two constellations shown above. Another way of making planar maneuvers is to thrust both satellites at the same time. The center of the constellation can follow a trajectory, or the constellation can just maintain constant distance from each other without worrying about following a trajectory. This is the most efficient way of maneuvering for formation flying since both satellites use the same amount of fuel in every maneuver. This will assure that one satellite won’t run out of fuel before the other. Below is a visual example of a double maneuver.
Non-coplanar maneuvers are more direct and less systematic than planar maneuvers since there’s only one thrust direction that affects it. Because of the sines and cosines in the inclination and ascending node rates of change equations, there are optimum maneuvering places as shown below.

\[
\frac{d\Omega}{dt} = \frac{\sqrt{r \cdot (1 + e \cdot \cos(v))}}{\mu} \cdot \sin(i) \cdot \sin(\omega + \nu) \cdot (P_W + U_W)
\]

\[
\frac{di}{dt} = \frac{\sqrt{r}}{\mu \cdot (1 + e \cdot \cos(v))} \cdot \cos(\omega + \nu) \cdot (P_W + U_W)
\]

Certain desired maneuvers are pointless in some areas of the orbit since they won’t have much of an affect. These areas change places for ascending node and inclination change maneuvers. There are
also areas where a desired maneuver to change one angle might have a negative effect on the other angle. One thrust direction is used to change two angular components. The inclination angle also has an affect on the rate of change of the ascending node. The closer the inclination is to zero, the higher the angular velocity of the ascending node. There is a singularity at zero inclination, where the ascending node ceases to exist and there’s only a planar position at the equator. That is why there is no ascending node change at $i = 0$. The optimum maneuvering positions are at the nodes for inclination change, and where the summation of the true anomaly and argument of perigee are at $90^\circ$ and $270^\circ$ for ascending node change. On the other hand, there is no ascending node change at the nodes, and no inclination change where the summation of the argument of perigee and true anomaly are $90^\circ$ and $270^\circ$. This is due to the sine in the ascending node rate being at a max at $90^\circ$ and $270^\circ$ and zero at the nodes. Also, the cosine in the inclination rate is at a max at the nodes and is zero at $90^\circ$ and $270^\circ$. Since fuel usage is maximized in these spots, orbital maneuvers will be primarily performed close to the optimum maneuvering positions. Thrusting will not occur at the exact maneuvering spot, which may have an undesirable affect on the opposite orbital angle. The effect is so small, it can be easily compensated for at the other maneuvering point. Figure 11 is an illustration of where the maneuvering areas are in the orbit.

These are not the only spots to make an out of plane maneuver. Double maneuvers can be accomplished anywhere in the orbit if the desired changes for both angles are the result of applying the thrust in the same direction. If not in a maneuvering area, and the desired thrust for one angle has an undesirable effect on the other, then no maneuver is performed until the satellite reaches a maneuvering area. Figure 12 shows the areas and conditions for out of plane maneuvering.
Since the Earth’s oblateness and drag perturbations do not affect the inclination angle, only the ascending node needs to be constantly changed to counter the perturbations. Inclination angle will...
remain the same and only needs to be acted upon when desiring to change it. There are maneuvers intended to change only the inclination angle, only the ascending node, or both. Non-coplanar maneuvers can be classified into those three categories. Which category is chosen depends on their planar angular position. These elements alone tell you how and where to make your maneuvers. Deciding the final factors determines your thrust output and time of thrust. These constraints for formation flying will determine the desired final position of each satellite. If using the trajectory following chief satellite with the deputy maintaining at a relative distance maneuver, the final desired position of the chief will be determined by the set desired trajectory, while the final desired position for the deputy will be determined by the relative distance. Figure 13 shows a drawing depicting a chief trajectory following formation.

![Figure 13: 1st order formation flying maneuvers](image-url)
The known elements for the chief satellite are the initial states and the final desired states. The initial states are obtained from onboard instrumentation readings and the final states are predetermined calculations. The satellite’s future position due to perturbations is estimated, and the thrust necessary to maintain the desired trajectory is determined. Below are calculations for a first order estimate for thrust output.

<table>
<thead>
<tr>
<th>Radial Maneuvering</th>
<th>Perpendicular Maneuvering</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_{\text{traj}} = r + V_{R_{\text{traj}}} \cdot \Delta t$</td>
<td>$\theta_{\text{traj}} = \theta + \frac{d\theta_{\text{traj}}}{dt} \cdot \Delta t$</td>
</tr>
<tr>
<td>$V_{R_{\text{traj}}} = V_R + (a_{R_g} + a_{R_{\text{pert}}}) \cdot \Delta t$</td>
<td>$\frac{d\theta_{\text{traj}}}{dt} = \frac{\theta_{\text{traj}} - \theta}{\Delta t}$</td>
</tr>
<tr>
<td>$V_{R_{\text{pert}}} = V_{R_{\text{per}}} + u_R \cdot \Delta t$</td>
<td>$V_{S_{\text{per}}} = V_S + (a_{S_{g}} + a_{S_{\text{pert}}}) \cdot \Delta t$</td>
</tr>
<tr>
<td></td>
<td>$V_{S_{\text{pert}}} = V_{S_{\text{per}}} + u_S \cdot \Delta t$</td>
</tr>
<tr>
<td></td>
<td>$\frac{d\omega_{\text{traj}}}{dt} = a_{\omega_R} \left( a_{R_{\text{pert}}} + u_R \right) + a_{\omega_S} \left( a_{S_{\text{pert}}} + u_S \right)$</td>
</tr>
<tr>
<td></td>
<td>$\frac{d\omega_{\text{traj}}}{dt} = \frac{V_{S_{\text{traj}}}}{r_{\text{traj}}} + \frac{d\omega_{\text{traj}}}{dt}$</td>
</tr>
</tbody>
</table>

Where $r$ and $r_{\text{traj}}$ are the present and desired radial positions, $V_R$ and $V_{R_{\text{traj}}}$ are the present and desired radial velocities, $V_S$ and $V_{S_{\text{traj}}}$ are the present and desired perpendicular velocities, $\theta$ and $\theta_{\text{traj}}$ are the present and desired planar angular positions and $\omega_{\text{traj}}$ is the desired argument of perigee. The radial and perpendicular accelerations due to gravity are $a_{R_g}$ and $a_{S_g}$. The radial and perpendicular components of the perturbation acceleration are $a_{R_{\text{pert}}}$ and $a_{S_{\text{pert}}}$, and $a_{\omega_R}$ and $a_{\omega_S}$ are orbital components to determine the effects of the perturbations to the argument of perigee. Therefore, to solve for the thrust and thrust time required to maintain the chief satellite in its desired trajectory, with a first order estimation, the equations are

$$ \frac{r_{\text{traj}} - r}{\Delta t} = V_R + \left( a_{R_g} + a_{R_{\text{pert}}} \right) \cdot \Delta t + u_R \cdot \Delta t $$

$$ \frac{\theta_{\text{traj}} - \theta}{\Delta t} = \frac{V_S + \left( a_{S_g} + a_{S_{\text{pert}}} \right) \cdot \Delta t + u_S \cdot \Delta t}{r_{\text{traj}}} + a_{\omega_R} \left( a_{R_{\text{pert}}} + u_R \right) + a_{\omega_S} \left( a_{S_{\text{pert}}} + u_S \right) $$
The same formulas are used to solve for the desired thrust and thrust time of the deputy satellite, except the desired future state is obtained from the estimated final position of the chief satellite. The derivation of the future desired states for the deputy vary depending on the mission requirements. Constraints are set forth to position the deputy satellite wherever desired.

The most desirable constraint is for minimum fuel usage per maneuver. This, however, is limited on the number of other constraints set forth for the mission. The less the mission’s constraints, the more favorable the minimum fuel equations will be. To determine the minimum fuel for the maneuver, take the time derivative of each equation, and set it equal to zero.

\[
\frac{\delta u_R}{dt} = 0 \quad \frac{\delta u_S}{dt} = 0
\]

Solving for the unknowns in the equation results in the minimum fuel per maneuver. The minimum fuel equations vary depending on other constraints used to determine the final desired positions. Double maneuvers open up the constraint equations to give a more favorable condition for minimum fuel usage. Both satellites thrust with respect to each other to overcome gravitational and perturbation forces and reach a desired state. Figure 14 is an example drawing of a double planar maneuver. A satellite’s thrust is pointed in the opposite direction of the other satellite to propel the satellites into their desired positions.

These estimated values are only of a first order magnitude and are not very accurate. Since the calculation error is proportional to the time step. If the time step is small enough the errors can be corrected in updating the thrust output after every time step. This kind of maneuver will eventually converge the satellites into their desired positions, but will waste extra fuel in making up for the errors.
The mission constraints determine what kind of algorithm to use to control the satellites. Using more accurate algorithms to solve for the outputs will save fuel in the long run. However, the calculations are more tedious and use up more computer power and memory to solve. In the following sections, two different control techniques for formation flying are developed and examined.
Second Order Taylor Series Approximation

The intention of the control system is to remove the error between the actual and desired states. A second order method can be constructed to more accurately reach a steady state, but still maintain simple derivations. The desired radial, perpendicular and out of plane positions can be manipulated by combining velocity and acceleration state equations for their corresponding state position. A second order performance index is developed to incur the state error from their rates in the form:

\[
J = \int_{t_0}^{t_f} \frac{dx}{dt} dt
\]

This is only in first order form, hence the acceleration can be included.

\[
\int_{t_0}^{t_f} \frac{d^2x}{dt^2} dt = \frac{dx}{dt} + C \quad C = -\frac{dx_0}{dt_0}
\]

\[
\frac{dx}{dt} = \int_{t_0}^{t_f} \frac{d^2x}{dt^2} dt + \frac{dx_0}{dt_0}
\]

And substitute in to make a second order equation

\[
J = \int_{t_0}^{t_f} \left( \int_{t_0}^{t_f} \frac{d^2x}{dt^2} dt + \frac{dx_0}{dt_0} \right) dt
\]

\[
J = \frac{dx_0}{dt_0} (t_f - t_0) + \int_{t_0}^{t_f} \int_{t_0}^{t_f} \frac{d^2x}{dt^2} dt dt
\]

The x is an orbital state vector, t is time and f and 0 denote the final and initial state within the time step. Calculating in the second order is important in determining the desired thrust since it is an acceleration. Constant acceleration is assumed within each calculation’s time step to simplify the
problem. This is not a correct assumption, but since the time span is small, the percent error will not be significant. The next calculation will maneuver to overcome the previous assumption error. Since constant acceleration is assumed throughout the time step, we’ll also assume the rates to be constant from their initial values.

\[ V_0 = \frac{dx_0}{dt_0}, \quad \frac{d^2x}{dt^2} = \frac{dV_0}{dt_0} \]

Each time step affects accuracy, hence time is viewed as a calculation weight \( wt \) and each calculation is started at time zero. Weight is an independent variable, and can be manipulated to enhance the calculation performance. This eliminates \( t_0 \), and the performance index becomes

\[ J = V_0 \cdot wt + \frac{dV_0 \cdot wt^2}{dt_0} \]

This gives an estimated indication of the difference in the orbital state after a certain time weight. In order to apply some kind of control to the system, there needs to be a desired performance index to converge to. The desired performance index is designed to eliminate the error between the actual and desired final orbital state, as shown below.

\[ J_{\text{des}} = x_{\text{des}} - x_0 \]

The value of the final position is desired to be less than the initial value. This is assured by relating the desired with the initial position by a gain \( K \).

\[ x_{\text{des}} = K \cdot x_0 \quad \text{where} \quad 0.0 < K < 1.0 \]

Therefore, the desired performance index becomes

\[ J_{\text{des}} = x_0(K - 1) \]

If the gain is below 1.0, this assures that the orbital state will decrease by a certain percentage of the initial value. This holds true as long as the gain \( K \) is within the stability limits of the accuracy of the second order equation. The gain \( K \) controls the rate of convergence of the orbital states. If \( K \) is too
close to zero then the spacecraft might overshoot its target, eventually settling into its steady state position or never settling at all. If it is too close to 1.0 then it might not be able to overcome the truncation error and never converge. The weight is another factor that determines the performance of the satellite’s convergence. If the weight is too small or too large, then it may not pair well with a gain $K$ to give a desired convergence. Running simulations with different pairs help determine the best suitable combination.

Combining the actual and desired performance indexes gives the controls equation below.

$$x_0(K - 1) = V_0 \cdot wt + \frac{dV_0}{dt_0} \cdot \frac{wt^2}{2}$$

In the case of out of plane orbital states, the thrust acceleration $u$ is in the first order angular velocity term. This simplifies the performance index to a first order equation which becomes.

$$x_0(K - 1) = V_0 \cdot wt$$

First order equations have a different convergence criterion. Running simulations helps determine the optimum convergence pair.

To apply this theory to control thrust, the orbital state equations are substituted into the performance index equation. The energy (Energy), radial position ($r$), planar angle ($\theta$), ascending node ($\Omega$) and argument of perigee ($\omega$) are used as the desired performance index. Out of plane maneuvering is determined by the planar orbital position, as indicated in the out of plane maneuvering section. Hence, the desired performance indexes are.
For Chief Satellite

### Ascending Node Maneuvers

\[
J_{\text{des}} = \begin{bmatrix}
\delta_{0} (K_{r} - 1) \\
\delta \text{Energy}_{0}(K_{0} - 1) \\
\delta \Omega_{0}(K_{\Omega} - 1)
\end{bmatrix}
\]

\[
J_{\text{des}} = \begin{bmatrix}
K_{r} - 1 & 0 & 0 \\
0 & K_{0} - 1 & 0 \\
0 & 0 & K_{\Omega} - 1
\end{bmatrix}
\begin{bmatrix}
\delta_{0} \\
\delta \text{Energy}_{0} \\
\delta \Omega_{0}
\end{bmatrix}
\]

### Inclination Maneuvers

\[
J_{\text{des}} = \begin{bmatrix}
\delta_{0} (K_{r} - 1) \\
\delta \text{Energy}_{0}(K_{0} - 1) \\
\delta_{0}(K_{i} - 1)
\end{bmatrix}
\]

\[
J_{\text{des}} = \begin{bmatrix}
K_{r} - 1 & 0 & 0 \\
0 & K_{0} - 1 & 0 \\
0 & 0 & K_{i} - 1
\end{bmatrix}
\begin{bmatrix}
\delta_{0} \\
\delta \text{Energy}_{0} \\
\delta_{0}
\end{bmatrix}
\]

For Deputy Satellite

### Ascending Node Maneuvers

\[
J_{\text{des}} = \begin{bmatrix}
\delta_{0} (K_{r} - 1) \\
\delta \theta_{0}(K_{0} - 1) \\
\delta \Omega_{0}(K_{\Omega} - 1)
\end{bmatrix}
\]

\[
J_{\text{des}} = \begin{bmatrix}
K_{r} - 1 & 0 & 0 \\
0 & K_{0} - 1 & 0 \\
0 & 0 & K_{\Omega} - 1
\end{bmatrix}
\begin{bmatrix}
\delta_{0} \\
\delta \theta_{0} \\
\delta \Omega_{0}
\end{bmatrix}
\]

### Inclination Maneuvers

\[
J_{\text{des}} = \begin{bmatrix}
\delta_{0} (K_{r} - 1) \\
\delta \theta_{0}(K_{0} - 1) \\
\delta_{0}(K_{i} - 1)
\end{bmatrix}
\]

\[
J_{\text{des}} = \begin{bmatrix}
K_{r} - 1 & 0 & 0 \\
0 & K_{0} - 1 & 0 \\
0 & 0 & K_{i} - 1
\end{bmatrix}
\begin{bmatrix}
\delta_{0} \\
\delta \theta_{0} \\
\delta_{0}
\end{bmatrix}
\]

The orbital states are applied to the actual performance index to extrapolate the desired thrusts. The angle \( \theta \) is the summation of the argument of perigee and the true anomaly (\( \nu + \omega \)). In the case of a chief satellite following a trajectory, the state equations applied are.

For Chief Satellite

### Ascending Node Maneuvers

\[
J = \begin{bmatrix}
\frac{\delta V_{R_{0}} \cdot wt_{r}}{\delta \text{Energy}_{0} \cdot wt_{0}} \\
\frac{\delta}{\delta dt_{0}} \cdot \frac{V_{R_{0}} \cdot wt_{l}^{2}}{2} \\
\frac{\delta \Omega_{0} \cdot wt_{l}}{\delta dt_{0}}
\end{bmatrix}
\]

### Inclination Maneuvers

\[
J = \begin{bmatrix}
\frac{\delta V_{R_{0}} \cdot wt_{r}}{\delta \text{Energy}_{0} \cdot wt_{0}} \\
\frac{\delta}{\delta dt_{0}} \cdot \frac{V_{R_{0}} \cdot wt_{l}^{2}}{2} \\
\frac{\delta \Omega_{0} \cdot wt_{l}}{\delta dt_{0}}
\end{bmatrix}
\]
For Deputy Satellite

Ascending Node Maneuvers

\[
J = \begin{pmatrix}
\delta V_R \cdot \omega_t \\
\delta \left( \frac{d\theta_0}{dt} \right) \cdot \omega_t0 \\
\delta \left( \frac{d\Omega_0}{dt} \right) \cdot \omega_{\Omega}
\end{pmatrix}
+ \begin{pmatrix}
\delta \left( \frac{V_R}{dt_0} \right) \cdot \omega_t^2 \\
\delta \left( \frac{d^2\theta_0}{dt_0^2} \right) \cdot \omega_t0^2 \\
\delta \left( \frac{d\Omega_0}{dt_0} \right) \cdot \omega_{\Omega}
\end{pmatrix}
\]

Inclination Maneuvers

\[
J = \begin{pmatrix}
\delta V_R \cdot \omega_t \\
\delta \left( \frac{d\theta_0}{dt} \right) \cdot \omega_t0 \\
\delta \left( \frac{d\Omega_0}{dt} \right) \cdot \omega_{\Omega}
\end{pmatrix}
+ \begin{pmatrix}
\delta \left( \frac{V_R}{dt_0} \right) \cdot \omega_t^2 \\
\delta \left( \frac{d^2\theta_0}{dt_0^2} \right) \cdot \omega_t0^2 \\
\delta \left( \frac{d\Omega_0}{dt_0} \right) \cdot \omega_{\Omega}
\end{pmatrix}
\]

Since the energy rate of change and the angular acceleration of the planar angle aren’t part of the state equations, they have to be derived separately. Below is the derivation of the rate of change of the energy equation.

\[
\text{Energy} = \frac{V_R^2}{2} + \frac{V_S^2}{2} - \frac{\mu}{r}
\]

\[
\frac{d\text{Energy}}{dt} = V_R \cdot \frac{dV_R}{dt} + V_S \cdot \frac{dV_S}{dt} + \mu \cdot \frac{dr}{dt}
\]

\[
\frac{d\text{Energy}}{dt} = V_R \left( \frac{V_S^2}{r} - \frac{\mu}{r^2} + P_R + U_R \right) + V_S \left( \frac{-V_R \cdot V_S}{r} + P_S + U_S \right) + \frac{\mu}{r^2} \cdot V_R
\]

\[
\frac{d\text{Energy}}{dt} = V_R \left( P_R + U_R \right) + V_S \left( P_S + U_S \right)
\]

Finding the angular acceleration of the planar angle requires a more intuitive approach. For cylindrical coordinates the acceleration vector et al [5] is.

\[
\frac{dV}{dt} = \left( \frac{\frac{d^2r}{dt^2}}{r} - \frac{\frac{d\theta}{dt}}{r^2} \right) \cdot e_r + \left( \frac{\frac{d^2\theta}{dt^2}}{r} + 2 \cdot \frac{dr}{dt} \cdot \frac{d\theta}{dt} \right) \cdot e_{\theta} + \left( \frac{\frac{dV_z}{dt}}{r} \right) \cdot e_z
\]

The acceleration in the \( \theta \) direction is the equivalent to the perpendicular velocity state equation. The perpendicular velocity state equation can be related to the angular acceleration by.
\[
\frac{dV_s}{dt} = r \frac{d^2 v}{dt^2} + 2\nu_{VR} \frac{dv}{dt}
\]
\[
\frac{d^2 v}{dt^2} = -\frac{3V_R \cdot V_S}{r^2} + \frac{P_S + U_S}{r}
\]
\[
\frac{d^2 V_s}{dt^2} - 2\nu_{VR} \frac{dv}{dt} = \left(\frac{-V_R \cdot V_S}{r} + P_S + U_S\right) - 2\nu_{VR} \left(\frac{V_s}{r}\right)
\]

Hence, the state equations used are.

\[
\frac{d\text{Energy}_0}{dt_0} = V_R \left(P_R + U_R\right) + V_S \left(P_S + U_S\right)
\]
\[
\frac{dr_0}{dt_0} = V_R
\]
\[
\frac{dV_{R0}}{dt_0} = \frac{V_s^2}{r} - \mu \frac{1}{r^2} + P_R + U_R
\]
\[
\frac{d\theta_0}{dt_0} = -\mu \frac{r^2}{e} \left(P_R + U_R + \frac{\sin(v)(2 + e \cdot \cos(v))}{e(1 + e \cdot \cos(v))}(P_S + U_S) + \frac{\sin(\omega + v)}{\tan(i)(1 + e \cdot \cos(v))}(P_W + U_W)\right)
\]
\[
\frac{d^2 V_0}{dt_0^2} = -\frac{3V_R \cdot V_S}{r^2} + \frac{P_S + U_S}{r}
\]
\[
\frac{d\Omega_0}{dt_0} = \frac{\cos(\omega + v) \cdot U_W}{V_S}
\]
\[
\frac{d\Omega_0}{dt_0} = \frac{\sin(\omega + v)}{V_S \sin(i)}(P_W + U_W)
\]

For the formation flying concept of a leader and follower satellite, the control algorithms used to each satellite are introduced. The chief satellite’s purpose is to maintain a constant orbit, hence its thrusters push to a desired radial distance while maintaining constant orbital energy. The radial distance is determined by the mission requirements, whether it is desired to maintain a circular or elliptical orbit. The out of plane maneuvers are the same as described before. The state equations used to maintain a constant orbit are.
\[
\begin{aligned}
\delta \left( \frac{dr_0}{dt_0} \right) &= V_{R_1} - V_{R_{\text{traj}}} \\
\delta \left( \frac{dV_R}{dt_0} \right) &= \frac{(V_S)^2}{r_1} - \frac{(V_{S_{\text{traj}}})^2}{r_{\text{traj}}} - \mu \left( \frac{1}{r_1^2} - \frac{1}{r_{\text{traj}}^2} \right) + P_{R_1} + U_{R_1} \\
\delta \left( \frac{d\text{Energy}_0}{dt_0} \right) &= V_R \cdot (P_{R_1} + U_{R_1}) + V_S \cdot (P_{S_1} + U_{S_1}) \\
\delta \left( \frac{d^2\nu_0}{dt_0^2} \right) &= 3 \frac{V_{R_{\text{traj}}} \cdot V_{S_{\text{traj}}}}{r_{\text{traj}}^2} - 3 \frac{V_{R_1} \cdot V_{S_1}}{r_1^2} + \frac{P_{S_1} + U_{S_1}}{r_1} \\
\delta \left( \frac{di_0}{dt_0} \right) &= \frac{\cos(\omega_1 + \nu_1) \cdot U_{W_1}}{V_{S_1}} \\
\delta \left( \frac{d\Omega_0}{dt_0} \right) &= \frac{\sin(\omega_1 + \nu_1)}{V_{S_1} \cdot \sin(i_1)} \left( P_{W_1} + U_{W_1} \right)
\end{aligned}
\]

The perpendicular velocity and the radial position of the trajectory \(V_{S_{\text{traj}}}\) and \(r_{\text{traj}}\) are obtained from the energy equation.

