Rendezvous Maneuver in LEO from Higher Orbit

Robert Duquette
University of Michigan
robduq@umich.edu

Patrick Poon
University of Michigan
patpoon@umich.edu

Joshua Strand
University of Michigan
sjosh@umich.edu

***Abstract*—This project aims to enhance satellite rendezvous capabilities, particularly for maneuvers from a higher to a lower orbit, to meet the increasing demands of complex satellite operations. We designed and implemented a dual-approach control system utilizing both Proportional-Integral-Derivative (PID) and Linear-Quadratic-Regulator (LQR) controllers to address the Drift Orbit B phase of rendezvous. The system's requirements were established to ensure precision and safety, including constraints on maximum overshoot, settling time, and rise time.**

In terms of methods, the project utilized a Simulink model that allows for a choice on control scheme (either PID or LQR), taking into account environmental disturbances such as drag and the J2 effect. The model was tested to validate the control system's performance, with the goal of improving orbital service operations for tasks that require rendezvous of satellites.

Analytical and experimental results confirmed the proficiency and distinct operational characteristics of the PID and LQR controllers within the proposed satellite rendezvous control system. The PID controller exhibited quicker responsiveness to trajectory alterations (during the impulsive burns), though it showed a tendency toward oscillation during steady state conditions. In contrast, the LQR controller demonstrated a smoother response profile, favoring a less reactive but more stable control input, as evidenced by less oscillatory behavior in its control efforts. The resulting balance of speed and efficiency between the two control strategies underlines the potential for use of this control system with specific mission requirements and highlights a significant advancement in the control of space vehicles.

I. Introduction

In today's booming space industry, satellite rendezvous capabilities are becoming increasingly necessary. As more and more commercial satellites are put into orbit, the demand for services such as on-orbit inspection, fuel delivery, and orbit boosting is rising. Programs like SpaceX's SmallSat Rideshare program have exponentially increased the accessibility of orbit to smaller entities [1]. However, while access to orbit is more readily available, the choice of final orbit destination and launch timing are limited, a compromise for the reduced launch cost. Services to boost a satellite's orbit to a desired final orbit or trajectory would increase accessibility to a broader range of orbits, including cislunar trajectories and interplanetary transfers. These capabilities necessarily require the servicing satellite to rendezvous with a satellite already in orbit.

A. Problem Introduction

In order to provide service to satellites in Earth orbit, it would be logical to reuse a vehicle as many times as possible. In the case of providing delta-v as a service, this means the vehicle would follow many different orbits in its lifetime. The problem of rendezvous from lower orbits has been solved regularly as demonstrated through regular resupply missions to the International Space Station. However, rendezvous with a spacecraft from a higher orbit to a lower orbit has not been demonstrated as thoroughly. This project aims to provide an investigation into rendezvous from a higher orbit to a target in a lower orbit.

II. Comprehensive Methodology

This section details our approach to the rendezvous problem. First we will outline the problem setup in detail, including the dynamics and kinematic models used

A. Problem Setup

For this problem, we consider a target satellite in low Earth orbit (LEO) at 500 km in altitude, with 45° of inclination. This is a reasonable orbit for a SpaceX Falcon 9 rideshare mission. The chaser spacecraft is in a circular orbit 30 km higher, at 530 km in altitude. We will consider orbital disturbances from atmospheric drag and the J2 effect in our analysis. Our approach involves:

- Six Degree-of-Freedom satellite model in Simulink.
- Use of the Clohessy-Wiltshire satellite dynamics linearization.
- Tracking control of the desired trajectory for rendezvous.
- Consideration of disturbances in the LEO environment such as J2 effect and atmospheric drag.

1) Linearized Translational Dynamics: The equations of motion for the 6 DOF system include rotational dynamics described by the Euler equations, and translational dynamics relative to the target spacecraft described by the Clohessy-Wiltshire (CW) equations. We separated the translational and attitude dynamics into separate state space models, as our models have no coupling effects between the two forms of motion. We first discuss the translational dynamics, then the attitude dynamics.

