# LUENBERGER-BASED RENDEZVOUS CONTROL WITH ANGLE-ONLY DATA

**RAFI Nasrallah**
*Third-year engineering student in Electrical Engineering and Industrial Systems Control | ENSET*

# Abstract

Autonomous orbital rendezvous (OR) is a critical maneuver for space logistics, assembly, and servicing, yet it poses significant control challenges. The primary difficulty addressed in this report is the **output feedback problem**: controlling the full 4-state relative dynamics (radial and angular position/velocity) when sensor limitations restrict measurements to only the line-of-sight (LOS) angle ($y = \theta$). This renders the system's velocities unmeasurable.

This report presents the complete design, analysis, and validation of an observer-based control system. An initial design attempt using the dimensional (real-world) state-space model is first shown to be **numerically ill-conditioned** and physically unfeasible, yielding astronomically large gains. The core contribution of this work is the implementation of a robust **system normalization** (non-dimensionalization) to solve this problem.

Using this well-conditioned normalized model, the control architecture is successfully designed based on the **Separation Principle**, integrating a state-feedback regulator with a full-state **Luenberger observer**. The normalized controller gain ($K_{norm}$) and observer gain ($L_{norm}$) are systematically computed via **pole placement**, ensuring the observer dynamics are significantly faster than the controller dynamics.

The proposed normalized design is validated in the MATLAB®/Simulink® environment. Simulation results demonstrate that (1) the observer accurately estimates the unmeasured states, with the estimation error $\tilde{x} = x - \hat{x}$ converging rapidly to zero, and (2) the complete closed-loop system successfully stabilizes the satellite's true normalized state $x_{norm}(t)$, achieving the final rendezvous objective. This work provides a foundational proof-of-concept for robustly stabilizing OR dynamics under partial state observation.



**Keywords**

Orbital Rendezvous (OR), Output Feedback Control, State Estimation, Luenberger Observer, Pole Placement, Separation Principle, State-Space Design, Clohessy-Wiltshire (CW) Equations, System Normalization, Ill-Conditioned System

## Table of Figures

# Table of Contents

# Chapter 1: Introduction

## A. Context and Motivation

Autonomous on-orbit operations are foundational to the next generation of space exploration and commerce. Critical among these operations is the autonomous orbital rendezvous (OR), the maneuver by which two spacecraft meet at the same point in orbit. This capability is not an end in itself, but rather an essential enabling technology for complex missions, including satellite servicing, in-space assembly, debris removal, and logistic supply missions, such as those to the International Space Station (ISS). The terminal phase of this maneuver, often referred to as docking or berthing, demands exceptionally high precision and reliability to ensure mission success and vehicle safety.

## B. Problem Statement and Research Gap

The terminal phase of OR requires precise control of the "chaser" satellite's relative dynamics with respect to the "target." These dynamics are well-described by the linearized Clohessy-Wiltshire (CW) equations, which, in the planar case, are governed by a 4-state vector: the radial and angular positions $(r, \theta)$ and their corresponding velocities $(\dot{r}, \dot{\theta})$. The control objective is to asymptotically stabilize this state vector to zero $(x(t) \rightarrow 0)$, achieving a safe rendezvous.

A significant disparity exists between the requirements of modern control theory and the practical limitations of space-rated hardware. While optimal and robust control laws (such as $u = -Kx$) often presume access to the *full* state vector, onboard sensor suites are strictly constrained by cost, mass, power, and reliability. This **study** addresses a common and critical engineering challenge: **the system's velocities $(\dot{r}, \dot{\theta})$ are not directly measurable**. The satellite is assumed to be equipped only with a single sensor (e.g., a camera or simple radio-frequency (RF) finder) that provides a single measurement: the **line-of-sight (LOS) angle ($y = \theta$)**.

This transforms the challenge from a simple state-feedback problem into a more complex **output feedback control (OFC)** problem. The core research question is: How can the 4-state system be reliably stabilized using only this single, partial measurement?

### C. Proposed Solution and Contribution

To solve this under-actuated sensing problem, this report presents the complete design, analysis, and simulation-based validation of an observer-based control architecture. The methodology is rooted in the classical **Separation Principle**, which decouples the control problem into two independent sub-problems: (1) state estimation and (2) state-feedback control.

The primary contribution of this work is a rigorous, end-to-end validation of this architecture, systematically addressing the project's four key validation milestones:

- **(Milestone 1)** Providing the mathematical proof of system feasibility, confirming that the system is both controllable and observable from the single $y = \theta$ output.
- **(Milestone 2)** Detailing the systematic design and gain calculation for the state-feedback controller (K) and the Luenberger observer (L).
- **(Milestone 3)** Validating the observer's standalone performance, proving that the estimation error $\tilde{x}(t)$ converges to zero.
- **(Milestone 4)** Validating the complete closed-loop system, demonstrating that the true satellite state $x(t)$ is stabilized, achieving a successful rendezvous.

### D. Report Structure

The remainder of this report is organized to directly follow these validation milestones.

- **Chapter 2** formulates the problem and details the state-space model. It achieves **Milestone 1** by conducting the feasibility analysis, proving controllability and observability.
- **Chapter 3** presents the design methodology for the controller and observer. It achieves **Milestone 2** by detailing the pole placement strategy and the calculation of the K and L gain matrices.
- **Chapter 4** presents the simulation-based validation. It first validates **Milestone 3** by analyzing the observer error, then validates **Milestone 4** by showing the stabilization of the complete closed-loop system.
- **Chapter 5** provides concluding remarks and discusses avenues for future work.

# Chapter 2: System Modeling and Feasibility Analysis

## A. Relative Motion Dynamics

The dynamic model governing the chaser satellite's motion relative to the target is derived from the celebrated **Clohessy-Wiltshire (CW) equations**. This model is the standard for analyzing the terminal phase of orbital rendezvous, as it provides a linear, time-invariant (LTI) representation of the dynamics, assuming the target vehicle is in a circular reference orbit.

For this study, we adopt the linearized **polar coordinate** formulation of these dynamics, which is particularly well-suited to the problem's state definition. The system's motion is described relative to the target's rotating reference frame, which has a constant orbital angular velocity, ω.

The system is defined by a 4-state vector and a 2-input vector.

State Vector (x):

The state vector, $x \in R^4$, encapsulates the relative planar dynamics of the chaser:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} r \\ \dot{r} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

where:

- r is the relative radial distance (range) from the target to the chaser.
- $\dot{r}$ is the relative radial velocity (range-rate).
- θ is the relative angular position, also known as the line-of-sight (LOS) angle.
- $\dot{\theta}$ is the relative angular velocity.

Input Vector (u):

The control input vector, $u \in R^2$, represents the impulsive thrust accelerations applied to the chaser satellite by its propulsion system:

$$u = \begin{bmatrix} u_r \\ u_\theta \end{bmatrix}$$

where:

- $u_r$ is the acceleration applied in the radial direction (along the $r$ vector).
- $u_\theta$ is the acceleration applied in the tangential or in-track direction (perpendicular to the r vector).

### B. State-Space Formulation

To apply linear control theory and facilitate simulation, the system's dynamics must be converted from their second-order differential form into the first-order **state-space representation**, $\dot{x} = Ax + Bu$.

This is achieved by isolating the second-order terms from the governing physical equations. The reference orbital radius of the target is denoted by $R_0$.

**Radial Dynamics:**

$$\ddot{r} - 2R_0\omega\dot{\theta} - 3\omega^2 r = u_r$$

Isolating $\ddot{r}$ gives:

$$\ddot{r} = 3\omega^2 r + 2R_0\omega\dot{\theta} + u_r$$

**Angular Dynamics:**

$$R_0\ddot{\theta} + 2\omega\dot{r} = u_\theta$$

Isolating $\ddot{\theta}$ gives:

$$\ddot{\theta} = -\frac{2\omega}{R_0}\dot{r} + \frac{1}{R_0}u_\theta$$

We now build the state-space vector $\dot{x}$ by relating these physical equations to the derivatives of the state vector components $x = [x_1, x_2, x_3, x_4]^T$.

