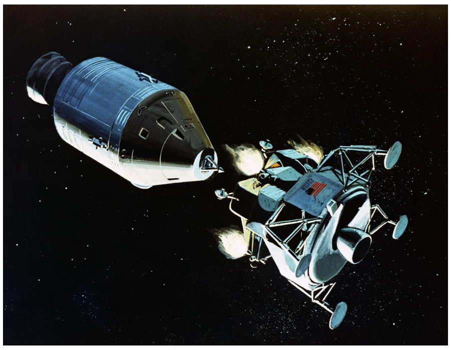

Guidance Navigation and Control System for a Rendezvous and Docking Mission

Guidance Report

Group Number 2

Joshua Redelbach	ist112470
William Bernholm	ist115600
Elena Francesca Cipriano	ist112514
Erica Vita	ist112267
João Almeida	ist103026
Gonçalo Carolino	ist103466

Report elaborated for the Curricular Unit

Guidance Navigation and Control

Supervisors: Prof. Alain de Souza

June 12th, 2025

Contents

Chapter 1

Introduction

In the context of autonomous spacecraft operations, particularly docking maneuvers, the effectiveness of a Guidance, Navigation, and Control (GNC) system is crucial. The previously completed work focused on the development and implementation of the *Navigation* and *Control* subsystems. This included sensor fusion strategies for state estimation implemented by a Kalman Filter and an LQR-based feedback controller that ensured stable tracking of target states. The current report extends that work by addressing the *Guidance* component, which is responsible for generating a feasible reference trajectory that the control system can track.

Guidance acts as the link between mission-level objectives and control-level execution. This phase of the project focuses on trajectory planning for the Rendezvous (Phase 2) and Docking (Phase 3) operations, both of which require high-precision translational motion under dynamic constraints. To achieve smooth and dynamically feasible reference paths, a fifth-order polynomial trajectory was implemented to define both position and linear velocity profiles over time. This approach ensures continuous and differentiable curves, which are essential for the LQR controller to perform accurate tracking.

This report describes the methodology for polynomial trajectory generation, the integration of these trajectories into the existing GNC framework, and the evaluation of their performance through simulation. The resulting guidance profiles provide a robust foundation for seamless transitions between phases and contribute significantly to the overall success of the autonomous docking mission.

Chapter 2

Methodology

This chapter presents the methodology used for generating reference trajectories and adjusting the controller for effective tracking performance. The goal is to ensure smooth transitions between mission phases and accurate convergence to target states during autonomous docking.

2.1 Trajectory Design

In order to enable smooth and dynamically feasible motion for the spacecraft, the reference trajectories for translational motion in Phase 2 (Rendezvous) and Phase 3 (Docking) were generated using fifth-order polynomial functions (also known as quintic polynomials). These trajectories define both position and velocity as a function of time in two dimensions. The docking manoeuvre mainly includes translational motion, and as the translational and rotational motion is decoupled, this work only focuses on guiding the translation of the spacecraft.

A fifth-order polynomial is well suited for this application as it allows to impose boundary conditions on the position, velocity, and acceleration at both the start and end of a trajectory. The general form of a fifth-order polynomial for a scalar variable $\mathrm{x}(t)$ is given by:

$$\mathrm{x}(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{2.1}$$

To determine the coefficients a_0 through a_5, a system of six equations is set up using the known initial and final values of position, velocity, and acceleration as well as the time stamp T of the final state:

$$\begin{aligned} \mathrm{x}(0) &= \mathrm{x}_0, \qquad & \dot{\mathrm{x}}(0) &= \mathrm{v}_0, \qquad & \ddot{\mathrm{x}}(0) &= \mathrm{a}_0 \\ \mathrm{x}(T) &= \mathrm{x}_f, \qquad & \dot{\mathrm{x}}(T) &= \mathrm{v}_f, \qquad & \ddot{\mathrm{x}}(T) &= \mathrm{a}_f \end{aligned} \tag{2.2}$$

These constraints guarantee smooth transitions that are physically realistic and well-suited for a spacecraft's propulsion and actuation capabilities. The same procedure is applied independently for both the x and y directions of motion.