We define the state space as:

$$\vec{x}_{trans} = (x, y, z, \dot{x}, \dot{y}, \dot{z})$$
$$\vec{x}_{att} = (\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi})$$

Where x is the relative radial position of the chaser spacecraft, y is the relative along-track position, z is the relative cross-track position, ϕ, θ, ψ are the Euler angles to describe the satellite's attitude. The remaining elements of the state space are the derivatives of the previous variables.

The Clohessy-Wiltshire equations are as follows:

$$\begin{cases} \ddot{x} = 3n^2x + 2n\dot{y} + u_x \\ \ddot{y} = -2n\dot{x} + u_y \\ \ddot{z} = -n^2z + u_z \end{cases} \tag{1}$$

Where n is the target's orbital rate in radians per second. Described in our translational state space model, the equations become:

$$\begin{cases} \dot{\vec{x}}_{trans} = A\vec{x} + B\vec{u} \\ \vec{y} = C\vec{x_{trans}} + D\vec{u} \end{cases}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} \mathbf{0}_{3\times3} \\ \frac{1}{m}\mathbf{I}_{3\times3} \end{bmatrix}_{6\times3} \quad C = \mathbf{I}_{6\times6} \;\; D = \mathbf{0}_{6\times3}$$

where m is the spacecraft mass, $\vec{u}$ is the applied force vector, and **I** is the identity matrix.

Traditionally for the CW equations, the dynamics are referenced relative to the target spacecraft. In our situation, we desire the chaser spacecraft to reach the same altitude as the target spacecraft, but 1 km in front of it in the orbit. In the reference frame relative to the target (shown in black in Figure 1), this would mean our final desired position would be $x = 0m$, $y = 1km$, $z = 0m$. Instead, we choose to define the 'target' reference frame as this final desired position, shown as the red dot in Figure 1. This linearization around the desired endpoint helps to make the analysis easier. Because the endpoint and the target craft are orbiting at the same altitude, they have the same orbital period and thus the offset between the two will stay constant at $1km$. To verify this, we can consider the CW equations at the endpoint. Here we only consider the translational dynamics, so $\vec{x} = (x, y, z, \dot{x}, \dot{y}, \dot{z})$.

$$\begin{aligned} \vec{x}_{endpoint} &= (0, 1km, 0, 0, 0, 0) \\ \ddot{x}_{endpoint} &= 3n^2(0) + 2n(0) + 0 = 0\, m/s^2 \\ \ddot{y}_{endpoint} &= -2n(0) + 0 = 0\, m/s^2 \\ \ddot{z}_{endpoint} &= -n^2(0) + 0 = 0\, m/s^2 \end{aligned}$$

Thus, the endpoint and the target spacecraft are stationary relative to each other in their respective reference frames (at equilibrium). Our new final position in this new linearized frame is now the origin. This is represented visually in Figure 1.

Euler's moment equations describe the attitude dynamics:

$$I\dot{\omega} + \omega \times (I\omega) = \tau_{ext} \tag{2}$$

$$\omega = [\omega_x, \omega_y, \omega_z]^T$$

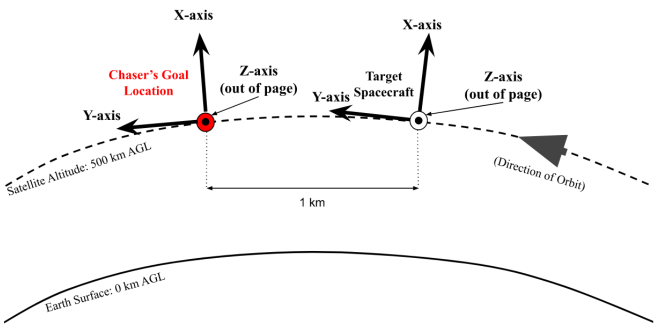

Fig. 1. Chaser satellite's goal location (red) and Target satellite (black) in CW frame (curvature of Earth exaggerated).

where ω is the instantaneous angular velocity of the spacecraft represented in the body-fixed frame, I is the spacecraft's moment of inertia matrix relative to it's principal axes, τ_ext are the external torques on the system.

We also need to define how the instantaneous angular velocity ω relates to the Euler angles - roll (ϕ), pitch (θ), and yaw (ψ) - which describe the orientation of the spacecraft with respect to the CW frame.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{3}$$

(4)

where $\omega_x, \omega_y, \omega_z$ are the instantaneous angular velocity components of the spacecraft in it's body fixed frame. Equation 4 relates the instantaneous angular velocity to the rates of change of the Euler angles, which fully defines the attitude dynamics model.