The performance index is applied to the state equations to obtain values for the thrusts as shown below for an ascending node maneuver.

\[
\begin{pmatrix}
K_{s,1} - 0 & 0 & 0 \\
0 & K_{s,1} - 0 & 0 \\
0 & 0 & K_{s,1} - 1
\end{pmatrix}
\begin{pmatrix}
\delta a_{R_1} \\
\delta \text{Energy}_0 \\
\delta \omega_1
\end{pmatrix} =
\begin{pmatrix}
\delta a_{R_1 \cdot \nu_1} \\
\delta \text{Energy}_0 \\
\delta \omega_1 \cdot \nu_1
\end{pmatrix}
\begin{pmatrix}
\frac{\delta V_{R_1 \cdot \nu_1}}{\delta t_0} \\
\frac{\delta \text{Energy}_0}{\delta t_0} \\
\frac{\delta \omega_1}{\delta t_0}
\end{pmatrix} +
\begin{pmatrix}
\frac{(V_S - V_{S_{\text{traj}}}) \cdot \nu_1}{r_1} \\
\frac{V_{R_1} \cdot (P_{R_1} + U_{R_1}) + V_{S_1} \cdot (P_{S_1} + U_{S_1}) \cdot \nu_1}{r_{\text{traj}}} \\
\frac{\sin(\omega_1 + \nu_1)}{V_{S_1} \cdot \sin(i_1)} \cdot (P_{W_1} + U_{W_1}) \cdot \nu_1
\end{pmatrix}
\begin{pmatrix}
\frac{\delta V_{R_1 \cdot \nu_1}}{\delta t_0} w_{\nu_1} \\
\frac{\delta \text{Energy}_0}{\delta t_0} w_{t_0} \\
\frac{\delta \omega_1}{\delta t_0} \cdot \nu_1 w_{\nu_1}
\end{pmatrix} +
\begin{pmatrix}
\frac{(V_S)^2}{r_1} - \mu \left( \frac{1}{r_1^2} - \frac{1}{r_{\text{traj}}^2} \right) + P_{R_1} + U_{R_1} \cdot \nu_1 w_{\nu_1} \\
0 \\
0
\end{pmatrix}
\begin{pmatrix}
\frac{\delta V_{R_1 \cdot \nu_1}}{\delta t_0} \\
\frac{\delta \text{Energy}_0}{\delta t_0} \\
\frac{\delta \omega_1}{\delta t_0}
\end{pmatrix} +
\begin{pmatrix}
\frac{(V_S)^2}{r_1} - \mu \left( \frac{1}{r_1^2} - \frac{1}{r_{\text{traj}}^2} \right) + P_{R_1} + U_{R_1} \cdot \nu_1 w_{\nu_1} \\
0 \\
0
\end{pmatrix}
\begin{pmatrix}
\frac{\delta \omega_1 \cdot \nu_1}{\delta t_0} \\
\frac{\delta \omega_1}{\delta t_0}
\end{pmatrix} +
\begin{pmatrix}
\frac{(V_S)^2}{r_1} - \mu \left( \frac{1}{r_1^2} - \frac{1}{r_{\text{traj}}^2} \right) + P_{R_1} + U_{R_1} \cdot \nu_1 w_{\nu_1} \\
0 \\
0
\end{pmatrix}
\begin{pmatrix}
\frac{\delta \omega_1 \cdot \nu_1}{\delta t_0} \\
\frac{\delta \omega_1}{\delta t_0}
\end{pmatrix}.
The states used to determine the maneuver are the radial position \(r\), the planar angle \(\theta\) and the out of plane inclination \(i\) and ascending node \((\Omega)\) angles. The planar angle is used to determine whether to accelerate or decelerate, the rest of the states are to maintain the same values as the chief satellite. The state equations used are

\[
\begin{bmatrix}
K_r - 1 & 0 & 0 \\
0 & K_\theta - 1 & 0 \\
0 & 0 & K_i - 1
\end{bmatrix}
\begin{bmatrix}
\frac{(r_1 - r_{0d})}{(E_{1} - E_{0d})} \\
\frac{\theta_1}{\theta_{0d}} \\
\frac{\frac{i_1}{i_{0d}}}{(\Omega_{1} - \Omega_{0d})}
\end{bmatrix}
\begin{bmatrix}
\frac{\sqrt{K_r (r_1 - r_{0d}) \mu}}{t_1} \\
\frac{\sqrt{K_\theta (\theta_1 \mu)}}{t_1} \\
\frac{\sqrt{K_i (\frac{i_1}{i_{0d}} \mu)}}{(\Omega_{1} - \Omega_{0d})}
\end{bmatrix}
+ \begin{bmatrix}
\frac{\frac{1}{2} w_1^2 r_{0d}}{t_1} \\
\frac{\frac{1}{2} w_1^2 \theta_{0d}}{t_1} \\
\frac{\frac{1}{2} w_1^2 \frac{i_1}{i_{0d}}}{(\Omega_{1} - \Omega_{0d})}
\end{bmatrix}
\end{align}
\]

For an inclination maneuver, the equations are similar, only in substituting inclination for ascending node parameters. The thrusts for an inclination maneuver are

\[
\begin{bmatrix}
K_r - 1 & 0 & 0 \\
0 & K_\theta - 1 & 0 \\
0 & 0 & K_i - 1
\end{bmatrix}
\begin{bmatrix}
\frac{(r_1 - r_{0d})}{(E_{1} - E_{0d})} \\
\frac{\theta_1}{\theta_{0d}} \\
\frac{\frac{i_1}{i_{0d}}}{(\Omega_{1} - \Omega_{0d})}
\end{bmatrix}
\begin{bmatrix}
\frac{\sqrt{K_r (r_1 - r_{0d}) \mu}}{t_1} \\
\frac{\sqrt{K_\theta (\theta_1 \mu)}}{t_1} \\
\frac{\sqrt{K_i (\frac{i_1}{i_{0d}} \mu)}}{(\Omega_{1} - \Omega_{0d})}
\end{bmatrix}
+ \begin{bmatrix}
\frac{\frac{1}{2} w_1^2 r_{0d}}{t_1} \\
\frac{\frac{1}{2} w_1^2 \theta_{0d}}{t_1} \\
\frac{\frac{1}{2} w_1^2 \frac{i_1}{i_{0d}}}{(\Omega_{1} - \Omega_{0d})}
\end{bmatrix}
\end{align}
\]

These equations maintain a satellite in a constant desired orbit, whether elliptical or circular. They can also be applied to maneuvering from one orbit to another. There are four singularity points which are eliminated by switching between ascending node only and inclination only out of plane maneuvers. There is also a singularity point when the inclination angle is zero. This only occurs in an equatorial orbit and only needs to be dealt with in that case.

For the deputy satellite, the equations become a lot more intense since corrective maneuvers are being constantly performed depending on the chief satellite’s equations of motion. This also creates more singularity points which are more difficult to deal with. The purpose of the deputy is to push or pull itself into a constant distance from the chief satellite. Since both satellites are to be released together, it is the job of the deputy to maneuver away and slow down to a desired relative position. The states used to determine the maneuver are the radial position \(r\), the planar angle \(\theta\) and the out of plane inclination \(i\) and ascending node \((\Omega)\) angles. The planar angle is used to determine whether to accelerate or decelerate, the rest of the states are to maintain the same values as the chief satellite. The state equations used are
the correct planar position. Figure 15 shows an illustration.

The difference in the radial and out of plane distances between the satellites to position it in maneuvers maintain the same radial position as the chief satellite. The planar maneuvers take into account the chief rates become

$$\frac{d\theta}{dt_0} = \frac{\nu_i}{v_1}$$

$$\frac{d\theta}{dt_0} = \frac{\nu_i + \nu_e}{v_1}$$

$$\frac{d\nu_i}{dt_0} = \frac{3\nu_i}{v_1} \left( \nu_i - \nu_e \right)$$

To shorten the derivations, since the unknowns are in the state equations of the deputy satellite, the parameters in the chief satellite are expressed as separate rates.

$$\frac{dV_{R_1}}{dt_0} = \frac{V_{S_1}^2}{r_1} - \mu \frac{1}{r_1^2} + P_{R_1} + U_{R_1}$$

$$\frac{dV_{S_1}}{dt_0} = \frac{r_1 V_{S_1}}{\mu} \left( \frac{\cos(v_i)}{e_1} \left( P_{R_1} + U_{R_1} \right) + \frac{\sin(v_i)(2 + e_1 \cos(v_i))}{e_1(1 + e_1 \cos(v_i))} \left( P_{S_1} + U_{S_1} \right) + \frac{\sin(o_1 + v_i)}{\tan(\theta_1)} \left( P_{W_1} + U_{W_1} \right) \right)$$

Hence the state equations with the chief rates become

$$\frac{dV_{R_0}}{dt_0} = \frac{V_{S_0}^2}{r_2} - \mu \frac{1}{r_2^2} + P_{R_2} + U_{R_2} - \frac{dV_{R_1}}{dt_0}$$

$$\frac{dV_{S_0}}{dt_0} = \frac{r_2 V_{S_0}}{\mu} \left( \frac{\cos(v_2)}{e_2} \left( P_{R_2} + U_{R_2} \right) + \frac{\sin(v_2)(2 + e_2 \cos(v_2))}{e_2(1 + e_2 \cos(v_2))} \left( P_{S_2} + U_{S_2} \right) + \frac{\sin(o_2 + v_2)}{\tan(\theta_2)} \left( P_{W_2} + U_{W_2} \right) \right) - \frac{dV_{S_1}}{dt_0}$$

The desired distance from the chief satellite is taken into account in the performance index of the planar angle. The out of plane maneuvers are designed to maintain the reference orbit. The radial maneuvers maintain the same radial position as the chief satellite. The planar maneuvers take into account the difference in the radial and out of plane distances between the satellites to position it in the correct planar position. Figure 15 shows an illustration.
Figure 15: Out of plane relative position of chief satellite from deputy satellite's orbital plane

From geometry two right triangles are obtained.

Figure 16: (a.) Radial projection of chief satellite on deputy satellite's orbital plane (b.) Desired relative distance projection on deputy satellite's orbital plane.

By the Pythagorean theorem, the projection of the radial distance of the chief satellite and the desired relative distance between them on the deputy plane is.
\[
\begin{align*}
 r_{\text{chief on plane}} &= \sqrt{\Delta W^2 - r_{\text{chief}}^2} \\
 \Delta d_{\text{des on plane}} &= \sqrt{\Delta d_{\text{des}}^2 - \Delta W^2}
\end{align*}
\]

From these projections the triangle below is obtained.

Figure 17: Desired relative angular difference projection on deputy satellite's orbital plane.

The correction planar maneuver angle \( \Delta \theta \) is obtained by using the law of cosines.

\[
\begin{align*}
\Delta d_{\text{des on plane}}^2 &= r_{\text{chief on plane}}^2 + r_{\text{deputy}}^2 - 2 r_{\text{chief on plane}} r_{\text{deputy}} \cos(\Delta \theta) \\
\Delta \theta &= \cos^{-1}\left( \frac{r_{\text{chief on plane}}^2 + r_{\text{deputy}}^2 - \Delta d_{\text{des on plane}}^2}{2 r_{\text{chief on plane}} r_{\text{deputy}}} \right)
\end{align*}
\]

This is the desired angular distance by which the deputy satellite must target in order to maintain the desired distance from the chief satellite. The out of plane position of the chief satellite on the deputy orbital plane is found by combining the Inertial to RSW coordinate conversion matrices et al \cite{1} of the deputy and chief satellites. The derivation of the relative matrix is shown below.

\[
\begin{align*}
 C_{2JKtoR2S2W2} &= C_{1JKtoR1S1W1} C_{12R1S1W1toR2S2W2} \\
 C_{12R1S1W1toR2S2W2} &= C_{1JKtoR1S1W1}^{-1} C_{2JKtoR2S2W2}
\end{align*}
\]

This relative conversion matrix is multiplied by the vectors in the chief coordinate to obtain their relative vectors in deputy coordinates.
Substituting the following matrices

\[
\begin{align*}
J_{\text{des}} &= \begin{pmatrix}
K_{\theta} - 1 & 0 & 0 \\
0 & K_{\theta} - 1 & 0 \\
0 & 0 & K_{\rho} - 1
\end{pmatrix}
\begin{pmatrix}
\theta_2 - \theta_1 \\
\theta_2 - \Delta \theta
\end{pmatrix}
\quad U = \begin{pmatrix}
U_{\theta \theta} \\
U_{\rho}
\end{pmatrix}
\end{align*}
\]

\[
B = \begin{pmatrix}
-\frac{\omega_t^2}{2} & 0 & 0 \\
-\frac{r_2 V_2 \cos \left( \frac{\omega_t^2}{2} \right)}{\mu \epsilon_2} & \frac{r_2 V_2 \sin \left( \frac{\omega_t^2}{2} \right)}{\mu \epsilon_2 \left( 1 + \epsilon_2 \cos \left( \frac{\omega_t^2}{2} \right) \right)} & \frac{V_2^2}{2r_2^2} \frac{r_2 V_2 \sin \left( \omega_2 + \omega_1 \right)}{\mu \tan \left( \frac{\omega_t^2}{2} \right) \left( 1 + \epsilon_2 \cos \left( \frac{\omega_t^2}{2} \right) \right)} \\
0 & 0 & \frac{V_2^2}{2r_2^2} \frac{1}{\nu_2} \frac{V_2 \sin \left( \omega_2 + \omega_1 \right)}{V_2 \sin \left( \omega_2 \right)} \omega_2^2
\end{pmatrix}
\]

\[
A = \begin{pmatrix}
\frac{r_2 V_2}{\mu} \left( r_2 \epsilon_1 \cos \left( \frac{r_2 V_2}{\mu} \right) \right) & \frac{r_2 V_2 \sin \left( r_2 V_2 \right)}{\epsilon_1 \left( 1 + \epsilon_2 \cos \left( r_2 V_2 \right) \right)} & \frac{V_2^2}{2r_2^2} \frac{1}{\nu_2} \frac{V_2 \sin \left( \omega_2 + \omega_1 \right)}{V_2 \sin \left( \omega_2 \right)} \omega_2^2
\end{pmatrix}
\]

The equation is transformed in the form

\[
J_{\text{des}} = A + B \cdot U
\]

This is where the value of \(\Delta W\) is obtained for the out of plane relative position of the chief satellite.

This can also be done with velocity and acceleration. The desired performance index for the planar angle of the deputy then becomes.

\[
J_{\text{des}} = \left( \theta_2 - \Delta \theta \right) \cdot \left( K_0 - 1 \right)
\]

This is applied to the rest of the state equations to obtain.
And solving for $U$ the desired thrusts are obtained.

$$U = B^{-1}(J\text{des} - A)$$

For inclination only maneuvers, simply replace cell block $B_{33}$ with the inclination rate parameters, and the matrices become

$$B = \begin{bmatrix}
\frac{w^2}{2} & 0 & 0 \\
-\frac{e_2 V_{\theta}}{\mu e_2} & \frac{\mu e_2}{\mu e_2 (1 + e_2 \cos(v_2))} & \frac{w^2}{2 e_2} \\
0 & 0 & \frac{\mu \tan(v_2) (1 + e_2 \cos(v_2))}{V_{\theta}}
\end{bmatrix}$$

$$A = \begin{bmatrix}
\frac{r_2 V_{\theta}^2}{\mu} & \frac{1}{r_2^2} & \frac{1}{r_2^2} \\
\frac{\mu e_2}{\mu e_2 (1 + e_2 \cos(v_2))} & \frac{\sin^2(v_2) + e_2 \cos(v_2)}{\tan(v_2) (1 + e_2 \cos(v_2))} & \frac{w^2}{2} \\
0 & 0 & \frac{\mu e_2}{\mu e_2 (1 + e_2 \cos(v_2))}
\end{bmatrix}$$

There are singularities in the resultant equation when the eccentricity and inclination angle are zero, and at every 90° intervals of the planar angle. There is also a singularity when the equation at cell block $B_{22}$ is zero. These need to be taken into consideration when running the simulation, and in applications. One way is to eliminate the section of the equation that is singular. If it is possible, it is best to find another equation that calculates the same unknown, or use a different maneuver method close to the singularity points. Another method is to let the satellites coast until the singularity point is passed. The method used is dependant on desired accuracy, mission requirements and the designer’s discrepancy.

The gain is a desired convergence rate determination. The target final output is a percentage of the orbital inputs, which is the gain. If the value of the initial input is too large, too much thrust might be required. This may result in instabilities or unreasonably high thrust outputs. In that sense it would be better to establish a way to determine a suitable gain dependant on the state errors ($\Delta X$). A
maximum (or minimum dependant on point of view) gain $K_{\text{max}}$ is determined, to be applied within a certain predetermined state error value $\Delta X_{\text{max}}$. Whenever the absolute value of an orbital parameter $X$ is less than its corresponding maximum state error value $\Delta X_{\text{max}}$, the maximum gain $K_{\text{max}}$ is applied. The sole purpose of $K_{\text{max}}$ is to smoothly converge the state error to zero. If the state error is greater than $\Delta X_{\text{max}}$, then the equation below is used to determine the desired gain.

$$K = \left[1 - \frac{\Delta X_{\text{max}}}{\Delta X}\right]$$

This will maintain a constant rate of convergence until the absolute value of $\Delta X$ is less than $\Delta X_{\text{max}}$.

Below is an illustration.

![Diagram of gain changes due to error](image.png)

**Figure 18: Gain changes due to error.**

You can set a constant convergence rate for $K$ using the following equation.

$$\frac{d\Delta X}{dt} = (1 - K_{\text{max}}) \frac{\Delta X_{\text{max}}}{\text{wt}}$$

This equation is determined by combining the desired performance index to the $K$ equation.
Fourth Order Runge-Kutta Approximation

The Second Order Taylor Series Method does not apply minimum fuel requirements since thrust time is replaced by a set calculation weight. For minimum fuel maneuvers, thrust time needs to be an unknown in order to determine the total fuel loss. In the Fourth Order Runge-Kutta Approximation Method each satellite has 4 unknowns as the control inputs in the R, S and W frame, and the thrust time $\Delta t$. To solve for the unknowns, an equal amount of constraint equations are needed.

One of the mission constraints is maintaining a constant distance between the satellites. Figure 19 describes the formation of the constant distance constraint equation.

$$\begin{align*}
\begin{pmatrix}
\delta r_{\text{R1}} \\
\delta r_{\text{S1}} \\
\delta r_{\text{W1}}
\end{pmatrix}
&= 
\begin{pmatrix}
C_{11} & C_{12} & C_{13} \\
C_{21} & C_{22} & C_{23} \\
C_{31} & C_{32} & C_{33}
\end{pmatrix}
\begin{pmatrix}
r_2 \\
0 \\
0
\end{pmatrix}
= 
\begin{pmatrix}
r_2 \cdot C_{11} \\
r_2 \cdot C_{21} \\
r_3 \cdot C_{32}
\end{pmatrix}
\end{align*}$$

$$\Delta d = 
\begin{pmatrix}
r_1 \\
0 \\
0
\end{pmatrix}
- 
\begin{pmatrix}
r_2 \cdot C_{11} \\
r_2 \cdot C_{21} \\
r_3 \cdot C_{32}
\end{pmatrix}
= 
\begin{pmatrix}
r_1 - r_2 \cdot C_{11} \\
r_2 \cdot C_{21} \\
r_3 \cdot C_{32}
\end{pmatrix}$$

$$\Delta d^2 = (r_1 - r_2 \cdot C_{11})^2 + (r_2 \cdot C_{21})^2 + (r_3 \cdot C_{32})^2$$

Figure 19: Desired relative difference constraint equation.

Another constraint is using equal fuel per maneuver. For two satellites we can develop 3 equal fuel equations. These are

$$U_{\text{R1}} = -U_{\text{R2}} \quad U_{\text{S1}} = -U_{\text{S2}} \quad U_{\text{W1}} = -U_{\text{W2}}$$

To obtain minimum fuel usage per maneuver, we find the partial derivative with respect to time of the acceleration inputs.

$$\frac{dU_{\text{R1}}}{dt} = 0 \quad \frac{dU_{\text{S1}}}{dt} = 0 \quad \frac{dU_{\text{W1}}}{dt} = 0$$
Solving for the control input $U$ will result in the minimum fuel to obtain the desired state between the satellites. Since each satellite uses the same amount of fuel, only the minimum fuel for one of the satellites needs to be found. Since each satellite burns for the same amount of time per maneuver, then $\Delta t_1 = \Delta t_2$ is the eighth constraint equation.

Each satellite has the 7 final orbital states, the control inputs in the 3 translation dimensions, and the maneuver time as 11 unknowns. For two satellites there are 14 orbital state unknowns and 8 control input unknowns. Hence, there are a total of 22 unknowns. The 14 state equations for both satellites can be used to solve for of the 14 final orbital states, and the 8 constraint equations can be used to solve for the 8 control input unknowns. For 22 equations there are 22 unknowns, therefore the problem can be solved.

Solving for these equations by hand is extremely tedious, and almost impossible to solve. Using a convergence solver method sometimes uses less memory and computing power than putting the solved equations by hand. Sometimes, the equations provide multiple answers. Finding the correct solution brings another cause for concern since the answers may vary depending on the inputs. The Newton-Raphson Method for Non-Linear Equations et al [7] has the advantage of usually converging fairly quickly. The way convergence solvers work is by providing an initial value through an algorithm that continuously updates the values until they stop changing. The value you get is the final answer. The Newton-Raphson Method continuously updates the unknown values by subtracting the previous value by the function divided by its derivative as shown below.

$$x_{n+1} = x_n - \frac{f(x_n)}{\frac{df(x_n)}{dx_n}}$$

The function $f(x)$ is obtained by bringing all the values to the left side and equaling the equation to zero. When the difference between the values $x_{n+1}$ and $x_n$ are within an acceptable error limit, the final value becomes $x_{n+1}$. If more than one answer exists, the final value will depend on the initial
value. In orbital calculations, equations are coupled depending on each other to run correctly. The Newton-Raphson Non-Linear Solver Method can be expanded to solve a system of equations in the form of matrices. All the equations are set to equal zero for the \([f(x)]\) matrix. The Jacobian becomes a square matrix, since the number of equations equal the number of unknowns. This is the equivalent of the derivative of \(f(x)\) and is inversed and multiplied by the \([f(x)]\) matrix to give.

\[
X_{n+1} = X_n - \left( \frac{dF(X_n)}{dX_n} \right)^{-1} \cdot F(X_n)
\]

Initial values are chosen for each unknown. Each unknown is subtracted by the inverse Jacobian function multiplier to become the new values of the unknowns. The process is repeated until the difference between the previous and updated unknowns are within a certain error limit. This yields the final value of the unknowns. One has to be careful in choosing the right initial values, or the equations may converge to erroneous values. The criterion is to have the initial values be fairly close to the anticipated final values. There are different ways of obtaining these initial values for every time step. The final state values and thrust unknowns can be estimated by solving for a simple first order estimation. These values should be close enough to the actual final values to cause the Newton-Raphson equations to converge. If the thrust time is small enough, then the final state values of the former calculations can be used as the initial values for the next iteration. The method used to determine the initial values depends on the longevity of the maneuver. The longer the maneuver, the higher the order of the approximations has to be for convergence. A criteria needs to be developed relating time of maneuver, iteration errors for the order of the initial value estimation and convergence criterion for the algorithms used to solve for the unknowns.

The algorithm used to estimate the satellite’s future position can vary, depending on the designer. The higher the accuracy of the algorithms, the more complex the iterations and the more computing power needed. The algorithms compute the estimated average time rate of change over a
time span \( t \). This average time derivative is multiplied by the time change \( t \) and added to the initial position. The solved final states become.

\[
X_{i+1} = X_i + X_{algorithm}(U, t)
\]

Where \( X_{i+1} \) is the final state, \( X_i \) is the initial state and \( X_{algorithm}(U, t) \) is the change in the states due to the average approximated derivative of the algorithm during time \( t \) The state equations and constraint equations in the Newton-Raphson mode are in the form.

\[
\begin{align*}
F_{state}(X_{i+1}, U, t) &= X_{i+1} - X_i - X_{algorithm}(U, t) = 0 \\
F_{constraints}(X_{i+1}, U, t) &= 0
\end{align*}
\]

The unknowns are the final states \( X_{i+1} \), the thrust \( U \) and the time of thrust \( t \). The \( F_{constraints} \) are the constraint equations in the Newton-Raphson solver mode and vary depending on the mission requirements. For the system to be solvable, the number of constraints needs to equal the number of unknowns. The Jacobian matrix of the functions \( F(X) \) is.

\[
\frac{dF(X_{i+1}, U, t)}{dX} = \begin{pmatrix}
\frac{dF_{state}(X_{i+1}, U, t)}{dX_{i+1}} & \frac{dF_{state}(X_{i+1}, U, t)}{dU} & \frac{dF_{state}(X_{i+1}, U, t)}{dt} \\
\frac{dF_{constraints}(X_{i+1}, U, t)}{dX_{i+1}} & \frac{dF_{constraints}(X_{i+1}, U, t)}{dU} & \frac{dF_{constraints}(X_{i+1}, U, t)}{dt}
\end{pmatrix}
\]

The inverse of the Jacobian becomes.

\[
\left( \frac{dF(X_{i+1}, U, t)}{dX} \right)^{-1} = \begin{pmatrix}
\frac{dX_{i+1}}{dF_{state}(X_{i+1}, U, t)} & \frac{dU}{dF_{state}(X_{i+1}, U, t)} & \frac{dt}{dF_{state}(X_{i+1}, U, t)} \\
\frac{dX_{i+1}}{dF_{constraints}(X_{i+1}, U, t)} & \frac{dU}{dF_{constraints}(X_{i+1}, U, t)} & \frac{dt}{dF_{constraints}(X_{i+1}, U, t)}
\end{pmatrix}
\]

Hence, the Newton Raphson matrix solver equation becomes.

\[
X_{n+1} = X_n - \begin{pmatrix}
\frac{dX_{i+1}}{dF_{state}(X_n)} & \frac{dU}{dF_{state}(X_n)} & \frac{dt}{dF_{state}(X_n)} \\
\frac{dX_{i+1}}{dF_{constraints}(X_n)} & \frac{dU}{dF_{constraints}(X_n)} & \frac{dt}{dF_{constraints}(X_n)}
\end{pmatrix} \begin{pmatrix}
F_{state}(X_n) \\
F_{constraints}(X_n)
\end{pmatrix}
\]
Where $X_n$ and $X_{n+1}$ are the estimated value and the updated value of the unknowns. If $X_0$ are the initial values, the Newton-Raphson solver loop is

$$X_{n+1} = X_0$$

while $X_{n+1} - X_n < \text{error}$

$$X_n = X_{n+1} - \left( \begin{array}{ccc} \frac{\text{d}X_{i+1}}{\text{d}F_{\text{state}}(X_n)} & \frac{\text{d}U}{\text{d}F_{\text{state}}(X_n)} & \frac{\text{d}t}{\text{d}F_{\text{state}}(X_n)} \\ \frac{\text{d}X_{i+1}}{\text{d}F_{\text{constraints}}(X_n)} & \frac{\text{d}U}{\text{d}F_{\text{constraints}}(X_n)} & \frac{\text{d}t}{\text{d}F_{\text{constraints}}(X_n)} \end{array} \right) \left( \begin{array}{c} F_{\text{state}}(X_n) \\ F_{\text{constraints}}(X_n) \end{array} \right)$$

end

To test the algorithm, it needs to be run through a simulation iteration to implement the outputs. The iterations are started by setting initial values for the orbital parameters. Using the initial values and the constraints, initial values are obtained for the unknowns to implement them into the algorithm. The outputs are run through a simulation iteration to output new orbital parameters. The orbital parameters and constraint equations are used to determine the initial values for the unknowns to implement into the algorithm. Output values are obtained and the process is repeated until desired. Figure 20 is a flow diagram of the process.

This concept was not tested since it is still being developed. However, the method was used in more simple derivations to verify and test maneuvering schemes. Ultimately the Second Order Taylor Series Approximation Method was used for testing maneuvering capabilities. The results and findings are shown in the next chapter.
Figure 20: Newton-Raphson and Runge-Kutta combination maneuvering flow diagram.
CHAPTER FIVE: FINDINGS

One challenge was to derive a set of equations to precisely simulate the atmospheric density due to height. These results affect the drag perturbation and need to have the ability to output atmospheric density with a height input.

Table 1: Program’s density output versus the 1976 U.S. Standard Atmosphere.

