AE 8133 – Space Mechanics

Project Phase III: Final Report

Dynamics of Docking of a Large Spacecraft to a Large Modular Orbiter in a Martian Orbit

Aman Gilani

500879895

Dynamics of Docking of a Large Spacecraft to a Large Modular Orbiter in a Martian Orbit

1. Introduction

The evolution of space exploration and docking systems has come a long way spanning several decades and numerous missions. These systems have played a critical role in enabling humans to conduct complex scientific missions in a sophisticated manner. As we move toward lunar exploration with the Artemis program and deep space exploration with the Mars rover missions, one of the main responsibilities of space exploration and interplanetary missions is the dynamics of docking. Future docking systems will play a fundamental role in assembling deep space and lunar habitats and transferring crew and cargo between Martian/lunar habitat and gateway orbit.

Figure 1: Mars Orbiter and Spacecraft Mission Vision. [1]

The above figure illustrates the vision put forth for the mission analyzed for this project. As seen, a large spacecraft is undergoing docking procedure with a large modular orbiter. This mission will enable us to set up a Martian colony thus furthering space exploration and scientific innovations.

The mission focuses on computing and analyzing the dynamics of orbit insertion of a large spacecraft and the dynamics of docking of the same spacecraft to a modular gateway orbiter. Additionally, this analysis will incorporate the computation of the launch dynamics of the spacecraft. The main objective of this mission is to compute and evaluate the satellite's equation of motion before, during, and after the docking procedure. This analysis will help us in studying the orbital and attitude dynamics of these satellites in the Martian orbit.

This report will emphasize developing and solving the dynamic mathematical model of systems defined for this mission. The dynamics model will incorporate the assumptions, coordinate frames, satellites and planetary bodies, external disturbances, and the desired trajectory defined in the phase I report. The approximations and the defined trajectory will enable us to simplify the complex computations of orbital and attitude dynamics for this mission. Further, this report will examine the numerical simulations completed based on the equations developed in the phase II report. These simulations include docking trajectory tracking, gravitational perturbations, and orbit change maneuvers for the respective satellites in this mission.

By the end of this project, one will be able to completely understand the complexities of orbital mechanics and attitude dynamics used to model the large spacecraft and a large modular gateway orbiter in a Martian orbit.

2. Mission Objectives & Performance Measures

This mission is evaluated based on the mission objectives, mission constrains, and mission performance measures set in the phase I report. These objectives and performance measures will help us in deriving the assumptions, and equations of motion for the systems defined for this mission.

2.1 Mission Objectives

2.1.1 Primary Objectives

The following are the main mission objectives set for the course of this project. By the end of this project, we aim to understand the complexities of the orbital mechanics of the given problem statement.

- To know the position, velocity, and acceleration of the large spacecraft and the large modular orbiter during each phase of the docking procedure.
- To know the momentums, and body forces acting on the two spacecraft during each phase of the docking procedure.
- To know all the steps needed to perform an accurate and safe docking in the Martian orbit.
- To understand the orbital dynamics of the two spacecraft before, during and after docking.

2.1.2 Secondary Objectives

The following are the secondary objectives set for the course of this project. These objectives are established to further optimize the docking procedure. The following protocols are aimed at maximizing safety and designing redundant systems to mitigate potential failure modes.

- Develop **Safety Protocols** to minimize the risk of misalignment and collision during the docking procedure.
- Design **Redundancy Protocols and Systems** to mitigate potential failures in orbit capture and during the docking procedures.
- **Optimize Trajectories** of the spacecraft and the orbiter to minimize energy consumption and maximize efficiency.

2.1.3 Mission Constraints

The following are the mission constraints established for the course of this project. These mission constraints will enable the desired trajectory design to account for the environmental variations and thus streamline the computations.

- **Resource Limitations:** These include computation restrictions and point mass and rigid body assumption restrictions for spacecraft and orbiters.
- **Spacecraft Orbital Variability:** Considering the various positions of the spacecraft launch location and the orbiter orbit, which may affect the available windows for docking between the spacecraft and orbiter.
- **Environmental Factors:** These include the Martian and its moon's environmental factors, such as dust storms or radiation and gravitational perturbation which could impact the trajectory and attitude.

2.2 Mission Performance Measures

The following are the mission performance measures established for the course of this project. These procedures will assist us in determining the performance of this mission and ensure safety during each phase of the docking procedure.

- **Docking Success Rate:** To calculate the number of successful dockings in diverse circumstances to evaluate the overall docking success rate.
- **Docking Time:** To calculate the average time required for a successful docking procedure.
- **Orbital Stability:** To evaluate the orbital stability of the spacecraft and modular orbiter after docking, ensuring they remain in a stable orbit.

3. System Model

3.1 Assumptions

Throughout this problem, approximating a celestial body or a satellite body as a point mass or a rigid body is critical to accurately compute and examine the trajectory and attitude of the desired entity. The planet Mars and the two spacecraft (active spacecraft and MGO) are approximated as point masses to calculate the spacecraft and the orbiter's trajectory accurately. However, this approximation works well only when the satellite body orbits the planet for a shorter duration. Therefore, to account for the gravitational field variation of Mars's shape onto the satellite's trajectories the planet Mars will be later analyzed as a rigid body. This approximation will be made to compute the perturbation effects on the orbiter as well as on the spacecraft after docking.

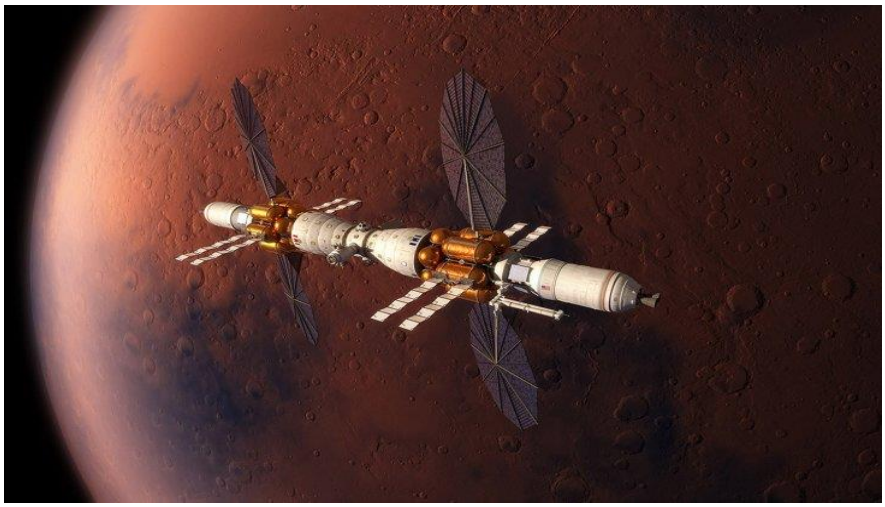

Figure 2: Gateway Orbiter in a Martian Orbit at 200Km. [2]

The planet Mars is approximated as a point mass and the satellites (active spacecraft and MGO) are modeled as rigid bodies to compute and analyze the attitude or orientation of the two satellites. For the course of this project, the gateway orbit altitude for the MGO is initially estimated to be 200 Km from the surface of Mars. The orbital plane inclination is defined at 45 degrees from the equatorial plane of Mars. The orbiter mass is approximated, similar to the mass of ISS, to be 420,000 Kg [3]. The mass of the spacecraft is initially estimated to be 45,000 Kg. After the preliminary velocity analysis, the orbiter has a velocity of 3450 m/s for it to maintain the circular orbit at 200 Km. However, moving on in the project, the stability of the orbit will be analyzed using the equations of motion.

3.2 System Description

3.2.1 Coordinate Frames

For the planet, firstly an inertial frame is utilized that is body-centered to Mars. This frame is non-rotating and fixed with the center of Mars. A second frame is used which is a rotating frame concentric with the center of Mars. This frame will account for the rotation of Mars. Third, a body-fixed Euler angle frame is utilized for the orbiter and the spacecraft to compute their respective equations of motion and eventually their attitude. This frame incorporates the pitch angle, roll angle, and yaw angle to illustrate the orientation of the satellite. Lastly, a Nadir-pointing body-fixed frame is used for the satellites to compute their trajectories. In this frame, one axis is always pointed towards the center of Mars, and the other two axes are aligned with the satellite body.

3.2.2 Satellites and Planetary Bodies

The systemic dynamic model developed this mission encompasses various primary planetary bodies and two satellite bodies. The primary celestial entities include the planet Mars and its moons Phobos and Deimos. The Spacecraft bodies comprise a large spacecraft and a modular gateway orbiter. All these entities are considered independent objects within the Martian orbit. However, the gravitational variations will induce perturbation effects on the two spacecraft as they orbit the planet for longer durations. This subject will be analyzed further to study the effect of gravitational variation on the two spacecraft.

3.3 System Equations of Motion

The systemic Dynamic Model for the mission is separated into three sub-sections. Each subsection will study the equations of motion relative to the assumptions, satellites, and planetary bodies defined for the system.

3.3.1 Phase I - MGO and Mars

The study of the Modular Gateway Orbiter's motion and trajectory around the planet Mars will be evaluated in this section. The motion of the MGO, specifically its orbital trajectory, will be expressed through the following equations of motion. These equations were developed by approximating the Orbiter and the planet Mars as point masses and estimating an in-plane circular orbit of 200 Km for the orbiter. The in-plane motion approximation for the orbiter is further modified to account for the inclination of the orbit into the equations of motion. This will be accomplished through using Rotation matrices to transform from body fixed cylindrical frame to body fixed rotating frame in terms of orbital elements.

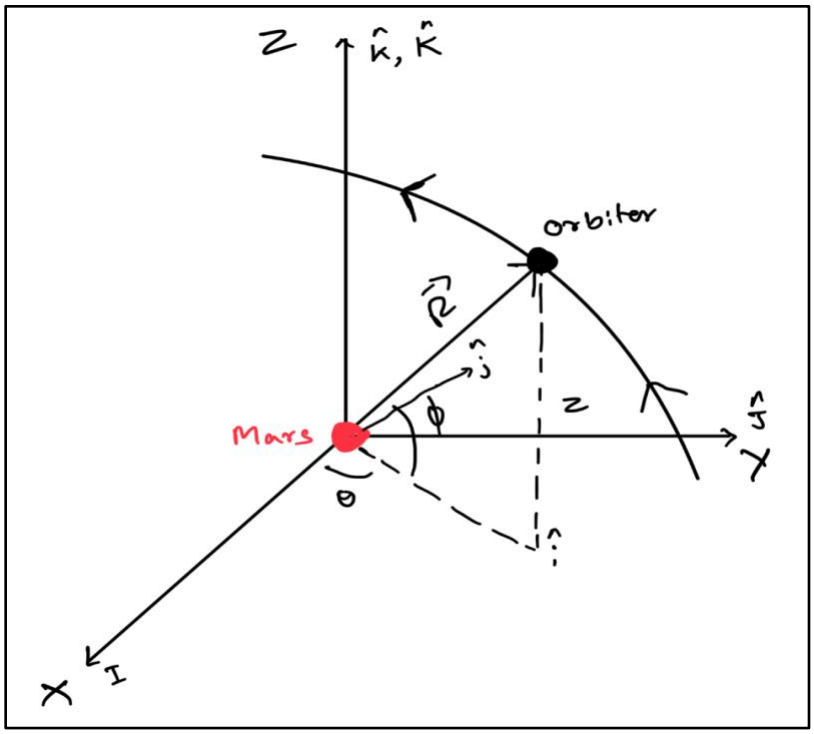

Figure 3: Mgo and Mars Perturbation Frame.