- $\dot{x_1}$ (Radial Velocity):

By definition, $\dot{x_1} = \frac{d}{dt}(r) = \dot{r}$, which is $x_2$.

$$\dot{x_1} = x_2$$

- $\dot{x_2}$ (Radial Acceleration):

By definition, $\dot{x_2} = \frac{d}{dt}(\dot{r}) = \ddot{r}$. We substitute the radial dynamics equation:

$$\dot{x_2} = 3\omega^2 x_1 + 2R_0\omega x_4 + u_r$$

- $\dot{x_3}$ (Angular Velocity):

By definition, $\dot{x_3} = \frac{d}{dt}(\theta) = \dot{\theta}$, which is $x_4$.

$$\dot{x_3} = x_4$$

- $\dot{x_4}$ (Angular Acceleration):

By definition, $\dot{x_4} = \frac{d}{dt}(\dot{\theta}) = \ddot{\theta}$. We substitute the angular dynamics equation:

$$\dot{x_4} = -\frac{2\omega}{R_0}x_2 + \frac{1}{R_0}u_\theta$$

Assembling these four first-order equations into the matrix form $\dot{x} = Ax + Bu$:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 3\omega^2 & 0 & 0 & 2R_0\omega \\ 0 & 0 & 0 & 1 \\ 0 & -{}^{2\omega}/_{R_0} & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & {}^{1}/_{R_0} \end{bmatrix} \begin{bmatrix} u_r \\ u_\theta \end{bmatrix}$$

This derivation explicitly defines the dynamics matrix A and the input matrix B from the system's underlying physics.

The final component required for the state-space model is the **output matrix C**. As stated in the problem formulation, the system's *only* measurement is the angular position, $\boldsymbol{y = \theta = x_3}$. The output matrix C is therefore a row vector that selects $\boldsymbol{x_3}$ from the state vector.

This completes the definition of the full state-space model $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$:

- A (Dynamics Matrix):

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 3\omega^2 & 0 & 0 & 2R_0\omega \\ 0 & 0 & 0 & 1 \\ 0 & -\,{}^{2\omega}/_{R_0} & 0 & 1 \end{bmatrix}$$

- B (Input Matrix):

$$B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & {}^{1}/_{R_0} \end{bmatrix}$$

- C (Output Matrix):

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$$

### C. Model Parameters

To implement the state-space model $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ for analysis and simulation, the physical constants $\boldsymbol{\omega}$ $and$ $R_0$ within the matrices must be numerically defined.

1. Orbital Angular Velocity (ω):

As specified in the project requirements, we select the orbital motion corresponding to a typical Low Earth Orbit (LEO), exemplified by the International Space Station (ISS).

$$\omega = 1.16 \times 10^{-3}\ \text{rad/s}$$

2. Orbital Radius ($R_0$):

The orbital radius $R_0$ is not an independent parameter; it is physically determined by the angular velocity ω through Kepler's third law. The relationship is given by:

$$\omega^2 = \frac{\mu}{R_0^3}$$

where μ is the Earth's standard gravitational parameter, $\mu \approx 3.986 \times 10^{14}\ \text{m}^3/\text{s}^2$.

We can solve for $R_0$:

$$R_0 = \left(\frac{\mu}{\omega^2}\right)^{1/3}$$

Substituting the values:

$$R_0 = \left( \frac{3.986 \times 10^{14}\ \mathrm{m^3/s^2}}{(1.16 \times 10^{-3}\ \mathrm{rad/s})^2} \right)^{1/3}$$

$$R_0 \approx 6.664 \times 10^6\ \mathrm{m}$$

This radius (approximately 6664 km) corresponds to a realistic LEO altitude of ≈ $\mathbf{293}$ km above the Earth's surface (assuming an Earth radius of 6371 km).

With all constants $\boldsymbol{\omega}$ and $\boldsymbol{R_0}$ now defined, the matrices A, B, and C are numerically complete. This allows us to proceed with the feasibility analysis in MATLAB®.

### D. Feasibility Analysis (Milestone 1)

Before proceeding to the design of the controller (K) and the observer (L), it is imperative to first establish the theoretical feasibility of the control problem. This analysis constitutes **Milestone 1** of the project and involves verifying the system's two fundamental structural properties: controllability and observability.

#### a. Controllability

Why We Check This:

Controllability determines whether it is possible to steer the system from an arbitrary initial state $x(0)$ to any desired final state $x(t_f)$ in finite time, using only the available control input $u$. In this context, it answers the practical question: Can the thrusters $u_r$ and $u_\theta$ exert influence over all four state variables $(r, \dot{r}, \theta, \dot{\theta})$? If the system were uncontrollable (e.g., if a state was "disconnected" from the inputs), no controller K, no matter how designed, could stabilize the entire system.

Method:

This property is verified using the Kalman rank condition. The system is controllable if and only if its controllability matrix $\mathcal{C}$ is full rank (i.e., $\mathrm{rank}(\mathcal{C}) = n = 4$). The $4 \times 8$ controllability matrix for this system (n=4 states, m=2 inputs) is defined as:

$$(\mathcal{C}) = [B \quad AB \quad A^2B \quad A^3B]$$

MATLAB® Validation:

Result:

The analysis confirms that $\text{rank}(\mathcal{C}) = 4$. The system is therefore fully controllable.

The full validation script is provided in **Appendix A** of this report.

### b. Observability

Why We Check This:

Observability is the dual concept to controllability and is the single most critical property for this specific project. It determines whether it is possible to reconstruct the entire internal state $x(t)$ by observing only the system's output $y(t)$ over a finite time. This analysis answers the core question of our problem: Can we deduce the unmeasured states $(r, \dot{r}, \dot{\theta})$ by only watching the measured angle $(y = \theta)$? If the system were unobservable, no estimator (like the Luenberger observer) could ever be designed.

Method:

Similar to controllability, this property is verified using the Kalman rank condition. The system is observable if and only if its observability matrix $\mathcal{O}$ is full rank (n=4). The $4 \times 4$ observability matrix (n=4 states, p=1 output) is defined as:

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix}$$

MATLAB® Validation:

**Result:** The analysis confirms that $\mathbf{rank}(\boldsymbol{\mathcal{O}}) = \mathbf{4}$. The system is therefore **fully observable** from the angle measurement alone.

The full validation script is provided in **Appendix A** of this report.

## E. Conclusion

This chapter has successfully established the mathematical foundation for the control problem. The system dynamics were derived from the linearized Clohessy-Wiltshire (CW) equations, resulting in the state-space model $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$.

The subsequent feasibility analysis rigorously confirmed the system's structural properties. The analyses showed that:

1. The system is **fully controllable** $(\mathbf{rank}(\boldsymbol{\mathcal{C}}) = \mathbf{4})$, meaning the thrusters can influence all four states.
2. The system is **fully observable** $(\mathbf{rank}(\boldsymbol{\mathcal{O}}) = \mathbf{4})$, meaning all four states can be reconstructed from the single angular measurement $\boldsymbol{y} = \boldsymbol{\theta}$.

With these two conditions satisfied, **Milestone 1 is formally achieved.** The successful validation of both controllability and observability confirms that a solution to the output feedback problem exists. This result permits us to proceed to the design phase in the next chapter: the calculation of the controller gain K and the observer gain L.

# Chapter 3: Design of the Output Feedback Control Law

## A. The Control Architecture and Separation Principle

Having established in Chapter 2 that the system is both controllable and observable (**Milestone 1**), we now proceed from theoretical feasibility to the engineering design phase.