<table>
<thead>
<tr>
<th>Height</th>
<th>A</th>
<th>B (Init Density)</th>
<th>C</th>
<th>Actual Density</th>
<th>Be^A(%H)^C</th>
<th>%difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>200000</td>
<td>-4.46</td>
<td>2.91E-10</td>
<td>0.88</td>
<td>2.91E-10</td>
<td>2.91E-10</td>
<td>0.000%</td>
</tr>
<tr>
<td>220000</td>
<td>-4.46</td>
<td>2.91E-10</td>
<td>0.88</td>
<td>1.66E-10</td>
<td>1.62E-10</td>
<td>-2.626%</td>
</tr>
<tr>
<td>240000</td>
<td>-4.46</td>
<td>2.91E-10</td>
<td>0.88</td>
<td>9.91E-11</td>
<td>9.86E-11</td>
<td>-0.483%</td>
</tr>
<tr>
<td>260000</td>
<td>-4.46</td>
<td>2.91E-10</td>
<td>0.88</td>
<td>6.16E-11</td>
<td>6.20E-11</td>
<td>0.671%</td>
</tr>
<tr>
<td>280000</td>
<td>-4.46</td>
<td>2.91E-10</td>
<td>0.88</td>
<td>3.94E-11</td>
<td>3.97E-11</td>
<td>0.824%</td>
</tr>
<tr>
<td>300000</td>
<td>-4.46</td>
<td>2.91E-10</td>
<td>0.88</td>
<td>2.58E-11</td>
<td>2.58E-11</td>
<td>-0.047%</td>
</tr>
<tr>
<td>300000</td>
<td>-5.34</td>
<td>2.58E-11</td>
<td>0.945</td>
<td>2.58E-11</td>
<td>2.58E-11</td>
<td>0.000%</td>
</tr>
<tr>
<td>320000</td>
<td>-5.34</td>
<td>2.58E-11</td>
<td>0.945</td>
<td>1.72E-11</td>
<td>1.71E-11</td>
<td>-0.768%</td>
</tr>
<tr>
<td>340000</td>
<td>-5.34</td>
<td>2.58E-11</td>
<td>0.945</td>
<td>1.16E-11</td>
<td>1.16E-11</td>
<td>0.394%</td>
</tr>
<tr>
<td>360000</td>
<td>-5.34</td>
<td>2.58E-11</td>
<td>0.945</td>
<td>7.99E-12</td>
<td>8.03E-12</td>
<td>0.535%</td>
</tr>
<tr>
<td>380000</td>
<td>-5.34</td>
<td>2.58E-11</td>
<td>0.945</td>
<td>5.55E-12</td>
<td>5.58E-12</td>
<td>0.521%</td>
</tr>
<tr>
<td>400000</td>
<td>-5.34</td>
<td>2.58E-11</td>
<td>0.945</td>
<td>3.89E-12</td>
<td>3.89E-12</td>
<td>0.110%</td>
</tr>
<tr>
<td>400000</td>
<td>-6.51</td>
<td>3.89E-12</td>
<td>0.98</td>
<td>3.89E-12</td>
<td>3.89E-12</td>
<td>0.000%</td>
</tr>
<tr>
<td>420000</td>
<td>-6.51</td>
<td>3.89E-12</td>
<td>0.98</td>
<td>2.75E-12</td>
<td>2.75E-12</td>
<td>0.121%</td>
</tr>
<tr>
<td>440000</td>
<td>-6.51</td>
<td>3.89E-12</td>
<td>0.98</td>
<td>1.96E-12</td>
<td>1.97E-12</td>
<td>0.379%</td>
</tr>
<tr>
<td>460000</td>
<td>-6.51</td>
<td>3.89E-12</td>
<td>0.98</td>
<td>1.40E-12</td>
<td>1.41E-12</td>
<td>0.770%</td>
</tr>
<tr>
<td>480000</td>
<td>-6.51</td>
<td>3.89E-12</td>
<td>0.98</td>
<td>1.01E-12</td>
<td>1.01E-12</td>
<td>0.387%</td>
</tr>
<tr>
<td>500000</td>
<td>-6.51</td>
<td>3.89E-12</td>
<td>0.98</td>
<td>7.30E-13</td>
<td>7.30E-13</td>
<td>-0.014%</td>
</tr>
<tr>
<td>500000</td>
<td>-7.34</td>
<td>7.30E-13</td>
<td>0.97</td>
<td>7.30E-13</td>
<td>7.30E-13</td>
<td>0.000%</td>
</tr>
<tr>
<td>520000</td>
<td>-7.34</td>
<td>7.30E-13</td>
<td>0.97</td>
<td>5.31E-13</td>
<td>5.28E-13</td>
<td>-0.507%</td>
</tr>
<tr>
<td>540000</td>
<td>-7.34</td>
<td>7.30E-13</td>
<td>0.97</td>
<td>3.88E-13</td>
<td>3.87E-13</td>
<td>-0.138%</td>
</tr>
<tr>
<td>560000</td>
<td>-7.34</td>
<td>7.30E-13</td>
<td>0.97</td>
<td>2.85E-13</td>
<td>2.86E-13</td>
<td>0.191%</td>
</tr>
<tr>
<td>580000</td>
<td>-7.34</td>
<td>7.30E-13</td>
<td>0.97</td>
<td>2.11E-13</td>
<td>2.11E-13</td>
<td>0.041%</td>
</tr>
<tr>
<td>600000</td>
<td>-7.34</td>
<td>7.30E-13</td>
<td>0.97</td>
<td>1.56E-13</td>
<td>1.56E-13</td>
<td>0.257%</td>
</tr>
<tr>
<td>600000</td>
<td>-7.66</td>
<td>1.56E-13</td>
<td>0.955</td>
<td>1.56E-13</td>
<td>1.56E-13</td>
<td>0.000%</td>
</tr>
<tr>
<td>620000</td>
<td>-7.66</td>
<td>1.56E-13</td>
<td>0.955</td>
<td>1.17E-13</td>
<td>1.16E-13</td>
<td>-0.983%</td>
</tr>
<tr>
<td>640000</td>
<td>-7.66</td>
<td>1.56E-13</td>
<td>0.955</td>
<td>8.79E-14</td>
<td>8.76E-14</td>
<td>-0.319%</td>
</tr>
<tr>
<td>660000</td>
<td>-7.66</td>
<td>1.56E-13</td>
<td>0.955</td>
<td>6.65E-14</td>
<td>6.67E-14</td>
<td>0.303%</td>
</tr>
<tr>
<td>680000</td>
<td>-7.66</td>
<td>1.56E-13</td>
<td>0.955</td>
<td>5.08E-14</td>
<td>5.10E-14</td>
<td>0.370%</td>
</tr>
<tr>
<td>700000</td>
<td>-7.66</td>
<td>1.56E-13</td>
<td>0.955</td>
<td>3.91E-14</td>
<td>3.91E-14</td>
<td>-0.013%</td>
</tr>
<tr>
<td>700000</td>
<td>-6.75</td>
<td>3.91E-14</td>
<td>0.915</td>
<td>3.91E-14</td>
<td>3.91E-14</td>
<td>0.000%</td>
</tr>
<tr>
<td>720000</td>
<td>-6.75</td>
<td>3.91E-14</td>
<td>0.915</td>
<td>3.04E-14</td>
<td>3.01E-14</td>
<td>-0.918%</td>
</tr>
<tr>
<td>740000</td>
<td>-6.75</td>
<td>3.91E-14</td>
<td>0.915</td>
<td>2.39E-14</td>
<td>2.39E-14</td>
<td>0.029%</td>
</tr>
<tr>
<td>760000</td>
<td>-6.75</td>
<td>3.91E-14</td>
<td>0.915</td>
<td>1.90E-14</td>
<td>1.92E-14</td>
<td>0.879%</td>
</tr>
<tr>
<td>780000</td>
<td>-6.75</td>
<td>3.91E-14</td>
<td>0.915</td>
<td>1.53E-14</td>
<td>1.55E-14</td>
<td>1.071%</td>
</tr>
<tr>
<td>800000</td>
<td>-6.75</td>
<td>3.91E-14</td>
<td>0.915</td>
<td>1.25E-14</td>
<td>1.25E-14</td>
<td>0.267%</td>
</tr>
</tbody>
</table>
Between 200km and 800km height there is only a 2.6% maximum deviation between the actual data and the algorithm density output. Most are under a 1% difference. The figures below show the results of two simulations, one with and one without the J2 term, after five periods around the Earth.

Figure 21: Orbit of two satellites without perturbations.

Figure 22: Orbits of two satellites with J2 and drag perturbations.
Figure 23 shows the relative distance between two satellites with the perturbation term.

For orbital maneuvers to be verified, a simulation program was developed to encompass the J2 perturbation and drag effects on the satellite. Iterations were solved using the Classical Runge-Kutta Fourth Order method and tested using different initial conditions. Results were compared with other published data to verify that the simulations work. The equations of motion were used due to their simplicity. They’re easy to visualize in 3 dimensions. The simulation program is very useful in determining a desired orbit and visualizing its reaction. You can also determine drifts after a certain time or orbit. Figure 24 shows graphs which demonstrate how orbital parameters affect the orbit of a satellite and relative orbit of a constellation.
The figure above shows the change in relative position of two satellites with different eccentricities. The satellites start at the same orbital position and are allowed to drift for 5 orbits. Having a relative acceleration in the opposite direction tangent to the swirl will cause a desired eccentricity change. Thrusting clockwise will increase the eccentricity while thrusting counter-clockwise will decrease it.

Figure 24: Relative position due to (a.) Decreasing eccentricity (b.) Increasing eccentricity.

Eccentricity change

decrease  
e1=0.01  e2=0.0

increase  
e1=0.01  e2=0.02

Height=300km  υ=10.3°  ω=33.0°  Ω=45.0°  i=35.0°
Figure 25: Relative position of satellite due to true anomaly difference.

This graph illustrates how a difference in the true anomaly between two satellites will affect their relative position. The cluster is started with the same orbital parameters except their true anomalies. They were allowed to drift for 5 orbits to demonstrate their relative position change. A lagging orbit will tend to decrease its relative position, while a leading orbit will increase. This means the relative difference in true anomaly will have a proportional effect on the relative orbit. In an eccentric orbit, if it is desired to lag another satellite, then a relative thrust in the relative radial direction needs to be
applied to maintain the same constant relative distance. A leader satellite would have to apply a relative thrust in the negative radial direction.

The graph above analyzes the effects an inclination difference has on the relative orbit of two satellites. There is only a cross plane difference hence there’s only a one dimensional path between the satellites. Relative inclination changes need to be pointed along the path line.

Blue decrease

\[ i_1 = 35.0° \quad i_2 = 15.0° \]

Green increase

\[ i_1 = 35.0° \quad i_3 = 55.0° \]

Height=300km \quad \nu = 10.3° \quad \omega = 33.0° \quad \Omega = 45.0° \quad e = 0.01

Figure 26: Relative position of satellite due to inclination difference.
The figure above illustrates two satellites with a difference in their argument of perigee positions. Since both satellites are started at their same orbital position, only with an offset in the same orbital plane, their energies and radial positions will remain the same. The result is the satellite will remain at a constant relative distance in a circular relative orbit.

Figure 27: Relative position of satellite due to argument of perigee difference.

\[
\begin{align*} 
\omega_1 &= 33.0^\circ \\
\omega_2 &= 63.0^\circ \\
\omega_1 &= 33.0^\circ \\
\omega_3 &= 123.0^\circ \\
\text{Height} &= 300\text{km} \\
\nu &= 10.3^\circ \\
i &= 35.0^\circ \\
\Omega &= 45.0^\circ \\
e &= 0.01 
\end{align*}
\]
Above we see what the relative effects are of ascending node differences between two satellites. An ascending node difference only affects in the relative plane parallel to the equatorial plane. This is because the ascending node angle is in the equatorial plane.

Figure 28: Relative position of satellite due to ascending node difference.

\[
\begin{align*}
\Omega_1 &= 45.0^\circ & \Omega_2 &= 75.0^\circ & \Omega_1 &= 45.0^\circ & \Omega_3 &= 135.0^\circ \\
\text{Height} &= 300\text{km} & \nu &= 10.3^\circ & i &= 45.0^\circ & \omega &= 33.0^\circ & e &= 0.01
\end{align*}
\]
Figure 29: Effects of perturbations with different eccentricities.

Perturbation changes every orbital parameter of an orbit, hence they are an important part of orbital maneuvering. Here it is seen how increasing the eccentricity decreases the amplitude of the perturbation force on the spacecraft. This is due to the decrease in gravitational effects due to an increase in radial position. A circular orbit has no apogee or perigee, adding an eccentricity gives the orbit one. If there was a decrease in velocity from a circular orbit, then the maneuvering position would become apogee. An increase would turn the maneuvering position into perigee. Changing the velocity to change the eccentricity would result in an inverse effect on perturbation forces.

Figure 30: Effects of perturbations after (a.) 2 orbits (b.) 5 orbits and (c.) 10 orbits.
Figure 30 shows how perturbations affect the argument of perigee and ascending node. The right graph is after 2 orbits, the middle graph is after 5 orbits and the left is after 10. There is an increase in the ascending node rate due to the lengthening out of the ascending node position from the Earth’s geometric center.

Below are other graphs showing the effects of perturbations due to the J2 term and drag.

Figure 31: Perturbation drift in true anomaly, argument of perigee and planar angle.

Below are perturbation effects of the true anomaly, argument of perigee and planar angle in different orbits.

Figure 32: Ascending node drift due to perturbations.
Figure 33: Perturbation drift on orbits with eccentricities of 0.01, 0.1 and 0.2 on (a.) radial velocity (b.) perpendicular velocity and (c.) out of plane velocity.

The graphs above show the velocity drift from the non-perturbed orbit for the RSW components. As is shown, an increase in eccentricity increases the amplitude of the drift in the radial and perpendicular direction, but decreases in the out of plane direction. The total velocity drift is

Figure 34: Total velocity perturbation drift on orbits with eccentricities of 0.01, 0.1 and 0.2.

The amplitude of the drift stays fairly constant since the predominant J2 term doesn’t decrease the energy of the orbit. A decrease in amplitude would be caused by the drag effects, which are very small compared to the J2 term. These simulation results are useful in designing a suitable controls.
algorithm. Being able to see the relative effects of the movement from one satellite to the other and the effects of perturbation aid a great deal in visualizing and designing the structure for the controls equations.

First the orbit of the chief satellite has to be determined. It is necessary for the chief satellite not only to be able to maintain its orbit, but to change it, depending on the mission requirements. For the KnightSat mission, the orbit will be circular or nearly circular. Below are the results for the simulation of a chief satellite station keeping in a circular orbit.

Figure 35: Orbit following simulation results chief satellite.

These are the graphs of the desired and the actual satellite orbit. Since they overlap they only appear as one orbit. On the three dimensional graph there is a slight bit of bluish color. This is due to the chief satellite’s slight out of plane deviation before entering a maneuvering area. The deviations in the orbital plane are too small to be seen as is demonstrated in the graphs below.
The energy error is within $1 \text{ m}^2/\text{s}^2$, and the radial distance error is within $1 \text{ mm}$. However, the out of plane error is much greater, as shown below.

The inclination angle shows a maximum error of approximately $0.0003$ radians. This may not seem like much, but that has the potential of being $2 \text{ km}$ off in the out of plane direction when at the planned KnightSat orbital height of $300\text{ km}$. The out of plane distance offset due to inclination and ascending node error is mostly dependant on how close to $90^\circ$ and $270^\circ$ along the orbital planar trajectory the satellite is, since that is the farthest distance from the ascending node. These position
errors can be taken into account when planning to fly over a certain area. The thrust outputs in the radial, perpendicular and out of plane direction are shown below.

(a.) (b.)

Figure 38: Orbit following thrust, perturbation and acceleration in (a.) Radial direction (b.) Perpendicular direction (c.) Out of plane direction.

The blue line represents the thrust, the green line represents the perturbations and the red line represents the total satellite acceleration. The radial and perpendicular thrusts only work to counteract the perturbation effects to maintain the satellite in a circular orbit. The satellite maintains a constant velocity in the radial and perpendicular direction, hence the satellite accelerations (red lines) in the radial and perpendicular direction are maintained at zero. The out of plane thrusts only work to bring
the satellite back to its original position in the maneuvering areas. That is where you see the pulses of thrusts. I set a constraint in the controls algorithms of 0.1 m/s² thrust acceleration. However this constraint can be changed down to approximately 0.012 m/s² to successfully have full control of the perturbation effects.

In an eccentric orbit, the maneuvers are the same as in a circular orbit except for the addition of argument of perigee maneuvers. These will be treated the same as out of plane maneuvers due to the $\sin(\nu)$ term connected to the perpendicular thrust in the argument of perigee state equation. Hence, argument of perigee maneuvers will take place near when the $\sin(\nu)$ term is at its max. Figure 39 shows the graphs showing the eccentricity (blue), argument of perigee error (red) and true anomaly error (green) of the chief satellite following a 0.1 eccentric orbit.

![Figure 39: 0.1 eccentricity orbit (a.) Eccentricity (b.) Argument of perigee error.](image)

As the graph above shows, the argument of perigee reaction is very similar to the ascending node reaction to the thrust. The true anomaly error remains very close to zero, hence why the eccentricity remains constant to the desired eccentricity. Out of plane and radial reactions are similar to the circular orbit keeping case. Below are the graphs of the radial, perpendicular and out of plane thrusts. The blue line represents the thrust acceleration, the green line represents the perturbations and the red line is the total acceleration of the satellite.
Figure 40: 0.1 eccentricity orbit thrust, perturbation and acceleration in (a.) Radial direction (b.) Perpendicular direction (c.) Out of plane direction.

Figure 41: Close-up of (a.) Figure 40a (b.) Figure 40b.
There are glitches in the smoothness of the perpendicular thrusts due to corrective maneuvers to the argument of perigee. This is due to out of plane thrusts affecting the argument of perigee.

It is necessary for the satellite to maneuver from one area to the other, even if minutely. The deputy satellite’s responsibility is to continuously maneuver itself to maintain its constant distance. Simulations were run to demonstrate the capabilities of the algorithms to change their orbits. Below is an orbit raising maneuver from 300km to 1000km height with a maximum relative radial rate of 20m/s as described in the algorithms above.

![Orbit raising maneuver 300km to 1000km.](image)

**Figure 42: Orbit raising maneuver 300km to 1000km.**

Below is the constant radial velocity of the chief satellite as it smoothly transitions to 1000km.

![Radial velocity of orbit raising maneuver.](image)

**Figure 43: Radial velocity of orbit raising maneuver.**
This is a smooth, low thrust maneuver for the MET with a suitable maximum thrust constraint. Shown in Figure 44 is the development of the radial distance while returning to a circular orbit once reaching its final trajectory height. The graph on the right is the eccentricity change and the graph on the left is of the radial distance error.

Figure 44: Orbit raising maneuver (a.) Radial position (b.) Eccentricity.

The maneuver took approximately 9-3/4 hours and 7 orbits. Thrust constraints can be placed to minimize the fuel consumption. Below are the radial, perpendicular and out of plane thrusts (blue) and perturbations (green).

(a.)

(b.)
Figure 45: Orbit raising maneuver thrust, perturbation and acceleration in (a.) Radial (b.) Perpendicular (c.) Out of Plane direction.

Notice the graph shows the out of plane perturbations obviously decreasing with respect to the increasing height.

Another set of simulations were to demonstrate low thrust, out of plane maneuvering. Below are the results of an inclination only maneuver to increase the inclination from $10^\circ$ to $15^\circ$.

Figure 46: Inclination only change maneuver (a.) inclination error (b.) Orbital path.

The graph above shows the inclination error. Out of plane maneuvers are performed every $180^\circ$, hence the step-like look of the graph. It took approximately 6 orbits and 8-1/2 hours to complete.
this maneuver. The graphs below show the ascending node and argument of perigee reaction since they’re affected by out of plane thrusts.

Figure 47: Inclination only change maneuver (a.) Argument of perigee (b.) Ascending node error.

As is seen from the graphs, the two orbital angles remain in their same position despite the inclination change. Below are the radial, perpendicular and out of plane thrusts.
Figure 48: Inclination only change maneuver thrust, perturbation and acceleration in (a.) Radial (b.) Perpendicular (c.) Out of plane direction.

The perturbations (green) increase as inclination increase. This is due to there being a sine of the inclination angle inside the RSW J2 perturbations. This causes a proportionate increase in the effects of the J2 perturbation as inclination increases.

To demonstrate control of the ascending node angle, simulations were run to change it from 30° to 45°. Below are graphs of the orbit (on the right) and the relative difference between the satellite and the trajectory (on the left).

Figure 49: Ascending node only change maneuver (a.) relative position from trajectory (b.) Orbital path.
It is interesting to note that an ascending node change in the inertial frame is equivalent to an inclination change of equal magnitude in the relative orbit frame. This demonstrates a relationship between the effects of out of plane maneuvers in the inertial frame and the inertial relative orbit frame. The graphs below demonstrates the transition of the ascending node error.

Figure 50: Ascending node only change maneuver (a.) Inclination (b.) Ascending node error.

As in the inclination only maneuver, there is a step-like change in the ascending node position due to only making maneuvers every 180°. Ascending node maneuvers very minutely affect planar parameters in changing the effects of the Earth’s oblateness perturbation, but have no direct intervention on them. They do directly affect the inclination angle and are maneuvered back to their original positions. Figure 51 shows the thrust outputs (blue) and their respective affect on accelerations (red) and perturbations (green).
Figure 51: Ascending node only change maneuver thrust, perturbation and acceleration in (a.) Radial
(b.) Perpendicular (c.) Out of plane direction.

Since there are no ascending node terms in the J2 perturbation, no affects are seen on the ascending
node. There are other much smaller terms affected by changing the ascending node, but are not
included in the simulations. There is a step-like input to the out of plane thrust with a maximum of
0.1 m/s².

Simulations involving formation flying were performed using different schemes. First
formation keeping was demonstrated with both satellites starting together. The deputy proceeds to
maneuver itself away from the chief until it reaches its desired relative distance of 1km. Then the
constellation remains in a constant relative orbit. Singularities are encountered near apogee and perigee, which cause the satellites to deviate from their desired relative positions as shown below.

Figure 52: Singularity effect on relative distance

The deviations are damped considerably by canceling the radial perturbations of the deputy. This allows the calculations to pass through the singularity points with very little effect on their relative orbits. Below is a demonstration of a separation and formation flying maneuver with the dampened effects.

Figure 53: Dampened singularity effect on relative distance
The result is only a maximum singularity deviation of 20 meters, compared to over 3,000 meters without the dampening effect. The constellation was released at a height of 300 kilometers and at an eccentricity of almost nearly zero. The figures below are the thrusts (blue), perturbations (green) and resultant accelerations on the satellite (red) in the radial, perpendicular and out of plane direction.

Figure 54: Formation flying radial thrust of (a.) Chief (b.) Deputy

Figure 55: Formation flying perpendicular thrust of (a.) Chief (b.) Deputy
To demonstrate the constellation’s ability to recover from being released at a different orbital energy than desired, simulations were completed to bring the constellation to a nearly circular desired orbit. The satellites were released together at an eccentric orbit of 0.01 and they both came to a nearly circular steady state orbit. The deputy satellite pushed itself away from the chief while changing its eccentricity, then slowing its relative speed to its desired constant relative distance behind the chief.

Simulations were run at 300km and 600km heights with relative desired distances of 600m and 1000m. The relative inertial position and relative distance graphs are shown below.
$\text{height}_{\text{chief}} = 300\text{km} \quad \Delta \text{dist}_{\text{deputy-chief}} = 1000\text{m}$

Figure 57: Relative graph of Deputy satellite's motion in IJK coordinates as seen from Chief Satellite.

Figure 58: Relative distance graph of formation flying satellites.
height_{chief} = 300\text{km} \quad \Delta\text{distance}_{chief-deputy} = 1000\text{m}

Figure 59: Radial, Perpendicular and Out of Plane thrust, perturbation and acceleration for Chief and Deputy Satellites
height_{chief} = 300\,km \quad \text{Δ}dist_{deputy-chief} = 600\,m

Figure 60: Relative graph of Deputy satellite's motion in IJK coordinates as seen from Chief Satellite.

Figure 61: Relative distance graph of formation flying satellites.
Figure 62: Radial, Perpendicular and Out of Plane thrust, perturbation and acceleration for Chief and Deputy Satellites

height\textsubscript{chief} = 300km \quad \Delta\text{distance}\textsubscript{chief-deputy} = 600m
height_{chief} = 600\text{km} \quad \Delta \text{dist}_{deputy-chief} = 1000\text{m}

Figure 63: Relative graph of Deputy satellite's motion in IJK coordinates as seen from Chief Satellite.

Figure 64: Relative distance graph of formation flying satellites.
Figure 65: Radial, Perpendicular and Out of Plane thrust, perturbation and acceleration for Chief and Deputy Satellites

height\textsubscript{chief} = 600\text{km} \quad \Delta\text{distance}\textsubscript{chief-deputy} = 1000\text{m}
height_{chief} = 600\text{km} \quad \Delta \text{dist}_{deputy-chief} = 600\text{m}

Figure 66: Relative graph of Deputy satellite's motion in IJK coordinates as seen from Chief Satellite.