2) Linearized Attitude Model: The linearized attitude state-space representation is given by the equation:

$$\dot{\vec{x}}_{att} = A\vec{x}_{att} + B\vec{u}$$

where the A and B matrices are given below:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{I_x} & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix}$$

with the input vector $\vec{u}$ representing each component of the potential external torque:

$$\vec{u} = \begin{bmatrix} \tau_{ext,x} \\ \tau_{ext,y} \\ \tau_{ext,z} \end{bmatrix}$$

This model is linearized around steady state conditions of zero Euler angles and zero angular rate.

3) Initial Conditions: We define the initial conditions to be:

$$\vec{x}_{trans} = \left(30km, 70.72km, 0km, 0\frac{m}{s}, -58.2\frac{m}{s}, 0\frac{m}{s}\right)$$

$$\vec{x}_{att} = (0, 0, 0, 0, 0, 0)$$

These initial conditions describe the satellite in a circular orbit $30km$ above the target altitude, starting $70.72km$ ahead of the target in the along-track direction. The cross-track position and velocity are both zero, which assumes the spacecraft are in the same orbit plane. This initial position is the location required for a Hohmann transfer to the target point. The Euler angles and rates are all assumed to be zero, indicating an initially controlled attitude, which is a reasonable assumption.

The Clohesy-Wiltshire equations are a reasonable assumption for this scenario, as the relative position of the satellites is in the order of tens of kilometers, which is much smaller than the orbital radii which is on the order of thousands of kilometers.

4) Reference Trajectory: The reference trajectory for our maneuver is a Hohmann transfer from the 530km orbit to the 500km orbit of the target satellite. Due to the linearization of the relative dynamics model, the true Hohmann transfer solution trajectory did not match the trajectory simulated in the unforced linear model. This meant that the controller would need to constantly 'fight' the dynamics in order to track the signal instead of the efficient two impulse trajectory that makes the Hohmann transfer so efficient. We decided to instead take the initial conditions of the Hohmann transfer solution, then tweak the along track velocity until the trajectory matched the endpoints of the Hohmann transfer solution. This allowed us to assess the performance of the controller in responding to the two step inputs (the impulse burns) as well as it's tracking performance in between. The reference trajectory is shown in Figure 2. The reference signal provided to the controller contains full position and velocity information as would be provided by a perfect estimator. The first step input happens at the start of the simulation, which is marked as the initial position in Figure 2. The initial condition is set different than the reference to create a step input. The second step input occurs when the chaser reaches the red square with the arrow and acts to cancel out any relative velocity. The reference signal continues on at the endpoint to assess the control in steady state conditions.

B. Controller Design

This section outlines our design and implementation of a Proportional-Integral-Derivative controller, and a Linear-Quadratic-Regulator controller for the transfer problem. First, requirements for the control system will be described. After this, the PID and LQR design and implementation will be discussed.

1) Controller Requirements: For our maneuver scenario, there are a few important requirements. Our maneuver falls under the Drift Orbit B phase of rendezvous and proximity operations, which is characterized by the range of distances between the satellite and the target. The range of distances for Drift Orbit B starts at $2\lambda_{max}$ and ends at about $1km$ from the target, where λ_{max} is the maximum Earth central angle from the altitude of the spacecraft [2]. This phase of rendezvous is approximated well by the CW equations, as the maximum relative distance between the target and chaser is on the order of tens of kilometers to hundreds of meters, while λ_{max} is on the order of thousands of kilometers. For our control system, we came up with the following requirements:

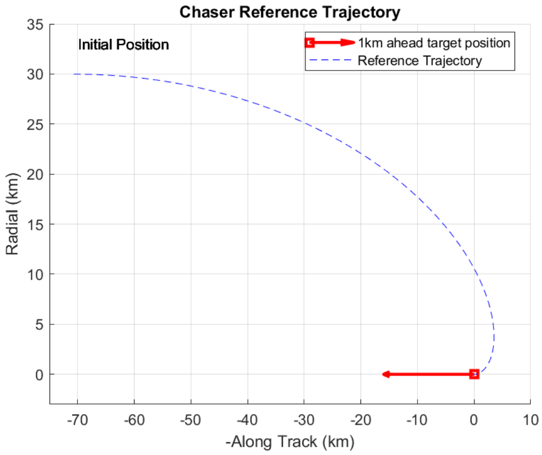

Fig. 2. Tracking signal for the control system

1) Maximum Overshoot = 100m
2) Maximum Settling Time = 1 minute
3) Maximum Rise Time = 1 minute