Position, Velocity and Acceleration

$$\vec{R} = R\hat{\imath} + Z\hat{k}$$

$$\dot{\vec{R}} = \dot{R}\hat{\imath} + \dot{\theta}R\hat{\jmath} + \dot{Z}\hat{k}$$

$$\ddot{\vec{R}} = \left[\ddot{R} - R\dot{\theta}^2\right]\hat{\imath} + \left[2\dot{R}\dot{\theta}\right]\hat{\jmath} + \ddot{Z}\hat{k}$$

Kinetic Energy

$$T = \frac{1}{2}m\dot{\vec{R}} = \frac{1}{2}M_o\left[\dot{R}^2 + \dot{\theta}^2R^2\right]$$

Potential Energy

$$U = -\frac{\mu M_o}{R}$$

R – Equation

$$\ddot{R} - \dot{\theta}^2R + \frac{\mu}{R^2} = 0$$

θ – Equation

$$\ddot{\theta}R^2 + 2R\dot{R}\dot{\theta} = 0$$

Frame transformation in terms of orbital elements.

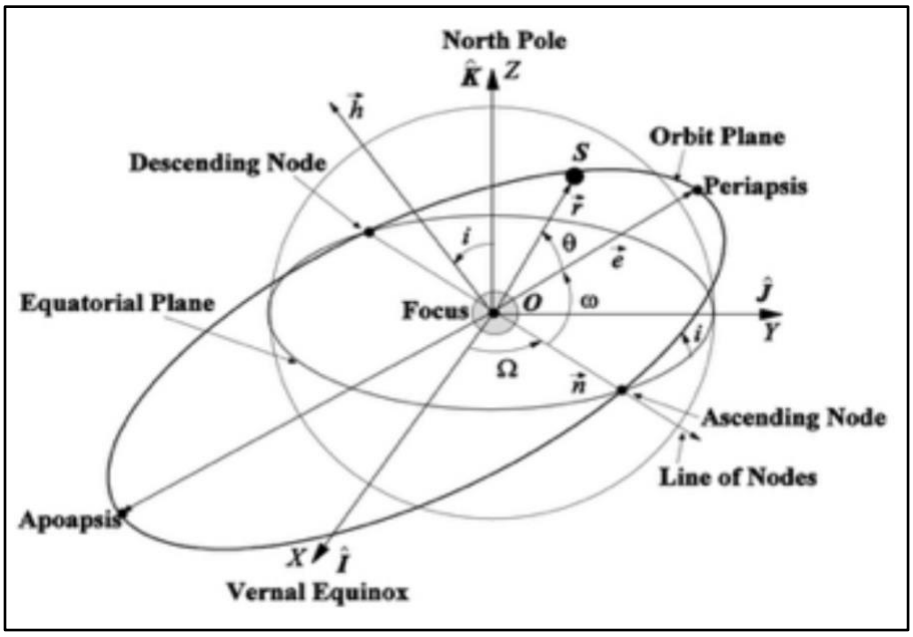

Figure 4: Orbital Frame.

$$\begin{Bmatrix} i \\ j \\ k \end{Bmatrix} = R_3(\omega + \theta)R_1(i)R_3(\Omega)\begin{Bmatrix} i_o \\ j_o \\ k_o \end{Bmatrix}$$

3.3.2 Phase II - Spacecraft and Mars

This section of the report will analyze the launch velocity and the delta-V equations for orbital insertion of an active spacecraft. Keeping the inclination of the orbit in mind, an azimuth launch angle will also be computed in order to insert the spacecraft into orbit. This phase of the mission is very critical since, along with the launch velocity the launch timing is also an important factor for a successful orbit insertion. The above parameters are designed in such a way that the spacecraft enters an elliptical orbit while the ascending or descending node is also the apogee of the elliptical orbit.

Launch Azimuth Angle [4], [5]

$$\beta = \arccos\left(\frac{\cos\left(inclination(i = 45^o)\right)}{\cos\left(latitude(\Phi = 23^o)\right)}\right)$$

Launch Timing and Orbital Velocity [4]

$$Orbital\ Velocity = \sqrt{\frac{G\left(M_{Mars} + M_{spacecraft}\right)}{R_m + R}}$$

$$Time\ to\ Orbit_{Circular}\ (t) = \frac{R^{\frac{3}{2}}}{\sqrt{\mu_{Mars}}} * 2\pi$$

$$Time\ to\ Orbit_{elliptical}\ (t) = \frac{a^{\frac{3}{2}}}{\sqrt{\mu_{Mars}}} * 2\pi$$

Therefore, the launch window is calculated by equating the time periods for the circular and elliptical orbits and computing the location of the orbiter at this time. A similar approach is used to compute the second launch window for the redundant docking system.

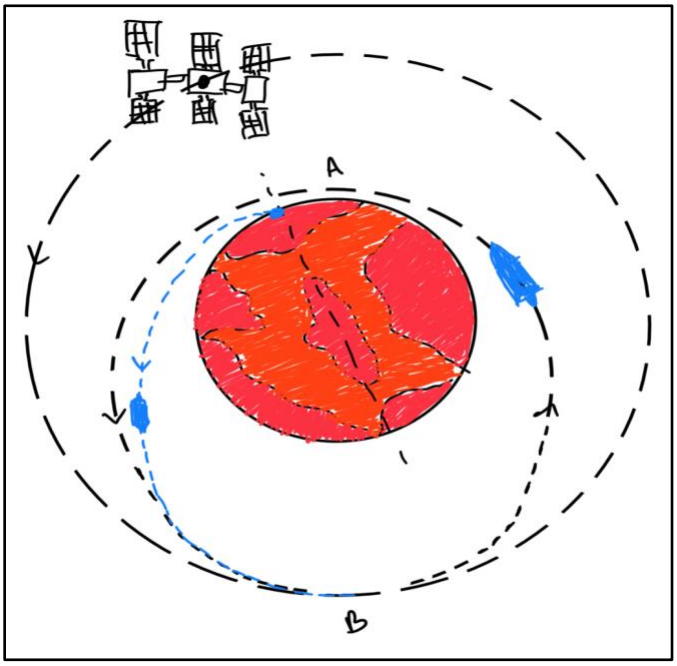

Figure 5: Hohmann Transfer.

The above figure illustrates the launch sequence and the orbit insertion of the spacecraft into the gateway orbit for docking to the MGO. The spacecraft launches from the Martian surface respective to the location of the MGO in the orbit. The launch window is calculated in such a way that both satellites are scheduled to meet at the apoapsis of the orbit. As the spacecraft approaches the modular orbiter, it interjects itself into the circular orbit and adjusts its attitude for the docking procedure.

However, as a redundant system, in the case the spacecraft misses the Modular gateway orbiter, the spacecraft modifies its orbit to an elliptical orbit as shown in the figure. Then the spacecraft changes its orbit from an elliptical one to a circular one in such a way that its relative position is in front of the MGO. Following this, the spacecraft slows down and adjusts its orientation for the docking procedure.

The following are the Delta-V equations for orbit insertion and return journey.

$$\mu_1 = G\left(M_{Mars} + M_{spacecraft}\right)$$

Delta-V to insert into elliptical orbit [6]

$$\Delta V_A = V_{A,ellipse} - V_{A,circle}$$

$$\Delta V_A = \sqrt{2\mu_1}\sqrt{\frac{R}{R_m(R+R_m)}} - \sqrt{\frac{\mu_1}{R_m}}$$

The above Delta – V_A equation is used for the redundant system; in the case the spacecraft misses the Modular gateway orbiter. This Delta-V will keep the spacecraft into an elliptical orbit until the next docking window.

Delta-V to insert into a new circular orbit [6]

$$\Delta V_B = V_{B,circle} - V_{B,ellipse}$$

$$\Delta V_B = \sqrt{\frac{\mu_1}{R}} - \sqrt{2\mu_1}\sqrt{\frac{R_m}{R(R+R_m)}}$$

Total Delta-V for orbit insertion [6]

$$\Delta V_{Total} = |\Delta V_A| + |\Delta V_B|$$

$$\Delta V_{Total} = \left|\sqrt{2\mu_1}\sqrt{\frac{R}{R_m(R+R_m)}} - \sqrt{\frac{\mu_1}{R_m}}\right| + \left|\sqrt{\frac{\mu_1}{R}} - \sqrt{2\mu_1}\sqrt{\frac{R_m}{R(R+R_m)}}\right|$$

Total Delta-V for Return Journey [6]

$$\Delta V_{Total} = |\Delta V_B| + |\Delta V_A|$$

$$\Delta V_{Total} = \left|\sqrt{2\mu_1}\sqrt{\frac{R_m}{R(R+R_m)}} - \sqrt{\frac{\mu_1}{R}}\right| + \left|\sqrt{\frac{\mu_1}{R_m}} - \sqrt{2\mu_1}\sqrt{\frac{R}{R_m(R+R_m)}}\right|$$

3.3.3 Phase III - Spacecraft and MGO – Approach (Trajectory) & Docking (Attitude)

This section of the report will analyze the equations of motion for the spacecraft during the docking procedure. The docking procedure is divided into two characteristics to linearize the equations of motion. The equations are computed and analyzed into the spacecraft's trajectory and attitude. The trajectory of the spacecraft also called the approach during docking can be computed by approximating the spacecraft and the orbiter as point mass bodies. The orientation of the spacecraft also known as its attitude equation of motion can be calculated by approximating the two bodies as rigid. Since the equations of motion for approach between the two satellites is coplanar, this motion can also be analyzed as in-plane motion to simplify the mathematical complications while solving the Lagrange equations.

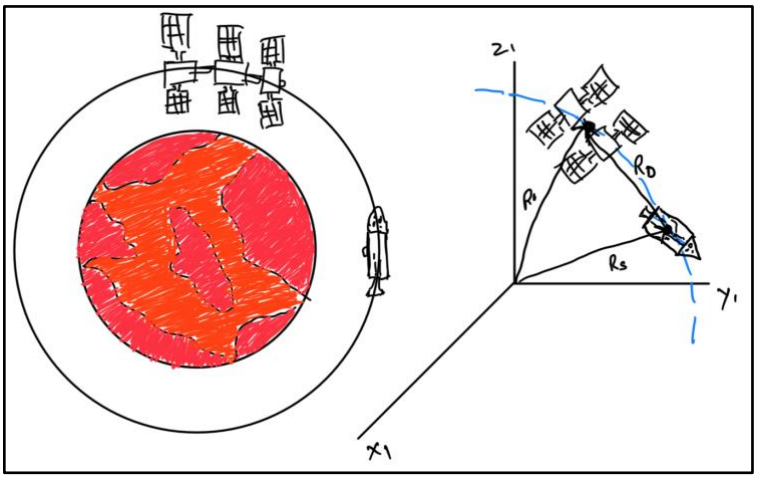

Figure 6: MGO and Spacecraft Docking Frame.

The above figure illustrates the coordinate system for the approach trajectory between the two satellites. The active spacecraft in this analysis is also referred as a follower satellite and the orbiter is referred as a leader satellite.

The aim of this analysis is to decrease the relative distance between the two satellites as the time goes on. This relationship can be defined with the following position, and acceleration equations relative to the two satellites.

Relative Equations of Motion [7]

$$\vec{\rho} = x\hat{\imath} + y\hat{\jmath} + z\hat{k}$$

$$\ddot{\vec{\rho}} = \left[\ddot{x} - x\dot{\theta}^2 - 2\dot{y}\dot{\theta}\right]\hat{\imath} + \left[\ddot{y} - y\dot{\theta}^2 + 2\dot{x}\dot{\theta}\right]\hat{\jmath} + [\ddot{z}]\hat{k}$$

Therefore, the equations of motion for the follower satellite (active spacecraft) with respect to the leader satellite (MGO) are presented as follows.

$$\ddot{x} - x\dot{\theta}^2 - 2\dot{y}\dot{\theta} = -\frac{\mu}{R^3}[R_o + x] + \frac{\mu}{R_o^2} + u(x)$$

$$\ddot{y} - y\dot{\theta}^2 + 2\dot{x}\dot{\theta} = -\frac{\mu}{R^3}y + u(y)$$

$$\ddot{z} = -\frac{\mu}{R^3}z + u(z)$$

Where $R = \sqrt{(R_o + x)^2 + (y)^2 + (z)^2}$ and $\dot{\theta} = \sqrt{\frac{\mu}{R^3}}$

The theta dependent linearized form the above equations are as follows,

$$x'' - 2y' - 3x = 0$$

$$y'' + 2x' = 0$$

$$z'' + z' = 0$$

The input control forces for the approach can be modelled with the following proportional and differential control system force inputs,

$$U_x = -K_p(X - X_C) - K_d(\dot{X} - \dot{X_C})$$

$$U_y = -K_p(Y - Y_C) - K_d(\dot{Y} - \dot{Y_C})$$

$$U_x = -K_p(Z - Z_C) - K_d(\dot{Z} - \dot{Z_C})$$

The above control input parameters are essential for the spacecraft to maintain the desired trajectory. The K_p and K_d control parameters are obtained from historic references. However, the K_p parameter is always greater than K_d parameter for the force inputs to have an order of 10^{-3}.