Figure 67: Relative distance graph of formation flying satellites.
height_{chief} = 600\text{km} \quad \Delta distance_{chief-deputy} = 600\text{m}

**Deputy Satellite**

**Chief Satellite**

Figure 68: Radial, Perpendicular and Out of Plane thrust, perturbation and acceleration for Chief and Deputy Satellites
The minor kinks that are seen in the relative orbits are due to chief satellite maneuvers that are being compensated for by the deputy satellite. Once the controls algorithm notices the change it starts making a maneuver of its own to counter-act the error. Since there is a gain involved in its transition, there will be minor variations before settling into its steady state position. This is a normal occurrence since the maneuvers are based on past data.
CHAPTER SIX: CONCLUSION

Derivation concepts for two satellites flying in formation were developed suitable for the KnightSat mission. A chief satellite can maneuver itself anywhere in the world with the deputy following it in a leader-follower satellite setup. The flexibility of the calculations are shown for maneuvering from one spot to another using low thrust capabilities. Some maneuver’s maximum thrust were upscaled to make the simulations faster, but the concept can still be applied to lower thrust. Once in the vicinity of the target area, the satellites can orient themselves to take the three dimensional pictures. The equations of motion can be solved using non-linear methods such as Taylor Series, Newton Raphson and Runge-Kutta to provide a good estimate for determining the dynamical outputs for spacecraft controls and orbital formation flying. Singularities become a problem when using these kinds of numerical methods, but can be dampened to where the effects are minimal. There are minor deviations from the steady state, but not more than less than a meter. The algorithms are a good way to see the solutions while visualizing the dynamical effects. Despite the minor problems, these methods are more than suitable for the KnightSat mission of formation flying.
APPENDIX A: 2 BODY PROBLEM
A. Derivation of 2 Body Problem

The vector force of gravity acting on a body due to another body is

\[ \vec{F_g} = G \frac{M}{r^3} \]

From these relations the equations of motion of the Earth and the satellite are

\[ \frac{d^2r_{earth}}{dt^2} = G \frac{m_{sat}}{r_{earth}^3} \frac{r_{earth}}{r_{earth}} + P_{earth} \]
\[ \frac{d^2r_{sat}}{dt^2} = G \frac{M_{earth}}{r_{sat}^3} \frac{r_{sat}}{r_{sat}} + P_{sat} + u \]

Where \( r_{earth} \) and \( r_{sat} \) are the radial distances from an arbitrary point in space to the center of the Earth and the satellite respectively, \( M_{earth} \) is the mass of the Earth, \( m_{sat} \) is the mass of the satellite, and \( r_{earthsat} \) is
the radial distance from the Earth to the satellite, \( r_{\text{sat-earth}} \) is the radial distance from the satellite to the Earth, and \( P_{\text{Earth}} \) and \( P_{\text{Sat}} \) are the perturbation accelerations. Perturbations will be discussed more in depth in Appendix D. The vector \( u \) is the effect of the thrusters on the satellite. To find their relative effects the equations are subtracted

\[
\frac{d^2 r_{\text{earth}}}{dt^2} - \frac{d^2 r_{\text{sat}}}{dt^2} = G \left( \frac{m_{\text{sat}}}{r_{\text{earth}}^3} - \frac{M_{\text{Earth}}}{r_{\text{sat-earth}}^3} \right) + \left( P_{\text{Earth}} - P_{\text{Sat}} \right) + u
\]

Since the difference of the effects is the relative behavior from one body to the other, to look for the effect on the satellite from the earth’s perspective the following can be set as true

\[
\frac{d^2 r_{\text{earth}}}{dt^2} - \frac{d^2 r_{\text{sat}}}{dt^2} = \frac{d^2 r}{dt^2}
\]

\( r_{\text{earth-sat}} = -r \)

\( r_{\text{sat-earth}} = r \)

\( \rightarrow P = P_{\text{Earth}} - P_{\text{Sat}} \)

Along with setting \( \mu = G(m_{\text{sat}} + M_{\text{Earth}}) \), the Earth centered equation of motion of the satellite is

\[
\frac{d^2 r}{dt^2} + \frac{\mu}{r^3} \frac{r}{dt^2} = P + u
\]
APPENDIX B: PLANAR ORBITAL STATE EQUATIONS
B. Derivation of Planar Orbital State Equations

The first state equation is the radial rate.

\[
\frac{dr}{dt} = V_R
\]

The time rate of change of the radial and perpendicular velocities are obtained from the gravitational motion equations. It is known that the position vector in the RSW frame is

\[
r = r e_R
\]

Taking the time derivative results in the velocity vector as follows

\[
\frac{\dot{r}}{dt} = r \frac{\dot{e}_R}{dt} + \frac{dr}{dt} e_R
\]

\[
\frac{\dot{e}_R}{dt} = \frac{\dot{d}v}{dt} e_S \quad \frac{\dot{e}_S}{dt} = -\frac{\dot{d}v}{dt} e_R
\]

\[
\frac{\dot{r}}{dt} = \frac{dr}{dt} e_R + r \frac{\dot{d}v}{dt} e_S = V_R e_R + V_S e_S
\]

The radial rate and true anomaly are obtained from the velocity vector

\[
\frac{dr}{dt} = V_R
\]

\[
V_S = r \frac{\dot{d}v}{dt} \quad \frac{\dot{d}v}{dt} = \frac{V_S}{r}
\]

Taking the time derivative of the velocity vector gives the acceleration vector

\[
\frac{\ddot{r}}{dt^2} = V_R \frac{\dot{d}v}{dt} e_S + \left( \frac{dV_R}{dt} e_R - \frac{V_S}{r} \frac{\dot{d}v}{dt} e_R + \left( \frac{dV_S}{dt} + \frac{V_R \cdot V_S}{r} \right) e_S
\]

\[
\frac{\ddot{e}_R}{dt^2} = V_R \frac{dV_S}{dt} e_S + \left( \frac{dV_R}{dt} e_R - \frac{V_S}{r} \frac{dV_S}{dt} e_R + \left( \frac{dV_S}{dt} + \frac{V_R \cdot V_S}{r} \right) e_S
\]

\[
\frac{\ddot{e}_S}{dt^2} = \left( \frac{dV_R}{dt} - \frac{V_S^2}{r} \right) e_R + \left( \frac{dV_S}{dt} + \frac{V_R \cdot V_S}{r} \right) e_S
\]
Inserting it into the gravity potential equation above results in

$$\left( \frac{dV_R}{dt} - \frac{V_S^2}{r} + \frac{\mu}{r^2} \right) \cdot e_R + \left( \frac{dV_S}{dt} + \frac{V_R \cdot V_S}{r} \right) \cdot e_S + \left( \frac{dV_W}{dt} \right) \cdot e_W = \left( P_R + U_R \right) e_R + \left( P_S + U_S \right) e_S + \left( P_W + U_W \right) e_W$$

Solving for the time rate changes and turning the vector equation into matrix form gives.

$$\begin{pmatrix}
\frac{dV_R}{dt} \\
\frac{dV_S}{dt} \\
\frac{dV_W}{dt}
\end{pmatrix} = \begin{pmatrix}
\frac{V_S^2}{r} - \frac{\mu}{r^2} + P_R + U_R \\
\frac{V_R \cdot V_S}{r} + P_S + U_S \\
\end{pmatrix}$$

The state equation for the true anomaly is taken from knowing the perpendicular velocity is the angular rate times the radial distance et al [1].

$$V_S = r \frac{dv}{dt} \quad \frac{dv}{dt} = \frac{V_S}{r}$$

Hence the state equations affecting the attributes of the orbit are

$$\frac{dr}{dt} = V_R$$

$$\frac{dV_R}{dt} = \frac{V_S^2}{r} - \frac{\mu}{r^2} + P_R + U_R$$

$$\frac{dV_S}{dt} = \frac{V_R \cdot V_S}{r} + P_S + U_S$$

$$\frac{dv}{dt} = \frac{V_S}{r}$$
C. Derivation of Angular State Equations from the Momentum Vectors

The angular momentum vector can be related to the ascending node and inclination angle by geometry.

\[ \cos(i) = \frac{h_z}{h}, \quad \tan(\Omega) = \left(\frac{-h_y}{h_x}\right) \]

Geometrically, the inclination and ascending node et al [3] are

To find their rates of change take their time derivatives

\[ \frac{\sin(i)}{\cos(i)^2} \frac{di}{dt} = \frac{\frac{dh_z}{dt} - h_z \frac{dh}{dt}}{h^2} \]

\[ \frac{1}{\cos^2(\Omega)} \frac{d\Omega}{dt} = \frac{\frac{dh_x}{dt} - h_x \frac{dh_y}{dt}}{h_x^2} \]

\[ \frac{d\Omega}{dt} = \cos^2(\Omega) \left(\frac{\frac{dh_x}{dt} - h_x \frac{dh_y}{dt}}{h_x^2}\right) \]

\[ \frac{di}{dt} = \frac{h \frac{dh_z}{dt} - h_z \frac{dh}{dt}}{\sin(i) h^2} \]
Since
\[ \cos^2(\Omega) = \frac{h_x^2}{h_x^2 + h_y^2} \]

The ascending node time rate of change can be written as.
\[ \frac{d\Omega}{dt} = \frac{h_y \frac{dh_x}{dt} - h_x \frac{dh_y}{dt}}{h_x^2 + h_y^2} \]

From inspection, it can be concluded that geometrically the angular momentum vectors et al [3] are
\[ h_x = h \cdot \sin(i) \cdot \sin(\Omega) \]
\[ h_y = -h \cdot \sin(i) \cdot \cos(\Omega) \]
\[ h_z = h \cdot \cos(i) \]

The time derivative of the angular momentum is found by the cross product of the radial position and
time derivative of the angular momentum is found by the cross product of the radial position and velocity et al [6]. Below are the calculations.

\[ \frac{dH}{dt} = r \times \frac{d^2r}{dt^2} \]
since \[ r \times r = 0 \]
\[ \frac{dH}{dt} = r \times \frac{d^2r}{dt^2} \]

\[ \frac{dH}{dt} = r \times \left( \frac{\mu \cdot r + P + U}{r^3} \right) = \frac{-\mu (r \times r)}{r^3} + r \times P + r \times U \]

\[ \frac{dH}{dt} = r \times P + r \times U \times \left[ \begin{array}{ccc} e_R & e_S & e_W \\ 0 & 0 & 0 \\ P_R & P_S & P_W \end{array} \right] + \left[ \begin{array}{ccc} e_R & e_S & e_W \\ r & 0 & 0 \\ U_R & U_S & U_W \end{array} \right] 
\]

\[ \frac{dH}{dt} = -r(P_W + U_W) \cdot e_S + r(P_S + U_S) \cdot e_W \]

the magnitude of the angular momentum only changes in the out of plane direction, hence the
magnitude of the angular momentum time derivative et al [3] is
\[ \frac{dh}{dt} = r(P_S + U_S) \]

Since the vector is in the RSW frame, a conversion is needed to represent it in the inertial frame.
The coordinate conversion is.

\[
\begin{bmatrix}
\cos(\omega + v) & \sin(\omega + v) & 0 \\
-sin(\omega + v) & \cos(\omega + v) & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 \\
0 & \cos(i) & \sin(i) \\
0 & -\sin(i) & \cos(i)
\end{bmatrix}
\begin{bmatrix}
\cos(\Omega) & \sin(\Omega) & 0 \\
-sin(\Omega) & \cos(\Omega) & 0 \\
0 & 0 & 1
\end{bmatrix}
\rightarrow
\begin{bmatrix}
\cos\omega & \sin\omega & 0 \\
-\sin\omega & \cos\omega & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

By replacing the following equations into the angular momentum vector etc [3]

\[
\begin{bmatrix}
\cos\omega \cdot \cos\Omega - \sin\omega \cdot \sin\Omega \cdot \cos(i) & \cos\omega \cdot \sin\Omega + \sin\omega \cdot \cos\Omega \cdot \cos(i) & -\sin\omega \cdot \sin\Omega \cdot \cos(i) \\
-\sin\omega \cdot \cos\Omega - \cos\omega \cdot \sin\Omega \cdot \cos(i) & \cos\omega \cdot \cos\Omega + \sin\omega \cdot \sin\Omega \cdot \cos(i) & \sin\omega \cdot \cos\Omega \cdot \cos(i)
\end{bmatrix}
\rightarrow
\begin{bmatrix}
\cos i & \sin i & 0 \\
-\sin i & \cos i & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

The time rates of change for the angular momentum vector become.

\[
\frac{dh_x}{dt} = r\left[ (P_W + U_W) \cdot \sin(\omega + v) \cdot \cos(\Omega) + \cos(\omega + v) \cdot \sin(\Omega) \cdot \cos(i) \right]
\]

\[
\frac{dh_y}{dt} = r\left[ (P_W + U_W) \cdot \sin(\omega + v) \cdot \sin(\Omega) - \cos(\omega + v) \cdot \cos(\Omega) \cdot \cos(i) \right]
\]

\[
\frac{dh_z}{dt} = r\left[ -\cos(\omega + v) \cdot \sin(i) + (P_S + U_S) \cdot \cos(i) \right]
\]

For the inclination angle time rate of change can be obtained by inserting these equations into the inclination time derivative as shown below.
\[
\frac{di}{dt} = \frac{h \cdot \frac{dh_z}{dt} - h_z \cdot \frac{dh}{dt}}{\sin(i) h^2}
\]

\[
h \cdot \frac{dh_z}{dt} = h \cdot r \cdot \cos(i) \cdot \left( P_S + U_S \right)
\]

\[
h_z \frac{dh}{dt} = h \left[ -r \left( P_W + U_W \right) \cdot \cos(\omega + \nu) \cdot \sin(i) + r \left( P_S + U_S \right) \cdot \cos(i) \right]
\]

\[
h \cdot \frac{dh_z}{dt} - h_z \frac{dh}{dt} = h \cdot r \cdot \left( P_W + U_W \right) \cdot \cos(\omega + \nu) \cdot \sin(i)
\]

\[
\frac{di}{dt} = h \cdot r \cdot \left( P_W + U_W \right) \cdot \cos(\omega + \nu) \cdot \sin(i) = \frac{r \cdot \cos(\omega + \nu) \cdot \left( P_W + U_W \right)}{h^2}
\]

The inclination rate can also be expressed as.

\[
\frac{di}{dt} = \frac{\cos(\omega + \nu) \cdot \left( P_W + U_W \right)}{V_S} \quad \text{or} \quad \frac{di}{dt} = \frac{r}{\mu \left( 1 + e \cdot \cos(\nu) \right)} \cdot \cos(\omega + \nu) \cdot \left( P_W + U_W \right)
\]

The ascending node can be obtained in a similar fashion et al [3], as shown below.

\[
\frac{d\Omega}{dt} = \frac{h_y \cdot \frac{dh_x}{dt} - h_x \cdot \frac{dh_y}{dt}}{h_x^2 + h_y^2}
\]

\[
h_x \cdot \frac{dh_y}{dt} = h \cdot \sin(i) \cdot \sin(\Omega) \cdot r \left[ \left( P_W + U_W \right) \cdot \left( \sin(\omega + \nu) \cdot \sin(\Omega) - \cos(\omega + \nu) \cdot \cos(\Omega) \cdot \cos(i) \right) - \left( P_S + U_S \right) \cdot \cos(\Omega) \cdot \sin(i) \right]
\]

\[
h_y \cdot \frac{dh_x}{dt} = -h \cdot \sin(i) \cdot \cos(\Omega) \cdot r \left[ \left( P_W + U_W \right) \cdot \left( \sin(\omega + \nu) \cdot \cos(\Omega) + \cos(\omega + \nu) \cdot \sin(\Omega) \cdot \cos(i) \right) + \left( P_S + U_S \right) \cdot \left( \sin(\Omega) \cdot \sin(i) \right) \right]
\]

\[
h_x \cdot \frac{dh_y}{dt} - h_y \cdot \frac{dh_x}{dt} = h \cdot r \left[ \left( P_W + U_W \right) \cdot \sin(\omega + \nu) \cdot \sin(i) \right]
\]

\[
h_x^2 + h_y^2 = h^2 \cdot \sin(i)^2
\]

\[
\frac{d\Omega}{dt} = \frac{h \cdot r \left[ \left( P_W + U_W \right) \cdot \sin(\omega + \nu) \cdot \sin(i) \right]}{h^2 \cdot \sin(i)^2} = \frac{r \cdot \sin(\omega + \nu)}{\sin(i) \cdot h} \left( P_W + U_W \right)
\]

The ascending node rate can also be expressed as.

\[
\frac{d\Omega}{dt} = \left[ \frac{\sin(\omega + \nu)}{V_S \cdot \sin(i)} \right] \left( P_W + U_W \right) \quad \text{or} \quad \frac{d\Omega}{dt} = \sqrt{\frac{r \left( 1 + e \cdot \cos(\nu) \right)}{\mu} \cdot \left[ \frac{\sin(\omega + \nu)}{\sin(i) \cdot \left( 1 + e \cdot \cos(\nu) \right)} \right] \left( P_W + U_W \right)}
\]
The argument of perigee is more complicated to obtain since it is not directly obtained geometrically from the angular momentum vector. The equation is used to find the magnitude of the angular momentum, by taking its time derivative and solving for the time rate of change of the argument of perigee The derivations below et al [3] explain this process.

\[ h^2 = \mu r \left(1 + e \cos(v)\right) \]

\[ 2h \frac{dh}{dt} = \mu \left[ \frac{-e \sin(v)}{r} \frac{dv}{dt} + \frac{de \cos(v)}{dt} + \frac{dr}{dt} \left(1 + e \cos(v)\right) \right] \]

\[ \frac{dv}{dt} = \frac{2h \frac{dh}{dt} + r \frac{de \cos(v)}{dt} - \frac{dr}{dt} \left(1 + e \cos(v)\right)}{r e \sin(v)} \]

Since geometrically the time rate of change of the planar angle is related to the time rate of change of the ascending node angle et al [3] by.

\[ \frac{d\theta}{dt} = \frac{-d\Omega}{dt} \cdot \cos(i) \]

And the planar angle \( \theta \) is.

\[ \theta = \nu + \omega \quad \frac{d\nu}{dt} = \frac{d\omega}{dt} - \frac{d\theta}{dt} \]

The time derivative of the argument of perigee becomes

\[ \frac{d\omega}{dt} = \frac{-d\Omega}{dt} \cdot \cos(i) + \frac{2h \frac{dh}{dt} + r \frac{de \cos(v)}{dt} - \frac{dr}{dt} \left(1 + e \cos(v)\right)}{r e \sin(v)} \]

The only unknown thus far is the time derivative of the eccentricity, which is

\[ \frac{de}{dt} = \frac{2 \left(V_R^2 + V_S^2 - \frac{2\mu}{r}\right) h \frac{dh}{dt} + \left(2V_{R}\frac{dV_{R}}{dt} + 2V_{S}\frac{dV_{S}}{dt} + \frac{2\mu}{r^2} V_{R}\right) h^2}{2 \mu^2 \cdot e} \]

The state equations for the radial and perpendicular velocities can be used to obtain

\[ \frac{de}{dt} = \frac{h}{\mu^2 \cdot e} \left[ \left( V_{R}^2 + V_{S}^2 - \frac{2\mu}{r} \right) + h V_{S} \left( P_{S} + U_{S} \right) + h V_{R} \left( P_{R} + U_{R} \right) \right] \]
The final state equation is obtained for the time rate of change of the argument of perigee.

\[
\frac{d\omega}{dt} = \frac{-d\Omega}{dt}\cos(i) + \left(\frac{h}{\mu e}\right)^2 \frac{V_R}{\sin(v)} (P_R + U_R) + \frac{h}{\mu e \sin(v)} \left[ 2 + \frac{\cos(v)}{\mu e} \left[ \frac{r}{V_R^2 + V_S^2 - \frac{2\mu}{r}} + h V_S \right] \right] (P_S + U_S)
\]

These time rates of change can be obtained in other ways. Other forms of the equation are

\[
\frac{d\omega}{dt} = \frac{r(1 + e \cos(v))}{\mu} \left[ \frac{-\cos(v)}{e} (P_R + U_R) + \frac{\sin(v)(2 + e \cos(v))}{e(1 + e \cos(v))} (P_S + U_S) + \frac{\sin(\omega + v)}{\tan(i)(1 + e \cos(v))} (P_W + U_W) \right]
\]

or

\[
\frac{d\omega}{dt} = \frac{r V_S}{\mu} \left[ \frac{-\cos(v)}{e} (P_R + U_R) + \frac{\sin(v)(2 + e \cos(v))}{e(1 + e \cos(v))} (P_S + U_S) + \frac{\sin(\omega + v)}{\tan(i)(1 + e \cos(v))} (P_W + U_W) \right]
\]
APPENDIX D: J2 PERTURBATIONS
D. Derivation of J2 Accelerations


\[ U = \frac{\mu}{r} \left[ 1 - \sum_{k=2}^{\infty} \left( \frac{R_{\text{earth}}}{r} \right)^k \cdot J_k \cdot P_k(\sin(\phi_{\text{lat}})) \right] \]

Where \( P_k \) are Legendre polynomial functions of order \( k \), \( J_k \) is a constant multiplier of the Legendre polynomials, and \( \phi_{\text{lat}} \) is the latitude position of the satellite projected to the surface of the Earth. When calculating only the J2 perturbation effect, the gravity potential difference due to the Earth’s oblateness et al [3] is.

\[ U_{J2} = \frac{-\mu}{r} \left( \frac{R_{\text{earth}}}{r} \right)^2 \cdot J_2 \cdot P_2(\sin(\phi_{\text{lat}})) \]

In Legendre’s polynomials \( 2P_2 \sin \phi_{\text{lat}} = 2 - 3 \cos^2 \phi_{\text{lat}} \), and geometrically \( r^2 \cos^2 \phi_{\text{lat}} = r^2 - z^2 \).

Therefore, the equation becomes

\[ U_{J2} = \frac{-\mu}{r} \left( \frac{R_{\text{earth}}}{r} \right)^2 \cdot J_2 \cdot P_2 \left[ \frac{1}{2} \left( \frac{3z^2}{r^2} - 1 \right) \right] \]

The force on the satellites due to this effect can be obtained by taking the gradients of the potential et al [3].

\[ F_{J2} = \text{Grad}(U_{J2}) = \left( \frac{dU_{J2}}{dr} \right) e_R + \left( \frac{dU_{J2}}{dz} \right) e_z \]

\[ \frac{dU_{J2}}{dr} = -\mu \cdot R_{\text{earth}} \cdot J_2 \cdot \left( \frac{3}{2r^4} - \frac{15z^2}{2r^6} \right) \]

\[ \frac{dU_{J2}}{dz} = -\frac{3\mu \cdot R_{\text{earth}}^2 \cdot J_2 \cdot z}{r^5} \]

\[ F_{J2} = -3\mu \cdot R_{\text{earth}} \cdot J_2 \cdot \left[ \frac{2z}{r^5} e_z + \left( \frac{1}{2r^4} - \frac{5z^2}{2r^6} \right) e_R \right] \]

In order to relate these acceleration terms to the state equations, they must be translated into the RSW frame to obtain the perturbations for the radial (R), perpendicular (S) and out of plane (W) direction.
As can be seen from the equation above, \( e_z \) and \( e_R \) are in different frames. To translate \( e_z \) from the inertial XYZ to the body centered RSW frame, the equations below et al [3] are used.

\[
\vec{e}_z = \sin(i) \cdot \sin(\omega + \nu) \cdot \vec{e}_R + \sin(i) \cdot \cos(\omega + \nu) \cdot \vec{e}_S + \cos(i) \cdot \vec{e}_W
\]

\[
z = r \cdot \sin(\phi_{lat}) = r \cdot \sin(i) \cdot \sin(\omega + \nu)
\]

Replacing the above equations into the J2 gravitational acceleration equation results in the vector perturbation in the RSW frame.

\[
F_{J2} = \frac{-3 \cdot \mu \cdot R_{\text{earth}} \cdot J_2}{r^4} \left[ \frac{1}{2} \left( 1 - 3 \cdot (\sin(i) \cdot \sin(\omega + \nu))^2 \right) \vec{e}_R + \sin(i)^2 \cdot \sin(\omega + \nu) \cdot \vec{e}_S + \sin(i) \cdot \sin(\omega + \nu) \cdot \cos(i) \cdot \vec{e}_W \right]
\]

Hence, in matrix form, the perturbation equations et al [2] are

\[
F_{J2 \text{RSW}} = \begin{bmatrix}
\frac{-3 \cdot \mu \cdot R_{\text{earth}} \cdot J_2}{2 \cdot r^4} \\
\frac{-3 \cdot \mu \cdot R_{\text{earth}} \cdot J_2}{r^4} \\
\frac{-3 \cdot \mu \cdot R_{\text{earth}} \cdot J_2}{r^4}
\end{bmatrix}
\begin{bmatrix}
\left( 1 - 3 \cdot (\sin(i) \cdot \sin(\omega + \nu))^2 \right) \\
\sin(i)^2 \cdot \sin(\omega + \nu) \cdot \cos(\omega + \nu) \\
\sin(i) \cdot \sin(\omega + \nu) \cdot \cos(i)
\end{bmatrix}
\]
APPENDIX E: DRAG PERTURBATIONS
E. Derivation of Drag Accelerations

Vectorizing the drag force equation et al [1], [3] gives

\[
\vec{\text{Drag}} = \frac{C_d \rho \cdot A \cdot V^2}{2m} \cdot \frac{\vec{V}}{|\vec{V}|}
\]

Since the norm of the vector is equal to the magnitude, therefore

\[
V = |\vec{V}| \quad \text{Drag} = \frac{C_d \rho \cdot A \cdot V}{2m} \cdot V
\]

The velocity vector in the RSW frame is

\[
\vec{V} = \vec{V}_R e_R + \vec{V}_S e_S + \vec{V}_W e_W
\]

The magnitude and square of the velocity is

\[
V^2 = V_R^2 + V_S^2 + V_W^2 \quad V = \sqrt{V_R^2 + V_S^2 + V_W^2}
\]

From these relations the radial, perpendicular and out of plane components of drag used in the state equations can be obtained

\[
\text{Drag}_R = \frac{C_d \rho \cdot A}{2m} \sqrt{V_R^2 + V_S^2 + V_W^2} \cdot V_R
\]

\[
\text{Drag}_S = \frac{C_d \rho \cdot A}{2m} \sqrt{V_R^2 + V_S^2 + V_W^2} \cdot V_S
\]

\[
\text{Drag}_W = \frac{C_d \rho \cdot A}{2m} \sqrt{V_R^2 + V_S^2 + V_W^2} \cdot V_W
\]
APPENDIX F: SIMULATION MATLAB PROGRAM
%NUMERICAL CONSTANTS
Rearth=6378130.0;  %Radius of Earth (meters)
u=3.986e+14;  %Universal Gravitational Constant (m^3/sec)
J2=0.001082627;  %Earth's oblateness J2 perturbation term

%INITIAL SETTINGS TRAJECTORY
e1traj=0.01;  %Eccentricity
Anomdeg1traj=10.1;  %Initial True Anomaly (degrees)
Anom1traj=Anomdeg1traj*pi()/180.0;  %Initial True Anomaly (radians)
wdeg1traj=33.0;  %Initial argument of perigee(degrees)
w1traj=wdeg1traj*pi()/180.0;  %Initial argument of perigee(radians)
omdeg1traj=45.0;  %Initial right ascension node(degrees)
om1traj=omdeg1traj*pi()/180.0;  %Initial right ascension node(radians)
incldeg1traj=15.0;  %Initial inclination angle (degrees)
im1traj=incldeg1traj*pi()/180.0;  %Initial inclination angle (radians)
height1traj=300000.0;  %Initial height (meters)
R1traj=Rearth+height1traj;  %Initial Radius (Meters)
cosE1traj=(e1traj+cos(Anom1traj))/(1+e1traj*cos(Anom1traj));  %Initial cosine of eccentric anomaly
fpa1traj=acos(sqrt((1-e1traj^2)/(1-e1traj^2*cosE1traj^2)));  %Initial flight path angle (Radians)
if e1traj==0.0
    fpa1traj=0.0;
end
V1traj=sqrt(mu*(2-(1-e1traj^2)/(1+e1traj*cos(Anom1traj)))/R1traj);  %Initial Velocity (m/sec)
VS1traj=V1traj*cos(fpa1traj);
VR1traj=V1traj*sin(fpa1traj);
h1traj=R1traj*VS1traj;