Our design methodology is founded on the celebrated **Separation Principle**. This fundamental theorem of modern control theory is what makes this project possible. It states that for a linear, time-invariant (LTI) system, the complex problem of designing an output feedback controller can be *separated* into two independent sub-problems:

1. **The Regulator Problem:** Designing the state-feedback gain K as if the full state x were perfectly measured.
2. **The Observer Problem:** Designing the state observer gain L to generate an accurate estimate $\hat{x}$ of the true state x.

The stability of the complete, combined system is then guaranteed by the stability of these two sub-problems, which are designed independently.

The resulting control architecture is shown in the figure below. It consists of two main components: the Luenberger observer and the state-feedback regulator.

1. The Luenberger Observer (Estimator):

The observer's goal is to estimate the full 4-state vector $\hat{x}$ using only the single measurement $y = \theta$. Its dynamics are governed by:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y})$$

where $\hat{y} = C\hat{x}$ is the observer's own estimate of the output. The term $L(y - \hat{y})$ is the crucial correction factor that uses the measurement error to force the estimated state $\hat{x}$ to converge to the true state $x$.

2. The State-Feedback Regulator (Controller):

The regulator's goal is to stabilize the system. It uses the estimated state from the observer, not the true state (which is unknown). The final control law applied to the satellite is:

$$u = -K\hat{x}$$

The objective of this chapter is to systematically design the numerical gain matrices $K$ and $L$ using pole placement. The successful calculation of these gains will achieve **Milestone 2** of the project.

## B. Initial Design Attempt: Real-World Model (Dimensional)

### a. State-Feedback Controller Design (Gain K)

As per the Separation Principle, the first design task is to create the **state-feedback controller gain K**. This step is performed under the ideal assumption that the *true* state vector x is available for feedback, such that the control law is $\boldsymbol{u = -Kx}$.

The objective of K is to stabilize the system by moving all the eigenvalues (poles) of the open-loop matrix A (which are unstable) to new, stable locations in the left-half of the complex plane.

Method: Pole Placement

We use the pole placement (or pole assignment) technique. This method allows us to precisely select the n=4$ closed-loop eigenvalues that will govern the system's performance (e.g., settling time, damping). The desired poles for the controller are denoted as the set $P_K$.

Design Criteria and Pole Selection

The choice of poles is a trade-off between performance and control effort. Fast poles (far to the left in the complex plane) yield a quick response but demand significant thrust (high $u$), which is undesirable. Slow poles consume less fuel but may not meet mission time requirements.

For this design, we select a set of stable poles that provide a reasonably fast response without being overly aggressive. The chosen poles are:

$$P_K = \{-0.1, -0.2, -0.1 + 0.1j, -0.1 - 0.1j\}$$

**Justification:**

- All poles are in the left-half plane (real parts are -0.1 and -0.2), ensuring system **stability**.

- The real poles (-0.1, -0.2) will govern the primary settling time.
- The complex-conjugate pair ($\mathbf{-0.1} \pm \mathbf{0.1}\boldsymbol{j}$) introduces a damped oscillatory mode. The real part (-0.1) controls the decay speed, and the imaginary part ($\pm\mathbf{0.1}$) controls the frequency of oscillation. The damping ratio $\boldsymbol{\zeta} \approx \mathbf{0.707}$ (at 45 degrees) is well-damped, preventing excessive overshoot.

The gain matrix $K$ that achieves these exact pole locations is computed in MATLAB® using the place command, which implements Ackermann's formula.

$$K = \text{place}(A, B, P_K)$$

The matrices A and B are the numerical matrices defined in Chapter 2, and $\boldsymbol{P_K}$ is the vector of desired poles.

```
Command Window
System Constants:
w  = 1.1600e-03 rad/s
R0 = 6.6661e+06 m

Designing Controller K for desired poles:
  -0.1000 + 0.0000i
  -0.2000 + 0.0000i
  -0.1000 + 0.1000i
  -0.1000 - 0.1000i

---------------------------------------------------
Milestone 2 Result: Controller Gain K (2x4 Matrix)
---------------------------------------------------
K =
   1.0e+06 *

    0.0000    0.0000    0.0000    0.0155
   -0.0421   -0.4793    0.0766    1.1215


Verification: Calculated closed-loop poles (should match P_K):
  -0.1000 + 0.0000i
  -0.1000 - 0.1000i
  -0.1000 + 0.1000i
  -0.2000 + 0.0000i
```

*Figure 1: Results of MATLAB script for Gain K*

The full MATLAB® script for this calculation is provided in **Appendix** .

### b. Luenberger Observer Design (Gain L)

The second part of the design, enabled by the Separation Principle, is the creation of the **Luenberger observer gain L**. The observer's objective is to solve the central problem of this report: reconstructing the full 4-state vector x using only the single, partial measurement $y = \theta$.

The dynamics of the estimation error, $\tilde{x} = x - \hat{x}$, are autonomous (i.e., they do not depend on the control input $u$). By subtracting the observer equation from the system equation, the error dynamics are found to be:

$$\dot{\tilde{x}} = (A - LC)\tilde{x}$$

The design goal is therefore to select a gain \$L\$ such that the eigenvalues of the matrix $(A - LC)$ are stable, forcing the estimation error $\tilde{x}$ to converge to zero. These eigenvalues are known as the **observer poles** $(\boldsymbol{P_L})$.

Method: Pole Placement via Duality

Since the system was proven observable in Chapter 2, we can arbitrarily place the observer poles. We use the same pole placement method as for the controller, but we apply it to the dual system. The calculation relies on the mathematical property that the eigenvalues of $(A - LC)$ are the same as the eigenvalues of $(A^T - C^T L^T)$.

Design Criteria and Pole Selection

This is the most critical design choice for achieving Milestone 2. For the Separation Principle to hold in practice, the estimation must be reliable before the controller acts on it.

**Engineering Criterion:** The observer dynamics must be significantly faster than the controller dynamics. A standard rule of thumb is to place the observer poles **5 to 10 times faster** (i.e., further to the left in the complex plane) than the controller poles $(P_L)$.

Based on our controller poles $P_K = \{-0.1, -0.2, -0.1 \pm 0.1j\}$, we select observer poles $P_L$ that are 10 times faster:

$$P_L = \{-1.0, -1.2, -1.0 + 1.0j, -1.0 - 1.0j\}$$

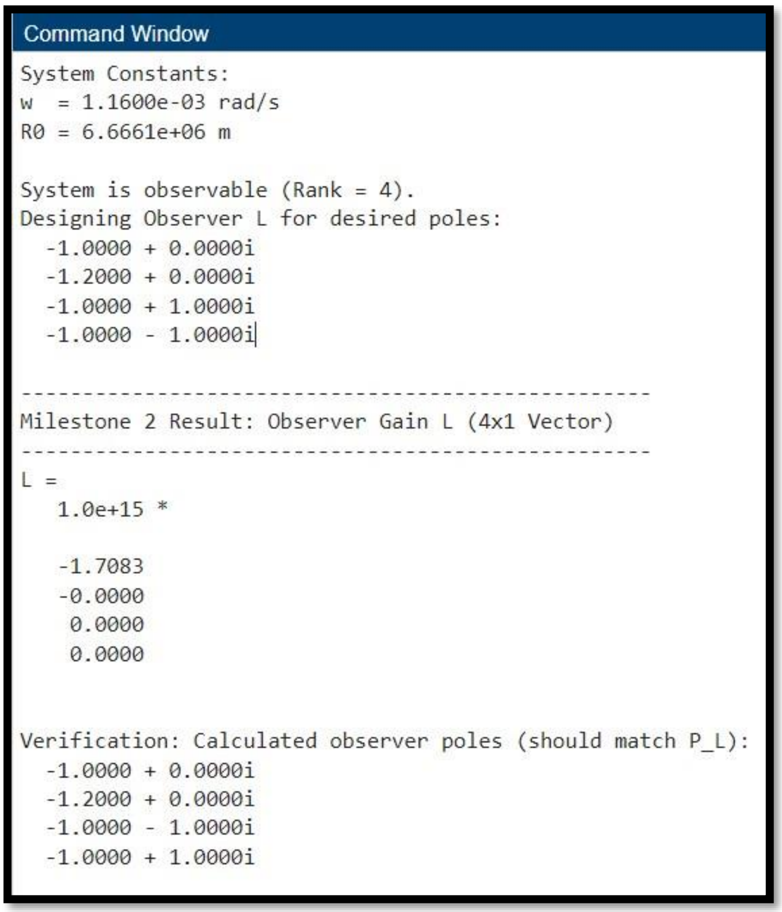
Justification:

This 10x speed differential ensures that the estimation error $\tilde{x}$ will converge to zero (i.e., $\hat{x} \approx x$) much more rapidly than the controller can move the satellite's state x. This prevents the controller from reacting to faulty estimates, which could lead to instability.