These controller requirements are derived from our maneuver scenario. As the chaser craft is approaching a point 1 km in front of the target spacecraft, the maximum overshoot must be smaller than this. We chose 10% of this final distance, as this gives a safety margin of 10. The maximum settling time was chosen to be approximately two percent of the transfer time of 47 minutes, as the chaser spacecraft should be settled into the reference trajectory by then. The same reasoning applies to rise time, as is important to quickly track the reference trajectory, especially at large changes like the burns at the start and end of the Hohmann transfer. We chose to ignore peak time, as this is not relevant to our system.

Translating these controller requirements into valid pole locations results into a region in the complex plane as shown in Figure 3. The rise time is translated into a minimum distance from the origin, seen as the circle (distorted to look like an oval) on the plot. The settling time is represented as the vertical line, which says the real part of the poles must be to the left of the line. The max overshoot represents the diagonal lines coming from the origin, prescribing a specific damping ratio for the system.

2) Simulation Design and Implementation: The Simulink model is designed to allow for easy modification of parameters. Our comparison necessitates variation of only the controller and disturbances, so keeping everything else the same was important. The model is designed to allow for varying

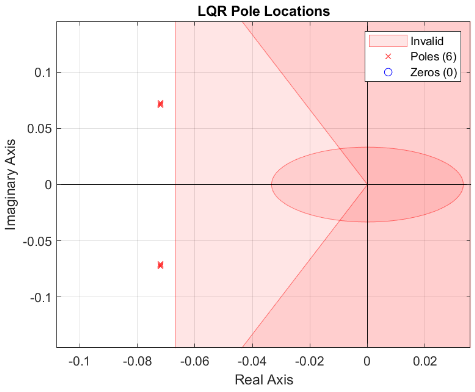

Fig. 3. LQR System poles fall outside the red area, which satisfies the control system requirements.

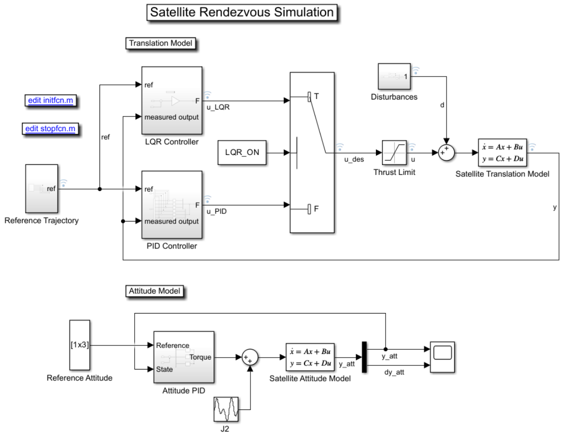

Fig. 4. Simulink Simulation Setup

initial conditions, different target and chaser parameters, and choice of PID or LQR control. These parameters are all set in an initialization function that runs before every simulation. The overarching model structure is shown in Figure 4. Separate subsystem blocks for the PID and LQR controllers allows the toggling of which controller is used. A saturation block is used to limit the thrust making the simulation more realistic. Disturbances of drag and J2 effect are included in the final force input to the state space translation model.

The attitude is designed to regulate to zero degrees in all Euler angles, utilizing PID control to stabilize the system.