3.4 External disturbance model

The systemic dynamic model of this problem in terms of body forces majorly constitutes gravitational force variation. In addition to the gravitational force, other external disturbances such as solar pressure radiation, aerodynamic drag from the Martian atmosphere, and Mars's magnetic field pressure play a crucial role in determining the spacecraft and orbiter's motion and orientation. These external force disturbances are usually addressed in the mathematical formulation of nonconservative forces thus providing a comprehensive study of the effect of these forces on the two spacecraft. For the course of this project, it will only analyze the effects gravitational perturbation on the modular gateway orbiter.

For the course of this mission, the aerodynamic drag on the orbiter is approximated to be negligible. This estimation is based on the defined orbit altitude and lower surface atmospheric density on Mars (approximately 0.02 Kg/m^3) []. This is a fair approximation for the orbiter as it stays in a circular orbit 200 Km above the Martian surface. On the other hand, the effects of the Martian atmosphere should be considered on the spacecraft while launching from the Martian surface.

3.4.1 Gravitational Perturbations

This section of the report will focus on computing the effects of gravitational perturbation from the shape of mars on the orbiter. This consideration is of significance because the orbiter will stay in the Martian orbit for longer durations. Therefore, the gravitational variation of mars has a significant effect on the orbiter's trajectory over longer durations. This analysis approximates the planet Mars as a rigid body and the orbiter as a point mass body in a circular orbit of 200 Km. The J_2 constant for the planet Mars was obtained from historic tables and is approximately $1960.45 * 10^{-6}$[3].

Potential Energy

$$U = -\frac{\mu M_o}{R}\left[1 - \frac{1}{2} * \frac{J_2 R_m^2}{R^2}\left(3\frac{Z^2}{R^2} - 1\right)\right]$$

Where $\mu = GM_{Mars}$, $J_2 = 1960.45 * 10^{-6}$ & $Z = \frac{3\sin^2\lambda - 1}{2}$, $\lambda = Satellite\ Latitude$

Gravitational Force per Unit Mass

$$f_g = -\frac{\mu}{R^2}\hat{\imath} + \frac{3}{2}\frac{J_2 R_m^2}{R^4}\left[\left(-\frac{5Z^2}{R^2} + 1\right)\hat{\imath} + \frac{2Z}{R}\hat{k}\right]$$

Equations of Motion

$$\ddot{x} = -\frac{\mu}{R^3}x - \frac{3\mu J_2 R_m^2}{2R^5}\left[1 - \frac{5Z^2}{R^2}\right]x$$

$$\ddot{y} = -\frac{\mu}{R^3}y - \frac{3\mu J_2 R_m^2}{2R^5}\left[1 - \frac{5Z^2}{R^2}\right]y$$

$$\ddot{z} = -\frac{\mu}{R^3}z - \frac{3\mu J_2 R_m^2}{2R^5}\left[3 - \frac{5Z^2}{R^2}\right]z$$

Where $R = \sqrt{(x)^2 + (y)^2 + (z)^2}$

3.5 Reference or Desired Trajectory

A crucial aspect of this mathematical problem formulation is to determine the desired trajectory of the orbiter and the active spacecraft. These trajectories are important to study the path and motion of the spacecraft as it launches the Martian surface, inserts itself into the gateway orbit, and finally docks to the modular gateway orbiter. The trajectory incorporates complex elements of orbital mechanics that is the plane inclination and eccentricity. Therefore, it is important to take the above-mentioned assumptions, coordinate frames, and external disturbances to compute the satellite's equation of motion and their stability during orbit insertion and the docking procedure.

The problem formulation for the desired trajectory of the satellites is divided into three following cases. These cases will focus on specific situations in the mission with specific satellites and the respective planetary body.

3.5.1 MGO and Mars – Orbit and Perturbation Effects

The first case for trajectory computation is between the Modular Gateway Orbiter and the planet Mars. As seen in the figure below, the orbiter orbits in a circular orbit with an altitude of 200 Km from the Martian surface. The approximation of both bodies as a point mass is valid for the computation of the circular orbit trajectory.

However, as discussed previously, the perturbation effects from the variation in the gravitational field will cause major trajectory fluctuations. Therefore, to account for this phenomenon, the planet Mars is approximated as a rigid body. This consideration is of significance because the orbiter will remain in the Martian orbit for longer durations. Therefore, the gravitational variation of mars has a significant effect on the orbiter's trajectory over longer durations. The equations of motion for this phenomenon are stated in the previous section under the gravitational perturbation subsection. In order to mitigate the effects of gravitational field variation on the orbiter, it is necessary for the orbiter to boost its trajectory from time to time in order to maintain a 200 km circular orbit at an inclination of 45 degrees.

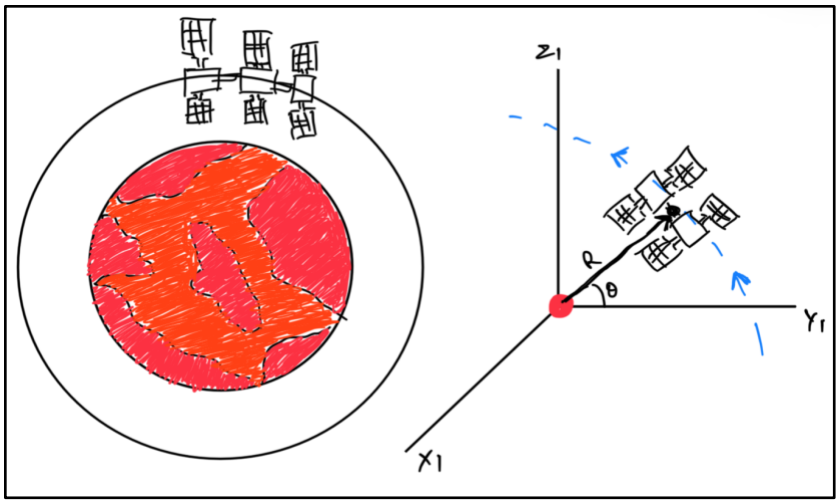

Figure 7: Gravitational Perturbation Frame.

3.5.2 Spacecraft – Launch and Orbit Insertion (Phase II)

The second case for this analysis is relative to the spacecraft, Modular Gateway Orbiter, and the planet Mars. The case incorporates the launch sequence and the orbit insertion of the spacecraft into the gateway orbit.

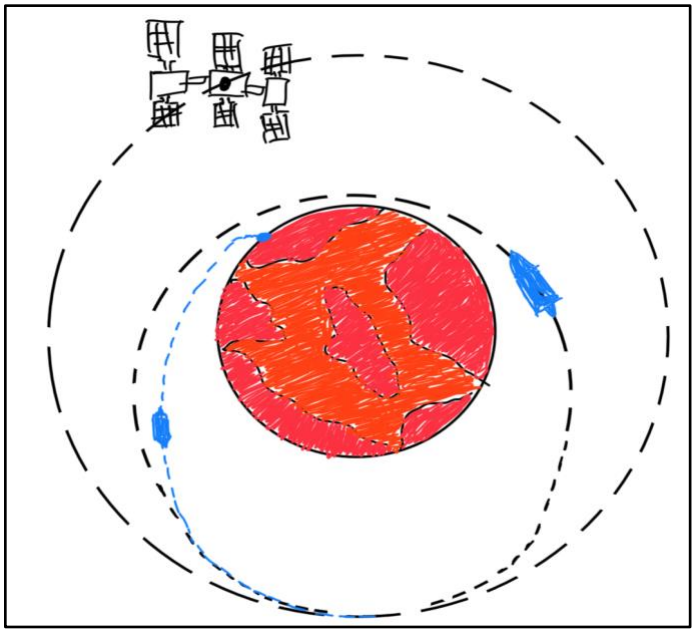

Figure 8: Launch and Orbit Transfer.

The above figure illustrates the launch sequence and the orbit insertion of the spacecraft into the gateway orbit for docking to the MGO. The spacecraft launches from the Martian surface respective to the location of the MGO in the orbit. The launch window is calculated in such a way that both satellites are scheduled to meet at the apoapsis of the orbit. As the spacecraft approaches the modular orbiter, it interjects itself into the circular orbit and adjusts its attitude for the docking procedure.

The equations governing the launch window, launch speed and Delta-V for orbit transfer are presented in the above section under Equations of motion of the spacecraft subsection. Along with this, the Azimuth launch angle for the spacecraft is also computed in order to insert the spacecraft into the inclined orbit from the launch location with an altitude of 23 degrees. The above parameters are designed in such a way that the spacecraft enters an elliptical orbit while the ascending or descending node is also the apogee of the elliptical orbit.

However, as a redundant system, in the case the spacecraft misses the Modular gateway orbiter, the spacecraft modifies its orbit to an elliptical orbit as shown in the figure. Then the spacecraft changes its orbit from an elliptical one to a circular one in such a way that its relative position is in front of the MGO. Following this, the spacecraft slows down and adjusts its orientation for the docking procedure.

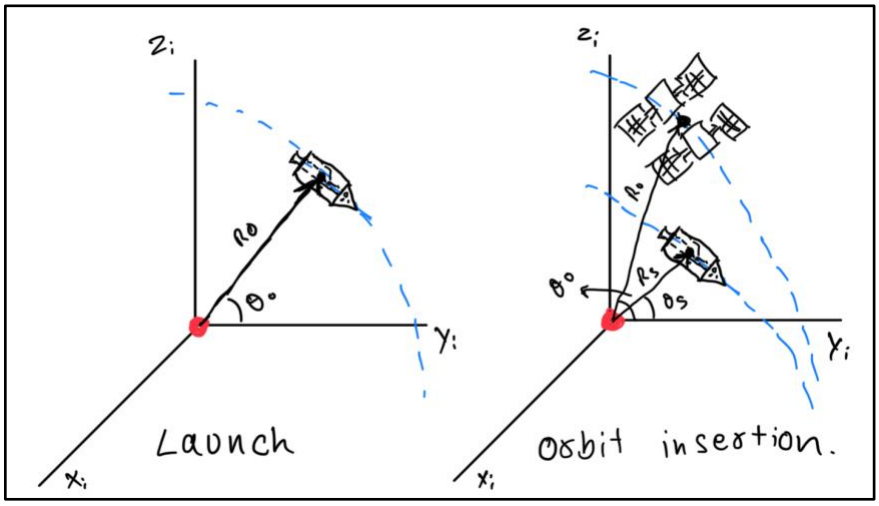

Figure 9: Launch and Orbit Insertion Frame.

The above figure illustrates the coordinate frame for this case. The first frame depicts the launch, and the second frame depicts the orbit insertion. The trajectory for the launch process, from the launch to orbit insertion for the spacecraft, approximates the planet Mars and the spacecraft as a point mass. The orbit insertion of the spacecraft into the gateway orbit also approximates the spacecraft and the MGO as a point mass. For the redundant case, a similar approximation is used to calculate the elliptical orbit of the spacecraft.

3.5.3 Spacecraft, MGO, and Mars – Approach and Docking (Phase III)

The third case in this problem formulation is concerning the spacecraft and the modular orbiter. Once the spacecraft enters the gateway orbit, this scenario majorly incorporates the attitude adjustment and corrections necessary for the docking procedure.