%INITIAL SETTINGS SATELLITE 1
m1=15.0;  %Satellite mass (kg)
A1=0.0001;  %Cross-Sectional Area (m^2)
Cd1=0.5;  %Coefficient of Drag
p1delt1d=1000;
outofp1delt1d=1000;
el1(1)=0.01;  %Eccentricity
Anomdeg1=10.1;  %Initial True Anomaly (degrees)
Anom1=Anomdeg1*pi()/180.0;  %Initial True Anomaly (radians)
wdeg1=33.0;  %Initial argument of perigee(degrees)
w1=wdeg1*pi()/180.0;  %Initial argument of perigee(radians)
omdeg1=45.0;  %Initial right ascension node(degrees)
om1=omdeg1*pi()/180.0;  %Initial right ascension node(radians)
incldeg1=15.0;  %Initial inclination angle (degrees)
im1=incldeg1*pi()/180.0;  %Initial inclination angle (radians)
height1=300000;
R1=Rearth+height1;
cosE1=(e1(1)+cos(Anom1))/((1+e1(1)*cos(Anom1)));  %Initial cosine of eccentric anomaly
fpa1=acos(sqrt((1-e1(1)^2)/(1-e1(1)^2*cosE1^2)));  %Initial flight path angle (Radians)
if el1(1)==0.0
    fpa1=0.0;
end
V1=sqrt(mu*(2-(1-e1(1)^2)/(1+e1(1)*cos(Anom1)))/R1);  %Initial Velocity (m/sec)
VS1 = V1 * cos(fpa1);
VR1 = V1 * sin(fpa1);

% INITIAL SETTINGS SATELLITE 2
m2 = 15.0; % Satellite mass (kg)
A2 = 0.0001; % Cross-Sectional Area (m^2)
Cd2 = 0.5; % Coefficient of Drag
pldeltd = 1000;
outofpldeltd = 1000;
e2(1) = 0.01; % Eccentricity
Anomdeg2 = 10.1; % Initial True Anomaly (degrees)
Anom2 = Anomdeg2 * pi() / 180.0; % Initial True Anomaly (radians)
wdeg2 = 33.0; % Initial argument of perigee (degrees)
w2(1) = wdeg2 * pi() / 180.0; % Initial argument of perigee (radians)

% INITIAL CONDITIONS SATELLITE 1

% Initial state conditions
r11(1) = R1;
r12(1) = VS1;
r13(1) = VR1;
r14(1) = Anom1;
theta1(1) = w1(1) + r14(1);

% Initial Thrust
TS1 = 0.0;
TR1 = 0.0;
TW1 = 0.0;
US1(1) = TS1;
\[ UR1(1) = TR1; \]
\[ UW1(1) = TW1; \]
\[ U1(1) = \sqrt{US1(1)^2 + UR1(1)^2 + UW1(1)^2}; \]

%Initial Perturbations
\[ sat = 1; \]
\[ k = 0; \]
\[ Density; \]
\[ dik1 = \sqrt{r11(1)^3 / (\mu * (1 + e1(1) * \cos(r14(1))))} * \sin(w1(1) + r14(1)) * \cos(w1(1) + r14(1)); \]
\[ P1J2W = -(3/2) * \mu * J2 * (Rearth/r11(1)^2)^2 * \sin(2*incl1(1)) * \sin(w1(1) + r14(1)); \]
\[ qV1 = -B*exp(A*((r11(1)-Rearth-Ho)/Ho)^C) * Cd1*A1/(2*m1); \]
\[ V1w = dik1*P1J2W; \]
\[ V1ws = 0.0; \]
%while V1w~=V1ws
\[ while \ abs(V1ws-V1w)>0.00001 \]
\[ V1ws = V1w; \]
\[ V1w = V1ws*(-1/dik1+qV1*sqrt(VR1^2+VS1^2+V1ws^2))/sqrt(VR1^2+VS1^2+V1ws^2); \]
end
\[ P1J2R = -(3/2) * \mu * J2 * (Rearth/r11(1)^2)^2 * (1-3*\sin(incl1(1))^2*\sin(w1(1) + r14(1))^2); \]
\[ P1J2S = -(3/2) * \mu * J2 * (Rearth/r11(1)^2)^2 * \sin(incl1(1))^2*\sin(2*(w1(1)+r14(1))); \]
\[ Drag1R = -sign(r13(1))*B*exp(A*((r11(1)-Rearth-Ho)/Ho)^C)*sqrt(r12(1)^2+r13(1)^2+V1w^2)*r13(1)*Cd1*A1/(2*m1); \]
\[ Drag1S = -sign(r12(1))*B*exp(A*((r11(1)-Rearth-Ho)/Ho)^C)*sqrt(r12(1)^2+r13(1)^2+V1w^2)*r12(1)*Cd1*A1/(2*m1); \]
\[ Drag1W = -sign(V1w)*B*exp(A*((r11(1)-Rearth-Ho)/Ho)^C)*sqrt(r12(1)^2+r13(1)^2+V1w^2)*V1w*Cd1*A1/(2*m1); \]
\[ PertR1(1) = P1J2R+Drag1R; \]
\[ PertS1(1) = P1J2S+Drag1S; \]
\[ PertW1(1) = P1J2W+Drag1W; \]
\[ Pert1(1) = \sqrt{PertR1(1)^2+PertS1(1)^2+PertW1(1)^2}; \]

%Initial Force on Satellite
\[ ForceR1(1) = r12(1)^2/r11(1)-\mu/r11(1)^2+PertR1(1)+TR1; \]
\[ ForceS1(1) = -r12(1)*r13(1)/r11(1)+PertS1(1)+TS1; \]
\[ ForceW1(1) = PertW1(1)+TW1; \]

%Initial Energy
\[ Energy1(1) = (r12(1)^2+r13(1)^2)/2-\mu/r11(1); \]
\[ dEnergy1dt(1) = r13(1) * (PertR1(1)+UR1(1))+r12(1) * (PertS1(1)+US1(1)); \]
\[ Energy1d = r12(1)^2/2.0-\mu/r11(1); \]

%Initial Angular Rates
\[ domg1dt(1) = \sqrt{r11(1)/(\mu * (1+e1(1) * \cos(r14(1))))} * (\sin(w1(1) + r14(1))/\sin(incl1(1)))* (PertW1(1)+U1(1)); \]
\[ if e1(1)<0.0000001 \]
\[ dw1dt(1) = -domg1dt(1) * \cos(incl1(1)); \]
\[ else \]
\[ dw1dt(1) = \]
\[ domg1dt(1) * \cos(incl1(1)) + \sqrt{(r11(1) * (1+e1(1) * \cos(r14(1))))/\mu} * ((\sin(r14(1)) * (2+e1(1) * \cos(r14(1)))/e1(1)) * (1+e1(1) * \cos(r14(1)))) * (PertS1(1)+US1(1)) - \]
\[ (\cos(r14(1))/e1(1)) * (PertR1(1)+UR1(1))); \]
end
\[ dincl1dt(1) = \sqrt{r11(1)/(\mu * (1+e1(1) * \cos(r14(1))))} * \cos(w1(1) + r14(1)) * (UW1(1)); \]
\[ dr14dt(1) = r12(1)/r11(1); \]
\[ dthetal1dt(1) = dw1dt(1)+dr14dt(1); \]
%Initial Orbital Position Coordinates (PQ)
P1(1)=r11(1)*cos(r14(1));
Q1(1)=r11(1)*sin(r14(1));

%Initial Orbital Velocity Coordinates(PQ)
VP1(1)=-sqrt(mu/(r11(1)*(1+e1(1)*cos(r14(1))))))*sin(r14(1));
VQ1(1)=sqrt(mu/(r11(1)*(1+e1(1)*cos(r14(1)))))*(e1(1)+cos(r14(1)));
VW1(1)=V1w;

%Transpose matrix from Orbital (PQ) to Inertial (IJK) Coordinates
T111=cos(omg1(1))*cos(w1(1))-sin(omg1(1))*sin(w1(1))*cos(incl1(1));
T112=-cos(omg1(1))*sin(w1(1))-sin(omg1(1))*cos(w1(1))*cos(incl1(1));
T113=sin(omg1(1))*cos(incl1(1));
T121=sin(omg1(1))*cos(w1(1))+cos(omg1(1))*sin(w1(1))*cos(incl1(1));
T122=-sin(omg1(1))*sin(w1(1))+cos(omg1(1))*cos(w1(1))*cos(incl1(1));
T123=-cos(omg1(1))*sin(incl1(1));
T131=sin(w1(1))*sin(incl1(1));
T132=cos(w1(1))*sin(incl1(1));
T133=cos(incl1(1));

%Initial Inertial Position Coordinates (XYZ)
x1(1)=P1(1)*T111+Q1(1)*T112;
y1(1)=P1(1)*T121+Q1(1)*T122;
z1(1)=P1(1)*T131+Q1(1)*T132;

%Initial Inertial Velocity Coordinates (XYZ)
Vx1(1)=VP1(1)*T111+VQ1(1)*T112+VW1(1)*T113;
Vy1(1)=VP1(1)*T121+VQ1(1)*T122+VW1(1)*T123;
Vz1(1)=VP1(1)*T131+VQ1(1)*T132+VW1(1)*T133;

%INITIAL CONDITIONS SATELLITE 2

%Initial state conditions
r21(1)=R2;
r22(1)=VS2;
r23(1)=VR2;
r24(1)=Anom2;
theta2(1)=w2(1)+r24(1);

%Initial Thrust
TR2=0.0;
TS2=0.0;
TW2=0.0;
US2(1)=0.0;
UR2(1)=0.0;
UW2(1)=0.0;
U2(1)=sqrt(US2(1)^2+UR2(1)^2+UW2(1)^2);

%Initial Perturbations
sat=2;
k=0;
Density:
\[ \text{dik2} = \sqrt{\frac{(r21(1)^3)}{(\mu*(1+e2(1)*\cos(r24(1)))))*\sin(w2(1)+r42(1))}\] \[*\cos(w2(1)+r24(1))}; \]

\[ P2J2W = -\frac{3}{2}\mu*J2*(Rearth/r21(1)^2)^2*\sin(2*incl2(1))*\sin(w2(1)+r24(1)); \]

\[ qV2= -B*exp(A*((r21(1)-Rearth-Ho)/Ho)^C)*Cd2*A2/(2*m2); \]

\[ V2w = dik2*P2J2W; \]

\[ V2ws = 0.0; \]

\%while V2ws~=V2w

\%while abs(V2ws-V2w)>0.00001

\[ V2ws = V2ws; \]

\[ V2w = V2ws*\left(-\frac{1}{dik2+qV2*sqrt(VR2^2+VS2^2+V2ws^2)}\right)+P2J2W/(-1/dik2+qV2*sqrt(VR2^2+VS2^2+2*V2ws^2)/sqrt(VR2^2+VS2^2+V2ws^2)); \]

end

\%Initial Force on Satellite

\[ \text{ForceR2}(1) = r22(1)^2/r21(1) - \mu/r21(1)^2 + \text{PertR2}(1) + \text{TR2}; \]

\[ \text{ForceS2}(1) = -r22(1)*r23(1)/r21(1) + \text{PertS2}(1) + \text{TS2}; \]

\[ \text{ForceW2}(1) = \text{PertW2}(1) + \text{TW2}; \]

\%Initial Energy

\[ \text{Energy2}(1) = \frac{(r22(1)^2+r23(1)^2)}{2} - \frac{\mu}{r21(1)}; \]

\[ \text{dEnergy2dt}(1) = r23(1)*(\text{PertR2}(1)+\text{UR2}(1))+r22(1)*(\text{PertS2}(1)+\text{US2}(1)); \]

\[ \text{Energy2d} = \frac{(r22(1)^2)}{2.0} - \frac{\mu}{r21(1)}; \]

\%Initial Angular Rates

\[ \text{domg2dt}(1) = \sqrt{\frac{r21(1)}{(\mu*(1+e2(1)*\cos(r24(1)))))*\sin(w2(1)+r24(1))}\] \[*/\sin(incl2(1))}\] \[*(\text{PertW2}(1)+\text{UW2}(1)); \]

\[ if \ e2(1) < 0.0000001 \]

\[ \text{dw2dt}(1) = -\text{domg2dt}(1)*\cos(incl2(1)); \]

\[ else \]

\[ \text{dw2dt}(1) = -\text{domg2dt}(1)*\cos(incl2(1))+\sqrt{(r21(1)^2)}*\cos(2*e2(1)*\cos(r24(1)))\] \[*/(e2(1)^2+e2(1)*\cos(r24(1))))\] \[*/(\text{PertS2}(1)+\text{US2}(1))\] \[*/(\cos(r24(1))+e2(1))\] \[*/(\text{PertR2}(1)+\text{UR2}(1))); \]

end

\[ \text{dincl2dt}(1) = \sqrt{(r21(1)^2)}*\cos(2*e2(1)*\cos(2*e2(1))}\] \[*/\cos(w2(1)+r24(1))\] \[*/(\text{UW2}(1)); \]

\[ \text{dr24dt}(1) = r22(1)/r21(1); \]

\[ \text{dtheta2dt}(1) = \text{dw2dt}(1)+\text{dr24dt}(1); \]

\%Initial Orbital Position Coordinates (PQ)

\[ P2(1) = r21(1)\] \[*\cos(r24(1)); \]

\[ Q2(1) = r21(1)*\sin(r24(1)); \]

\%Initial Orbital Velocity Coordinates (PQ)

\[ \text{VP2}(1) = -\sqrt{(\mu*(1+e2(1)\] \[*\cos(r24(1))))}\] \[*/\sin(r24(1)); \]
VQ2(1) = \sqrt{\mu/(r21(1)*(1+e2(1)\cos(r24(1))))} \cdot (e2(1)+\cos(r24(1)));
VW2(1) = V2w;

% Transpose matrix from Orbital (PQ) to Inertial (IJK) Coordinates
T211 = \cos(omg2(1))\cos(w2(1)) - \sin(omg2(1))\sin(w2(1))\cos(incl2(1));
T212 = -\cos(omg2(1))\sin(w2(1)) - \sin(omg2(1))\cos(w2(1))\cos(incl2(1));
T213 = \sin(omg2(1))\cos(incl2(1));
T221 = \sin(omg2(1))\cos(w2(1)) + \cos(omg2(1))\sin(w2(1))\cos(incl2(1));
T222 = -\sin(omg2(1))\sin(w2(1)) + \cos(omg2(1))\cos(w2(1))\cos(incl2(1));
T223 = -\cos(omg2(1))\sin(incl2(1));
T231 = \sin(w2(1))\sin(incl2(1));
T232 = \cos(w2(1))\sin(incl2(1));
T233 = \cos(incl2(1));

% Initial Inertial Position Coordinates (XYZ)
x2(1) = P2(1) \cdot T211 + Q2(1) \cdot T212;
y2(1) = P2(1) \cdot T221 + Q2(1) \cdot T222;
z2(1) = P2(1) \cdot T231 + Q2(1) \cdot T232;

% Initial Inertial Velocity Coordinates (XYZ)
Vx2(1) = VP2(1) \cdot T211 + VQ2(1) \cdot T212 + VW2(1) \cdot T213;
Vy2(1) = VP2(1) \cdot T221 + VQ2(1) \cdot T222 + VW2(1) \cdot T223;
Vz2(1) = VP2(1) \cdot T231 + VQ2(1) \cdot T232 + VW2(1) \cdot T233;

% INITIAL CONDITIONS TRAJECTORY

% Initial state conditions
r11traj(1) = R1traj;
r12traj(1) = VS1traj;
r13traj(1) = VR1traj;
r14traj(1) = Anom1traj;

% Initial Energy
Etraj(1) = (r12traj(1)^2 + r13traj(1)^2)/2.0 - \mu/r11traj(1);

% Initial Eccentricity
etraj(1) = sqrt(1.0 + 2.0 * Etraj(1) * (r11traj(1)*r12traj(1)/\mu)^2);

% Orbital Position Coordinates (PQ)
P1traj(1) = r11traj(1) \cdot \cos(r14traj(1));
Q1traj(1) = r11traj(1) \cdot \sin(r14traj(1));

% Orbital Velocity Coordinates (PQ)
VP1traj(1) = \sqrt{\mu/(r11traj(1)*(1+etraj\cdot\cos(r14traj(1))))} \cdot \sin(r14traj(1));
VQ1traj(1) = \sqrt{\mu/(r11traj(1)*(1+etraj\cdot\cos(r14traj(1))))} \cdot (etraj + \cos(r14traj(1)));
VW1traj(1) = 0.0;

% Transpose matrix from Orbital (PQ) to Inertial (IJK) Coordinates
T111 = \cos(omgl1traj)\cos(wl1traj) - \sin(omgl1traj)\sin(wl1traj)\cos(incl1traj);
T112 = -\cos(omgl1traj)\sin(wl1traj) - \sin(omgl1traj)\cos(wl1traj)\cos(incl1traj);
T113 = \sin(omgl1traj)\cos(incl1traj);
T121 = \sin(omgl1traj)\cos(wl1traj) + \cos(omgl1traj)\sin(wl1traj)\cos(incl1traj);
T122 = -\sin(omgl1traj)\sin(wl1traj) + \cos(omgl1traj)\cos(wl1traj)\cos(incl1traj);
T123 = -\cos(omgl1traj)\sin(incl1traj);
T131 = \sin(wl1traj)\sin(incl1traj);
T132=cos(w1traj)*sin(incl1traj);
T133=cos(incl1traj);

%Inertial Position Coordinates (IJK)
x1traj(1)=P1traj(1)*T111+Q1traj(1)*T112;
y1traj(1)=P1traj(1)*T121+Q1traj(1)*T122;
z1traj(1)=P1traj(1)*T131+Q1traj(1)*T132;

%Inertial Velocity Coordinates (XYZ)
Vx1traj(1)=VP1traj(1)*T111+VQ1traj(1)*T112+VW1traj(1)*T113;
Vy1traj(1)=VP1traj(1)*T121+VQ1traj(1)*T122+VW1traj(1)*T123;
Vz1traj(1)=VP1traj(1)*T131+VQ1traj(1)*T132+VW1traj(1)*T133;

%Initial Relative States Satellite 1
r11d(1)=r11(1)-r11traj(1);
r12d(1)=r12(1)-r12traj(1);
r13d(1)=r13(1)-r13traj(1);
r14d(1)=r14(1)-r14traj(1);
w1d(1)=w1(1)-w1traj;
theta1d(1)=r14(1)+w1(1)-r14traj(1)-w1traj;
incl1d(1)=incl1(1)-incl1traj;
x1d(1)=x1(1)-x1traj(1);
y1d(1)=y1(1)-y1traj(1);
z1d(1)=z1(1)-z1traj(1);
d1(1)=sqrt((x1(1)-x1traj(1))^2+(y1(1)-y1traj(1))^2+(z1(1)-z1traj(1))^2);
dorbit1(1)=sqrt((P1(1)-P1traj(1))^2+(Q1(1)-Q1traj(1))^2);
dEnergy1(1)=Energy1(1)-Etraj(1);
de1(1)=e1(1)-e1traj(1);

%Initial Relative States Satellite 2
r21d(1)=r21(1)-r11(1);
r22d(1)=r22(1)-r12(1);
r23d(1)=r23(1)-r13(1);
r24d(1)=r24(1)-r14(1);
w2d(1)=w2(1)-w1(i);
theta2d(1)=r24(1)+w2(1)-r14(1)-w1(1);
incl2d(1)=incl1(1)-incl1traj;
x2d(1)=x2(1)-x1traj(1);
y2d(1)=y2(1)-y1traj(1);
z2d(1)=z2(1)-z1traj(1);
d2(1)=sqrt((x2d(1))^2+(y2d(1))^2+(z2d(1))^2);
dorbit2(1)=sqrt((P2(1)-P1(1))^2+(Q2(1)-Q1(1))^2);
dEnergy2(1)=Energy2(1)-Energy1(1);
de2(1)=e2(1)-e1(1);

%Loop Initialization
while true
    if r14(i)<r14(1)+2*orbits*pi

%SIMULATED TRAJECTORY CONDITIONS SATELLITE 1

%Classical Runge-Kutta Fourth Order State equations
% Initializing step 1
k11traj(1)=r13traj(i);
k12traj(1)=-r12traj(i)*r13traj(i)/r11traj(i);
k13traj(1)=r12traj(i)^2/r11traj(i)-mu/r11traj(i)^2;
k14traj(1)=r12traj(i)/r11traj(i);

r11ktraj(1)=r11traj(i)+wt*k11traj(1)/2;
r12ktraj(1)=r12traj(i)+wt*k12traj(1)/2;
r13ktraj(1)=r13traj(i)+wt*k13traj(1)/2;
r14ktraj(1)=r14traj(i)+wt*k14traj(1)/2;

% Initializing step 2
k11traj(2)=r13ktraj(1);
k12traj(2)=-r12ktraj(1)*r13ktraj(1)/r11ktraj(1);
k13traj(2)=r12ktraj(1)^2/r11ktraj(1)-mu/r11ktraj(1)^2;
k14traj(2)=r12ktraj(1)/r11ktraj(1);

r11ktraj(2)=r11traj(i)+wt*k11traj(2)/2;
r12ktraj(2)=r12traj(i)+wt*k12traj(2)/2;
r13ktraj(2)=r13traj(i)+wt*k13traj(2)/2;
r14ktraj(2)=r14traj(i)+wt*k14traj(2)/2;

% Initializing step 3
k11traj(3)=r13ktraj(2);
k12traj(3)=-r12ktraj(2)*r13ktraj(2)/r11ktraj(2);
k13traj(3)=r12ktraj(2)^2/r11ktraj(2)-mu/r11ktraj(2)^2;
k14traj(3)=r12ktraj(2)/r11ktraj(2);

r11ktraj(3)=r11traj(i)+wt*k11traj(3);
r12ktraj(3)=r12traj(i)+wt*k12traj(3);
r13ktraj(3)=r13traj(i)+wt*k13traj(3);
r14ktraj(3)=r14traj(i)+wt*k14traj(3);

% Initializing step 4
k11traj(4)=r13ktraj(3);
k12traj(4)=-r12ktraj(3)*r13ktraj(3)/r11ktraj(3);
k13traj(4)=r12ktraj(3)^2/r11ktraj(3)-mu/r11ktraj(3)^2;
k14traj(4)=r12ktraj(3)/r11ktraj(3);

r11traj(i+1)=r11traj(i)+wt*(k11traj(1)+2.0*(k11traj(2)+k11traj(3))+k11traj(4))/6;
r12traj(i+1)=r12traj(i)+wt*(k12traj(1)+2.0*(k12traj(2)+k12traj(3))+k12traj(4))/6;
r13traj(i+1)=r13traj(i)+wt*(k13traj(1)+2.0*(k13traj(2)+k13traj(3))+k13traj(4))/6;
r14traj(i+1)=r14traj(i)+wt*(k14traj(1)+2.0*(k14traj(2)+k14traj(3))+k14traj(4))/6;

% Energy
Etraj(i+1)=(r12traj(i+1)^2+r13traj(i+1)^2)/2.0-mu/r11traj(i+1);

% Eccentricity
eltraj(i+1)=sqrt(1.0+2.0*Etraj(i+1)*(r11traj(i+1)*r12traj(i+1)/mu)^2);

% Orbital Position Coordinates (PQ)
P1traj(i+1)=r11traj(i+1)*cos(r14traj(i+1));
Q1traj(i+1)=r11traj(i+1)*sin(r14traj(i+1));

%Orbital Velocity Coordinates (PQ)
VP1traj(i+1)=-sqrt(mu/r11traj(i+1))*sin(r14traj(i+1));
VQ1traj(i+1)=sqrt(mu/r11traj(i+1))*cos(r14traj(i+1));
VW1traj(i+1)=0.0;

%Transpose matrix from Orbital (PQ) to Inertial (IJK) Coordinates
T111=cos(omg1traj)*cos(w1traj)-sin(omg1traj)*sin(w1traj)*cos(incl1traj);
T112=-cos(omg1traj)*sin(w1traj)-sin(omg1traj)*cos(w1traj)*cos(incl1traj);
T113=sin(omg1traj)*cos(incl1traj);
T121=sin(omg1traj)*cos(w1traj)+cos(omg1traj)*sin(w1traj)*cos(incl1traj);
T122=-sin(omg1traj)*sin(w1traj)+cos(omg1traj)*cos(w1traj)*cos(incl1traj);
T123=-cos(omg1traj)*sin(incl1traj);
T131=sin(w1traj)*sin(incl1traj);
T132=cos(w1traj)*sin(incl1traj);
T133=cos(incl1traj);

%Inertial Position Coordinates (IJK)
x1traj(i+1)=P1traj(i+1)*T111+Q1traj(i+1)*T112;
y1traj(i+1)=P1traj(i+1)*T121+Q1traj(i+1)*T122;
z1traj(i+1)=P1traj(i+1)*T131+Q1traj(i+1)*T132;

%Inertial Velocity Coordinates (XYZ)
Vx1traj(i+1)=VP1traj(i+1)*T111+VQ1traj(i+1)*T112+VW1traj(i+1)*T113;
Vy1traj(i+1)=VP1traj(i+1)*T121+VQ1traj(i+1)*T122+VW1traj(i+1)*T123;
Vz1traj(i+1)=VP1traj(i+1)*T131+VQ1traj(i+1)*T132+VW1traj(i+1)*T133;

%CONTROLS INPUT SATELLITE 1
TR1=0.0;
TS1=0.0;
TW1=0.0;

%Out of plane maneuver
if r14(i)+w1(i)>2*(n1+1)*pi()
    n1=n1+1;
end

Komgl=0.5;
wtomgl=1.0;
Fomgl=(omg1(i)-omgltraj)*(Komgl-1.0);
UWomgl=Fomgl*r12(i)*sin(incl1(i))/(wtomgl*sin(w1(i)+r14(i)))-PertW1(i);