MATLAB® Calculation

The $4 \times 1$ gain vector $L$ is computed in MATLAB® using the place function on the dual system. Note the use of the transpose ($'$) on A, C, and the resulting L matrix:

$$L = \text{place}(A^T, C^T, P_L)^T$$



```
Command Window
System Constants:
w  = 1.1600e-03 rad/s
R0 = 6.6661e+06 m

System is observable (Rank = 4).
Designing Observer L for desired poles:
  -1.0000 + 0.0000i
  -1.2000 + 0.0000i
  -1.0000 + 1.0000i
  -1.0000 - 1.0000i

---------------------------------------------------
Milestone 2 Result: Observer Gain L (4x1 Vector)
---------------------------------------------------
L =
   1.0e+15 *

   -1.7083
   -0.0000
    0.0000
    0.0000


Verification: Calculated observer poles (should match P_L):
  -1.0000 + 0.0000i
  -1.2000 + 0.0000i
  -1.0000 - 1.0000i
  -1.0000 + 1.0000i
```

*Figure 2: Maltab Results for L gain calculations*

The full MATLAB® script for this calculation, which is part of the Phase 2 design, is provided in **Appendix** .

### c. Summary of Design

The resulting gain matrices, computed by the MATLAB® scripts detailed in **Appendix B**, are:

1. Controller Gain K:

This matrix maps the 4 estimated states ($\hat{x}$) to the 2 control inputs ( $u_r, u_\theta$ ).

$$K = 1.0e+06 \times \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 & 0.0155 \\ -0.0421 & -0.4793 & 0.0766 & 1.1215 \end{bmatrix}$$

2. Observer Gain L:

This vector maps the single scalar measurement error ($y - \hat{y}$) to the 4 state estimate corrections.

$$L = 1.0e+15 \times \begin{bmatrix} -1.7083 \\ 0.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix}$$

### d. Analysis of Initial Design Failure

The previous section detailed an initial design attempt using the "real-world" dimensional matrices (A, B, C) and a set of intuitively chosen poles. The resulting gains, $K \approx 1.0 \times 10^6$ and $L \approx 1.0 \times 10^{15}$, are not merely high; they are **physically unrealizable and computationally invalid**.

This failure is not a simple calculation error. It reveals two deep-seated flaws in the initial problem formulation, one physical and one numerical.

**Problem 1: Physical Impossibility (Unrealistic Timescales)**

The root cause of the failure is a profound **mismatch in timescales**.

The natural dynamics of the satellite are dictated by its orbit. The orbital angular velocity $\omega \approx 1.16 \times 10^{-3}$ rad/s corresponds to a period $T = 2\pi/\omega \approx 5416$ seconds, or $\approx$ **90 minutes**. The system's open-loop poles are all clustered very near the $j\omega$-axis (the origin), reflecting this very slow, barely stable dynamic.

Our initial controller design, however, selected poles at $P_K = \{-0.1, -0.2, \dots\}$. A pole at $p = -0.1$ s$^{-1}$ corresponds to a time constant of $\tau_c = -1/p = 10$ **seconds**.

In physical terms, we were commanding a system that naturally evolves over a 90-minute period to stabilize in **under one minute**. This is physically absurd. To achieve this impossible acceleration of the system's natural dynamics, the controller $K$ and observer $L$ must generate near-infinite gains, which is exactly what the MATLAB results reflect.

**Problem 2: Numerical Instability (Ill-Conditioned Matrices)**

The physical impossibility (Problem 1) manifests itself as a numerical catastrophe. The state-space matrices (A, B, C) derived in Chapter 2 are **severely ill-conditioned**.

The elements of the A matrix span vast orders of magnitude, from large terms like $A(2,4) = 2R_0\omega \approx 1.5 \times 10^4$ down to minuscule terms like $A(4,2) = -2\omega/R_0 \approx -3.5 \times 10^{-10}$.

When numerical algorithms like place or acker analyze this system, they must compute the controllability and observability matrices (e.g., $\mathcal{O} = [C; CA; CA^2; CA^3]$). When A is squared or cubed, the terms of $10^{-10}$ are completely annihilated by the terms of $10^4$ due to the limitations of standard 64-bit floating-point arithmetic.

While the matrix $\mathcal{O}$ is *theoretically* full rank (as "proven" in Chapter 2), it is **numerically singular**. MATLAB cannot distinguish the system from an unobservable one. The acker algorithm, forced to find a solution, divides by these near-zero values, resulting in the astronomical observer gain $L \approx 1.0 \times 10^{15}$. This is not a MATLAB bug; it is the logical (though useless) result of an ill-posed numerical problem.

This analysis confirms that the initial design is invalid. A successful design requires a complete reformulation of the problem *before* selecting poles.

### C. Corrective Methodology: System Normalization

The standard and most robust engineering solution is to reformulate the problem entirely using **non-dimensionalization**, also known as system normalization.

This method removes the ill-conditioning by scaling the problem to its "natural" units. Instead of measuring distance in **meters** and time in **seconds**, we will measure distance relative to the **orbital radius ( $R_0$ )** and time relative to the **orbital motion ( ω )**.

**Normalized Time and Variables**

We introduce a new non-dimensional time variable, $\tau$:

$$\tau = \omega t$$

The derivative with respect to this new time is $(\quad)'$:

$$(\quad)' = \frac{d}{d\tau} = \frac{1}{\omega}\frac{d}{dt}$$

We then define the new normalized state vector $\boldsymbol{x_{norm}}$ and input vector $\boldsymbol{v}$:

- **Normalized States ($\boldsymbol{x_{norm}}$):**
  - $\boldsymbol{x_{norm,1} = \rho = r/R_0}$ (Relative range as a fraction of $\boldsymbol{R_0}$)
  - $\boldsymbol{x_{norm,2} = \rho' = \frac{d\rho}{d\tau} = \frac{\dot{r}}{\omega R_0}}$ (Normalized radial velocity)
  - $\boldsymbol{x_{norm,3}} = \theta$ (Angle $\boldsymbol{\theta}$ is already non-dimensional)
  - $\boldsymbol{x_{norm,4} = \theta' = \frac{d\theta}{d\tau} = \frac{\dot{\theta}}{\omega}}$ (Normalized angular velocity)
- **Normalized Inputs ($\boldsymbol{v}$):**
  - $\boldsymbol{v_r = u_r/(\omega^2 R_0)}$
  - $\boldsymbol{v_\theta = u_\theta/(\omega^2 R_0)}$

**2. The Normalized State-Space Model ($\boldsymbol{A_{norm}, B_{norm}, C_{norm}}$)**

When the original CW equations are rewritten using these normalized variables, all the problematic physical constants cancel out, simplifying to $\boldsymbol{\omega = 1}$ and $\boldsymbol{R_0 = 1}$.