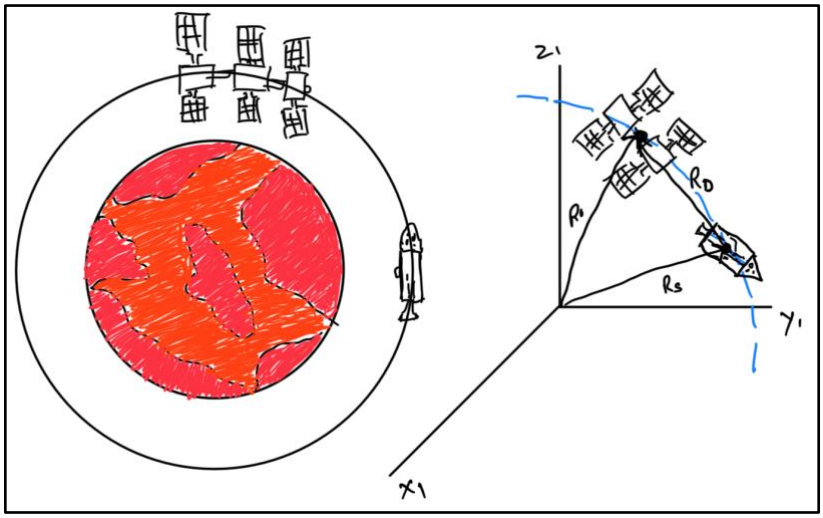

Figure 10: Approach and Docking Desired Trajectory.

The above figure illustrates the coordinate frame for the docking procedure. In this case, both satellites, the active spacecraft, and the MGO, are approximated as rigid bodies. This approximation is necessary to compute the attitude changes and adjustments of the spacecraft before the docking procedure. The relative position, velocity, and acceleration between the two satellites are very crucial to study the spacecraft's motion relative to the MGO.

The equations of motion of spacecraft section above analyzes the equations of motion for the spacecraft during the docking procedure. The docking procedure is divided into two characteristics to linearize the equations of motion. The equations are computed and analyzed into the spacecraft's trajectory and attitude. The trajectory of the spacecraft also called the approach during docking can be computed by approximating the spacecraft and the orbiter as point mass bodies. The orientation of the spacecraft also known as its attitude equation of motion can be calculated by approximating the two bodies as rigid.

4. Mathematical Analysis

This section of the report focuses on analyzing the docking process for the spacecraft with respect to the orbiter. This will include developing of a desired trajectory of the path the spacecraft will take during the docking procedure. This analysis incorporates the assumptions and constraints defined in the previous sections.

Following the docking analysis, this section will also analytically study the Delta V calculations to determine the launch window and velocity for the active spacecraft. As presented in the previous sections, the launch location for the active spacecraft on Mars is pre-determined along with the spacecraft and orbiter's mass. These parameters will further enable us to compute the launch timing, orbital velocity, and orbital time period for the two satellites.

4.1 Docking Analysis.

The docking sequence for the satellite system begins when the active spacecraft is within a 5 Km radial distance with the modular gateway orbiter. Thus, following this we define various 3D functions in X, Y, and Z coordinates for a line to illustrate a desired path for the active spacecraft.

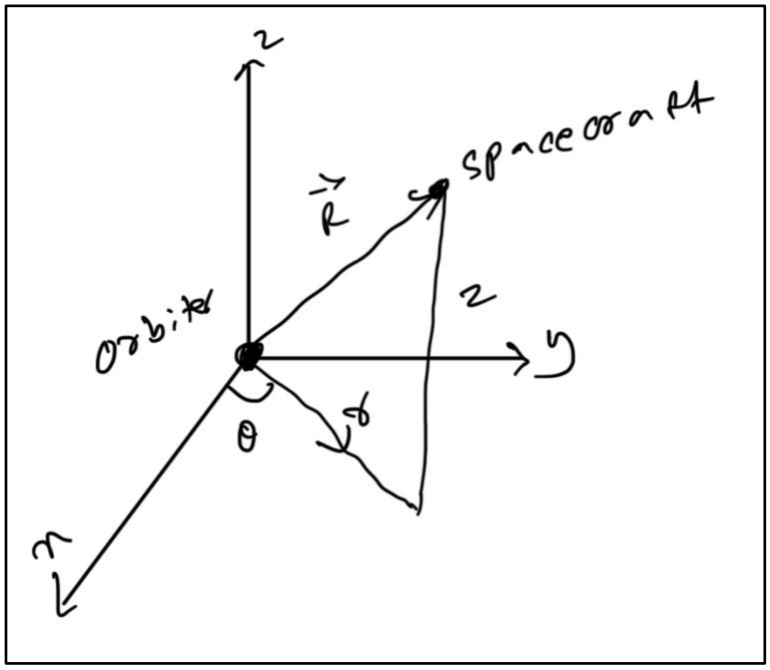

Figure 11: Docking Analysis Frame.

The desired position for the active spacecraft is presented following in sinusoidal functions and in cylindrical coordinate frame. The cylindrical coordinate frame was considered because of its versatility while transforming the equations to cartesian, non-inertial frame if needed. The function uses sinusoidal functions to plot the trajectory from time = 0 until docking is complete.

The time required for the entire docking sequence is also predefined and is approximated to be 60 min.

$$r_d = 5000\left(1 - \sin\left(\frac{\pi t}{2T}\right)\right)$$

$$Z_d = 5000\left(1 - \sin\left(\frac{\pi t}{2T}\right)\right)$$

$$theta_d = 45^o$$

$$x_d = r_d * \cos(theta_d)$$

$$y_d = r_d * \sin(theta_d)$$

The desired velocity for the active spacecraft is computed by taking the derivative for the above given position functions in the cylindrical coordinate frame.

$$\dot{r}_d = 5000 * \frac{\pi}{2T} * \cos\left(\frac{\pi t}{2T}\right)$$

$$\dot{z}_d = 5000 * \frac{\pi}{2T} * \cos\left(\frac{\pi t}{2T}\right)$$

$$\dot{theta}_d = 0$$

$$\dot{x}_d = 5000 * \frac{\pi}{2T} * \cos\left(\frac{\pi t}{2T}\right) * \cos(theta_d)$$

$$\dot{y}_d = 5000 * \frac{\pi}{2T} * \cos\left(\frac{\pi t}{2T}\right) * \sin(theta_d)$$

Following developing the above-described desired functions for the trajectory of the active spacecraft during docking, the initial conditions are computed based on when time is 0. The initial parameters for the desired position and velocity are tabulated below.

Table 1: Desired Trajectory Parameters at T = 0.

Parameter	Initial Values
r_d	5000 m
Z_d	5000 m
$theta_d$	45 deg
x_d	3537 m
y_d	3534 m
$\dot{r}_d$	-2.1806
$\dot{z}_d$	-2.1806
$\dot{theta}_d$	0
$\dot{x}_d$	-1.5425
$\dot{y}_d$	-1.5413

As seen, the radial distance at the initial conditions is 5000 m. This indicates the docking sequence start location for the active spacecraft. Following this, the spacecraft has 60 minutes to approach and dock to the modular gateway orbiter. Additionally, the cartesian parameters were also computed using the relevant rotation matrices. This was done because the equations of motion for the docking analysis were derived in cartesian frame as well. The initial condition presented above are then utilized in the equations of motion developed for the docking procedure to compute the control input forces. These forces are estimated to have an order of 10^{-3} for the docking system to be stable.

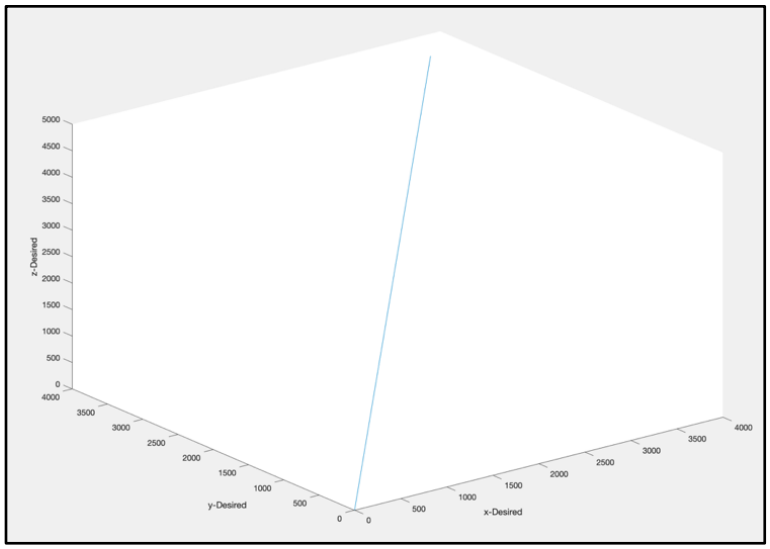

Figure 12: Desired Trajectory, Position Function.

The figure above illustrates the desired position trajectory for the active spacecraft during docking. The initial position parameters are tabulated in the above table for reference. The position functions in the Cartesian frame coordinates (X, Y, and Z) and the force feedback control loop function keeps the spacecraft following the trajectory as prescribed above.

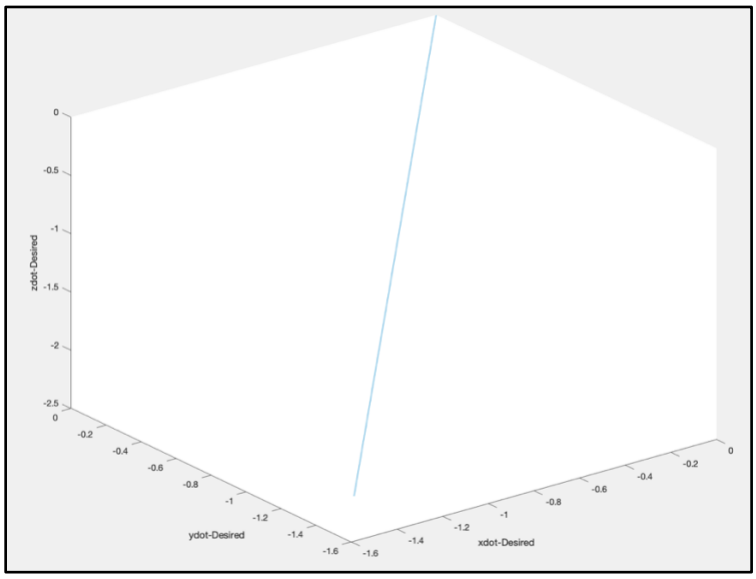

Figure 13: Desired Trajectory, Velocity Function.

The Figure above illustrates the velocity function of the desired trajectory. The initial parameters for the velocity in X, Y and Z directions are tabulated in the above table. The velocity

is a function of time similar to the position function and enable the force control system to maintain the spacecraft's velocity close to the desired values during the docking procedure.

4.2 Delta V and Launch Window

This section of the report will present the calculations completed that are required before simulating the Hohmann Orbit transfer. These computations include the azimuth angle for the spacecraft launch. The computation for this angle utilizes the launch location and orbit inclinations as seen in the formula presented in previous section.

$$Azimuth\ Angle\ (\beta) = 40\ Deg.$$

Following this, the orbital velocity for the modular orbiter and orbital time period are also computed using the equations provided in the previous section. These parameters are computed to calculate the launch timing for the spacecraft. This launch window will determine the launch location for the modular gateway orbiter at the time when the active spacecraft is at the perigee of the elliptical orbit. The time period for the circular orbit and the elliptical orbit are also utilized to compute the launch timing and the relevant location of the orbiter during this time.

$$Orbital\ Velocity = 3454\frac{m}{s}$$

$$Circular\ Orbit\ Time\ Period\ (T_c) = 109$$

$$Elliptical\ Orbit\ Time\ Period\ (T_e) = 104$$

Using the above computed parameters, the launch timing for the spacecraft and the location of the orbiter are computed and presented in the next section. The next section will also illustrate and analyze the Hohmann orbit transfer simulation whilst using these parameters as initial conditions. Additionally, the delta Velocity for the orbit transfer maneuvers is also presented in the next section. These velocities include the change in velocity required for the active spacecraft to change its orbit from an elliptical to a circular orbit. The trajectory for the Hohmann transfer and docking is designed in such a way that, the active spacecraft approaches the orbiter 8 Km behind of the orbiter as it approaches the apogee location for the circular orbit.