Ki1=0.5;
wti1=10.0;
Fi1=(incl1(i)-incl1traj)*(Ki1-1.0);
UWincl1=Fi1*r12(i)/(wti1*cos(w1(i)+r14(i)));

if r14(i)+w1(i)<(n1*360.0+20.0)*pi()/180.0 |
r14(i)+w1(i)>(n1*360.0+340.0)*pi()/180.0
    TW1=UWincl1;
end
if r14(i)+wl(i)<(n1*360.0+70.0)*pi()/180.0 &
  r14(i)+wl(i)>(n1*360.0+20.0)*pi()/180.0
  if (omg1(i)<omg1traj & incl1(i)<incl1traj) | (omg1(i)>omg1traj &
  incl1(i)>incl1traj)
    if abs(UWincl1)>abs(UWomg1)
      TW1=UWomg1;
    else
      TW1=UWincl1;
    end
  end
end

if r14(i)+wl(i)<(n1*360.0+110.0)*pi()/180.0 &
  r14(i)+wl(i)>(n1*360.0+70.0)*pi()/180.0
  TW1=UWomg1;
end

if r14(i)+wl(i)<(n1*360+160.0)*pi()/180.0 &
  r14(i)+wl(i)>(n1*360+110.0)*pi()/180.0
  if (omg1(i)>omg1traj & incl1(i)<incl1traj) | (omg1(i)<omg1traj &
  incl1(i)>incl1traj)
    if abs(UWincl1)>abs(UWomg1)
      TW1=UWomg1;
    else
      TW1=UWincl1;
    end
  end
end

if r14(i)+wl(i)<(n1*360.0+200.0)*pi()/180.0 &
  r14(i)+wl(i)>(n1*360.0+160.0)*pi()/180.0
  TW1=UWomg1;
end

if r14(i)+wl(i)<(n1*360.0+250.0)*pi()/180.0 &
  r14(i)+wl(i)>(n1*360.0+200.0)*pi()/180.0
  if (omg1(i)<omg1traj & incl1(i)<incl1traj) | (omg1(i)>omg1traj &
  incl1(i)>incl1traj)
    if abs(UWincl1)>abs(UWomg1)
      TW1=UWomg1;
    else
      TW1=UWincl1;
    end
  end
end

if r14(i)+wl(i)<(n1*360.0+290.0)*pi()/180.0 &
  r14(i)+wl(i)>(n1*360.0+250.0)*pi()/180.0
  TW1=UWomg1;
end

if abs(TW1)>0.1
  TW1=sign(TW1)*0.1;
end
%Planar maneuvering

%Radial Distance
Kr1=0.707;
w8r1=10.0;
ctdistr1=1000.0;
if abs(r11(i)-r11traj(i))>ctdistr1
    Kr1=1.0-(1.0-Kr1)*ctdistr1/abs(r11(i)-r11traj(i));
end
FF1=(r11(i)-r11traj(i))*(Kr1-1.0);
TR1=(FF1/w8r1-r13(i)+r13traj(i))*2/w8r1-
r12(i)^2/r11(i)+r12traj(i)^2/r11traj(i)+mu*(1.0/r11(i)^2-1.0/r11traj(i)^2)-
PertR1(i);

%Energy
KE1=0.8;
w8E1=10.0;
ctEnergy1=5000.0;
if abs(Energy1(i)-Etraj(i))>ctEnergy1
    KE1=1.0-(1.0-KE1)*ctEnergy1/abs(Energy1(i)-Etraj(i));
end
FE1=(Energy1(i)-Etraj(i))*(KE1-1.0);
TS1=(FE1/w8E1-r13(i)*(PertR1(i)+TR1))/r12(i)-PertS1(i);

%Argument of Perigee
w8w1=10.0;
Kw1=0.8;
cdistw1=1000.0;
ctw1=acos(1-cdistw1^2/(2.0*abs(r11(i))));
if abs(w1(i)-w1traj)>ctw1
    Kw1=1.0-(1.0-Kw1)*ctw1/abs(w1(i)-w1traj);
end
Fw1=(w1(i)-w1traj)*(Kw1-1.0); for i=1:length(r14)
    if r14(i)<(j1*360.0+120.0)*pi()/180.0 & r14(i)>(j1*360.0+60.0)*pi()/180.0
        TS1=USw1;
        end
    if r14(i)<(j1*360.0+300.0)*pi()/180.0 & r14(i)>(j1*360.0+240.0)*pi()/180.0
        TS1=USw1;
        end
    end
US1(i+1)=TS1;
UR1(i+1)=TR1;
UW1(i+1)=TW1;
U1(i+1)=sqrt(US1(i+1)^2+UR1(i+1)^2+UW1(i+1)^2);

%CONTROLS INPUT SATELLITE 2

TR2=0.0;
TS2=0.0;
TW2=0.0;

%Out of plane maneuver
if r24(i)+w2(i)>2*(n2+1)*pi()
    n2=n2+1;
end

Komg2=0.5;
wtomg2=1.0;
Fomg2=(omg2(i)-omg1traj)*(Komg2-1.0);
UWomg2=Fomg2*r22(i)*sin(incl2(i))/(wtomg2*sin(w2(i)+r24(i)))-PertW2(i);

Ki2=0.5;
wti2=10.0;
Fi2=(incl2(i)-incl1traj)*(Ki2-1.0);
UWincl2=Fi2*r22(i)/(wti2*cos(w2(i)+r24(i)));

if r24(i)+w2(i)<(n2*360.0+20.0)*pi()/180.0 |
    TW2=UWincl2;
end

if r24(i)+w2(i)<(n2*360.0+20.0)*pi()/180.0 &
    r24(i)+w2(i)>(n2*360.0+340.0)*pi()/180.0
    if omg2(i)<omg1traj & incl2(i)<incl1traj
        if abs(UWincl2)>abs(UWomg2)
            TW2=UWomg2;
        else
            TW2=UWincl2;
        end
    end
if omg2(i)>omg1traj & incl2(i)>incl1traj
    if abs(UWincl2)>abs(UWomg2)
        TW2=UWomg2;
    else
        TW2=UWincl2;
    end
end

if r24(i)+w2(i)<(n2*360.0+110.0)*pi()/180.0 &
    r24(i)+w2(i)>(n2*360.0+150.0)*pi()/180.0
    if omg2(i)<omg1traj & incl2(i)<incl1traj
        if abs(UWincl2)>abs(UWomg2)
            TW2=UWomg2;
        else
            TW2=UWincl2;
        end
    end
if omg2(i)>omg1traj & incl2(i)>incl1traj
    if abs(UWincl2)>abs(UWomg2)
        TW2=UWomg2;
    else
        TW2=UWincl2;
    end
end
if abs(UWincl2)>abs(UWomg2)
   TW2=UWomg2;
else
   TW2=UWincl2;
end
end

del 200.0*pi()/180.0 &
   TW2=UWincl2;
end

if r24(i)+w2(i)<(n2*360.0+250.0)*pi()/180.0 &
   if omg2(i)<omg1traj & incl2(i)<incl1traj
      if abs(UWincl2)>abs(UWomg2)
         TW2=UWomg2;
      else
         TW2=UWincl2;
      end
   end
   if omg2(i)>omg1traj & incl2(i)>incl1traj
      if abs(UWincl2)>abs(UWomg2)
         TW2=UWomg2;
      else
         TW2=UWincl2;
      end
   end
end

if abs(TW2)>0.1
   TW2=sign(TW2)*0.1;
end

%Planar maneuvering

%Radial Distance
Kr2=0.707;
w8r2=10.0;
ctdistr2=10.0;
if abs(r21(i)-r11(i))>ctdistr2
   Kr2=1.0-(1.0-Kr1)*ctdistr2/abs(r21(i)-r11(i));
end
FR2=(r21(i)-r11(i))*(Kr2-1.0);
TR2=2*(FR2-w8r2*(r23(i)-r13(i)))/w8r2^2-
r22(i)^2/r21(i)+r12(i)^2/r11(i)+mu*(1.0/r21(i)^2-1.0/r11(i)^2)-
PertR2(i)+PertR1(i)+TR1;

%Planar Angle


\[ \text{KS2} = 0.707; \]
\[ \text{w8S2} = 10.0; \]
\[ \text{desdist} = 1000.0; \]

\[ \text{deltaW} = r11(i) * (\sin(\text{omg2}(i)) * \sin(\text{inc12}(i)) * (\cos(w1(i)+r14(i)) * \cos(\text{omg1}(i)) - \cos(\text{omg2}(i)) * \sin(\text{inc12}(i)) * (\cos(w1(i)+r14(i)) * \sin(\text{omg1}(i)) + \sin(w1(i)+r14(i)) * \cos(\text{omg1}(i))) \right) \]
\[ r1lon2 = \sqrt{r11(i)^2 - \text{deltaW}^2}; \]
\[ \text{deltaPlane} = \text{desdist}^2 - \text{deltaW}^2; \]
\[ \text{theta1on2} = \arctan(\cos(\text{inc1}(i)-\text{inc2}(i)) * \tan(r14(i)+w1(i))); \]
\[ \text{ThetaS2} = \arccos((r1lon2^2+r21(i)^2-\text{deltaPlane})/(2*r1lon2*r21(i)))); \]
\[ \text{cttheta1} = \arccos(1-\text{desdist}^2/(2.0*abs(r21(i)))); \]
\[ \text{if abs(r24(i)+w2(i)-r14(i)-w1(i)-ThetaS2) > cttheta1} \]
\[ \text{KS2} = 1.0 - (1.0-\text{KS2})*\text{cttheta1}/\text{abs(r24(i)+w2(i)-r14(i)-w1(i)-ThetaS2}); \]
\[ \text{end} \]
\[ \text{FS2} = (r24(i) + w2(i)-r14(i)-w1(i)-ThetaS2) * (KS2-1.0); \]

\[ \text{if e1(i)<0.0000001 | e2(i)<0.0000001} \]
\[ \text{if e1(i)<0.0000001} \]
\[ TS2 = 2.0*\mu*r21(i) * e2(i) * (1.0+e2(i) * \cos(r24(i))) * (\text{FS2/w8S2-} \]
\[ (r21(i) * r22(i) * (\mu/r21(i)^2 - \cos(r24(i)) * (\text{PertR2(i) + TR2})/e2(i) + \sin(w2(i)+r24(i)) * (\text{PertW2(i) + TW2})/(\tan(\text{inc12}(i)) * (1.0 + e1(i) * \cos(r14(i))))))) \right) \]
\[ r11(i) * r12(i) * (\mu/r11(i)^2 - \cos(r14(i)) * (PertR1(i)+TR1)/e1(i) + \sin(w1(i)+r14(i)) * (\text{PertW1(i) + TW1})/(\tan(\text{inc11}(i)) * (1.0 + e1(i) * \cos(r14(i)))) + \sin(r14(i)) * (2.0+e1(i) * \cos(r14(i))) * (\text{PertS1(i) + TS1})/(e1(i) \right) \]
\[ (\mu+3*(r22(i)*r23(i)/r21(i)^2-(r12(i)*r13(i)/r11(i)^2)+(PertS1(i)+TS1)/r11(i))*w8S2/2.0)/(2*r21(i)^2*r22(i)*sin(r24(i)))*mu*2(e2(i)))/(1.0+e2(i) * \cos(r24(i)))/(\text{w8S2}) - PertS2(i); \]

\[ \text{if DenmS2(i+1)<InSing & DenmS2(i+1)>OutSing} \]
\[ TsS2 = 2.0*\mu*r21(i) * e2(i) * (1.0+e2(i) * \cos(r24(i))) * (\text{FS2/w8S2-} \]
\[ (r21(i) * r22(i) * (\mu/r21(i)^2 - \cos(r24(i)) * (\text{PertR2(i) + TR2})/e2(i) + \sin(w2(i)+r24(i)) * (\text{PertW2(i) + TW2})/(\tan(\text{inc12}(i)) * (1.0 + e2(i) * \cos(r24(i))))) - r11(i) * r12(i) * (\mu/r11(i)^2 - \cos(r14(i)) * (PertR1(i)+TR1)/e1(i) + \sin(w1(i)+r14(i)) * (\text{PertW1(i) + TW1})/(\tan(\text{inc11}(i)) * (1.0 + e1(i) * \cos(r14(i)))) + \sin(r14(i)) * (2.0+e1(i) * \cos(r14(i))) * (\text{PertS1(i) + TS1})/(e1(i) \right) \]
\[ (\mu+3*(r22(i)*r23(i)/r21(i)^2-(r12(i)*r13(i)/r11(i)^2)+(PertS1(i)+TS1)/r11(i))*w8S2/2.0)/(2*r21(i)^2*r22(i)*sin(r24(i)))*mu*2(e2(i)))/(1.0+e2(i) * \cos(r24(i)))/(\text{w8S2}) - PertS2(i); \]

\[ \text{if e2(i)<0.0000001} \]
\[ TS2 = 2.0*r21(i) * (\text{FS2/w8S2-} \]
\[ (r21(i) * r22(i) * (\mu/r21(i)^2 - \cos(r24(i)) * (\text{PertR2(i) + TR2})/e2(i) + \sin(w2(i)+r24(i)) * (\text{PertW2(i) + TW2})/(\tan(\text{inc12}(i)) * (1.0 + e2(i) * \cos(r24(i)))))) - r11(i) * r12(i) * (\mu/r11(i)^2 - \cos(r14(i)) * (PertR1(i)+TR1)/e1(i) + \sin(w1(i)+r14(i)) * (\text{PertW1(i) + TW1})/(\tan(\text{inc11}(i)) * (1.0 + e1(i) * \cos(r14(i)))) + \sin(r14(i)) * (2.0+e1(i) * \cos(r14(i))) * (\text{PertS1(i) + TS1})/(e1(i) \right) \]
\[ (\mu+3*(r22(i)*r23(i)/r21(i)^2-(r12(i)*r13(i)/r11(i)^2)+(PertS1(i)+TS1)/r11(i))*w8S2/2.0)/(2*r21(i)^2*r22(i)*sin(r24(i)))*mu*2(e2(i)))/(1.0+e2(i) * \cos(r24(i)))/(\text{w8S2}) - PertS2(i); \]

\[ \text{if e2(i)<0.0000001} \]
\[ TS2 = 2.0*r21(i) * (\text{FS2/w8S2-} \]
\[ (r21(i) * r22(i) * (\mu/r21(i)^2 - \cos(r24(i)) * (\text{PertR2(i) + TR2})/e2(i) + \sin(w2(i)+r24(i)) * (\text{PertW2(i) + TW2})/(\tan(\text{inc12}(i)) * (1.0 + e2(i) * \cos(r24(i)))))) - r11(i) * r12(i) * (\mu/r11(i)^2 - \cos(r14(i)) * (PertR1(i)+TR1)/e1(i) + \sin(w1(i)+r14(i)) * (\text{PertW1(i) + TW1})/(\tan(\text{inc11}(i)) * (1.0 + e1(i) * \cos(r14(i)))) + \sin(r14(i)) * (2.0+e1(i) * \cos(r14(i))) * (\text{PertS1(i) + TS1})/(e1(i) \right) \]
\[ (\mu+3*(r22(i)*r23(i)/r21(i)^2-(r12(i)*r13(i)/r11(i)^2)+(PertS1(i)+TS1)/r11(i))*w8S2/2.0)/(2*r21(i)^2*r22(i)*sin(r24(i)))*mu*2(e2(i)))/(1.0+e2(i) * \cos(r24(i)))/(\text{w8S2}) - PertS2(i); \]

\[ \text{end} \]
/r21(i)^2-r12(i)*r13(i)/r11(i)^2)+(PertS1(i)+TS1)/r11(i))*w8S2/2.0)/w8S2-PertS2(i);
                   end
              if e1(i)<0.0000001 & e2(i)<0.0000001
                             TS2=2*r21(i)*(FS2/w8S2-(r21(i)*r22(i)*(mu/r21(i)^2+sin(w2(i)+r24(i)))*
                             (PertW2(i)+TW2)/(tan(incl2(i))*(1.0+e2(i)*cos(r24(i))))-r11(i)*r12(i)*
                             (mu/r11(i)^2+sin(w1(i)+r14(i))*(PertW1(i)+TW1)/(tan(incl1(i)))*
                             (1.0+e1(i)*cos(r14(i)))))}/mu+(3*(r22(i)*r23(i)/r21(i)^2-r12(i)*r13(i)/r11(i)^2)+
                             (PertS1(i)+TS1)/r11(i))*w8S2/2.0)/w8S2-PertS2(i);
                   end
              else
                             TS2=2*mu*r21(i)*e2(i)*(1.0+e2(i)*cos(r24(i)))*(FS2/w8S2-(r21(i)*r22(i)*
                             (mu/r21(i)^2-cos(r24(i))*(PertR2(i)+TR2)/e2(i)+sin(w2(i)+r24(i))))*
                             (mu/r11(i)^2+sin(w1(i)+r14(i))*(PertR1(i)+TR1)/e1(i)+sin(w1(i)+r14(i)))*
                             (2.0+e1(i)*cos(r14(i))*(PertS1(i)+TS1)/(1.0+e1(i)*cos(r14(i)))))}/mu+
                             (3*(r22(i)*r23(i)/r21(i)^2-r12(i)*r13(i)/r11(i)^2)+(PertS1(i)+TS1)/r11(i))*
                             w8S2/2.0)/(2*r21(i)^2*r22(i)*sin(r24(i))*(2.0+e2(i)*cos(r24(i)))+
                             mu*e2(i)*(1.0+e2(i)*cos(r24(i)))*w8S2)-PertS2(i);
                             DenmS2(i+1)=2*r21(i)^2*r22(i)*sin(r24(i))*(2.0+e2(i)*cos(r24(i)))+mu*e2(i)*(1.0+e
                             2(i)*cos(r24(i)))/w8S2;
                               if r24(i)<2.0*pi()*n2>pi()/2.0 & r24(i)<2.0*pi()*n2<3.0*pi()/2.0
                                 InSing=10.0^17;
                                 OutSing=-0.89*10.0^18;
                               else
                                 InSing=4.0*10.0^17;
                                 OutSing=-3.5*10.0^17;
                               end
                             if DenmS2(i+1)<InSing & DenmS2(i+1)>OutSing
                                TR2=-PertR2(i);
                                w8S2=10.0;
                                TS2=2*mu*r21(i)*e2(i)*(1.0+e2(i)*cos(r24(i)))*(FS2/w8S2-(r21(i)*r22(i)*
                                (mu/r21(i)^2-cos(r24(i))*(PertR2(i)+TR2)/e2(i)+sin(w2(i)+r24(i))))*
                                (mu/r11(i)^2+sin(w1(i)+r14(i))*(PertR1(i)+TR1)/e1(i)+sin(w1(i)+r14(i)))*
                                (2.0+e1(i)*cos(r14(i))*(PertS1(i)+TS1)/(1.0+e1(i)*cos(r14(i)))))}/mu+
                                (3*(r22(i)*r23(i)/r21(i)^2-r12(i)*r13(i)/r11(i)^2)+(PertS1(i)+TS1)/r11(i))*
                                w8S2/2.0)/(2*r21(i)^2*r22(i)*sin(r24(i))*(2.0+e2(i)*cos(r24(i)))+
                                mu*e2(i)*(1.0+e2(i)*cos(r24(i)))*w8S2)-PertS2(i);
                               end
            end
        US2(i+1)=TS2;
        UR2(i+1)=TR2;
        UW2(i+1)=TW2;
        U2(i+1)=sqrt(US2(i+1)^2+UR2(i+1)^2+UW2(i+1)^2);

        %SIMULATED CONDITIONS SATELLITE 1

        %Classical Runge-Kutta Fourth Order State equations

        %Initializing Atmospheric density step 1
        sat=1;
        k=0;
Density:
P1J2R=-(3/2)*mu*J2*(Rearth/r11(i)^2)^2*(1-3*sin(incl1(i))^2)*sin(w1(i)+r14(i))^2);
P1J2S=-(3/2)*mu*J2*(Rearth/r11(i)^2)^2*sin(incl1(i))^2*sin(2*(w1(i)+r14(i)));
P1J2W=-(3/2)*mu*J2*(Rearth/r11(i)^2)^2*sin(2*incl1(i))*sin(w1(i)+r14(i));
Drag1R=-sign(r13(i))*B*exp(A*((r11(i)-Rearth-Ho)/Ho)^C)*sqrt(r12(i)^2+r13(i)^2+V1w^2)*r13(i)*Cd1*A1/(2*m1);
Drag1S=-sign(r12(i))*B*exp(A*((r11(i)-Rearth-Ho)/Ho)^C)*sqrt(r12(i)^2+r13(i)^2+V1w^2)*r12(i)*Cd1*A1/(2*m1);
Drag1W=-sign(V1w)*B*exp(A*((r11(i)-Rearth-Ho)/Ho)^C)*sqrt(r12(i)^2+r13(i)^2+V1w^2)*V1w*Cd1*A1/(2*m1);
P1R=P1J2R+Drag1R;
P1S=P1J2S+Drag1S;
P1W=P1J2W+Drag1W;
k11(1)=r13(i);
k12(1)=-r12(i)*r13(i)/r11(i)+P1S+TS1;
k13(1)=r12(i)^2/r11(i)-mu/r11(i)^2+P1R+TR1;
k14(1)=r12(i)/r11(i);
komp1(1)=sqrt(r11(i)/(mu*(1+e1(i)*cos(r14(i)))))*(sin(w1(i)+r14(i))/sin(incl1(i)))*(P1W+TW1);
    if e1(i)<0.000001
        kw1(1)=komp1(1)*cos(incl1(i));
    else
        kw1(1)=komp1(1)*cos(incl1(i))+sqrt(r11(i)*(1+e1(i)*cos(r14(i)))/(mu)*(sin(r14(i))*cos(r14(i)))/(el(i))*(1+el(i)*cos(r14(i))))*(P1S+TS1)-(cos(r14(i))/el(i))*(P1R+TR1));
    end
    kinc1(1)=sqrt(r11(i)/(mu*(1+el(i)*cos(r14(i)))))*cos(w1(i)+r14(i))*(TW1);
r11k(1)=r11(i)+wt*k11(1)/2;
r12k(1)=r12(i)+wt*k12(1)/2;
r13k(1)=r13(i)+wt*k13(1)/2;
r14k(1)=r14(i)+wt*k14(1)/2;
qV1=-B*exp(A*((r11k(1)-Rearth-Ho)/Ho)^C)*Cd1*A1/(2*m1);
V1w=dik1*P1J2W;

%Initializing Atmospheric density step 2
sat=1;
k=1;
Density:
    if 1.0+(r12k(1)^2+r13k(1)^2-2*mu/r11k(1))*(r11k(1)*r12k(1))^2/mu^2<0.0
        elk=0.0;
    else
        elk=sqrt(1+(r12k(1)^2+r13k(1)^2-2*mu/r11k(1))*(r11k(1)*r12k(1))^2/mu^2);
    end
    P1J2R=-(3/2)*mu*J2*(Rearth/r11k(1)^2)^2*(1-3*sin(incl1k(1))^2)*sin(w1k(1)+r14k(1))^2);
    P1J2S=-(3/2)*mu*J2*(Rearth/r11k(1)^2)^2*sin(incl1k(1))^2*sin(2*(w1k(1)+r14k(1)));
    P1J2W=-(3/2)*mu*J2*(Rearth/r11k(1)^2)^2*sin(2*incl1k(1))*sin(w1k(1)+r14k(1));
dik1=sqrt(r11k(1)^3/(mu*(1+elk*cos(r14k(1)))))*sin(w1k(1)+r14k(1))*cos(w1k(1)+r14k(1));
qV1=-B*exp(A*((r11k(1)-Rearth-Ho)/Ho)^C)*Cd1*A1/(2*m1);
V1w=dik1*P1J2W;
V1ws = 0.0;
    while abs(V1ws - V1w) > 0.00001
        V1ws = V1w;
        V1w = V1ws - (V1ws * (-1 / dikl + qV1 * sqrt(r13k(1)^2 + r12k(1)^2 + V1ws^2) + P1J2W) / (-1 / dikl + qV1 * (r13k(1)^2 + r12k(1)^2 + 2 * V1ws^2) / sqrt(r13k(1)^2 + r12k(1)^2 + V1ws^2)));
    end
    Drag1R = sign(r13k(1)) * B * exp(A * ((r11k(1) - Rearth - Ho) / Ho) ^ C) * sqrt(r12k(1)^2 + r13k(1)^2 + V1w^2) * r13k(1) * Cd1 * A1 / (2 * m1);
    Drag1S = sign(r12k(1)) * B * exp(A * ((r11k(1) - Rearth - Ho) / Ho) ^ C) * sqrt(r12k(1)^2 + r13k(1)^2 + V1w^2) * r12k(1) * Cd1 * A1 / (2 * m1);
    Drag1W = sign(V1w) * B * exp(A * ((r11k(1) - Rearth - Ho) / Ho) ^ C) * sqrt(r12k(1)^2 + r13k(1)^2 + V1w^2) * V1w * Cd1 * A1 / (2 * m1);
    P1R = P1J2R + Drag1R;
    P1S = P1J2S + Drag1S;
    P1W = P1J2W + Drag1W;
    k11(2) = r13k(1);
    k12(2) = -r12k(1) * r13k(1) / r11k(1) + P1S + TS1;
    k13(2) = r12k(1)^2 / r11k(1) - mu / r11k(1)^2 + P1R + TR1;
    k14(2) = r12k(1) / r11k(1);

    komg1(2) = sqrt(r11k(1) / (mu * (1 + elk * cos(r14k(1)))))) * (sin(w1k(1) + r14k(1)) / sin(incl1k(1))) * (P1W + TW1);
    if elk < 0.000001
        kw1(2) = -komg1(1) * cos(incl1k(1));
    else
        kw1(2) = -komg1(1) * cos(incl1k(1)) + sqrt(r11k(1) * (1 + elk * cos(r14k(1)))) / mu * ((sin(r14k(1)) * (2 + elk * cos(r14k(1)))) / elk * (1 + elk * cos(r14k(1)))) * (P1S + TS1) - (cos(r14k(1)) / elk) * (P1R + TR1);
    end
    kincl1(2) = sqrt(r11k(1) / (mu * (1 + elk * cos(r14k(1)))))) * cos(w1k(1) + r14k(1)) * (TW1);

    r11k(2) = r11(i) + wt * k11(2) / 2;
    r12k(2) = r12(i) + wt * k12(2) / 2;
    r13k(2) = r13(i) + wt * k13(2) / 2;
    r14k(2) = r14(i) + wt * k14(2) / 2;
    omg1k(2) = omg1(i) + wt * komg1(2) / 2;
    w1k(2) = w1(i) + wt * kw1(2) / 2;
    incl1k(2) = incl1(i) + wt * kincl1(2) / 2;