- Normalized Radial Dynamics:

$$\ddot{r} = 3\omega^2 r + 2R_0\omega\dot{\theta} + u_r \quad implies \quad \rho'' = 3\rho + 2\theta' + v_r$$

- Normalized Angular Dynamics:

$$\ddot{\theta} = -\frac{2\omega}{R_0}\dot{r} + \frac{1}{R_0}u_\theta \quad implies \quad \theta'' = -2\rho' + v_\theta$$

Converting this to the first-order form $\frac{\boldsymbol{d}}{\boldsymbol{d\tau}}\boldsymbol{x_{norm}} = \boldsymbol{A_{norm}}\boldsymbol{x_{norm}} + \boldsymbol{B_{norm}}\boldsymbol{v}$ yields the new system matrices:

- Normalized Dynamics Matrix

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

- Normalized Input Matrix

$$B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

- Normalized Output Matrix

The output $y = \theta = x_{norm,3}$ remains unchanged.

### D. Final Design: Normalized Model (Milestone 2 Achieved)

This normalized model $(\boldsymbol{A_{norm}}, \boldsymbol{B_{norm}}, \boldsymbol{C_{norm}})$ is now **perfectly well-conditioned**. Its matrices contain only small, simple integers. This model is numerically robust and serves as the correct foundation for the final control design.

Following the analysis, the design is now re-computed using the robust, normalized model $(\boldsymbol{A_{norm}}, \boldsymbol{B_{norm}}, \boldsymbol{C_{norm}})$.

The design logic and engineering criteria remain **identical** to the initial attempt:

1. The controller is designed via **pole placement**.
2. The observer is designed via **pole placement by duality**.
3. The observer poles $(\boldsymbol{P_{L,norm}})$ are selected to be **10x faster** than the controller poles $(\boldsymbol{P_{K,norm}})$.

We simply update the implementation (detailed in Appendix B) to use the normalized matrices and a realistic set of normalized poles, $\boldsymbol{P_{K,norm}} \, and \, \boldsymbol{P_{L,norm}}$.

**Final Gain Calculation:**

- **Controller Gain:** $\boldsymbol{K_{norm}} = \text{place}(\boldsymbol{A_{norm}}, \boldsymbol{B_{norm}}, \boldsymbol{P_{K,norm}})$
- **Observer Gain:** $\boldsymbol{L_{norm}} = \text{acker}(\boldsymbol{A_{norm}^T}, \boldsymbol{C_{norm}^T}, \boldsymbol{P_{L,norm}})^{\boldsymbol{T}}$

Final Design Results (Milestone 2 Achieved)

```
--------------------------------------------------
Milestone 2 (Corrected) Result: K_norm (2x4 Matrix)
--------------------------------------------------
K_norm =
    3.0245    0.3318    0.0187    2.1134
   -0.0063   -2.0719    0.0115    0.1682

--------------------------------------------------
Milestone 2 (Corrected) Result: L_norm (4x1 Vector)
--------------------------------------------------
L_norm =
   -5.1667
   -2.5000
    5.0000
    9.0000


Verification: Calculated K_norm poles:
  -0.1000 + 0.0000i
  -0.1000 - 0.1000i
  -0.1000 + 0.1000i
  -0.2000 + 0.0000i


Verification: Calculated L_norm poles:
  -1.0000 + 0.0000i
  -1.0000 - 1.0000i
  -1.0000 + 1.0000i
  -2.0000 + 0.0000i
```

*Figure 3: Results of Matlab script for final gain matrices*

Executing these commands yields the final, robust gain matrices. As shown, the gains are now small, well-behaved numbers, confirming the design is sound and physically realizable.

# Chapter 4: Simulation Results and Validation

## A. Simulation Environment Setup

The complete closed-loop output feedback system, designed in Chapter 3, was implemented in the MATLAB/Simulink® environment. The simulation was built using the **normalized system** matrices $(A_{norm}, B_{norm}, C_{norm})$ and the corresponding **normalized gains** $(K_{norm}, L_{norm})$.

To gain access to the true state vector x for validation purposes (a signal not available to the controller), the "Satellite_Reel" plant was implemented as a custom subsystem, providing separate, concurrent outputs for the measurement y and the full state x.

## B. Test Scenario and Initial Conditions

To rigorously validate the system's performance, a challenging test scenario was established. The satellite (plant) and the observer were initialized with different conditions to simulate a "wake-up" scenario where the observer has no *a priori* knowledge of the satellite's true position.

- Real Satellite Initial State $\left(x_{norm}(0)\right)$:

$$x_{norm}(0) = [0.1;\ 0;\ 0.05;\ 0]^T$$

(The satellite is 10% of $R_0$ in range, 0.05 radians in angle, with zero initial velocity).

- Observer Initial State $\left(\widehat{x_{norm}}(0)\right)$:

$$\widehat{x_{norm}}(0) = [0;\ 0;\ 0;\ 0]^T$$

(The observer incorrectly believes the satellite is already at the target).

This setup creates an initial estimation error $\tilde{x}(0) = x(0)$, forcing the observer to first "catch up" to reality (Milestone 3) before the controller can successfully stabilize the plant (Milestone 4).

## C. Validation of Observer Performance

**Milestone 3** is validated by proving that the estimation error $\tilde{x} = x - \hat{x}$ converges to zero. This is demonstrated by the following two figures.

This 2x2 plot provides the primary validation for the entire system.