3) PID Design and Implementation: Our PID controller uses 6 separate unity feedback PID controllers to control the spacecraft. Each of the satellites axes has two PID controllers, one for the position and one for the velocity. The controllers aim to drive the error between the reference state and the measured state to zero. We decided to control both the position and velocity because the reference signal varies in time, and if the satellite is in the right position but with the wrong velocity, it won't go where we want it to.

	x	y	z	u	v	w	ϕ	θ	ψ
P	10	10	10	10	10	10	0.733	0.733	0.733
I	0.01	0.01	0.01	0.1	0.1	0.1	0.002	0.002	0.002
D	50	50	40	0	0	0	66.1	66.1	66.1
N	0.2	0.2	0.2	100	100	100	7.62	7.62	7.62

TABLE I
TUNED PID GAINS FOR EACH REFERENCE STATE VARIABLE

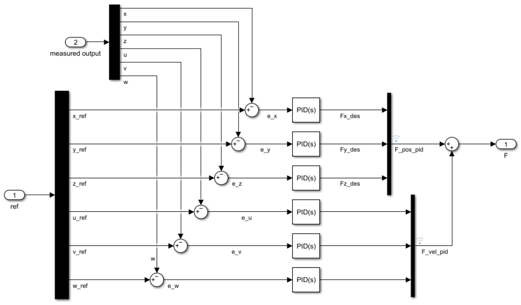

Fig. 5. Simulink Subsystem Block implementation of the PID controller

We used Simulink's PID Controller block for our PID controllers. This formulation includes parameters for the proportional, integral, and derivative gains, as well as a low-pass filter cutoff frequency. The transfer function of the PID blocks from error to force is shown in eqn 5. The transfer functions were used to derive the transfer function matrix of the closed loop system for Bode plot analysis.

$$G_{PID}(s) = K_P + K_I \frac{1}{s} + K_D \frac{N}{1 + N\frac{1}{s}} \tag{5}$$

The implementation of the full controller is shown in Figure 5. The contributions of each PID block for each axis summed together to produce the force vector output $F = [u_x, u_y, u_z]^T$. This converts the 6×1 vector of state errors into a 3×1 vector of force output. The final PID gains for each controller are shown in Table I. PID controllers are also used to control each Euler angle, which was linearized around Euler angles and rates of zero.

4) LQR Design and Implementation: The LQR controller was designed using Matlab. Using the state cost matrix Q, the control cost matrix R, and the state space translation model, the gain matrix and poles of the system were calculated. The state cost matrix were initially set as the 6×6 identity matrix, and the control cost matrix was set as the 3 identity matrix. We then normalized both matrices, and then multiplied Q by a priority factor ρ, and R by $(1 - \rho)$:

$$Q = \rho Q_{normalized}$$
$$R = (1 - \rho) R_{normalized}$$

This allowed adjustment to tune the trade-off of speed versus efficiency. The priority, $\rho = 0.6$, is set to prioritize speed slightly more than efficiency.

The LQR controller is designed to regulate the state error to zero. The error is then multiplied by the 3×6 gain matrix K_{LQR} to produce the 3×1 output force vector. This results in a very simple and elegant controller, as shown in Figure 6. The beauty of this control scheme is that the full state information of position and velocity essentially turns the control output into an optimized PD controller. The position feedback is the proportional control, and since the velocity is the derivative of the position, the velocity acts as derivative feedback.

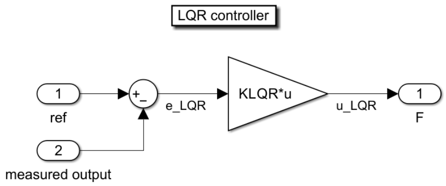

Fig. 6. LQR Controller Implementation

III. Results and Discussion

This section will detail analytical and experimental results from our work. First, simulation results will be covered, then Bode plot and Nyquist plot analyses.

A. Simulation Results

From the simulation results, both controllers were able to track the reference signal effectively with and without disturbances. The PID controller was quicker to respond to the impulsive burns in the reference trajectory, but tended to oscillate more in steady state. The LQR controller had a much smoother response, but took longer to respond. From Figure 7 it is clear that the PID controller had much tighter control than the LQR controller. However, the LQR controller was much more efficient with it's control input. As seen in Figure 8, the LQR controller has an impulse at the beginning, then has a comparatively smoother control input than the PID controllers, which oscillates throughout the entire transfer.

B. Bode Plot Analysis

Bode Plot frequency analysis was done on the full system transfer function for both the PID and LQR controllers. The frequency spectrum used encompassed 10^{-6} rad/s to 10^4 rad/s. Converted to a period, it covers a range of 72 days to 0.06 seconds. We limited the frequency spectrum as Matlab's bode() function decided that the 'interesting' dynamics were happening around a rate on the order of magnitude of roughly 10^-13 rad/s, but this equates to a period of approximately 2 million years, which is very out of the scope of this 47 minute maneuver.