5. Numerical Simulation Results and Discussion

This section of the report presents and discusses the numerical simulation results whilst applying the above discussed system model, equations of motion and various assumptions made to simplify the problem statement. The simulation results include Hohmann transfer and docking for the active spacecraft. The simulation will illustrate various stages of the mission by graphing the spacecraft's position and velocity vectors from t = 0 till the end of docking. Additionally, this section will also present the external disturbance numerical model for gravitational perturbation effects on the modular gateway orbiter.

5.1 Phase I – Perturbation Simulation.

This section of the report presents and discusses the numerical simulation for the gravitational perturbation model for the modular gateway orbiter. The simulation was simplified by considering the orbital perturbation in plane rather than inclined at the orbital angle. This was justified in order to study the effect of J2 on the orbit disturbances for the orbiter.

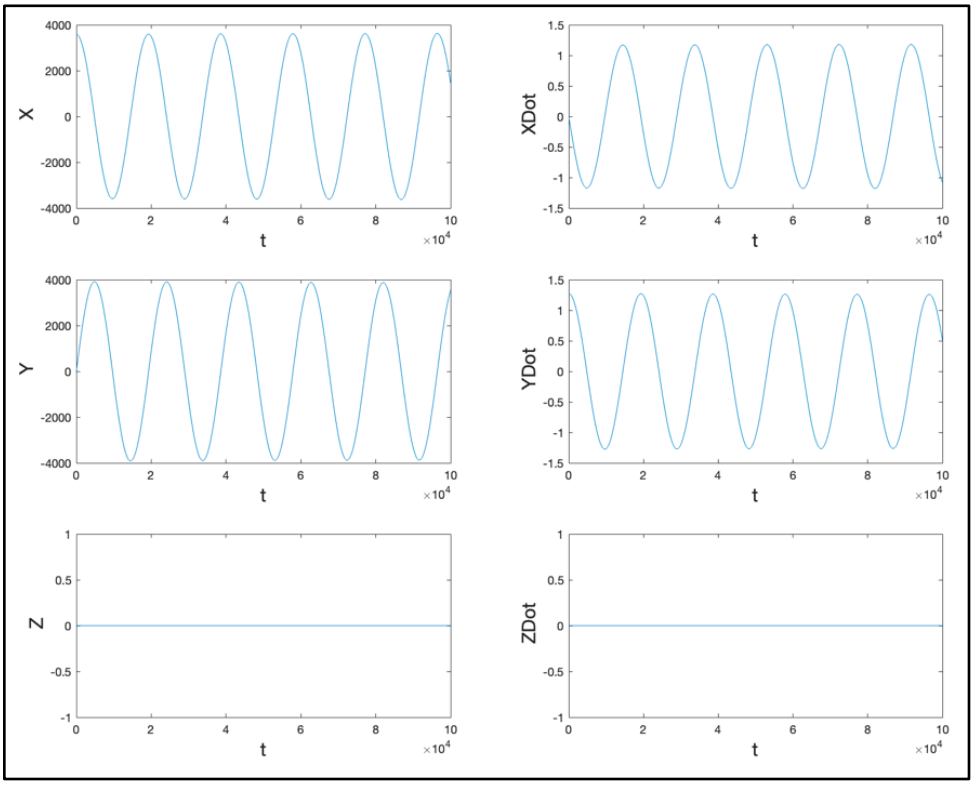

Figure 14: J2 Perturbation Position and Velocity of the Orbiter.

The figure above illustrates the position and velocity of the orbiter in Cartesian parameters. The position graphs indicate the disturbance in X and Y direction with a sinusoidal response. The Z direction is constant a 0 since the analysis considers only in-plane motion of the orbiter. The velocity vectors in X and Y direction also follow a sinusoidal function.

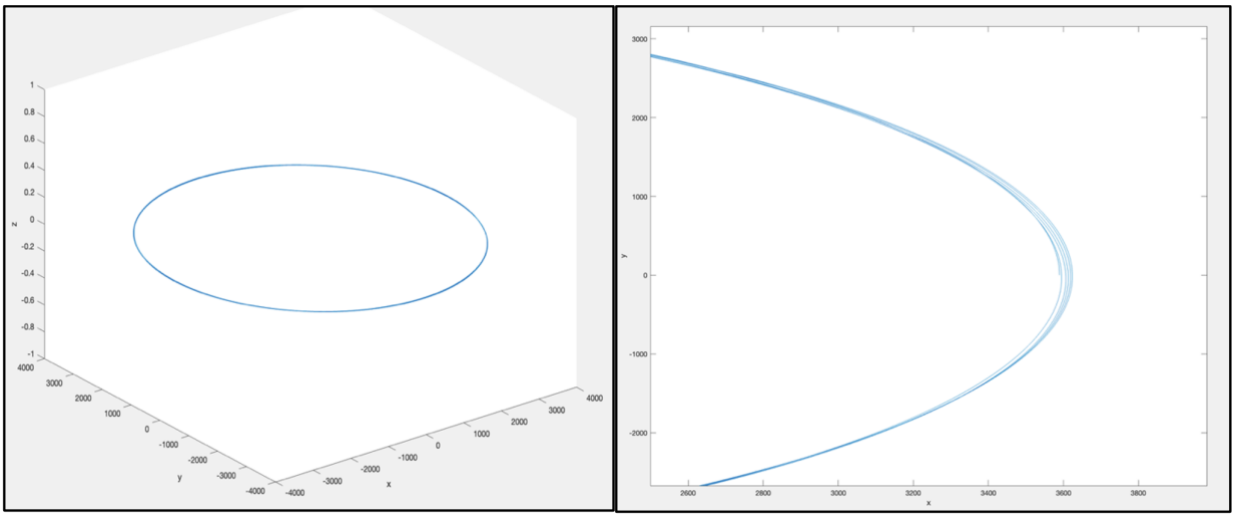

Figure 15: Perturbation Model Render in 3D and Zoomed 3D.

The figure above illustrates the in-plane orbital motion for the modular gateway orbiter. The graph on the left plots the orbital position and the effect of gravitational perturbation on the orbiter in 3D. The graph on the right is a zoomed in, 3D render of the same perturbation model. As seen, the orbital variations are clearly visible on the right plot. These variations are visible when any satellite stays in the orbit for longer durations. These variations in orbit arise due to the irregular shape of the planet the satellite revolves around. The gravitational perturbation effects are approximated by J2 for Mars.

5.2 Phase II – Delta V Simulations.

This section of the report will present and review the Phase II section for the mission. This phase concerns with Hohmann transfer and the necessary delta velocity required for this mission. Additionally, the simulation also computes the launch window for the transfer and the location for the orbiter during the transfer. The location is calculated such that the spacecraft arrives at the apogee of the circular orbit 8 Km behind the orbiter.

The time to transfer the active spacecraft from an elliptical orbit to a circular orbit is calculated to be approximately 52.1 minutes. A T = 0, the active spacecraft is located at the perigee

of the elliptical orbit. At the same time, the orbiter launch location was calculated to be at 7.6 degrees West from the active spacecraft.

$$Orbiter\ Location\ during\ launch = 7.6\ deg$$

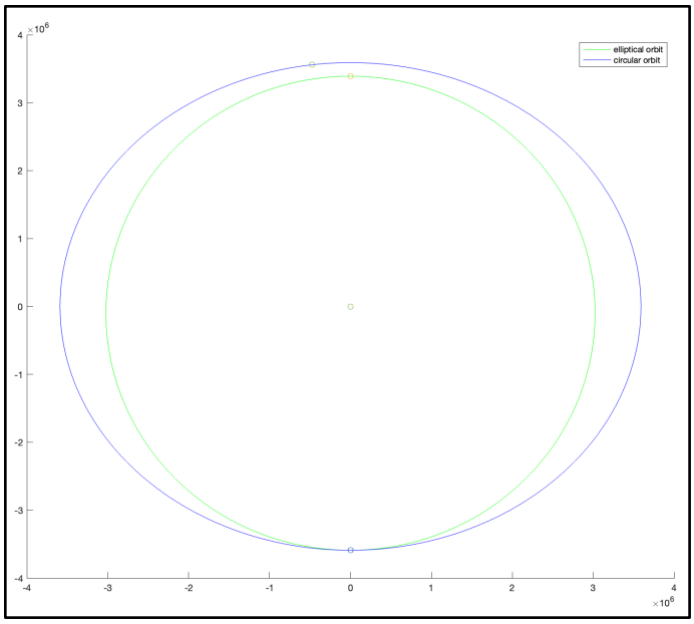

Figure 16: Hohmann Transfer Simulation.

The above figure illustrates the two orbits and the location of the two satellites at T = 0. The elliptical orbit for the active spacecraft is illustrated in green and the circular orbit for the orbiter is represented in purple. The launch of the spacecraft from the Martian surface is such that the active spacecraft is at the perigee of the elliptical orbit and the orbiter is at 7.6 degrees west from the spacecraft before the Hohmann transfer procedure commences.

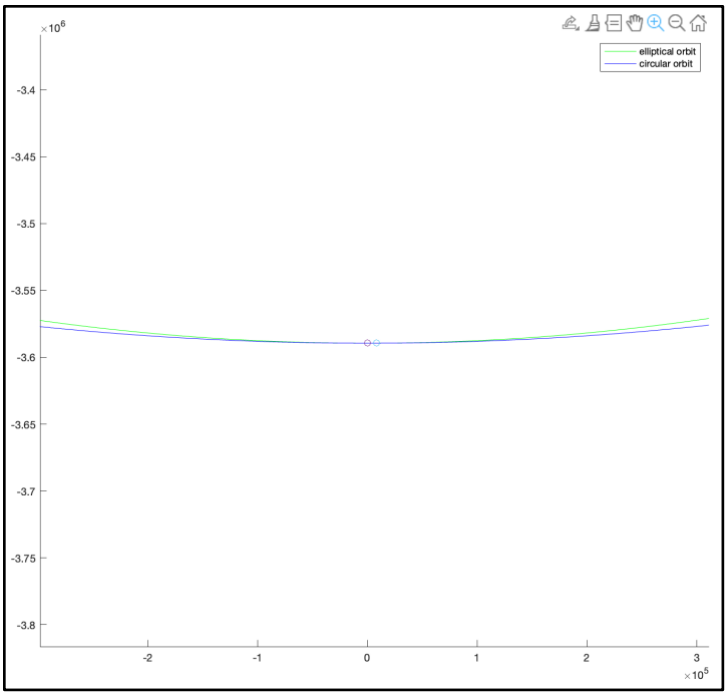

Figure 17: Hohmann Transfer at Final Time, before Docking.

The above figure illustrates the final position of the two spacecrafts as the active spacecraft approaches the modular gateway orbiter at the end of the Hohmann transfer. The Hohmann transfer for the active spacecraft takes approximately 52 minutes from the initial position. At the end time, the modular gateway orbiter is approximately 8 Km ahead of the active spacecraft. At this point, the active spacecraft commences the first stage of docking when it approaches the orbiter and prepares for docking.