The benefits of this method include:

- *Smoothness:* Continuous derivatives up to acceleration ensure actuator-friendly reference signals.

- *Feasibility:* Constraints are met without requiring trajectory replanning.
- *Predictability:* The entire trajectory is known ahead of time, which simplifies integration with the controller.

This approach avoids the complications of real-time replanning while providing high confidence in the trajectory's dynamic feasibility.

2.2 Controller Tuning

The original controller implemented for the spacecraft was designed to track static reference points. However, with the introduction of time-varying reference trajectories, the controller had to be adjusted to maintain optimal tracking performance.

A Linear Quadratic Regulator (LQR) was used as the main control strategy. The control law is given by:

$$\mathbf{u}(t) = -\mathbf{K}\left(\mathbf{x}(t) - \mathbf{x}_{\text{ref}}(t)\right) \tag{2.3}$$

where $\mathbf{x}(t)$ is the current state, $\mathbf{x}_{\text{ref}}(t)$ is the time-dependent reference trajectory, and K is the optimal gain matrix obtained by solving the algebraic Riccati equation for a given pair of weighting matrices $\mathbf{Q}$ and $\mathbf{R}$.

With time-varying references, the error signal becomes dynamic, and the magnitude and frequency of state deviations vary compared to static-point control. Therefore, the matrices $\mathbf{Q}$ and $\mathbf{R}$ must be re-tuned to:

- Penalize deviations from the moving reference trajectory appropriately.
- Ensure smooth control effort and prevent too aggressive as well as too slow corrections.
- Maintain a trade-off between response speed and control energy.

Note that, special attention is given to the velocity tracking component of the reference, as poor velocity matching may result in oscillatory or unstable behaviour during docking. The tuning was carried out through iterative simulation, balancing position accuracy with actuator limitations.

Chapter 3

Implementation

3.1 Trajectory Creation

To support the guidance component of the GNC system, time-dependent reference trajectories were created offline in advance for both Phase 2 (Rendezvous) and Phase 3 (Docking). These trajectories were generated using a self-developed *Python* implementation that computes fifth-order polynomial profiles for 2D translational motion. The goal was to define smooth and dynamically feasible position and velocity profiles over a specified time horizon.

The underlying method solves a set of boundary conditions for each axis (x and y), ensuring that initial and final positions, velocities, and accelerations are met. The use of quintic polynomials provides continuous second-order derivatives, which is particularly beneficial for actuator efficiency and control stability.

Phase 2 trajectory transitions the spacecraft from its initial position to a pre-docking configuration near the target. Phase 3 trajectory performs the final approach and alignment with the docking port. A key design decision in this updated guidance approach is that the spacecraft no longer comes to a full stop at the end of Phase 2. Instead, a small, non-zero velocity component in the y direction is intentionally preserved. This modification reflects a shift in strategy: the aim is now to achieve a smooth gliding transition between phases, rather than decelerating to zero and re-accelerating in the next phase.

This choice offers several advantages that complement the tuning of the **Q** and **R** matrices:

- It reduces the total manoeuvrer time by eliminating unnecessary stop-and-go behaviour.
- It decreases control effort and propellant usage by maintaining continuous motion.
- It ensures that the velocity profile remains compatible with the acceleration limits of Phase 3.