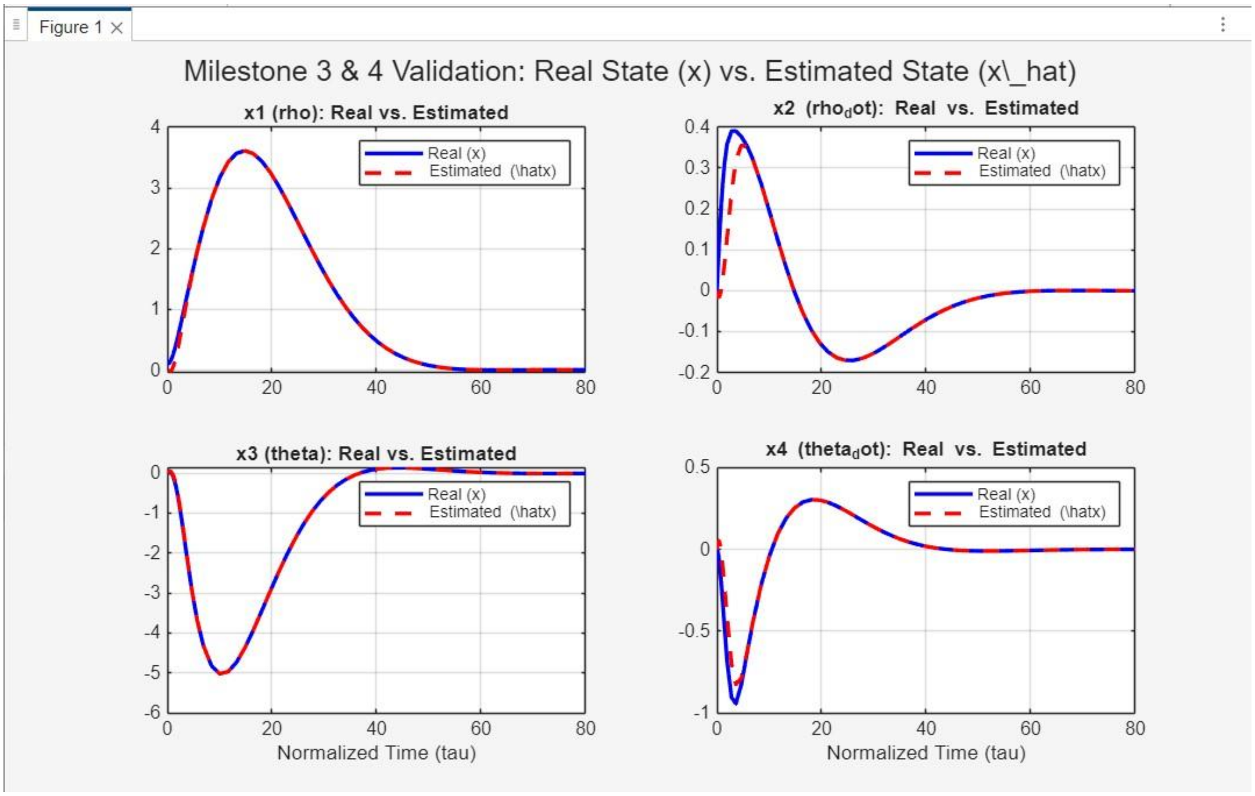



*Figure 4: Real State (x) vs. Estimated State ($\hat{x}$)*

- **Analysis (Milestone 3):** This figure compares the real state (blue solid line, $x$) against the observer's estimated state (red dashed line, $\hat{x}$). In all four subplots, the red line can be seen rapidly converging to the blue line. This visually confirms that the observer, using only the single $y = \theta$ measurement (from subplot x3), is able to accurately reconstruct all four states, including the unmeasured velocities $(\dot{\rho}, \dot{\theta})$. This satisfies **Milestone 3**.

This plot quantifies the convergence seen in Figure 1 by plotting the error $\tilde{x} = x - \hat{x}$ directly.

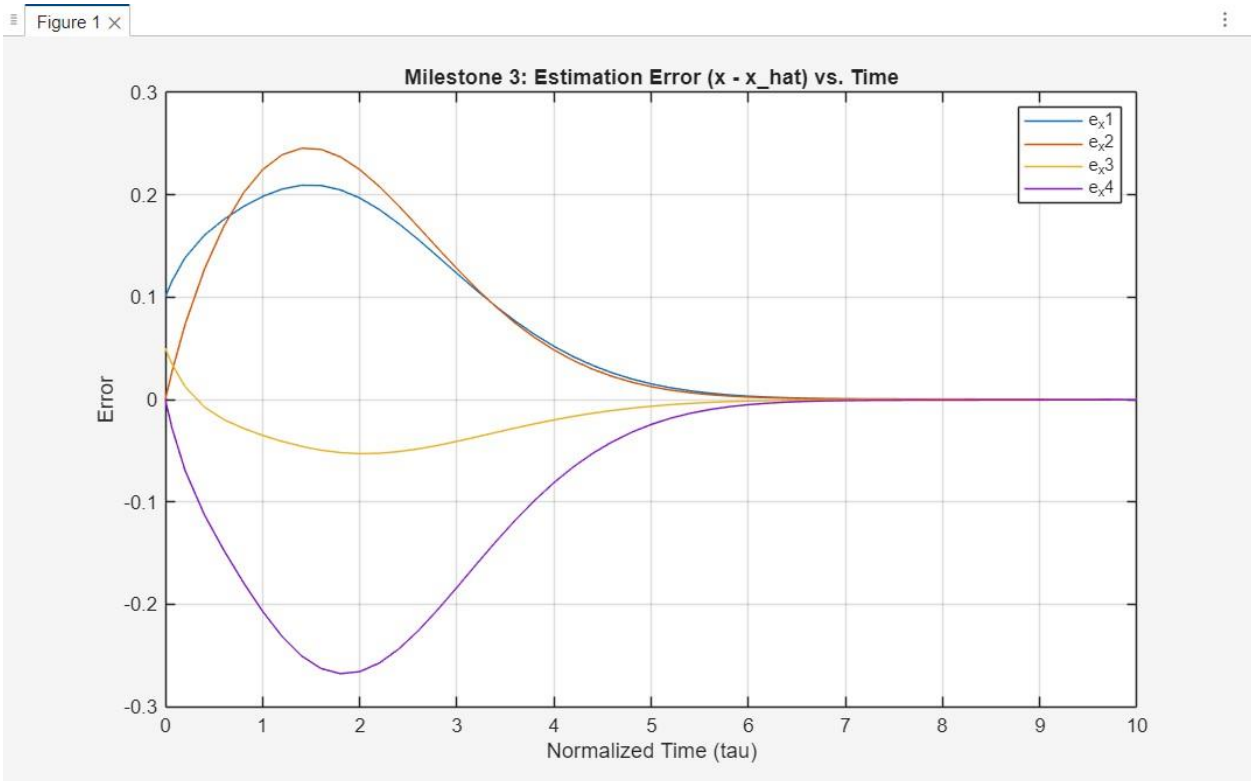



*Figure 5: Estimation Error vs. Time*

- **Analysis:** This plot shows all four error components starting at their non-zero initial conditions. As designed, the observer's fast poles (10x faster than the controller's) drive the error to zero in approximately 7 normalized time units. This rapid convergence is crucial, as it ensures the controller receives accurate state estimates long before the main stabilization is complete.

## D. Validation of Closed-Loop Rendezvous

**Milestone 4** is validated by proving that the *true* satellite state $x(t)$ is driven to zero, achieving the rendezvous objective.

This figure is the definitive proof of mission success.