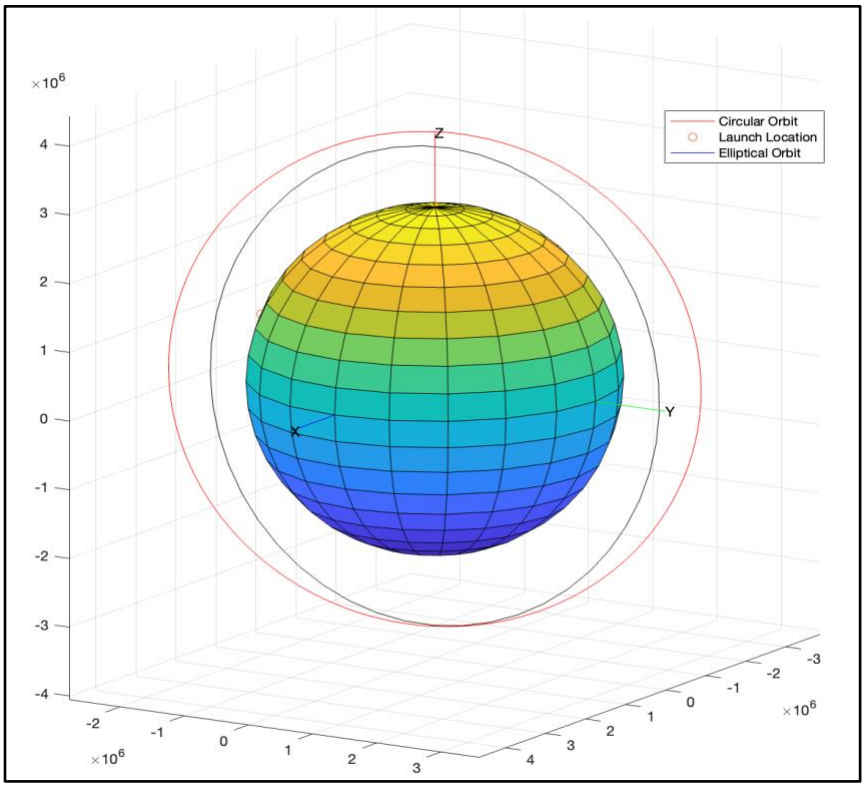

Figure 18: 3D render of Hohmann Transfer Simulation with Orbits.

The above figure illustrates the elliptical orbit and the circular orbit. The figure also captures the location of the launch site on Mars at 23-degree latitude and longitude 305 degrees. The figure shows the inclination of the two orbits and the apogee point where the active spacecraft rendezvous with the modular gateway orbiter.

The Delta velocity computation for the active spacecraft to transfer into a circular spacecraft is approximately 2.68 Km/s. This is the change in velocity needed for the active spacecraft to perform the Hohmann transfer.

5.3 Phase III – Docking Simulation.

This section of the report presents and analyzes the phase III simulation, also known as docking simulation. The docking of the active spacecraft to the modular gateway orbiter is a complex procedure that requires various force feedback control inputs. These control input enables the active spacecraft to stay on the prescribed desired trajectory along with the desired velocity for the approach.

This section will compare the actual and the desired reference trajectories for the active spacecraft along with the change in control input with respect to time. At T = 0, the active spacecraft is radial 5000 m from the gateway orbiter and the time taken for the entire docking procedure is approximately 60 min.

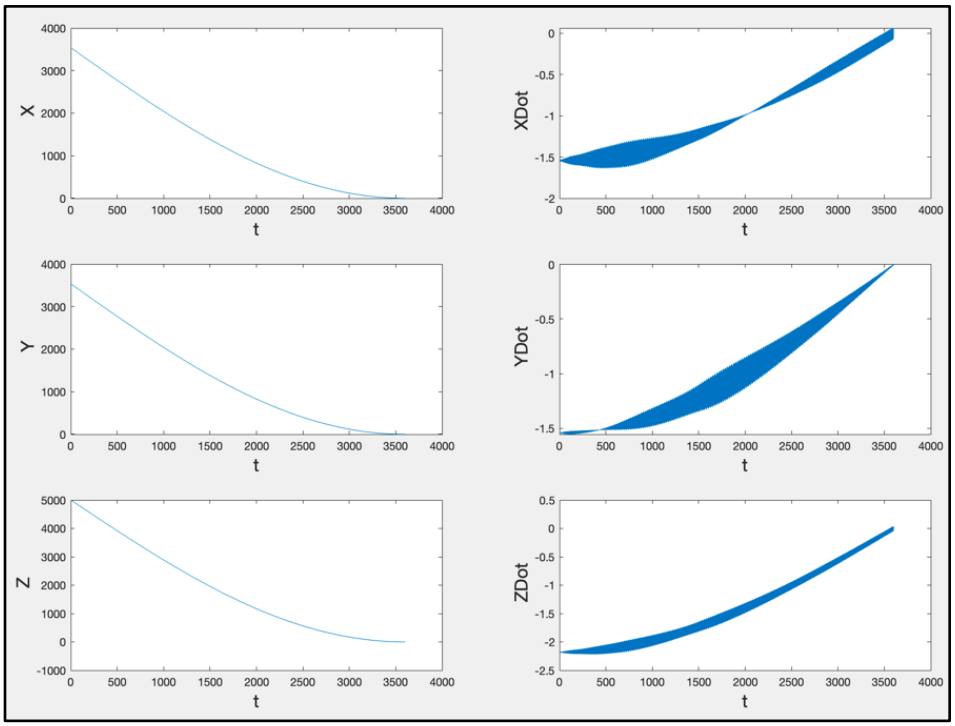

Figure 19: Actual Trajectory Position and Velocity Functions.

The above figure illustrates the actual position and velocity for the active spacecraft with respect to the time it takes to dock. As seen, the position plots start at some value and approaches 0 as time goes on. This is what was expected since as the active spacecraft approaches the orbiter, the radial distance between the two decreases. This postulate was also true for the actual velocities as seen on the plot at the right of the figure. As time goes on, the actual velocity of the orbiter in X, Y, and Z approaches to zero.

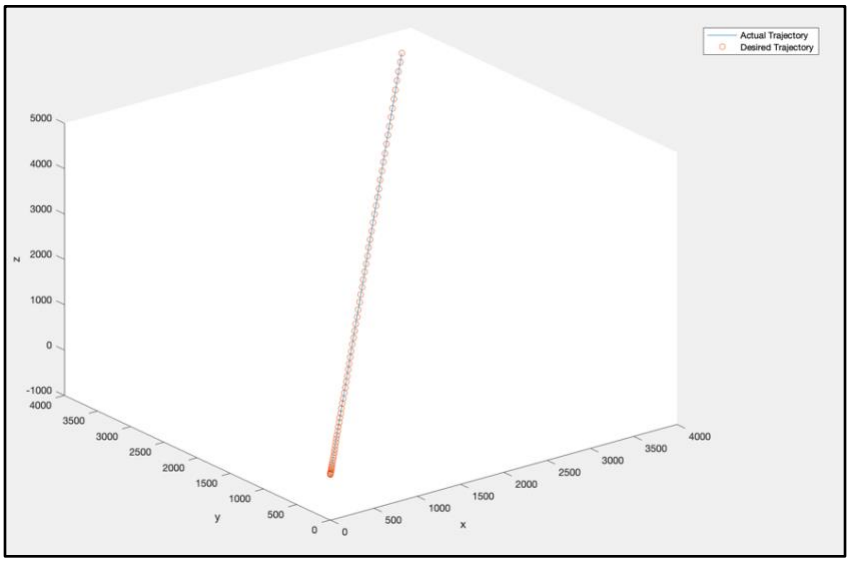

Figure 20: Actual Trajectory and Desired Trajectory.

The above figure compares the actual trajectory and the desired trajectory for the active spacecraft. The blue solid line represents the actual trajectory, and the red dotted line represents the desired trajectory. As evident, both the trajectories start with a radial difference between the start and end point as 5000 m. Additionally, the trends for the two trajectories imitate each other which means that the actual trajectory follows the desired prescribed path. This is an important step in ensuring that the spacecraft does not have a miss alignment or any over correction that would lead to a fatal accident.

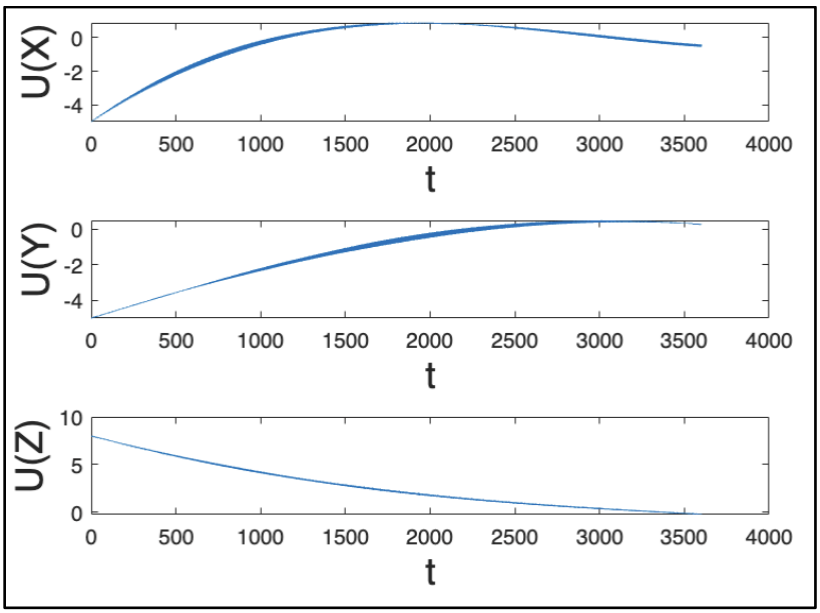

Figure 21: Control Force Inputs WRT Time.

The above figure illustrates the control force inputs that ensures that the active spacecraft follows the desired path prescribed in the above sections. As seen, the control force inputs start at a random value and as the time goes on, they approach zero. This behavior was expected since, it makes sense for the control inputs to have decreasing force inputs as the active spacecraft approaches the modular gateway orbiter for docking. As the active spacecraft gets closer to the orbiter, the force input, and the position and the velocity of the spacecraft also approaches zero. This behavior signifies a successful docking procedure and the fact that the equations of motion developed for the docking mission delivers its task.

6. Future Recommendations

This section will explore further missions based on the math and analysis developed in this report for this mission. One of the future developments for this mission could be to carry out interplanetary missions. These missions could collect samples from the Martian surface, dock to the modular gateway orbiter and finally transported back to earth for further research and analysis.

Similarly, this mission can also be extended beyond the Martian surface. A similar concept with modified calculations could be applied to explore exoplanets, moons and large asteroids. This will enable an increase in exploration. Furthermore, the model can be redeveloped to have an automatic launch and docking sequence thus eliminating the need for a human astronaut.

7. Conclusions

The objective of this mission is to study the motions of the equation and stability of the active spacecraft launching from the Martian surface, orbit insertion into the gateway orbit, and finally docking to a large modular gateway orbiter. This mission focuses on computing the orbital trajectory of the two satellites and the attitude of the two satellites for the docking procedure. The mathematical formulation put forth in this report will be used as a guiding tool throughout this project.

Additionally, the mathematical formulation postulates the approximations, coordinate frames, and external disturbances necessary for developing the equations of motion for the two satellites. The report contains the equations of motion for the two satellites during various stages of the mission. The equations of motion and the Delta-V equations enable us to analyze the two satellites during launch, orbit insertion and finally during the docking procedure. Along with this the external disturbances enforced on the two satellites are also derived and analyzed in this report. Specifically, the gravitational perturbation on the orbiter and the solar radiation pressure on the two satellites are modelled respectively.

The desired trajectory for the mission is also demonstrated along with illustrations of the mission profile. The desired trajectory constitutes the mathematical formulation of a circular orbit for MGO around the planet Mars. Additionally, it formulates the launch and orbit insertion of the active spacecraft into the gateway orbit. This formulation also examines the redundant system designed in case the spacecraft fails the orbit insertion phase of the mission. Lastly, the desired trajectory introduces the coordinate frame and plan for the docking process of the two satellites.

In conclusion, this report presents the mission assumptions, equations of motion, desired trajectory, mathematical analysis and the numerical simulations conducted to analyze the position, velocities, and acceleration of the two satellites. As seen in the numerical simulation section, the mission is successfully simulated in MATLAB as the active spacecraft safely performs a Hohmann transfer and docks to the modular orbiter.