The trajectory generation process is fully deterministic and avoids the need for online optimization during simulation, which simplifies integration and reduces computational load. To test the robustness of the guidance system, the worst-case initial state scenario of the previous work was assumed for simulation:

$$\mathbf{x}_0 = \begin{bmatrix} -20\,\mathrm{m} & -220\,\mathrm{m} & 0.1\,\mathrm{m\,s^{-1}} & 0.1\,\mathrm{m\,s^{-1}} & 180\,\mathrm{deg} & 0.5\,\mathrm{deg/s} \end{bmatrix} \tag{3.1}$$

The three reference states that trigger the switch of control phases are given as:

$$\mathbf{x}_{\text{ref},1} = \begin{bmatrix} - & - & - & - & 0\,\text{deg} & 0\,\text{deg/s} \end{bmatrix} \tag{3.2}$$

$$\mathbf{x}_{\text{ref},2} = \begin{bmatrix} 0\,\text{m} & -10\,\text{m} & 0\,\text{m}\,\text{s}^{-1} & 0.5\,\text{m}\,\text{s}^{-1} & 0\,\text{deg} & 0\,\text{deg/s} \end{bmatrix} \tag{3.3}$$

$$\mathbf{x}_{\text{ref},3} = \begin{bmatrix} 0\,\text{m} & 0\,\text{m} & 0\,\text{m}\,\text{s}^{-1} & 0\,\text{m}\,\text{s}^{-1} & 0\,\text{deg} & 0\,\text{deg/s} \end{bmatrix} \tag{3.4}$$

In order to ensure feasibility and control efficiency, the total trajectory duration T for both phases is iteratively tuned. The tuning process is guided by the following criteria:

- No overshoot in position trajectories
- Velocity remains within acceptable limits for tracking
- Acceleration remains bounded to avoid excessive control effort

The duration T for each trajectory is finally set to $T_{\text{Phase1}} = 500\,\text{s}$ and $T_{\text{Phase2}} = 40\,\text{s}$. The resulting trajectories for each phase are shown in Fig. 3.1 and Fig. 3.2. Each figure contains the time evolution of position, velocity, and acceleration in both translational axes. These plots verify that the planned motion profiles remain smooth and within specification throughout the manoeuvrer.

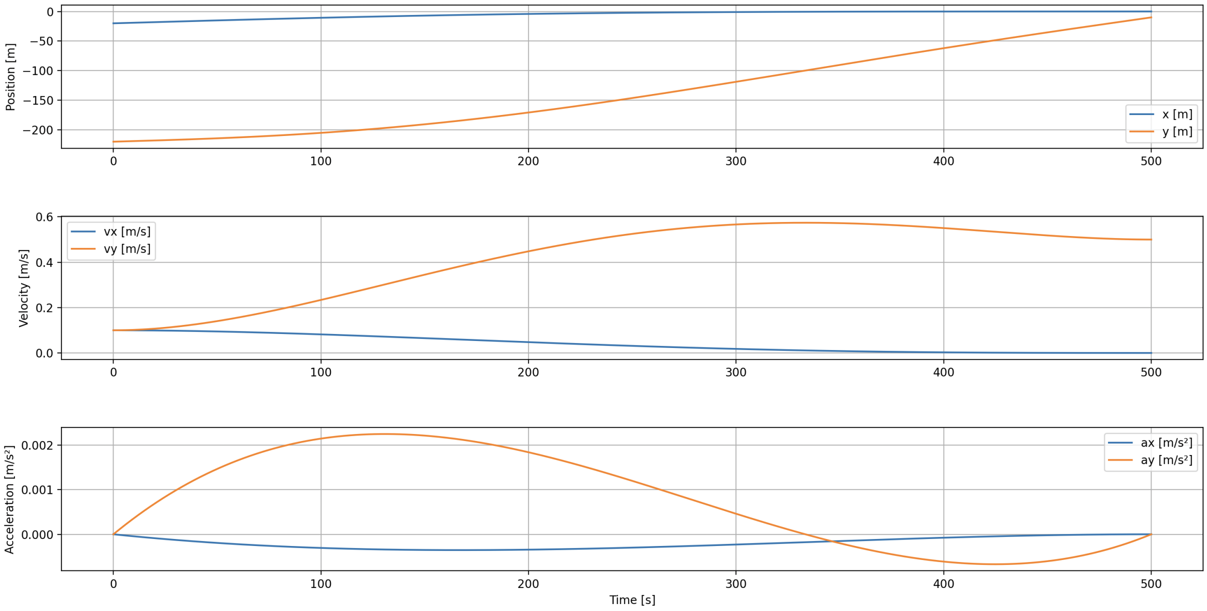

Figure 3.1: Position, velocity, and acceleration profiles for Phase 2 trajectory.