    % Initializing Atmospheric density step 3
    sat = 1;
    k = 2;
    Density:
    if 1.0 + (r12k(2)^2 + r13k(2)^2 - 2 * mu / r11k(2)) * (r11k(2) * r12k(2))^2 / mu^2 < 0.0
        elk = 0.0;
    else
        elk = sqrt(1 + (r12k(2)^2 + r13k(2)^2 - 2 * mu / r11k(2)) * (r11k(2) * r12k(2))^2 / mu^2);
    end
    P1J2R = -((3/2) * mu * J2 * (Rearth / r11k(2)^2)^2 * (1 - 3 * sin(incl1k(2))^2 * sin(w1k(2) + r14k(2))^2);
Vlws=0.0;
while abs(Vlws-Vlw)>0.00001
Vlws=Vlw;
Vlw=Vlw*(-1/dik1+qV1*sqrt(r13k(2)^2+r12k(2)^2+Vlws^2)) + P1J2W)/(-
1/dik1+qV1*(r13k(2)^2+r12k(2)^2+2*Vlws^2)/sqrt(r13k(2)^2+r12k(2)^2+Vlws^2));
end
Drag1R=-sign(r13k(2))*B*exp(A*(r11k(2)-Rearth-
Ho)/Ho)^C)*sqrt(r12k(2)^2+r13k(2)^2+Vlw^2)^2*mu/r11k(2)^2+P1R+TR1;
Drag1S=-sign(r12k(2))*B*exp(A*S*(r11k(2)-Rearth-
Ho)/Ho)^C)*sqrt(r12k(2)^2+r13k(2)^2+Vlw^2)*r12k(2)*Cd1A1/(2*m1);
Drag1W=-sign(Vlw)*B*exp(A*S*(r11k(2)-Rearth-
Ho)/Ho)^C)*sqrt(r12k(2)^2+r13k(2)^2+Vlw^2)*Vlw*Cd1A1/(2*m1);
P1R=P1J2R+Drag1R;
P1S=P1J2S+Drag1S;
P1W=P1J2W+Drag1W;
k14=3=r13k(2);
k12=-r12k(2)*r13k(2)/r11k(2)+P1S+TS1;
k13=r12k(2)^2/r11k(2)^2-mu/r11k(2)^2+P1R+TR1;
k14=r12k(2)/r11k(2);

if sat<0.00001
   kwl=-komg(1)*cos(incl1k(2));
else
   kwl=komg(1)*cos(incl1k(2))+sqrt(r11k(2)*(1+e1k*cos(r14k(3))))*(sin(wlk(2)+r14k(3)))*
   (P1R+TWl);
end

r11k(3)=r11k(i)+wt*k11(3);
r12k(3)=r12k(i)+wt*k12(3);
r13k(3)=r13k(i)+wt*k13(3);
r14k(3)=r14k(i)+wt*k14(3);
ompk(3)=omp(1)+wt*komg(3);
w1k(3)=w1(1)+wt*kwl(3);
incl1k(3)=incl1(1)+wt*kincl1(2)/2;

%Initializing Atmospheric density step 4
sat=1;
k=3;
Density:
if 1.0+(r12k(3)^2+r13k(3)^2-2*mu/r11k(3))*r11k(3)*r12k(3))^2<0.0
e1k=0.0;
else
e1k=sqrt(1+(r12k(3)^2+r13k(3)^2-2*mu/r11k(3))*r11k(3)*r12k(3))^2/mu^2;
end
P1J2R=-(3/2)*mu*J2*(Rearth/r11k(3)^2)^2*(1-
3*sin(incl1k(3))^2*sin(wlk(3)+r14k(3))^2);
P1J2S=-(3/2)*mu*J2*(Rearth/r11k(3)^2)^2*sin(incl1k(3))^2*sin(2*(w1k(3)+r14k(3)));
P1J2W=-(3/2)*mu*J2*(Rearth/r11k(3)^2)^2*sin(2*incl1k(3))*sin(wlk(3)+r14k(3));
dik1=sqrt(r11k(3)^3/(mu*(1+e1k*cos(r14k(3)))))*sin(wlk(3)+r14k(3))*cos(wlk(3)+r14
k(3));
qVl=-B*exp(A*S*(r11k(3)-Rearth-Ho)/Ho)^C)*Cd1A1/(2*m1);
Vlw=dik1*P1J2W;
Vlws=0.0;
while abs(Vlws-Vlw)>0.00001
    Vlws=Vlw;
    Vlw=Vlws-(-1/dik1+qV1*sqrt(r13k(3)^2+r12k(3)^2+Vlws^2)+P1J2W)/(-1/dik1+qV1*(r13k(3)^2+r12k(3)^2+2*Vlw^2)/sqrt(r13k(3)^2+r12k(3)^2+Vlws^2));
end
Drag1R=-sign(r13k(3))*B*exp(A*((r11k(3)-Rearth-Ho)/Ho)^C)*sqrt(r12k(3)^2+r13k(3)^2+Vlw^2)*r13k(3)*Cd1*A1/(2*m1);
Drag1S=-sign(r12k(3))*B*exp(A*((r11k(3)-Rearth-Ho)/Ho)^C)*sqrt(r12k(3)^2+r13k(3)^2+Vlw^2)*r12k(3)*Cd1*A1/(2*m1);
Drag1W=-sign(Vlw)*B*exp(A*((r11k(3)-Rearth-Ho)/Ho)^C)*sqrt(r12k(3)^2+r13k(3)^2+Vlw^2)*Vlw*Cd1*A1/(2*m1);
P1R=P1J2R+Drag1R;
P1S=P1J2S+Drag1S;
P1W=P1J2W+Drag1W;
k11(4)=r13k(3);
k12(4)=-r12k(3)*r13k(3)/r11k(3)+P1S+TS1;
k13(4)=r12k(3)^2/r11k(3)-mu/r11k(3)^2+P1R+TR1;
k14(4)=r12k(3)/r11k(3);
komg1(4)=sqrt(r11k(3)/(mu*(1+e1k*cos(r14k(3)))))*(sin(w1k(3)+r14k(3))/sin(incl1k(3)))*(P1W+TW1);
    if e1k<0.0000001
        kw1(4)=-komg1(4)*cos(incl1k(3));
    else
        kw1(4)=komg1(4)*cos(incl1k(3))+sqrt(r11k(3)*(1+e1k*cos(r14k(3)))/mu)*((sin(r14k(3))*(2+e1k*cos(r14k(3)))/e1k*(1+e1k*cos(r14k(3))))*(P1S+TS1)-(cos(r14k(3))/e1k)*(P1R+TR1));
        kinc1(4)=sqrt(r11k(3)/(mu*(1+e1k*cos(r14k(3)))))*cos(w1k(3)+r14k(3))*(TW1);
        if el(i)<0.0000001
            kw1(1)=komg1(1)*cos(incl1(i));
            kw1(2)=-komg1(2)*cos(incl1k(1));
            kw1(3)=-komg1(3)*cos(incl1k(2));
            kw1(4)=-komg1(4)*cos(incl1k(3));
        end
        r11(i+1)=r11(i)+wt*(k11(i)+2.0*(k11(2)+k11(3))+k11(4))/6.0;
        r12(i+1)=r12(i)+wt*(k12(i)+2.0*(k12(2)+k12(3))+k12(4))/6.0;
        r13(i+1)=r13(i)+wt*(k13(i)+2.0*(k13(2)+k13(3))+k13(4))/6.0;
        r14(i+1)=r14(i)+wt*(k14(i)+2.0*(k14(2)+k14(3))+k14(4))/6.0;
        omg1(i+1)=omg1(i)+wt*(komg1(i)+2.0*(komg1(2)+komg1(3))+komg1(4))/6.0;
        w1(i+1)=w1(i)+wt*(kw1(i)+2.0*(kw1(2)+kw1(3))+kw1(4))/6.0;
        thetal1(i+1)=w1(i+1)+r14(i+1);
        incl1(i+1)=incl1(i)+wt*(kinc1(i)+2.0*(kinc1(2)+kinc1(3))+kinc1(4))/6.0;
    end
%Orbital Position Coordinates (PQ)
P1(i+1)=r11(i+1)*cos(r14(i+1));
Q1(i+1)=r11(i+1)*sin(r14(i+1));
%Eccentricity
    if 1+(r12(i+1)^2+r13(i+1)^2-2*mu/r11(i+1))*(r11(i+1)*r12(i+1))^2/mu^2<0.0
        el(i+1)=0.0;
    else
        el(i+1)=sqrt(1+(r12(i+1)^2+r13(i+1)^2-2*mu/r11(i+1))*(r11(i+1)*r12(i+1))^2/mu^2);
    end
\% Out of plane velocity
i=i+1;
sat=1;
k=0;
Density;

dik1=sqrt(r1k(3)^3/(mu*(1+el(i)*cos(r14(i)))))*sin(wl(i)+r14(i))*cos(wl(i)+r14(i));
P1J2W=-3*mu*J2*(Rearth/r11(i)^2)*2*sin(2*incl(i))*sin(wl(i)+r14(i));
qV1=-B*exp(A*((r11(i)-Rearth-Ho)/Ho)^C)*Cd1*A1/(2*m1);
Vlw=dik1*P1J2W;
Vls=0.0;
while abs(Vlws-Vlw)>0.00001
Vls=Vlw;
Vlw=Vlws-(1/dik1+qV1*sqrt(r13(i)^2+r12(i)^2+Vlws^2))/sqrt(r13(i)^2+r12(i)^2+Vlws^2));
end
P1J2R=-(3/2)*mu*J2*(Rearth/r11(i)^2)^2*(1-3*sin(incl(i))^2*sin(wl(i)+r14(i))^2);
P1J2S=-(3/2)*mu*J2*(Rearth/r11(i)^2)^2*sin(incl(i))^2*sin(2*(w1(i)+r14(i)));
Drag1R=sign(r13(i))*B*exp(A*((r11(i)-Rearth-Ho)/Ho)^C)*sqrt(r12(i)^2+r13(i)^2+Vlw^2)*r13(i)*Cd1*A1/(2*m1);
Drag1S=sign(r12(i))*B*exp(A*((r11(i)-Rearth-Ho)/Ho)^C)*sqrt(r12(i)^2+r13(i)^2+Vlw^2)*r12(i)*Cd1*A1/(2*m1);
Drag1W=-sign(Vlw)*B*exp(A*((r11(i)-Rearth-Ho)/Ho)^C)*sqrt(r12(i)^2+r13(i)^2+Vlw^2)*Vlw*Cd1*A1/(2*m1);
i=i-1;
PertR1(i+1)=P1J2R+Drag1R;
PertS1(i+1)=P1J2S+Drag1S;
PertW1(i+1)=P1J2W+Drag1W;
Pert1(i+1)=sqrt(PertR1(i+1)^2+PertS1(i+1)^2+PertW1(i+1)^2);
ForceR1(i+1)=r12(i+1)^2/r11(i+1)-mu/r11(i+1)^2+PertR1(i+1)+TR1;
ForceS1(i+1)=-r12(i+1)*r13(i+1)/r11(i+1)+PertS1(i+1)+TS1;
ForceW1(i+1)=PertW1(i+1)+TW1;

% Energy
Energy1(i+1)=(r12(i+1)^2+r13(i+1)^2)/2-mu/r11(i+1);
dEnergy1dt(i+1)=r13(i+1)*(PertR1(i+1)+TR1)+r12(i+1)*(PertS1(i+1)+TS1);

% Angular rates

domg1dt(i+1)=sqrt(r11(i+1)/(mu*(1+el(i+1)*cos(r14(i+1)))))*sin(wl(i+1)+r14(i+1))/sin(incl(i+1)))*(PertW1(i+1)+TW1);
if el(i+1)<0.0000001
dwldt(i+1)=-domg1dt(i+1)*cos(incl(i+1));
else
dwldt(i+1)=
-domg1dt(i+1)*cos(incl(i+1))+sqrt(r11(i+1)*(l+el(i+1)*cos(r14(i+1)))/mu)*((sin(r14(i+1))*(2+el(i+1)*cos(r14(i+1)))/(el(i+1)*(l+el(i+1)*cos(r14(i+1)))))*(PertS1(i+1)+TS1)-(cos(r14(i+1))/el(i+1))*(PertR1(i+1)+TR1));
end

dincl1dt(i+1)=sqrt(r11(i+1)/(mu*(1+el(i+1)*cos(r14(i+1)))))*cos(wl(i+1)+r14(i+1))*(TW1);
dr14dt(i+1)=r12(i+1)/r11(i+1);
dtheta1dt(i+1)=dwldt(i+1)+dr14dt(i+1);
%Orbital Velocity Coordinates (PQ)
VP1(i+1)=sqrt(μ/(r11(i+1)*(1+e1(i+1)*cos(r14(i+1))))) * sin(r14(i+1));
VQ1(i+1)=sqrt(μ/(r11(i+1)*(1+e1(i+1)*cos(r14(i+1))))) * (e1(i+1) + cos(r14(i+1)));
VW1(i+1)=V1w;

%Transpose matrix from Orbital (PQ) to Inertial (IJK) Coordinates
T111=cos(omg1(i+1))*cos(w1(i+1))-sin(omg1(i+1))*sin(w1(i+1))*cos(incl1(i+1));
T112=-cos(omg1(i+1))*sin(w1(i+1))-sin(omg1(i+1))*cos(w1(i+1))*cos(incl1(i+1));
T113=sin(omg1(i+1))*cos(incl1(i+1));
T121=sin(omg1(i+1))*cos(w1(i+1))+cos(omg1(i+1))*sin(w1(i+1))*cos(incl1(i+1));
T122=-sin(omg1(i+1))*sin(w1(i+1))+cos(omg1(i+1))*cos(w1(i+1))*cos(incl1(i+1));
T123=cos(omg1(i+1))*sin(incl1(i+1));
T131=sin(w1(i+1))*sin(incl1(i+1));
T132=cos(w1(i+1))*sin(incl1(i+1));
T133=cos(incl1(i+1));

%Inertial Position Coordinates (IJK)
x1(i+1)=P1(i+1)*T111+Q1(i+1)*T112;
y1(i+1)=P1(i+1)*T121+Q1(i+1)*T122;
z1(i+1)=P1(i+1)*T131+Q1(i+1)*T132;

%Inertial Velocity Coordinates (XYZ)
Vx1(i+1)=VP1(i+1)*T111+VQ1(i+1)*T112+VW1(i+1)*T113;
Vy1(i+1)=VP1(i+1)*T121+VQ1(i+1)*T122+VW1(i+1)*T123;
Vz1(i+1)=VP1(i+1)*T131+VQ1(i+1)*T132+VW1(i+1)*T133;

%Relative States
r11d(i+1)=r11(i+1)-r11traj(i+1);
r12d(i+1)=r12(i+1)-r12traj(i+1);
r13d(i+1)=r13(i+1)-r13traj(i+1);
r14d(i+1)=r14(i+1)-r14traj(i+1);
w1d(i+1)=w1(i+1)-w1traj;
theta1d(i+1)=r14(i+1)+w1(i+1)-r14traj(i+1)-w1traj;
omg1d(i+1)=omg1(i+1)-omg1traj;
incl1d(i+1)=incl1(i+1)-incl1traj;
x1d(i+1)=x1(i+1)-x1traj(i+1);
y1d(i+1)=y1(i+1)-y1traj(i+1);
z1d(i+1)=z1(i+1)-z1traj(i+1);
d1(i+1)=sqrt((x1(i+1)-x1traj(i+1))^2+(y1(i+1)-y1traj(i+1))^2+(z1(i+1)-z1traj(i+1))^2);
dorbit1(i+1)=sqrt((P1(i+1)-P1traj(i+1))^2+(Q1(i+1)-Q1traj(i+1))^2);
dEnergy1(i+1)=Energy1(i+1)-Etraj(i+1);
del1(i+1)=w1(i+1)-eltraj(i+1);

%SIMULATED CONDITIONS SATELLITE 2

%Classical Runge-Kutta Fourth Order State equations

%Initializing Atmospheric density step 1
sat=2;
k=0;
Density;
P2J2R=-(3/2)*mu*J2*(Rearth/r21(i)^2)^2*(1-3*sin(incl2(i))^2*sin(w2(i)+r24(i))^2);
P2J2S=-(3/2)*mu*J2*(Rearth/r21(i)^2)^2*sin(incl2(i))^2*sin(2*(w2(i)+r24(i)));
P2J2W=-(3/2)*mu*J2*(Rearth/r21(i)^2)^2*sin(2*incl2(i))*sin(w2(i)+r24(i));
P2J2R=-sign(r23(i))*B*exp(A*((r21(i)-Rearth-Ho)^C)*sqrt(r22(i)^2+r23(i)^2+V2w^2)*r23(i)*Cd2*A2/(2*m2));
P2J2S=-sign(r22(i))*B*exp(A*((r21(i)-Rearth-Ho)^C)*sqrt(r22(i)^2+r23(i)^2+V2w^2)*r22(i)*Cd2*A2/(2*m2));
P2J2W=-sign(r21(i)-Rearth-Ho)^C)*sqrt(r22(i)^2+r23(i)^2+V2w^2)*V2w*Cd2*A2/(2*m2);
P2R=P2J2R+Drag2R;
P2S=P2J2S+Drag2S;
P2W=P2J2W+Drag2W;
k21(1)=r23(i);
k22(1)=-r22(i)*r23(i)/r21(i)+P2S+TS2;
k23(1)=r22(i)^2/r21(i)-mu/r21(i)^2+P2R+TR2;
k24(1)=r22(i)/r21(i);
komg2(1)=sqrt(r21(i)/(mu*(1+e2(i)*cos(r24(i)))))*sin(w2(i)+r24(i))*/sin(incl2(i)))*(P2W+TW2);
   if e2(i)<0.0000001
     kw2(1)=-komg2(1)*cos(incl2(i));
    else
     kw2(1)=komg2(1)*cos(incl2(i))+sqrt(r21(i)/(mu*(1+e2(i)*cos(r24(i)))))*((sin(r24(i)))*(2+e2(i)*cos(r24(i)))/(e2(i)*(1+e2(i)*cos(r24(i)))))*(P2S+TS2) -
     (cos(r24(i))/e2(i))*(P2R+TR2));
    end
    kincl2(1)=sqrt(r21(i)/(mu*(1+e2(i)*cos(r24(i)))))*cos(w2(i)+r24(i))*TW2);
r21k(1)=r21(i)+wt*k21(1)/2;
r22k(1)=r22(i)+wt*k22(1)/2;
r23k(1)=r23(i)+wt*k23(1)/2;
r24k(1)=r24(i)+wt*k24(1)/2;
    omg2k(1)=omg2(i)+wt*komg2(1)/2;
w2k(1)=w2(i)+wt*kw2(1)/2;
    incl2k(1)=incl2(i)+wt*kincl2(1)/2;

%Initializing Atmospheric density step 2
sat=2;
k=1;
Density;
   if 1.0+(r22k(1)^2+r23k(1)^2-2*mu/r21k(1))*(r21k(1)*r22k(1))^2/mu^2<0.0
     e2k=0.0;
    else
     e2k=sqrt(1+(r22k(1)^2+r23k(1)^2-2*mu/r21k(1))*(r21k(1)*r22k(1))^2/mu^2);
    end
P2J2R=-(3/2)*mu*J2*(Rearth/r21k(1)^2)^2*(1-3*sin(incl2k(1)))*2*sin(w2k(1)+r24k(1))^2);
P2J2S=-(3/2)*mu*J2*(Rearth/r21k(1)^2)^2*sin(incl2k(1))^2*sin(2*(w2k(1)+r24k(1)));
P2J2W=-(3/2)*mu*J2*(Rearth/r21k(1)^2)^2*sin(incl2k(1))^2*sin(w2k(1)+r24k(1));
dik2=sqrt(r21k(1)^3/(mu*(1+e2k*cos(r24k(1)))))*sin(w2k(1)+r24k(1))*cos(w2k(1)+r24k(1));
qV2=-B*exp(A*((r21k(1)-Rearth-Ho)/Ho)^C)*Cd2*A2/(2*m2);
V2w=dik2*P2J2W;
V2ws=0.0;
while abs(V2ws-V2w)>0.00001
    V2ws=V2w;
V2w=V2ws*(1/dik2+qV2*sqrt(r23k(1)^2+r22k(1)^2+V2ws^2)+P2J2W)/(-1/dik2+qV2*(r23k(1)^2+r22k(1)^2+V2ws^2)/sqrt(r23k(1)^2+r22k(1)^2+V2ws^2));
end
Drag2R = -sign(r23k(1)) * B * exp(A*((r21k(1)-Rearth-Ho)/Ho)^C) * sqrt(r22k(1)^2+r23k(1)^2+V2w^2) * r23k(1) * Cd2*A2/(2*m2);

Drag2S = -sign(r22k(1)) * B * exp(A*((r21k(1)-Rearth-Ho)/Ho)^C) * sqrt(r22k(1)^2+r23k(1)^2+V2w^2) * r22k(1) * Cd2*A2/(2*m2);

Drag2W = -sign(V2w) * B * exp(A*((r21k(1)-Rearth-Ho)/Ho)^C) * sqrt(r22k(1)^2+r23k(1)^2+V2w^2) * V2w * Cd2*A2/(2*m2);

P2R = P2J2R + Drag2R;
P2S = P2J2S + Drag2S;
P2W = P2J2W + Drag2W;

k21(2) = r23k(1);
k22(2) = -r22k(1) * r23k(1)/r21k(1) + P2S + TS2;
k23(2) = r22k(1)^2/r21k(1) - mu/r21k(1)^2 + P2R + TR2;
k24(2) = r22k(1)/r21k(1);

komg2(2) = sqrt(r21k(1)/(mu*(1+e2k*cos(r24k(1)))))*(sin(w2k(1)+r24k(1))/sin(incl2k(1)))*((P2W+TW2));
if e2k<0.0000001
    kw2(2) = -komg2(1)*cos(incl2k(1));
else
    kw2(2) = komg2(1)*cos(incl2k(1)) + sqrt(r21k(1)*(1+e2k*cos(r24k(1)))/mu)*((sin(r24k(1)))*(2+e2k*cos(r24k(1)))/(e2k*(1+e2k*cos(r24k(1)))))*((P2S+TS2) - (cos(r24k(1))/e2k)*(P2R+TR2));
end
kincl2(2) = sqrt(r21k(1)/(mu*(1+e2k*cos(r24k(1)))))*cos(w2k(1)+r24k(1))*(TW2);

r21k(2) = r21(i) + wt*k21(2)/2;
r22k(2) = r22(i) + wt*k22(2)/2;
r23k(2) = r23(i) + wt*k23(2)/2;
r24k(2) = r24(i) + wt*k24(2)/2;
ombg2k(2) = omg2(i) + wt*komg2(2)/2;
w2k(2) = w2(i) + wt*kw2(2)/2;
incl2k(2) = incl2(i) + wt*kincl2(2)/2;

% Initializing Atmospheric density step 3
sat = 2;
k = 2;
Density;
if 1.0 + (r22k(2)^2+r23k(2)^2-2*mu/r21k(2))*(r21k(2)*r22k(2))^2/mu^2 < 0.0
    e2k = 0.0;
else
    e2k = sqrt(1 + (r22k(2)^2+r23k(2)^2-2*mu/r21k(2))*(r21k(2)*r22k(2))^2/mu^2);
end
P2J2R = -(3/2) * mu * J2 * (Rearth/r21k(2))^2 * (1 - 3*sin(incl2k(2))^2) * sin(w2k(2)+r24k(2))^2);
P2J2S = -(3/2) * mu * J2 * (Rearth/r21k(2))^2 * sin(incl2k(2))^2 * sin(2*(w2k(2)+r24k(2)));
P2J2W = -(3/2) * mu * J2 * (Rearth/r21k(2))^2 * sin(2*(incl2k(2))) * sin(w2k(2)+r24k(2));

dik2 = sqrt((2*r21k(2)^3)/(mu*(1+e2k*cos(r24k(2)))))*sin(w2k(2)+r24k(2))*cos(w2k(2)+r24k(2));
qV2 = -B * exp(A*((r21k(2)-Rearth-Ho)/Ho)^C) * Cd2*A2/(2*m2);
V2w = dik2 * P2J2W;
V2ws = 0.0;
while abs(V2ws-V2w) > 0.00001
    V2ws = V2w;
    V2w = V2ws - (V2ws*(-1/dik2+qV2*sqrt(r23k(2)^2+r22k(2)^2+V2ws^2))+P2J2W)/(-1/dik2+qV2*(r23k(2)^2+r22k(2)^2+2*V2ws^2)/sqrt(r23k(2)^2+r22k(2)^2+V2ws^2));
end
Drag2R = -\text{sign}(r23k(2)) \times B \times e^{A \times ((r21k(2)-\text{Rearth}-Ho)/Ho)^C} \times \sqrt{r22k(2)^2+r23k(2)^2+V2w^2} \times V2w \times C_d2 \times A^2/(2 \times m2); \\
Drag2S = -\text{sign}(r22k(2)) \times B \times e^{A \times ((r21k(2)-\text{Rearth}-Ho)/Ho)^C} \times \sqrt{r22k(2)^2+r23k(2)^2+V2w^2} \times V2w \times C_d2 \times A^2/(2 \times m2); \\
Drag2W = -\text{sign}(V2w) \times B \times e^{A \times ((r21k(2)-\text{Rearth}-Ho)/Ho)^C} \times \sqrt{r22k(2)^2+r23k(2)^2+V2w^2} \times V2w \times C_d2 \times A^2/(2 \times m2); \\