8. References

[1] M. Boucher, “Lockheed Martin Planning a Mars Base Camp,” SpaceRef, Sep. 29, 2017. https://spaceref.com/science-and-exploration/lockheed-martin-planning-a-mars-base-camp/

[2] D. Etherington, “Lockheed Martin’s Mars Base Camp would stage Mars surface missions,” TechCrunch, Sep. 29, 2017. https://techcrunch.com/2017/09/28/lockheed-martins-mars-base-camp-would-stage-mars-surface-missions/

[3] ‘International Space Station Facts and Figures - NASA’. Accessed: Nov. 08, 2023. [Online]. Available: https://www.nasa.gov/international-space-station/space-station-facts-and-figures/

[4] ‘Mathematics of Satellite Motion’. Accessed: Nov. 08, 2023. [Online]. Available: https://www.physicsclassroom.com/class/circles/Lesson-4/Mathematics-of-Satellite-Motion

[5] ‘Launch Azimuth - OrbiterWiki’. Accessed: Nov. 08, 2023. [Online]. Available: https://www.orbiterwiki.org/wiki/Launch_Azimuth#:~:text=The%20launch%20azimuth%20is%20the,the%20celestial%20body%27s%20reference%20plane.

[6] Dr. J. Yokota, Orbital Mechanics - Lecture Notes 5.

[7] Dr. K. D. Kumar, Fundamentals of Dynamics and Control of Space Systems. 2012.

[8] ‘Mars Fact Sheet’. Accessed: Nov. 08, 2023. [Online]. Available: https://nssdc.gsfc.nasa.gov/planetary/factsheet/marsfact.html

[9] ‘Mars Exploration: Multimedia’. Accessed: Nov. 08, 2023. [Online]. Available: https://mars.nasa.gov/gallery/global/PIA01249.html#:~:text=The%20Martian%20north%20pole%20is,longitude%20is%20near%20305%20degrees.

[10] ‘Elliptic Orbits’. Accessed: Nov. 08, 2023. [Online]. Available: https://galileo.phys.virginia.edu/classes/152.mf1i.spring02/EllipticOrbits.htm

[11] “Math for Orbits,” galileoandeinstein.phys.virginia.edu. https://galileoandeinstein.phys.virginia.edu/7010/CM_14_Math_for_Orbits.html#:~:text=Eccentricity&text=e%3DF1C%2Fa.

9. Appendix

9.1 Phase I – Perturbation.

```
clc
clear
close

global mu Rm J2 Z X_lo i omega_l
```

Parameters.

```
mu = 6.6743*(10^-11)*7.34767309*(10^22)/(1000^3);    % km^3/s^2
Rm = 3389.5;              % Km
J2 = 1960.45*10^-6;
e = 0;                   % Orbital eccentricty
a = Rm/(1-e);            % Semi-major axis (km)
i = deg2rad(45);
omega_l = deg2rad(0);
theta_l = deg2rad(0);
Z = Rm*sin(i)*sin(omega_l + theta_l);

thetadot = sqrt(mu/a^3);
thetadotm = sqrt(mu/a^3);
```

Leader Satellite - Initial Conditions.

```
X_lo = 200 + Rm;         % Km
Y_lo = 0;
Z_lo = 0;
theta_lo = theta_l;

Xlo_Dot = 0;         % Km
Ylo_Dot = (200 + Rm)*thetadot;
Zlo_Dot = 0;

orbiter_o = [X_lo; Xlo_Dot; Y_lo; Ylo_Dot; Z_lo; Zlo_Dot];
```

Declaring Time and initial Matrix.

```
t_start = 0;
t_end = 100000;
t_step = 1;
```

```
t_span = t_start:t_step:t_end;
```

Numerical Simulation

```
options = odeset('abstol',1e-9,'reltol',1e-9);
[t,x] = ode45(@Eqn_Motion,t_span,orbiter_o);
```

Plotting the Results - Follower Satellite.

```
figure(1)
hold on
zoom on

% X and XDot for Follower
subplot(3,2,1),plot(t,x(:,1));
xlabel('\fontsize{16 }t');
ylabel('\fontsize{16} X');

subplot(3,2,2),plot(t,x(:,2));
xlabel('\fontsize{16 }t');
ylabel('\fontsize{16} XDot');

% Y and YDot for Follower
subplot(3,2,3),plot(t,x(:,3));
xlabel('\fontsize{16 }t');
ylabel('\fontsize{16} Y');

subplot(3,2,4),plot(t,x(:,4));
xlabel('\fontsize{16 }t');
ylabel('\fontsize{16} YDot');

% Z and ZDot for Follower
subplot(3,2,5),plot(t,x(:,5));
xlabel('\fontsize{16 }t');
ylabel('\fontsize{16} Z');

subplot(3,2,6),plot(t,x(:,6));
xlabel('\fontsize{16 }t');
ylabel('\fontsize{16} ZDot');

hold off
zoom off

figure('Name','3D - Cartesion Coordinate Perturbation')
plot3(x(:,1),x(:,3),x(:,5))
```

```
xlabel('x')
ylabel('y')
zlabel('z')

figure('Name','2D - Cartesion Coordinate Perturbation')
plot(x(:,1),x(:,3))
xlabel('x')
ylabel('y')
```

Functions.

```
function  [Orbiter] = Eqn_Motion(t,x)

global mu Rm J2 Z X_lo i omega_l

r_l = sqrt((x(1))^2 + (x(3))^2 + (x(5))^2);
thetadot = sqrt(mu/((r_l)^3));

theta = thetadot*t;
xl = X_lo*cos(theta);
yl = X_lo*sin(theta);
zl = 0;

Z = Rm*sin(i)*sin(omega_l + theta);

r_l = sqrt((xl)^2 + (yl)^2 + (zl)^2);

XDot_1 = x(2);
XDot_2 = -((mu/(r_l^3)) + ((3*mu*J2*(Rm^2))/(2*(r_l^5)))*(1 -
((5*(Z^2))/(r_l^2))))*(x(1));

YDot_1 = x(4);
YDot_2 = -((mu/(r_l^3)) + ((3*mu*J2*(Rm^2))/(2*(r_l^5)))*(1 -
((5*(Z^2))/(r_l^2))))*(x(3));

ZDot_1 = x(6);
ZDot_2 = -((mu/(r_l^3)) + ((3*mu*J2*(Rm^2))/(2*(r_l^5)))*(1 -
((5*(Z^2))/(r_l^2))))*(x(5));

Orbiter = [XDot_1; XDot_2; YDot_1; YDot_2; ZDot_1; ZDot_2];

end
```

9.2 Phase II – Delta V and Hohmann Transfer.

Parameters

```
i = deg2rad(45);
latitude = deg2rad(23);
lon = deg2rad(305);

G = 6.6743*10^-11;      % N*m^2/Kg^2.
M_mars = 0.64169*10^24; % Kg.
M_spacecraft = 45000;   % Kg.
M_Orbiter = 420000;     % Kg.

R_mars = 3389.5*1000;   % m.
R_Orbit = 200*1000;     % m.
R = R_mars + R_Orbit;

e = 0.5;                % Orbital eccentricty
a = (R + R_mars)/2;             % Semi-major axis (m)
b = sqrt(a^2 - (e*a)^2);    % Semi-minor axis (m)
```

Azimuth, Launch Vel. and Timing.

```
beta = rad2deg(acos((cos(i))/(cos(latitude))));                    % Launch
Azimuth.

Vm_orbit = sqrt(((G)*(M_mars + M_Orbiter))/(R_mars + R_Orbit));     % Orbiter
Velocity.

T_Circle_Orbit = (2*pi*(((R_Orbit+R_mars)^(3/2))/(sqrt(G*(M_mars +
M_Orbiter)))))/60;   % Orbiter Time Period.
T_Ellipse_Orbit = (2*pi*(((a)^(3/2))/(sqrt(G*(M_mars)))))/60;
```

Plotting the Orbits and Launch Timings.

```
% Elliptical Orbit.
y_e = 0:100:a;
x_e = sqrt((1 - (y_e/a).^2)*(b.^2));

hold on
plot(x_e,y_e - a + R_mars,'-G')
plot(-x_e,y_e - a + R_mars,'-G')
plot(-x_e,-y_e - a + R_mars,'-G')
plot(x_e,-y_e - a + R_mars,'-G')
```

```
plot(0,0,'o')                    % Origin (Centre of Mars)

% Circular Orbit.
y_c = 0:100:R;
x_c = sqrt(R^2 - y_c.^2);

plot(x_c,y_c,'-B')
plot(-x_c,y_c,'-B')
plot(x_c,-y_c,'-B')
plot(-x_c,-y_c,'-B')

theta = deg2rad(7.6);                        % Assume location of orbiter at t =
0;

arc = theta*R;

Distance_Travelled = pi*R - arc;

Time_Taken = (Distance_Travelled/(pi*R))*(T_Circle_Orbit/2)    % From 15 degree
location.

time_diff = T_Ellipse_Orbit/2 - Time_Taken;

Extra_Dis = ((pi*R)*(time_diff))/((T_Circle_Orbit/2));
theta_F = Extra_Dis/R;

% Spacecraft Position.
plot(0,R_mars,'o')                           % Initial Position.
plot(0,-R,'o')                               % Final Position.

% Orbiter Position.
plot(-R*sin(theta),R*cos(theta),'o')         % Initial Position.
plot(R*sin(theta_F),-R*cos(theta_F),'o')     % Final Position.

legend('elliptical orbit','','','','','circular orbit')
hold off
```

Plotting launch location and orbit

```
figure (2)

[x_s,y_s,z_s] = sphere;
x_s = x_s*(2500000);
y_s = y_s*(2500000);
z_s = z_s*(2500000);
```

```
 surf (x_s, y_s, z_s)
 axis equal

 hold on

 plotCircle3D([0,0,0],[1/cos(deg2rad(45)),1/sin(deg2rad(45)),1],R)


 x_L = (2500000) * cos(latitude) * cos(lon);
 y_L = (2500000) * cos(latitude) * sin(lon);
 z_L = (2500000) * sin(latitude);

 plot3(x_L,y_L,z_L,'o')

 [ellipseCoords, h] = ellipse3D(a,b,0,0,-
a+R_mars,50,deg2rad(0),deg2rad(63),deg2rad(45),1,'k' );



 legend('','Circular Orbit','Launch Location','Elliptical Orbit')
```