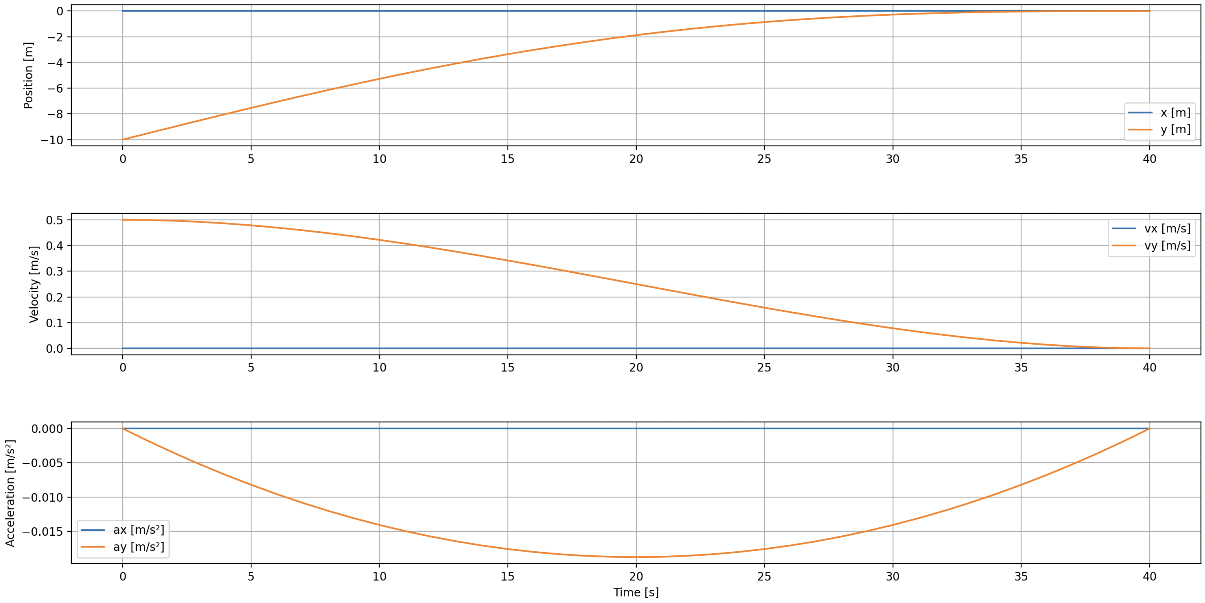

Figure 3.2: Position, velocity, and acceleration profiles for Phase 3 trajectory.

3.2 Integration in Simulation

Creating the trajectory results in two *csv*-files consisting of the position, velocity and acceleration in x and y direction for each time step. The step size for the time intervals is equal to one second. For testing the integration of the guidance trajectory, the same *Matlab Simulink* simulation is used as described in the previous work but the *Guidance* block is updated (see Fig. 3.3).

The two trajectories are loaded and stored in the beginning of the simulation. The block *phase_switch* determines whether the current reference state $\mathbf{x}_{\text{ref},n}$ is successfully reached and, if so, triggers the phase switch. In difference to the previous implementation, different tolerances ϵ are defined for the position and velocity for