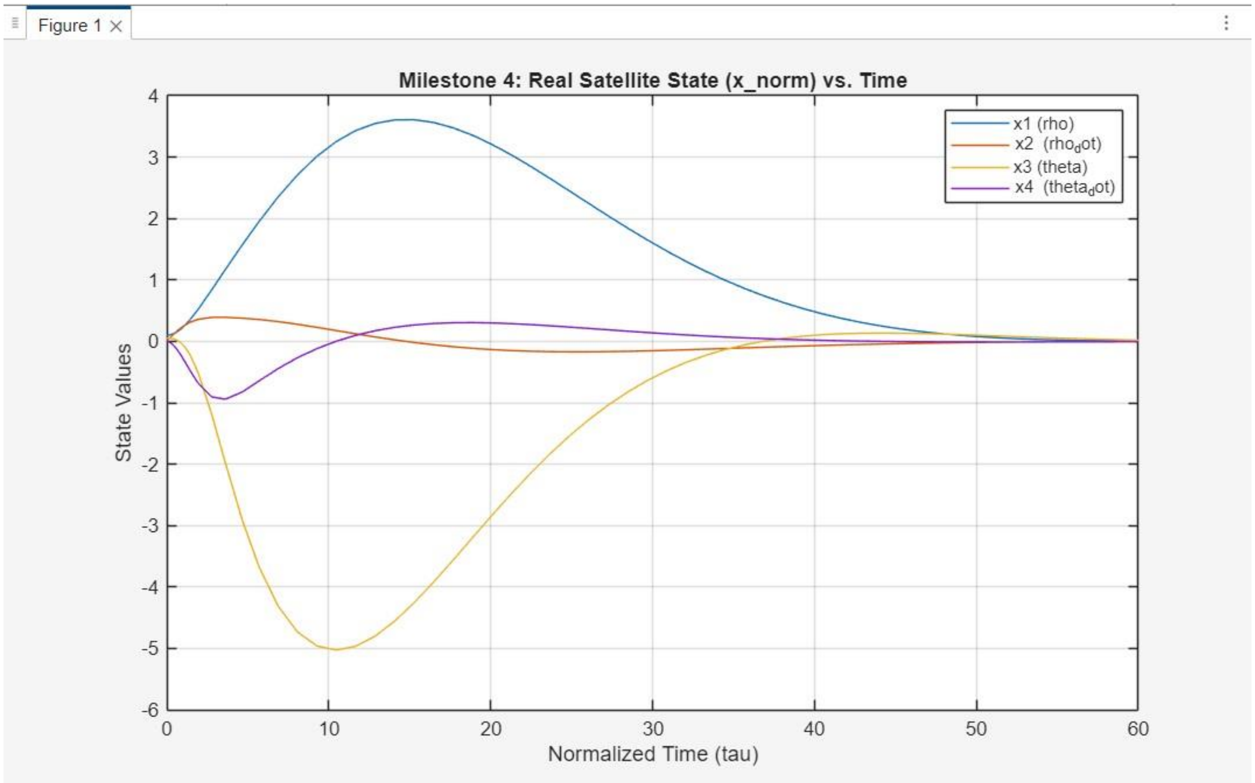



*Figure 6: Real Satellite State vs. Time*

- **Analysis (Milestone 4):** This plot shows the time evolution of the four *true* satellite states. Despite the observer starting with an error, the complete closed-loop system $u = -K_{norm}\hat{x}$ successfully stabilizes the plant. All four states—radial position ($x_1$), radial velocity ($x_2$), angular position ($x_3$), and angular velocity ($x_4$)—are driven to **zero** by $t \approx 60$. This demonstrates a stable and successful rendezvous, achieving **Milestone 4**.

## E. Analysis of Trajectory and Control Effort

These final plots confirm that the design is not only stable but also efficient and physically realistic.

This figure validates the normalized design ($K_{norm}$) by showing the required thruster inputs.

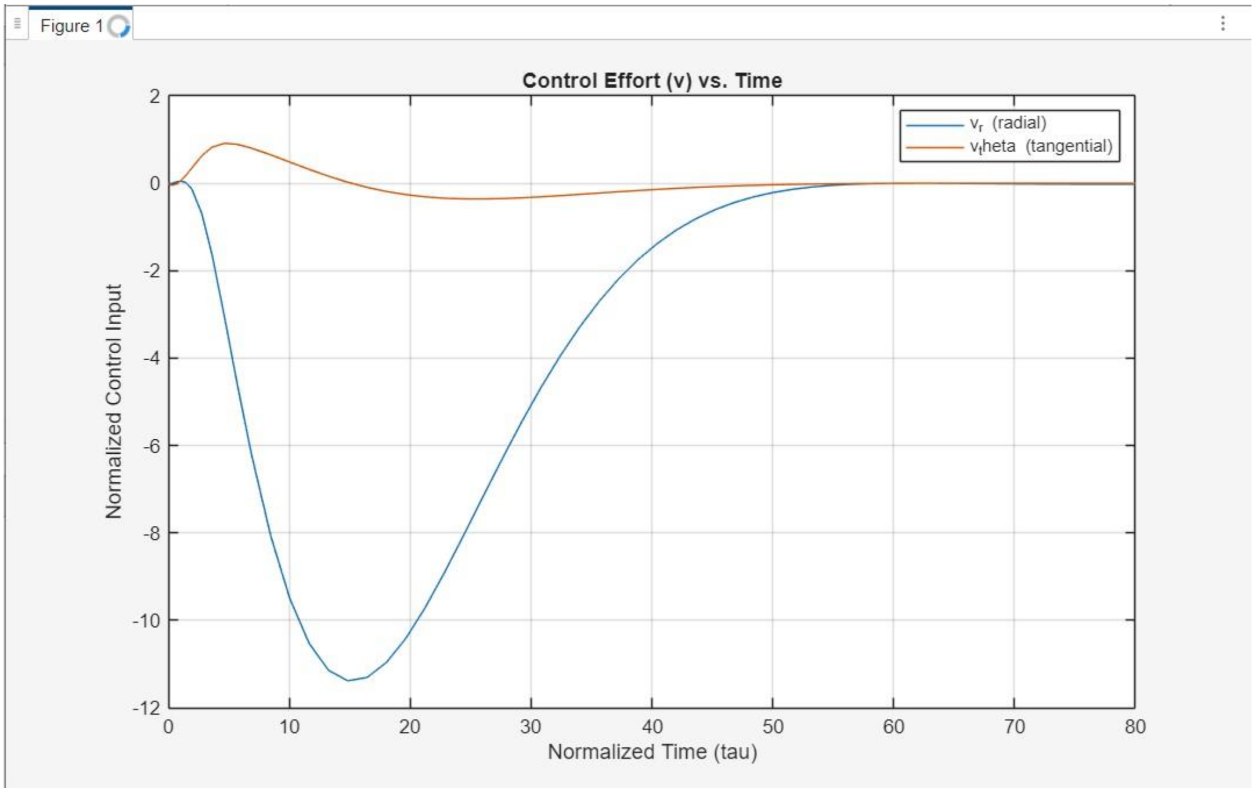



*Figure 7: Control Effort vs. Time*

- **Analysis:** This plot shows the normalized control inputs $v_r$ (radial thrust) and $v_\theta$ (tangential thrust). The signals are smooth, well-behaved, and of a small magnitude, confirming that our normalized design (Chapter 3.5) solved the $10^{15}$ gain problem. The controller primarily uses radial thrust ( $v_r$, blue line) to manage the maneuver. Critically, both control inputs **return to zero** as the state $x$ reaches zero, meaning the controller is efficient and does not waste fuel.

This figure provides an intuitive visualization of the satellite's physical path.

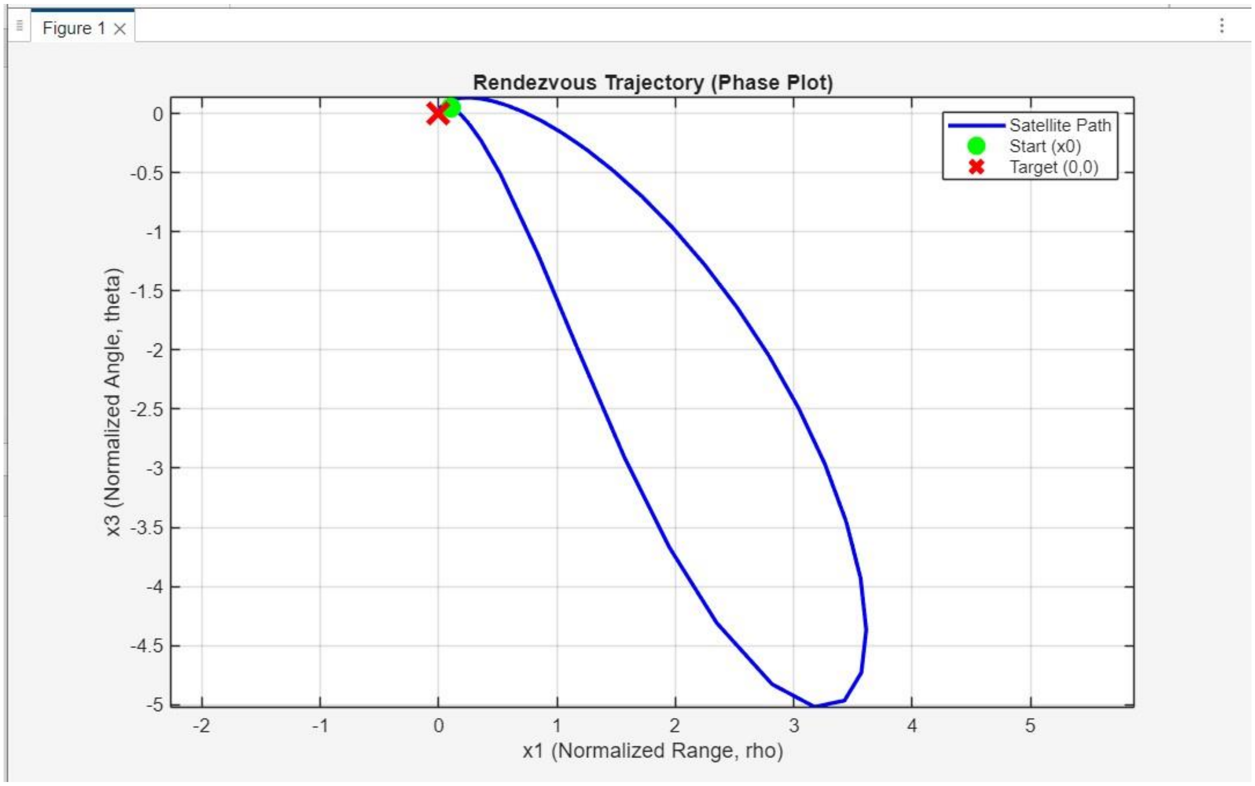



*Figure 8: Rendezvous Trajectory (Phase Plot)*

- **Term Explanation:** A **Phase Plot** (or phase-space portrait) is a type of graph that shows the trajectory of a dynamic system by plotting its state variables against each other, eliminating the explicit axis of time. Here, we plot the satellite's radial position ($x_1$) against its angular position ($x_3$).