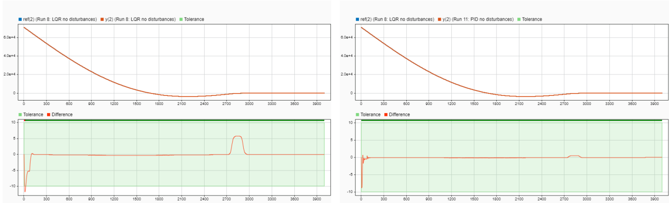

Fig. 7. LQR (left) vs PID (right) controller's y position compared to the reference.

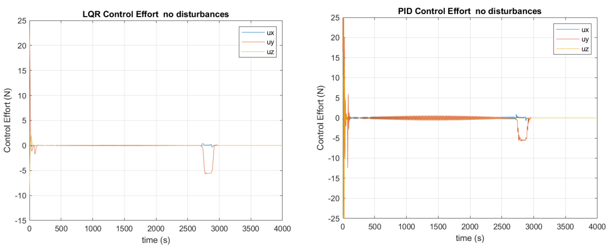

Fig. 8. LQR (left) controller is much more efficient than PID (right) controller in terms of effort.

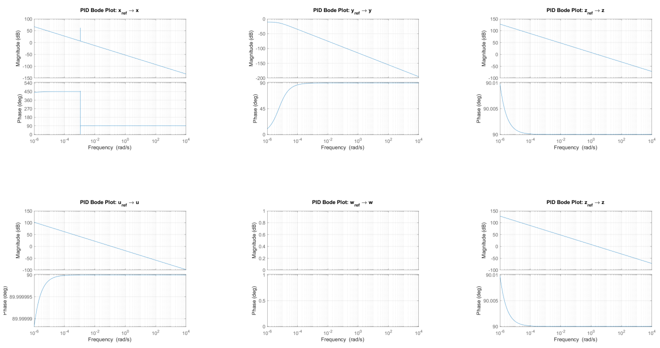

Fig. 9. Bode Frequency Plots for the closed loop PID controller.

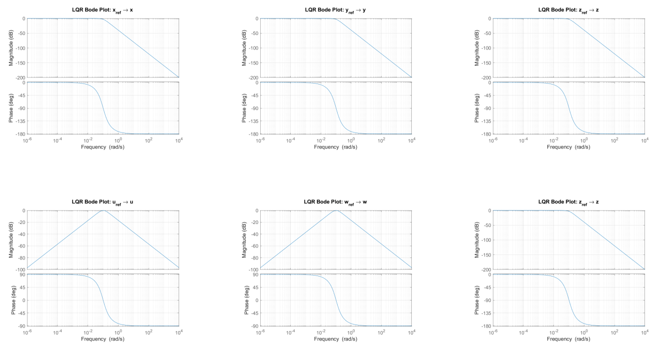

Fig. 10. Bode Frequency Plots for the closed loop LQR controller.

The stability margin can be seen in the bode plots of the PID and LQR systems from figures 9 and 10 respectively. The Bode Plots of the PID controller of x to x show a phase margin of 90 degreces but a gain margin of 6 dB on the conservative side. The Bode plots of the LQR controller show stability as they do not cross -180 degrees of phase.

C. Nyquist Plot Analysis

The Nyquist plots of the control systems are shown in Figure 12 for the LQR and Figure 11 for the PID controller. For the LQR controller, the number of open loop poles in the right half plane is zero, thus the zero loops around the critical point -1 indicates the controller is stable, supporting the evidence from the simulations. However, for the PID controller, there were 2 open loop poles in the right half plane, and the nyquist plot does not make any loops around -1. This suggests that the PID controller has closed loop poles