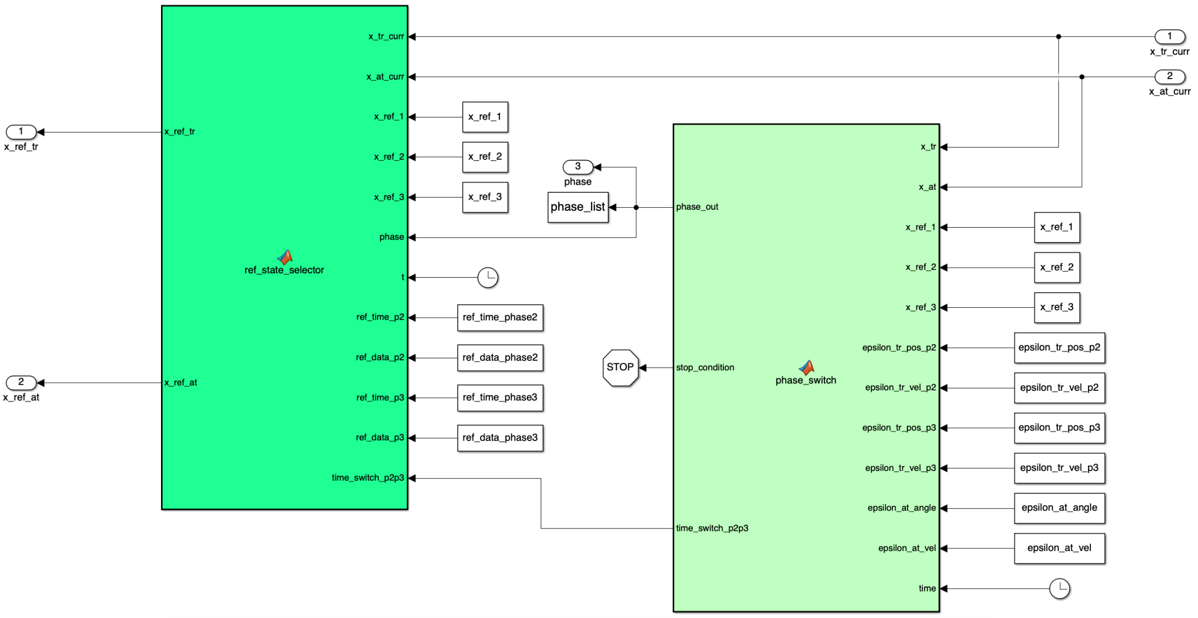

Figure 3.3: Guidance block of the *Simulink* simulation.

Phase	Tolerance Attitude $[\epsilon_\theta, \epsilon_{\dot\theta}]$	Tolerance Position $[\epsilon_x, \epsilon_y, \epsilon_{\dot x}, \epsilon_{\dot y}]$	$\mathbf{x}_{ref}$
1	$[4, 0.2]$	$[-, -, -, -]$	$[-, -, -, -, 0, 0]$
2	$[4, 0.2]$	$[0.25, 0.25, 0.1, 0.5]$	$[0, -10, 0, 0.5, 0, 0]$
3	$[4, 0.2]$	$[0.1, 0.01, 0.04, 0.05]$	$[0, 0, 0, 0, 0, 0]$

Table 3.1: List of the tolerances and reference states of the guidance block for each phase of the mission. The units are not added for better readability. Positions are to be interpreted in $[\mathrm{m}]$, angles in $[\mathrm{deg}]$, linear velocities as $[\mathrm{m\,s^{-1}}]$ and angular velocities as $[\mathrm{deg/s}]$.

Phase 2 and 3. They are given in Tab. 3.1. To ensure robust phase transitions and avoid unnecessary control effort, larger tolerances are applied around $\mathbf{x}_{\text{ref},2}$, where precise convergence is less critical.

The block *ref_state_selector* takes the current time t since the start of the manoeuvrer and, depending on the current phase, the reference for position and velocity $\mathbf{x}_{\text{ref,trans}}$ is determined. It is computed by interpolating the stored, discretized trajectory around the current time t. As the guidance only focuses on the translational motion, as reasoned in 2.1, for the attitude and angular velocity, the respective components of the current $\mathbf{x}_{\text{ref},n}$ are selected and assigned to $\mathbf{x}_{\text{ref,rot}}$. Those two reference vectors are then fed to the controller.

Chapter 4

Results

The worst-case docking scenario is simulated including the adapted guidance system. As mentioned in Sec. 2.2, the $\mathbf{Q}$ and $\mathbf{R}$ matrices for the translational motion controller have to be retuned. This is done by an iterative trial-and-error approach. The final chosen values are summarized in Tab. 4.1.