P2R = P2J2R + Drag2R; \\
P2S = P2J2S + Drag2S; \\
P2W = P2J2W + Drag2W; \\
k21(3) = r23k(2); \\
k22(3) = -r22k(2) \times r23k(2)/r21k(2)+P2S+TS2; \\
k23(3) = r22k(2)^2/r21k(2)-\mu/r21k(2)^2+P2R+TR2; \\
k24(3) = r22k(2)/r21k(2); \\
komg2(3) = \sqrt{r21k(2)/(\mu \times (1+e2k \times \cos(r24k(2))))} \times (\sin(w2k(2)+r24k(2))/\sin(incl2k(2))) \times (P2W+TW2); \\
\text{if } e2k<0.0000001 \\
\quad k2(3) = -komg2(1) \times \cos(incl2k(2)); \\
\text{else} \\
\quad kw2(3) = komg2(1) \times \cos(incl2k(2))+\sqrt{r21k(2)/(\mu \times (1+e2k \times \cos(r24k(2))))} \times ((\sin(r24k(2)) \times (2 \times e2k \times \cos(r24k(2))))/e2k \times (1+e2k \times \cos(r24k(2)))) \times (P2S+TS2)-
\quad \cos(r24k(2))/e2k \times (P2R+TR2)); \\
\quad \end{if} \\
\quad kincl2(3) = \sqrt{r21k(2)/(\mu \times (1+e2k \times \cos(r24k(2))))} \times \cos(w2k(2)+r24k(2)) \times (TW2); \\
\quad r21k(3) = r21(i)+wt \times k21(3); \\
r22k(3) = r22(i)+wt \times k22(3); \\
r23k(3) = r23(i)+wt \times k23(3); \\
r24k(3) = r24(i)+wt \times k24(3); \\
omeg2k(3) = omg2(i)+wt \times komg2(3); \\
w2k(3) = w2(i)+wt \times kw2(3); \\
incl2k(3) = incl2(i)+wt \times kincl2(2)/2; \\
\%Initializing Atmospheric density step 4 \\
sat = 2; \\
k = 3; \\
Density: \\
\text{if } 1.0+(r22k(3)^2+r23k(3)^2-2 \times \mu/r21k(3))^2/(\mu^2)<0.0 \\
\quad e2k = 0.0; \\
\text{else} \\
\quad e2k = \sqrt{1+(r22k(3)^2+r23k(3)^2-2 \times \mu/r21k(3))^2/\mu^2}; \\
\quad \end{if} \\
P2J2R = -(3/2) \times \mu \times J2 \times (\text{Rearth}/r21k(3)^2)^2 \times (1-3 \times \sin(incl2k(3))^2 \times \sin(w2k(3)+r24k(3))^2); \\
P2J2S = -(3/2) \times \mu \times J2 \times (\text{Rearth}/r21k(3)^2)^2 \times \sin(incl2k(3))^2 \times \sin(2 \times (w2k(3)+r24k(3))); \\
P2J2W = -(3/2) \times \mu \times J2 \times (\text{Rearth}/r21k(3)^2)^2 \times \sin(2 \times incl2k(3))) \times \sin(w2k(3)+r24k(3)); \\
dik2 = \sqrt{(r21k(3)^3/(\mu \times (1+e2k \times \cos(r24k(3)))) \times \sin(w2k(3)+r24k(3))^2 \times \cos(w2k(3)+r24k(3))^2); \\
qV2 = B \times e^{A \times ((r21k(3)-\text{Rearth}-Ho)/Ho)^C} \times C_d2 \times A^2/(2 \times m2); \\
V2w = dik2 \times P2J2W; \\
\text{while } \abs(V2w-V2w)>0.00001 \\
\quad V2w = V2w; \\
\text{end}
Drag2R = \text{sign}(r23k(3)) * B \cdot \exp(A \cdot ((r21k(3) - \text{Rearth} - Ho)/Ho)^C) \cdot \sqrt[r22k(3)^2 + r23k(3)^2 + \text{V2w}^2]{r23k(3)} \cdot \text{Cd2} \cdot A^2/(2 \cdot m^2);
Drag2S = \text{sign}(r22k(3)) \cdot \text{B} \cdot \exp(A \cdot ((r21k(3) - \text{Rearth} - Ho)/Ho)^C) \cdot \sqrt[r22k(3)^2 + r23k(3)^2 + \text{V2w}^2]{r22k(3)} \cdot \text{Cd2} \cdot A^2/(2 \cdot m^2);
Drag2W = \text{sign}(\text{V2w}) \cdot \text{B} \cdot \exp(A \cdot ((r21k(3) - \text{Rearth} - Ho)/Ho)^C) \cdot \sqrt[r22k(3)^2 + r23k(3)^2 + \text{V2w}^2]{\text{V2w} \cdot \text{Cd2} \cdot A^2/(2 \cdot m^2)};

P2R = P2J2R + Drag2R;
P2S = P2J2S + Drag2S;
P2W = P2J2W + Drag2W;
k21(4) = r23k(3);
k22(4) = -r22k(3) \cdot r23k(3)/r21k(3) + P2S + TS2;
k23(4) = r22k(3)^2/r21k(3) - \mu/r21k(3)^2 + P2R + TR2;
k24(4) = r22k(3)/r21k(3);

komg2(4) = \sqrt[r21k(3)/(\mu*(1+e2k*cos(r24k(3))))]{(\sin(w2k(3) + r24k(3)) / \sin(incl2k(3)))} \cdot (P2W + TW2);
if e2k < 0.0000001
  kw2(4) = -komg2(4) \cdot \cos(incl2k(3));
else
  kw2(4) = komg2(4) \cdot \cos(incl2k(3)) + \sqrt[r21k(3)/(\mu*(1+e2k*cos(r24k(3))))]{((\sin(r24k(3)) \cdot (2+e2k*cos(r24k(3)))) / (e2k*(1+e2k*cos(r24k(3))))}* (P2S + TS2) - (\cos(r24k(3)) / e2k) * (P2R + TR2);
end

kincl2(4) = \sqrt[r21k(3)/(\mu*(1+e2k*cos(r24k(3))))]{(\sin(w2k(3) + r24k(3)) / \sin(incl2k(3)))} \cdot (P2W + TW2);
if e2(i) < 0.0000001
  kw2(1) = -komg2(1) \cdot \cos(incl2(i));
  kw2(2) = -komg2(2) \cdot \cos(incl2k(1));
  kw2(3) = -komg2(3) \cdot \cos(incl2k(2));
  kw2(4) = -komg2(4) \cdot \cos(incl2k(3));
end

r21(i+1) = r21(i) + \text{wt} \cdot (k21(1) + 2.0 \cdot (k21(2) + k21(3)) + k21(4))/6.0;
r22(i+1) = r22(i) + \text{wt} \cdot (k22(1) + 2.0 \cdot (k22(2) + k22(3)) + k22(4))/6.0;
r23(i+1) = r23(i) + \text{wt} \cdot (k23(1) + 2.0 \cdot (k23(2) + k23(3)) + k23(4))/6.0;
r24(i+1) = r24(i) + \text{wt} \cdot (k24(1) + 2.0 \cdot (k24(2) + k24(3)) + k24(4))/6.0;

w2(i+1) = w2(i) + \text{wt} \cdot (kw2(1) + 2.0 \cdot (kw2(2) + kw2(3)) + kw2(4))/6.0;
theta2(i+1) = \text{wt} \cdot (w2(i+1) + \text{wt} \cdot (kw2(1) + 2.0 \cdot (kw2(2) + kw2(3))) + kw2(4))/6.0;

% Orbital Position Coordinates (PQ)
P2(i+1) = r21(i+1) \cdot \cos(r24(i+1));
Q2(i+1) = r21(i+1) \cdot \sin(r24(i+1));

% Eccentricity
if 1.0 + (r22(i+1)^2 + r23(i+1)^2 - 2*\mu/r21(i+1))^2 > 0.0
  e2(i+1) = 0.0;
else
  e2(i+1) = \sqrt[r22(i+1)^2 + r23(i+1)^2 - 2*\mu/r21(i+1)]{(r21(i+1) * r22(i+1))}^2 / \mu^2;
end

% Out of plane velocity
i = i+1;
sat = 2;
k = 0;
Density;
\[ \text{dik}_2 = \sqrt{r_{21k}(3)^3/(\mu(1+e_2(i)\cos(r_{24}(i))))} \cdot \sin(w_2(i)+r_{24}(i)) \cdot \cos(w_2(i)+r_{24}(i)) \];
\[ P_{2J2W} = -3\mu\cdot J_2 \cdot (\text{Rearth} / r_{21(i)}^2)^2 \cdot \sin(2\cdot \text{incl}_2(i)) \cdot \sin(w_2(i)+r_{24}(i)); \]
\[ q_{V2} = -B \cdot \exp(A \cdot ((r_{21(i)} - \text{Rearth} - H_0) / H_0)^C) \cdot \text{Cd}_2 \cdot A_2 / (2 \cdot m_2); \]
\[ V_{2w} = \text{dik}_2 \cdot P_{2J2W}; \]
\[ V_{2ws} = 0.0; \]
\[ \text{while abs}(V_{2ws} - V_{2w}) > 0.00001 \]
\[ V_{2ws} = V_{2w}; \]
\[ V_{2w} = V_{2ws} - (-1 / \text{dik}_2 + q_{V2} \cdot \sqrt{r_{23(i)^2 + r_{22(i)^2} + V_{2ws}^2}}) \cdot P_{2J2W} / (-1 / \text{dik}_2 + q_{V2} \cdot (r_{23(i)}^2 + r_{22(i)}^2 + 2 \cdot V_{2ws}^2) / \sqrt{r_{23(i)^2 + r_{22(i)^2} + V_{2ws}^2}}); \]
\[ \text{end} \]
\[ P_{2J2R} = -(3/2) \cdot \mu \cdot J_2 \cdot (\text{Rearth} / r_{21(i)}^2)^2 \cdot (1 - 3 \cdot \sin(\text{incl}_2(i))^2 \cdot \sin(w_2(i)+r_{24}(i))^2); \]
\[ \text{if } e_2(i+1) < 0.0000001 \]
\[ \frac{dw}{dt} = -\text{domg}_2 / \sin(\text{incl}_2(i+1)); \]
\[ \text{else} \]
\[ \frac{dw}{dt} = -\text{domg}_2 \cdot \cos(\text{incl}_2(i+1)) + \sqrt{(r_{21(i)}^2 \cdot e_2(i+1) \cdot \cos(r_{24}(i+1))) / (\mu \cdot (2 \cdot e_2(i+1) \cdot \cos(r_{24}(i+1))) / (e_2(i+1) \cdot (1 + e_2(i+1) \cdot \cos(r_{24}(i+1)))) \cdot \text{Pert}_2 - (\cos(r_{24}(i+1))/e_2(i+1)) \cdot (\text{Pert}_2 + \text{TR}); \]
\[ \text{end} \]
\[ \frac{\text{dIncl}}{\text{dt}} = \sqrt{r_{21(i)} / (\mu(1+e_2(i)\cos(r_{24}(i))))} \cdot \cos(w_2(i)+r_{24}(i)) \cdot \cos(w_2(i)+r_{24}(i)) \cdot (\text{Pert}_2 + \text{TR}); \]
\[ \frac{dv}{dt} = r_{22(i)} \cdot r_{21(i)}/r_{21(i)} - \mu / r_{21(i)}^2 + \text{Pert}_2 + \text{TR}; \]
\[ \text{Pert}_2 = \text{Pert}_2 + \text{TR}; \]
\[ \text{Energy}_2 = (r_{22(i)}^2 + r_{23(i)}^2) / 2 - \mu / r_{21(i)}; \]
\[ \text{dEnergy}_2 / \text{dt} = r_{23(i)} \cdot (\text{Pert}_2 + \text{TR}) + r_{22(i)} \cdot (\text{Pert}_2 + \text{TR}); \]
\[ \text{angular rates} \]
\[ \text{domg}_2 / \text{dt} = \sqrt{r_{21(i)} / (\mu(1+e_2(i)\cos(r_{24}(i))))} \cdot \sin(r_{24}(i)) \cdot \sin(w_2(i)+r_{24}(i))/\sin(\text{incl}_2(i+1)) \cdot (\text{Pert}_2 + \text{TR}); \]
\[ \text{if } e_2(i+1) < 0.0000001 \]
\[ \frac{dw}{dt} = -\text{domg}_2 / \cos(\text{incl}_2(i+1)); \]
\[ \text{else} \]
\[ \frac{dw}{dt} = -\text{domg}_2 \cdot \cos(\text{incl}_2(i+1)) + \sqrt{r_{21(i)} \cdot e_2(i+1) \cdot \cos(r_{24}(i+1)) / (\mu \cdot (2 \cdot e_2(i+1) \cdot \cos(r_{24}(i+1)) / (e_2(i+1) \cdot (1 + e_2(i+1) \cdot \cos(r_{24}(i+1)))) \cdot \text{Pert}_2 - (\cos(r_{24}(i+1))/e_2(i+1)) \cdot (\text{Pert}_2 + \text{TR}); \]
\[ \text{end} \]
\[ \text{V}_{2w} = \text{V}_{2w} + \text{V}_{2ws} \]
\[ \text{Pert}_2 = \text{Pert}_2 + \text{TR}; \]
\[ \text{Energy}_2 = (r_{22(i)}^2 + r_{23(i)}^2) / 2 - \mu / r_{21(i)}; \]
\[ \text{dEnergy}_2 / \text{dt} = r_{23(i)} \cdot (\text{Pert}_2 + \text{TR}) + r_{22(i)} \cdot (\text{Pert}_2 + \text{TR}); \]
\[ \text{angular rates} \]
\[ \text{domg}_2 / \text{dt} = \sqrt{r_{21(i)} / (\mu(1+e_2(i)\cos(r_{24}(i))))} \cdot \sin(r_{24}(i)) \cdot \sin(w_2(i)+r_{24}(i))/\sin(\text{incl}_2(i+1)) \cdot (\text{Pert}_2 + \text{TR}); \]
\[ \text{if } e_2(i+1) < 0.0000001 \]
\[ \frac{dw}{dt} = -\text{domg}_2 / \cos(\text{incl}_2(i+1)); \]
\[ \text{else} \]
\[ \frac{dw}{dt} = -\text{domg}_2 \cdot \cos(\text{incl}_2(i+1)) + \sqrt{r_{21(i)} \cdot e_2(i+1) \cdot \cos(r_{24}(i+1)) / (\mu \cdot (2 \cdot e_2(i+1) \cdot \cos(r_{24}(i+1)) / (e_2(i+1) \cdot (1 + e_2(i+1) \cdot \cos(r_{24}(i+1)))) \cdot \text{Pert}_2 - (\cos(r_{24}(i+1))/e_2(i+1)) \cdot (\text{Pert}_2 + \text{TR}); \]
\[ \text{end} \]
\[ \text{dIncl}_2 / \text{dt} = \sqrt{r_{21(i)} / (\mu(1+e_2(i)\cos(r_{24}(i))))} \cdot \cos(w_2(i)+r_{24}(i)) \cdot \cos(w_2(i)+r_{24}(i)) \cdot (\text{Pert}_2 + \text{TR}); \]
\[ \text{dv}/\text{dt} = r_{22(i)} \cdot r_{21(i)}/r_{21(i)} - \mu / r_{21(i)}^2 + \text{Pert}_2 + \text{TR}; \]
\[ \text{Pert}_2 = \text{Pert}_2 + \text{TR}; \]
\[ \text{Energy}_2 = (r_{22(i)}^2 + r_{23(i)}^2) / 2 - \mu / r_{21(i)}; \]
\[ \text{dEnergy}_2 / \text{dt} = r_{23(i)} \cdot (\text{Pert}_2 + \text{TR}) + r_{22(i)} \cdot (\text{Pert}_2 + \text{TR}); \]
\[ \text{angular rates} \]
\[ \text{domg}_2 / \text{dt} = \sqrt{r_{21(i)} / (\mu(1+e_2(i)\cos(r_{24}(i))))} \cdot \sin(r_{24}(i)) \cdot \sin(w_2(i)+r_{24}(i))/\sin(\text{incl}_2(i+1)) \cdot (\text{Pert}_2 + \text{TR}); \]
\[ \text{if } e_2(i+1) < 0.0000001 \]
\[ \frac{dw}{dt} = -\text{domg}_2 / \cos(\text{incl}_2(i+1)); \]
\[ \text{else} \]
\[ \frac{dw}{dt} = -\text{domg}_2 \cdot \cos(\text{incl}_2(i+1)) + \sqrt{r_{21(i)} \cdot e_2(i+1) \cdot \cos(r_{24}(i+1)) / (\mu \cdot (2 \cdot e_2(i+1) \cdot \cos(r_{24}(i+1)) / (e_2(i+1) \cdot (1 + e_2(i+1) \cdot \cos(r_{24}(i+1)))) \cdot \text{Pert}_2 - (\cos(r_{24}(i+1))/e_2(i+1)) \cdot (\text{Pert}_2 + \text{TR}); \]
\[ \text{end} \]
T213 = \sin(\omega_2(i+1)) \cos(\incl_2(i+1)); 
T221 = \sin(\omega_2(i+1)) \cos(w_2(i+1)) + \cos(\omega_2(i+1)) \sin(w_2(i+1)) \cos(\incl_2(i+1)); 
T222 = -\cos(\omega_2(i+1)) \sin(\incl_2(i+1)); 
T223 = -\cos(\omega_2(i+1)) \sin(\incl_2(i+1)); 
T231 = \sin(w_2(i+1)) \sin(\incl_2(i+1)); 
T232 = \cos(w_2(i+1)) \sin(\incl_2(i+1)); 
T233 = \cos(\incl_2(i+1));

% Inertial Position Coordinates (IJK)
\[ x_2(i+1) = P_2(i+1) \times T211 + Q_2(i+1) \times T212; \]
\[ y_2(i+1) = P_2(i+1) \times T221 + Q_2(i+1) \times T222; \]
\[ z_2(i+1) = P_2(i+1) \times T231 + Q_2(i+1) \times T232; \]

% Inertial Velocity Coordinates (XYZ)
\[ Vx_2(i+1) = V\cdot P_2(i+1) \times T211 + V\cdot Q_2(i+1) \times T212 + V\cdot W_2(i+1) \times T213; \]
\[ Vy_2(i+1) = V\cdot P_2(i+1) \times T221 + V\cdot Q_2(i+1) \times T222 + V\cdot W_2(i+1) \times T223; \]
\[ Vz_2(i+1) = V\cdot P_2(i+1) \times T231 + V\cdot Q_2(i+1) \times T232 + V\cdot W_2(i+1) \times T233; \]

% Relative States
\[ r_{21}(i+1) = r_{21}(i+1) - r_{11}(i+1); \]
\[ r_{22}(i+1) = r_{22}(i+1) - r_{12}(i+1); \]
\[ r_{23}(i+1) = r_{23}(i+1) - r_{13}(i+1); \]
\[ r_{24}(i+1) = r_{24}(i+1) - r_{14}(i+1); \]
\[ w_2(i+1) = w_2(i+1) - w_1(i+1); \]
\[ \theta_{2d}(i+1) = \theta_{2}(i+1) + w_2(i+1) - r_{14}(i+1) - w_1(i+1); \]
\[ \omg_{2d}(i+1) = \omg_2(i+1) - \omg_1(i+1); \]
\[ \incl_{2d}(i+1) = \incl_2(i+1) - \incl_1(i+1); \]
\[ x_2(i+1) = x_2(i+1) - x_1(i+1); \]
\[ y_2(i+1) = y_2(i+1) - y_1(i+1); \]
\[ z_2(i+1) = z_2(i+1) - z_1(i+1); \]
\[ d_2(i+1) = \sqrt{(x_2(i+1)^2 + y_2(i+1)^2 + z_2(i+1)^2); \]
\[ d\text{ Orbit}_2(i+1) = \sqrt{(P_2(i+1) - P_1(i+1))^2 + (Q_2(i+1) - Q_1(i+1))^2); \]
\[ d\text{ Energy}_2(i+1) = E_2(i+1) - E_1(i+1); \]
\[ dE_2(i+1) = e_2(i+1) - e_1(i+1); \]

% SIMULATION PARAMETERS
\[ T(i+1) = T(i) + t\delta; \]
\[ i = i + 1; \]
\[ \text{else break end} \]

figure(1)
plot(theta1/(2*pi), UR1, theta1/(2*pi), PertR1, theta1/(2*pi), ForceR1), xlabel('Orbits'), ylabel('Chief Radial thrust acceleration (m/s^2)')
figure(2)
plot(theta2/(2*pi), UR2, theta2/(2*pi), PertR2, theta2/(2*pi), ForceR2), xlabel('Orbits'), ylabel('Deputy Radial thrust acceleration (m/s^2)')
figure(3)
plot(theta1/(2*pi), US1, theta1/(2*pi), PertS1, theta1/(2*pi), ForceS1), xlabel('Orbits'), ylabel('Chief Perpendicular perturbation acceleration (m/s^2)')
figure(4)
plot(theta2/(2*pi), US2, theta2/(2*pi), PertS2, theta2/(2*pi), ForceS2), xlabel('Orbits'), ylabel('Deputy Perpendicular perturbation acceleration (m/s^2)')
figure(5)
plot(\theta_1/(2\pi), U_{W1}, \theta_1/(2\pi), P_{erW1}), xlabel('Orbits'), ylabel('Chief Out of Plane perturbation acceleration (m/s^2)')
figure(6)
plot(\theta_2/(2\pi), U_{W2}, \theta_2/(2\pi), P_{erW2}), xlabel('Orbits'), ylabel('Deputy Out of Plane perturbation acceleration (m/s^2)')
figure(7)
plot(\theta_1/(2\pi), U_1, \theta_1/(2\pi), P_{er1}), xlabel('Orbits'), ylabel('Total Chief accelerations (m/s^2)')
figure(8)
plot(\theta_2/(2\pi), U_2, \theta_2/(2\pi), P_{er2}), xlabel('Orbits'), ylabel('Total Deputy accelerations (m/s^2)')
figure(9)
plot(\theta_1/(2\pi), r_{11d}), xlabel('Orbits'), ylabel('Chief radial distance from trajectory (m)')
figure(10)
plot(\theta_2/(2\pi), r_{21d}), xlabel('Orbits'), ylabel('Deputy radial distance from trajectory (m)')
figure(11)
plot(\theta_1/(2\pi), r_{12d}), xlabel('Orbits'), ylabel('Chief perpendicular velocity from trajectory (m/s)')
figure(12)
plot(\theta_2/(2\pi), r_{22d}), xlabel('Orbits'), ylabel('Deputy perpendicular velocity from trajectory (m/s)')
figure(13)
plot(\theta_1/(2\pi), r_{13d}), xlabel('Orbits'), ylabel('Chief radial velocity from trajectory (m/s)')
figure(14)
plot(\theta_2/(2\pi), r_{23d}), xlabel('Orbits'), ylabel('Deputy radial velocity from trajectory (m/s)')
figure(15)
plot(\theta_1/(2\pi), \theta_{1d}, \theta_1/(2\pi), r_{14d}, \theta_1/(2\pi), w_{1d}), xlabel('Orbits'), ylabel('Chief planar angular distance from trajectory (rads)')
figure(16)
plot(\theta_2/(2\pi), \theta_{2d}, \theta_2/(2\pi), r_{24d}, \theta_2/(2\pi), w_{2d}), xlabel('Orbits'), ylabel('Deputy planar angular distance from trajectory (rads)')
figure(17)
plot(\theta_1/(2\pi), d\theta_{1d}/dt, \theta_1/(2\pi), dr_{14d}/dt, \theta_1/(2\pi), dw_{1d}/dt), xlabel('Orbits'), ylabel('Chief planar angular velocity from trajectory (rads/s)')
figure(18)
plot(\theta_2/(2\pi), d\theta_{2d}/dt, \theta_2/(2\pi), dr_{24d}/dt, \theta_2/(2\pi), dw_{2d}/dt), xlabel('Orbits'), ylabel('Deputy planar angular velocity from trajectory (rads/s)')
figure(19)
plot(\theta_1/(2\pi), \omega_{1d}), xlabel('Orbits'), ylabel('Chief ascending node angular distance from trajectory (rads)')
figure(20)
plot(\theta_2/(2\pi), \omega_{2d}), xlabel('Orbits'), ylabel('Deputy ascending node angular distance from trajectory (rads)')
figure(21)
plot(\theta_1/(2\pi), i_{1d}), xlabel('Orbits'), ylabel('Chief inclination angular distance from trajectory (rads)')
figure(22)
plot(\theta_2/(2\pi), i_{2d}), xlabel('Orbits'), ylabel('Deputy inclination angular distance from trajectory (rads)')
figure(23)
plot(P_2, Q_2, P_1, Q_1, P_{1traj}, Q_{1traj}), xlabel('P position (m)'), ylabel('Q position (m)')
axis equal; axis square; grid on
figure(24)
plot3(x2,y2,z2,x1,y1,z1,x1traj,y1traj,z1traj),xlabel('x position (m)'),ylabel('y position (m)'),zlabel('z position (m)')
axis equal; axis square; grid on
figure(25)
plot(theta1/(2*pi),e1),xlabel('Orbits'),ylabel('Chief Eccentricity')
figure(26)
plot(theta2/(2*pi),e2),xlabel('Orbits'),ylabel('Deputy Eccentricity')
figure(27)
plot(theta1/(2*pi),dEnergy1),xlabel('Orbits'),ylabel('Chief Energy')
figure(28)
plot(theta2/(2*pi),dEnergy2),xlabel('Orbits'),ylabel('Deputy Energy')
figure(29)
plot(r11d,r13d),xlabel('Radial'),ylabel('Chief Bode graph')
figure(30)
plot(r21d,r23d),xlabel('Radial'),ylabel('Deputy Bode graph')
figure(31)
plot(de1,ade1dt),xlabel('Eccentricity'),ylabel('Chief Bode graph')
figure(32)
plot(de2,ade2dt),xlabel('Eccentricity'),ylabel('Deputy Bode graph')
figure(33)
plot(dEnergy1,dEnergy1dt),xlabel('Energy'),ylabel('Chief Bode graph')
figure(34)
plot(dEnergy2,dEnergy2dt),xlabel('Energy'),ylabel('Deputy Bode graph')
figure(35)
plot(omg1d,domg1dt),xlabel('Ascending Node'),ylabel('Chief Bode graph')
figure(36)
plot(omg2d,domg2dt),xlabel('Ascending Node'),ylabel('Deputy Bode graph')
figure(37)
plot(incl1d,dincl1dt),xlabel('Inclination'),ylabel('Chief Bode graph')
figure(38)
plot(incl2d,dincl2dt),xlabel('Inclination'),ylabel('Deputy Bode graph')
figure(39)
plot(theta2/(2*pi),d2),xlabel('Orbits'),ylabel('Deputy distance (m)')
figure(40)
plot3(x2d,y2d,z2d),xlabel('Orbits'),ylabel('Deputy distance (IJK) (m)')
axis equal; axis square; grid on
return
LIST OF REFERENCES