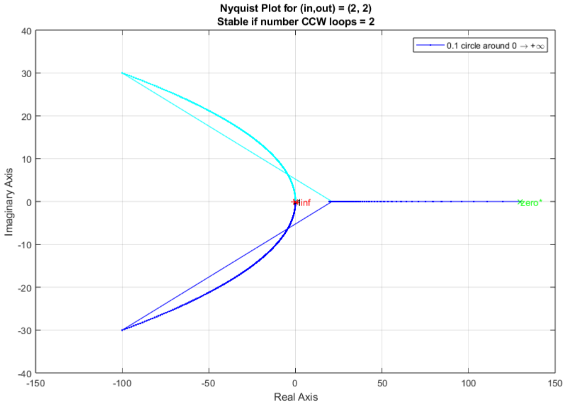

Fig. 11. PID Nyquist plot

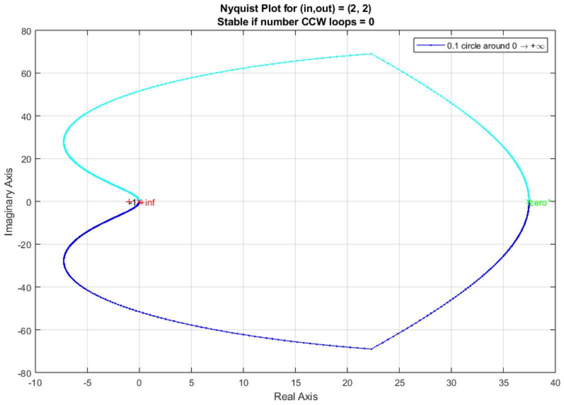

Fig. 12. LQR Nyquist plot

in the RHP, which suggests instability. However, after looking at those closed loop poles in the RHP, the real part was on the order of 10^{-6}, which could indicate that the effect is negligible.

IV. Conclusion and Future Directions

The pursuit of innovative satellite rendezvous techniques, as described in this report, spotlights the pivotal role of advanced control designs in enhancing orbital service capabilities for the future of the space industry. This project demonstrates a novel approach to the challenges associated with rendezvous maneuvers from a higher to lower orbit, which are increasingly critical as satellite operations become much more frequent and complex.

Through the development and testing of a control system for use on an orbit-servicing chaser satellite, we have shown that intricate orbital rendezvous maneuvers can be performed with high precision. Our results confirm the effectiveness of our control system, even in the face of several interfering variables common in LEO space flight. The robustness of both the PID and LQR controllers in maintaining course corrections and adjusting to the dynamic space environment highlights the positive results of this project and sets the foundation for the next generation of autonomous rendezvous technology.

Looking to the future, the areas this project could impact are wide-reaching. The ability to efficiently and safely alter satellite orbits has usefulness for orbital debris management, satellite constellation/orbit maintenance, and extending mission lifespans through on-orbit servicing. Each of these areas would directly benefit from the precise control techniques that have been developed and refined in this study.

As the space industry continues to grow exponentially, the integration of better simulation computations as well as the exploration of AI-based control algorithms will be huge factors in advancing our ability to manage spacecraft. Enhanced computational models will provide deeper insight into the nuances of orbital dynamics, while AI algorithms promise to revolutionize control systems with unprecedented adaptability and responsiveness. These advancements will enable the undertaking of more ambitious missions, paving the way for a future where space operations are both more reliable and safe.

V. References

[1] SpaceX, "Falcon 9," SpaceX, 2023. https://www.spacex.com/vehicles/falcon-9/

[2] Wertz, James R.; Bell, Robert (2003). Tchoryk, Jr., Peter; Shoemaker, James (eds.). "Autonomous Rendezvous and Docking Technologies – Status and Prospects" (PDF). SPIE AeroSense Symposium. Space Systems Technology and Operations Conference, Orlando Florida, April 21–25, 2003. 5088: 20. Bibcode:2003SPIE.5088...20W. doi:10.1117/12.498121. S2CID 64002452. Paper 5088-3. Archived from the original (PDF) on April 25, 2012. Retrieved August 3, 2019.

[3] "20N Monopropellant Thruster - Orbital Propulsion Centre — Spacecraft Thruster," Satnow.com, 2024. https://www.satnow.com/products/thrusters/orbital-propulsion-centre/36-1158-20n-monopropellant-thruster (accessed Apr. 29, 2024).

VI. Appendix

A. Code

The code and extra plots for this project is available at https://github.com/robqduquette/aero470_RPOS_project.