The values of the $\mathbf{Q}$ and $\mathbf{R}$ matrices in the translational controller are selected to balance accuracy and control effort. Higher values in $\mathbf{Q}$ give more importance to reducing position and velocity errors, while higher values in $\mathbf{R}$ penalize the use of control inputs. The tuning is adapted for different mission phases:

- *Phase 2 (Rendezvous):* High $\mathbf{Q}$ values ensure accurate tracking of position and velocity. Moderate $\mathbf{R}$ values help limit excessive control effort while maintaining good performance.

- *Phase 3 (Final docking):* $\mathbf{Q}$ values remain high, but in one axis a lower weight is used to reflect reduced priority or less sensitivity. The $\mathbf{R}$ matrix values are reduced to allow more responsive and precise control during this critical phase.

These changes reflect the need for more precise manoeuvring as the spacecraft gets closer to the target, while still managing actuator effort and system stability.

In order to analyse the performance of the trajectory tracking and if the computed trajectory benefits to the performance of the docking mission, several metrics and plots are presented in the following.

Fig. 4.1 shows the time evolution of the system's state variables and control inputs throughout the docking maneuver. Tab 4.2 presents the final state values at the docking instant, corresponding to the end of Phase 3, and compares them to the results obtained prior to the implementation of the guidance system. Additionally, the table includes information on the time required to complete each phase, the total docking time, and the

Controller	**Q Matrix**	**$\mathbf{R}$ Matrix**
Attitude 1 $(att, 1)$	$\mathrm{diag}(1,\ 1)$	10
Position 2 $(pos, 2)$	$\mathrm{diag}(1000,\ 1000,\ 1000,\ 1000)$	$\mathrm{diag}(1,\ 1)$
Attitude 2 $(att, 2)$	$\mathrm{diag}(10,\ 10)$	1
Position 3 $(pos, 3)$	$\mathrm{diag}(1000,\ 1000,\ 1,\ 1000)$	$\mathrm{diag}(0.1,\ 0.001)$
Attitude 3 $(att, 3)$	$\mathrm{diag}(10,\ 10)$	1

Table 4.1: Final tuned values of $\mathbf{Q}$ and $\mathbf{R}$ matrices for the controller.