Delta Velocity.

```
 mu_1 = G*(M_mars + M_spacecraft);
 R_a = 100 * 10^3; % m.

 % To enter into an elliptical orbit.
 Delta_Va = (sqrt(2*mu_1))*(sqrt((R_Orbit)/((R_a)*(R_Orbit + R_a)))) -
sqrt((mu_1)/(R_a))

 % To entre into a circular orbit.
 Delta_Vb = sqrt((mu_1)/(R_Orbit)) -
(sqrt(2*mu_1))*(sqrt((R_a)/((R_Orbit)*(R_Orbit + R_a))))

 Total_DV = Delta_Va + Delta_Vb;

 function plotCircle3D(center,normal,radius)

 theta=0:0.01:2*pi;
 v=null(normal);
 points=repmat(center',1,size(theta,2))+radius*(v(:,1)*cos(theta)+v(:,2)*sin(the
ta));
 plot3(points(1,:),points(2,:),points(3,:),'r-');
```

```
 end

 function [ellipseCoords, h] = ellipse3D(
rx,ry,x0,y0,z0,Nb,pitch,roll,yaw,graph,C )
 %
 %   1. Plot an ellipse on the XY plane with semimajor axis of radius rx
 %      along X axis and semimajor axis of radius ry along Y axis
 %
 %   2. Rotate ellipse CCW about X axis by 'pitch' radians. 0 leaves ellipse
 %      on XY plane. pi/2 rotates CCW about positive X-axis and puts ellipse
 %      on XZ plane
 %
 %   3. Rotate ellipse CCW about Y axis by 'roll' radians. 0 leaves
 %      ellipse on XY plane. pi/2 rotates CCW about positive Y-axis and puts
 %      ellipse on YZ plane
 %
 %   4. Rotate ellipse CCW about Z axis by 'yaw'
 %      radians. 0 leaves ellipse on XY plane. pi/2 rotates CCW about
 %      positive Z-axis and (leaving ellipse on XY plane)
 %
 %   5. Apply offsets of x0, y0, and z0 to ellipse
 %
 %   6. Optionally generate 3D plot of ellipse
 %
 %% Inputs
 %
 % rx - length of radius of semimajor axis of ellipse along x-axis
 % ry - length of radius of semimajor axis of ellipse along y-axis
 % x0 - origin of ellipse with respect to x-axis
 % y0 - origin of ellipse with respect to y-axis
 % z0 - origin of ellipse with respect to z-axis
 % Nb - number of points used to define ellipse
 % pitch - angle of pitch in radians of ellipse wRt +x-axis
 % roll - angle of roll in radians of ellipse wRt +y-axis
 % yaw - angle of yaw in radians of ellipse wRt +z-axis
 % graph - flag that tells function whether or not to graph.
 %   0 - do not graph
 %   1 - graph and let this function define graphing options
 %   >1 - graph, but allow other functions to define graphing options
 % C - color of ellipse to be plotted. Acceptable input either in character
 %   form ('r') or RGB form ([0 .5 1])

 %% Parse Inputs
 % Check the number of input arguments
 if nargin<=1,error('Not enough arguments');end;
 if nargin<1,rx=[];end;
```

```
if nargin<2,ry=[];end;
if nargin<3,x0=[];end;
if nargin<5,x0=[];y0=[];z0=[];end;
if nargin<6,Nb=[];end
if nargin<7,pitch=[];end
if nargin<8,roll=[];end
if nargin<9,yaw=[];end
if nargin<10,graph=[];end
if nargin<11,C=[];end
% set up the default values
if isempty(rx),rx=1;end;
if isempty(ry),ry=1;end;
if isempty(x0),x0=0;end;
if isempty(y0),y0=0;end;
if isempty(z0),z0=0;end;
if isempty(Nb),Nb=300;end;
if isempty(pitch),pitch=0;end;
if isempty(roll),roll=0;end;
if isempty(yaw),yaw=0;end;
if isempty(graph),graph=0;end;
if isempty(C),C='b';end;
h=0;
if ischar(C)
    C=C(:);
else
    C=[0,0,1]; % Make blue
end
% Ensure that all input values are scalars
if length(rx)>1,error('too many values for rx');end;
if length(ry)>1,error('too many values for ry');end;
if length(x0)>1,error('too many values for x0');end;
if length(y0)>1,error('too many values for y0');end;
if length(z0)>1,error('too many values for z0');end;
if length(Nb)>1,error('too many values for Nb');end;
if length(pitch)>1,error('too many values for pitch');end;
if length(roll)>1,error('too many values for roll');end;
if length(yaw)>1,error('too many values for yaw');end;
if length(graph)>1,error('too many values for graph');end;
%% Mathematical Formulation
% Declare angle vector theta (t in parametric equation of ellipse)
the=linspace(0,2*pi,Nb);
% Create X and Y vectors using parametric equation of ellipse
X=rx*cos(the);
Y=ry*sin(the);
% Declare the Z plane as all zeros before transformation
Z=zeros(1, length(X));
```

```
% Define rotation matrix about X axis. 0 leaves ellipse on XY plane. pi/2
% rotates CCW about X-axis and puts ellipse on XZ plane
Rpitch = [1          0           0          ;...
          0          cos(pitch)  -sin(pitch);...
          0          sin(pitch)  cos(pitch)];
% Define rotation matrix about Y axis. 0 leaves ellipse on XY plane. pi/2
% rotates CCW about Y-axis and puts ellipse on YZ plane
Rroll = [cos(roll)   0   sin(roll)  ;...
         0           1   0          ;...
         -sin(roll)  0   cos(roll)] ;
% Define rotation matrix about about Z axis. 0 leaves ellipse on XY plane.
% pi/2 rotates CCW about Z-axis and (leaving ellipse on XY plane)
Ryaw = [cos(yaw)   -sin(yaw)  0;...
        sin(yaw)   cos(yaw)   0;...
        0           0         1];
% Apply transformation
for i=1:length(X)

    xyzMat = [X(i);Y(i);Z(i)]; % create temp values
    temp = Rpitch*xyzMat; % apply pitch
    temp = Rroll*temp; % apply roll
    temp = Ryaw*temp; % apply yaw
    X(i) = temp(1); % store results
    Y(i) = temp(2);
    Z(i) = temp(3);

end
% Apply offsets
X = X + x0;
Y = Y + y0;
Z = Z + z0;
%% Graphing
if graph > 0

    if graph == 1
        axis equal;
        axis auto;
        axis vis3d;
        grid on;
        view(3);

        axisLength = max([rx ry]) + max([abs(x0) abs(y0) abs(z0)]);
        h3=line([0,axisLength],[0,0],[0,0]);
        set(h3,'Color','b');
        text(axisLength,0,0,'X');
```

```
        h2=line([0,0],[0,axisLength],[0,0]);
        set(h2,'Color','g');
        text(0,axisLength,0,'Y');

        h1=line([0,0],[0,0],[0,axisLength]);
        set(h1,'Color','r');
        text(0,0,axisLength,'Z');

        h = line(X,Y,Z); % plot ellipse in 3D
        set(h,'color',C); % Set color of ellipse
    else

        h = line(X,Y,Z); % plot ellipse in 3D
        set(h,'color',C); % Set color of ellipse
    end

end
ellipseCoords = [X;Y;Z]; % define return values
end
```

9.3 Phase III – Docking.

```
global mu Rm r_o kp kd TDock_end pi
```

Parameters

```
mu = 0.042828*(10^6)*(1000^3);    % m^3/s^2
Rm = 3389.5*1000;
r_o = 200*1000 + Rm;
thetadot = sqrt(mu/r_o^3);

kp = 1;
kd = 0.001;
```

Desired Trajectory

```
pi = 3.14;

% Too calculate deratives.
syms tDock_span TDock_end

% Prescribed in clyndrical coordinates.
z_d = 5000*(1-sin(pi*tDock_span/(2*TDock_end)));
r_d = 5000*(1-sin(pi*tDock_span/(2*TDock_end)));
theta_d = 45*(pi/180);

% Calculating cartesian coordinates.
x_d = r_d*cos(theta_d);
y_d = r_d*sin(theta_d);

% Calculating diff components.
xd_dot = diff(x_d);
rd_dot = diff(r_d);
thetad_dot = 0;
yd_dot = diff(y_d);
zd_dot = diff(z_d);

TDock_end = 60*60;
tDock_span = 0:50:TDock_end;

% Desired Position in X,Y,Z.
z_d = 5000*(1-sin(pi*tDock_span/(2*TDock_end)));
r_d = 5000*(1-sin(pi*tDock_span/(2*TDock_end)));
theta_d = 45*(pi/180);
x_d = r_d*cos(theta_d);
```

```
y_d = r_d*sin(theta_d);

figure ('Name','DesredTrajectory-Position')
title('Desired Trajectory')
plot3(x_d,y_d,z_d,'-')
xlabel('x-Desired')
ylabel('y-Desired')
zlabel('z-Desired')

% Desired Velocity in X,Y,Z.
xd_dot = subs(xd_dot,tDock_span);
xd_dot = double(subs(xd_dot,TDock_end));

yd_dot = subs(yd_dot,tDock_span);
yd_dot = double(subs(yd_dot,TDock_end));

zd_dot = subs(zd_dot,tDock_span);
zd_dot = double(subs(zd_dot,TDock_end));

figure ('Name','DesredTrajectory-velocity')
title('Desired Trajectory')
plot3(xd_dot,yd_dot,zd_dot,'-')
xlabel('xdot-Desired')
ylabel('ydot-Desired')
zlabel('zdot-Desired')

Relative_o = [x_d(1,1);xd_dot(1,1);y_d(1,1);yd_dot(1,1);z_d(1,1);zd_dot(1,1);-
5;-5;8];
```

ODE Solver

```
t_start = 0;
t_end = TDock_end;
t_step = 1;
t_span = t_start:t_step:t_end;

options = odeset('abstol',1e-9,'reltol',1e-9);
[t,x] = ode45(@Eqn_Motion,t_span,Relative_o);
```

Plot

```
figure ('Name','Actual Position  & Velocity')
zoom on

% X and XDot for Follower
```

```
subplot(3,2,1),plot(t,x(:,1));
xlabel('\fontsize{16 }t');
ylabel('\fontsize{16} X');

subplot(3,2,2),plot(t,x(:,2));
xlabel('\fontsize{16 }t');
ylabel('\fontsize{16} XDot');

% Y and YDot for Follower
subplot(3,2,3),plot(t,x(:,3));
xlabel('\fontsize{16 }t');
ylabel('\fontsize{16} Y');

subplot(3,2,4),plot(t,x(:,4));
xlabel('\fontsize{16 }t');
ylabel('\fontsize{16} YDot');

% Z and ZDot for Follower
subplot(3,2,5),plot(t,x(:,5));
xlabel('\fontsize{16 }t');
ylabel('\fontsize{16} Z');

subplot(3,2,6),plot(t,x(:,6));
xlabel('\fontsize{16 }t');
ylabel('\fontsize{16} ZDot');
```

Plot

```
figure ('Name','Actual Trajectory Vs. Desired Trajectory')
plot3(x(:,1),x(:,3),x(:,5),'-')
hold on
plot3(x_d,y_d,z_d,'o')
xlabel('x')
ylabel('y')
zlabel('z')
legend('Actual Trajectory','Desired Trajectory')
hold off
```

Plot

```
figure ('Name','Feedback Control')
zoom on
hold on
subplot(3,1,1),plot(t,x(:,7));
```

```
 xlabel('\fontsize{16 }t');
 ylabel('\fontsize{16} U(X)');

 subplot(3,1,2),plot(t,x(:,8));
 xlabel('\fontsize{16 }t');
 ylabel('\fontsize{16} U(Y)');

 % Y and YDot for Follower
 subplot(3,1,3),plot(t,x(:,9));
 xlabel('\fontsize{16 }t');
 ylabel('\fontsize{16} U(Z)');
 hold off
 zoom off

 function  [Relative] = Eqn_Motion(t,x)

 global mu r_o kp kd TDock_end pi

 z_d = 5000*(1-sin(pi*t/(2*TDock_end)));
 r_d = 5000*(1-sin(pi*t/(2*TDock_end)));
 theta_d = 45*(pi/180);
 x_d = r_d*cos(theta_d);
 y_d = r_d*sin(theta_d);

 zd_dot = ((-5000*pi)/(2*TDock_end))*cos((pi*t)/(2*TDock_end));
 xd_dot = ((-5000*pi*(cos(theta_d)))/(2*TDock_end))*cos((pi*t)/(2*TDock_end));
 yd_dot = ((-5000*pi*(sin(theta_d)))/(2*TDock_end))*cos((pi*t)/(2*TDock_end));

 R = sqrt((r_o + x(1))^2 + (x(3))^2 + (x(5))^2);
 thetadot = sqrt(mu/((R)^3));
 % u(x)  should be 10^-3 order. reduce kd.kp>kd o
 x_dot1 = x(2);
 u_x = (kp*(x(1) - x_d) + kd*(x(2) - xd_dot));
 x_dot2 = mu/(r_o^2) - ((mu*(r_o + x(1)))/(R^3)) + x(1)*(thetadot^2) +
2*x(4)*thetadot - u_x;

 y_dot1 = x(4);
 u_y = (kp*(x(3) - y_d) + kd*(x(4) - yd_dot));
 y_dot2 = -((mu*(x(3)))/(R^3)) + x(3)*(thetadot^2) - 2*x(2)*thetadot - u_y;

 z_dot1 = x(6);
 u_z = (kp*(x(5) - z_d) + kd*(x(6) - zd_dot));
 z_dot2 = -((mu*(x(5)))/(R^3)) - u_z;
 Relative = [x_dot1;x_dot2;y_dot1;y_dot2;z_dot1;z_dot2;u_x;u_y;u_z];
 end
```