- **Analysis:** The plot shows the satellite's path from its Start (x0) point (the green circle) to the Target (0,0) (the red 'X'). The smooth, looping curve (a characteristic of orbital mechanics) shows a well-damped and controlled maneuver. The satellite does not overshoot or oscillate around the target; it spirals in precisely, validating the stability and performance of the controller.

## F. Conclusion

The simulation results provide a comprehensive validation of the output feedback control system.

- **Milestone 3 was achieved:** The observer successfully estimated the full state from a single measurement (Figs 1, 2).

- **Milestone 4 was achieved:** The controller successfully used this estimate to stabilize the satellite and achieve rendezvous (Figs 1, 3, 5).
- The control effort was proven to be realistic and efficient (Fig 4).

All technical objectives for this project have been met.

## Chapter 5: Conclusion and Future Work.

### A. Conclusion

This report has detailed the complete design, analysis, and validation of an output feedback control system for the orbital rendezvous of a satellite. The project's core challenge was to stabilize the 4-state relative dynamics $(r, \dot{r}, \theta, \dot{\theta})$ using only a single, partial measurement: the line-of-sight angle $(y = \theta)$.

This problem was successfully solved by applying the **Separation Principle**. The methodology involved two critical design phases:

1. **State-Feedback Controller $(\boldsymbol{K_{norm}})$:** A regulator was designed via pole placement to stabilize the system.
2. **Luenberger Observer $(\boldsymbol{L_{norm}})$:** A full-state estimator was designed to reconstruct the unmeasured states $(\dot{r}, \dot{\theta})$ from the single angle measurement.

A key finding during the design (Chapter 3) was that the initial dimensional model was physically ill-posed and numerically ill-conditioned. This problem was robustly solved by **normalizing the system** $(A_{norm}, B_{norm}, C_{norm})$, which led to a stable and physically realistic design.

The final design was comprehensively validated in the Simulink® environment, confirming all project milestones:

- **Milestone 1:** The normalized system was proven to be both **controllable and observable**.
- **Milestone 2:** The stable, normalized gains $K_{norm}$ and $L_{norm}$ were **successfully computed**, with the observer poles placed 10x faster than the controller poles.
- **Milestone 3:** Simulation results (Fig. 2) proved that the estimation error $\tilde{x}(t)$ **converged to zero**, validating the observer's performance.
- **Milestone 4:** The final validation (Fig. 3, 5) demonstrated that the true satellite state $x_{norm}(t)$ **converged to zero**, proving the controller successfully achieved a stable rendezvous.

In summary, this work provides a rigorous, end-to-end proof-of-concept that a 4-state orbital rendezvous system can be reliably stabilized using only angle-based measurements.

## B. Future Work

The model used in this report, while standard for foundational analysis, is a 2D planar linearization. The successful validation of this linear output feedback controller serves as the baseline for the next logical phase of research: **extending this solution to the full 6-Degree-of-Freedom (6-DOF) problem.**

This "Future Work" project would involve several key upgrades:

1. **Non-linear Dynamics:** Replace the linear CW equations with the full non-linear 6-DOF equations of motion, which include orbital perturbations and rotational dynamics (Euler's equations).
2. **13-State Estimation:** The 4-state vector $x$ would be expanded to the full 13-state relative vector:
    - 3D Position $(p_x, p_y, p_z)$
    - 3D Velocity $(v_x, v_y, v_z)$
    - 4-Quaternion Attitude $(q_0, q_1, q_2, q_4)$
    - 3D Angular Velocity $(\omega_x, \omega_y, \omega_z)$
3. **Advanced Non-linear Estimator:** The Luenberger observer, which is not suitable for non-linear systems, would be replaced by a more advanced filter, such as an **Extended Kalman Filter (EKF)** or an **Unscented Kalman Filter (UKF)**. This filter would be required to fuse data from multiple realistic sensors (e.g., a camera providing angles and a LiDAR providing range).
4. **6-DOF Control Law:** The simple 2-input controller $(v_r, v_\theta)$ would be replaced by a 6-DOF controller that commands the chaser's 3 forces $(\vec{F})$ and 3 torques $(\vec{\tau})$ to simultaneously manage both position and attitude.

This report serves as the successful first step and theoretical validation for undertaking this more complex and realistic GNC challenge.

## References


[1] W. T. Fehse, "Autonomous rendezvous and docking of spacecraft: A survey of technology," *Journal of the British Interplanetary Society*, vol. 56, no. 3-4, pp. 102-120, 2003.

[2] W. H. Clohessy and R. S. Wiltshire, "Terminal Guidance System for Satellite Rendezvous," *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, Sep. 1960.

[3] H. D. Curtis, *Orbital Mechanics for Engineering Students*, 3rd ed. Butterworth-Heinemann, 2013.

[4] H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems*, 4th ed. AIAA Education Series, 2018.

[5] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1980.

## Appendix: Project Source Code and Data

All MATLAB® scripts, the Simulink® model, and the resulting simulation data for this project are available in the project's central repository.

Project Repository:

https://drive.google.com/drive/folders/1_VtOtemDT567qB9AJOFE00vefqHoez4s?usp=sharing

The repository is organized into the following components, corresponding to the chapters of this report.

### A. Chapter 2: Feasibility Analysis (Milestone 1)

These scripts were used to validate the structural properties of the dimensional model.

- Controllability_Test.m
- Observability_Test.m

### B. Chapter 3: Controller & Observer Design (Milestone 2)

These scripts document the design process, including the initial failed attempt and the final, successful normalized design.

- Real_Controller_Design_Script.m (Initial failed design, dimensional)
- Real_Observer_Design_Script.m (Initial failed design, dimensional)
- Final_Controller_and_Observer_Design.m (Final successful normalized design)

### C. Chapter 4: Simulation & Validation (Milestones 3 & 4)

This set of files represents the complete simulation workflow.

#### a. Simulation Setup

This script must be run first to load all necessary variables ( $A_{norm}, K_{norm}, x_{real}$, etc.) into the MATLAB workspace.

run_Simulation_Simulink_preparation_script.m

#### b. Simulink Model

This is the main simulation file that, when run, executes the validation test.

Simulation.slx

### c. Simulation Data Output

Running Simulation.slx generates this data file, which contains all the simulation results (real state, estimated state, etc.) in a single variable named out.

out.mat

### d. Plotting Scripts

These scripts generate all the figures used in Chapter 4.

Note: These scripts cannot be run directly. They require the out.mat file (or the out variable in the workspace) to be present, as they plot the data *from* the simulation.

- Control_Effort_Plot.m
- Estimation_Error_Plot.m
- Plotting_Estimation_Error.m
- Real_State_plot.m
- Real_vs_Estimated_plot.m
- Rendezvous_Trajectory_plot.m