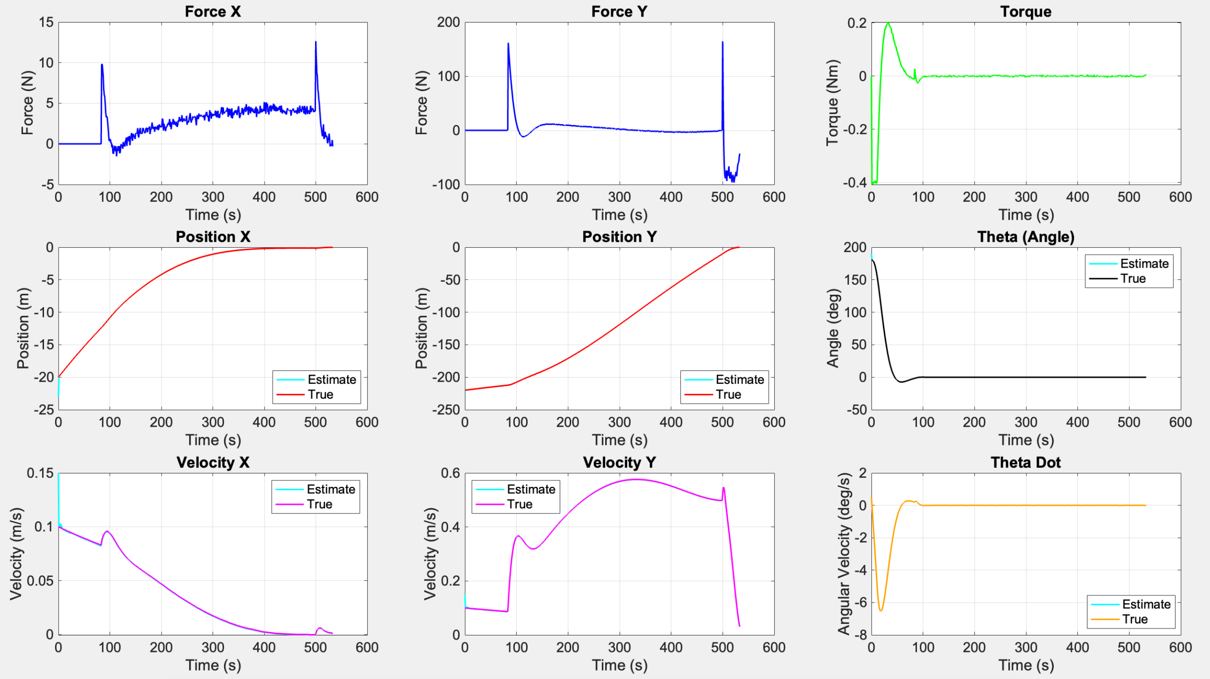

Figure 4.1: Plot of the input and state variables over time.

corresponding expended Δv values. These Δv values are computed by integrating the absolute value of the specific thrust over time for each manoeuvrer.

Subsequently, Tab. 4.3 presents the root mean square error between the actual and reference trajectories for each state variable. In relation to this, the evolution of the position and velocity errors over time is illustrated in Fig. 4.2 and Fig. 4.3.

Finally, Fig. 4.4, Fig. 4.5, and Fig. 4.6 show the true and reference trajectories throughout the entire docking maneuver, as well as detailed views of the initial and final approach segments. It is also noted that the state estimates provided by the Kalman filter are included in the plots.

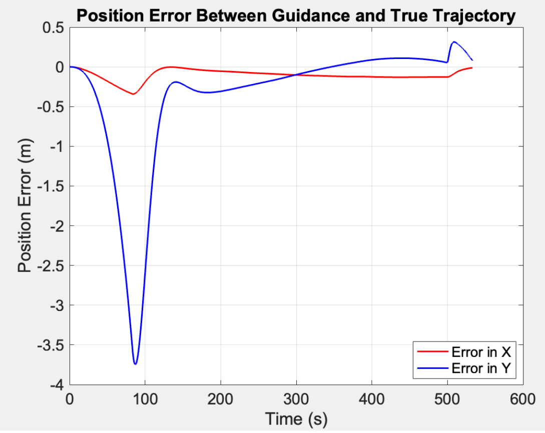

Figure 4.2: Position error over time between true and desired trajectory.

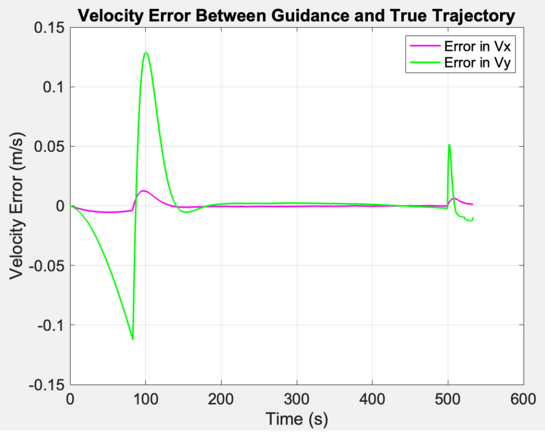

Figure 4.3: Velocity error over time between true and desired trajectory.

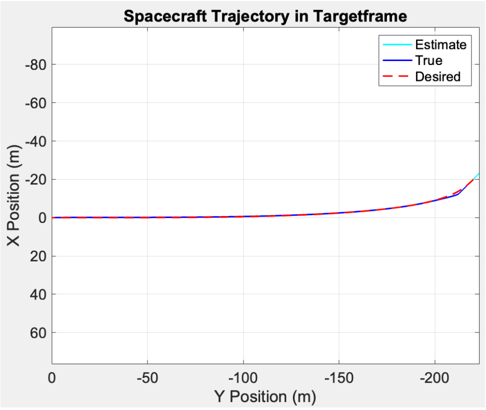

Figure 4.4: Trajectory plot - full.

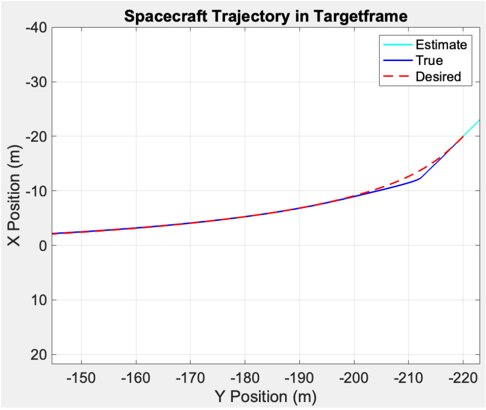

Figure 4.5: Trajectory plot - zoom begin.

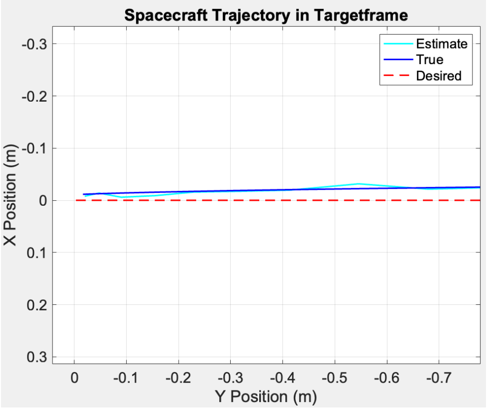

Figure 4.6: Trajectory plot - zoom end.

Metric	**With Guidance**	**Without Guidance**
x_{final} $[\text{m}]$	-0.0157	0.0072
y_{final} $[\text{m}]$	-0.0099	-0.0091
$\dot{x}_{\text{final}}$ $[\text{m}\,\text{s}^{-1}]$	0.0011	-0.0001
$\dot{y}_{\text{final}}$ $[\text{m}\,\text{s}^{-1}]$	0.0286	0.0002
θ_{final} $[\text{deg}]$	0.0175	0.0245
$\dot{\theta}_{\text{final}}$ $[\text{deg/s}]$	-0.0030	0.0007
Reference 1 reached after $[\text{s}]$	83.26	83.26
Reference 2 reached after $[\text{s}]$	499.41	2001.80
Docking completed after $[\text{s}]$	533.20	2626.20
Δv in x $[\text{m}\,\text{s}^{-1}]$	0.3045	0.4100
Δv in y $[\text{m}\,\text{s}^{-1}]$	1.2473	1.0027
Total Δv $[\text{m}\,\text{s}^{-1}]$	1.2839	1.0833

Table 4.2: Comparison of the docking: with vs. without guidance.

Metric	**RMSE**
Trajectory X Position $[\text{m}]$	0.1247
Trajectory Y Position $[\text{m}]$	0.8945
Trajectory X Velocity $[\text{m}\,\text{s}^{-1}]$	0.0032
Trajectory Y Velocity $[\text{m}\,\text{s}^{-1}]$	0.0340

Table 4.3: Tracking RMSE results.

Chapter 5

Discussion

The results presented in Fig. 4.1 for the state variables and control inputs show the expected behaviour of the system, confirming that the controller is able to successfully track the reference trajectories generated by the guidance module. However, from a closer analysis of the time evolution of the system, an interesting observation emerges which directly affects both the controller behaviour and the actuator activity. Although the duration assigned for Phase 2, where the guidance trajectory is applied, was set to $500\,\mathrm{s}$, the simulation shows a small timing deviation of approximately $0.5\,\mathrm{s}$. This discrepancy arises from the fact that, due to imperfect trajectory tracking, the system satisfies the switching tolerances slightly earlier than expected, leading to a premature transition into Phase 3. As a consequence, the initial reference state for Phase 3 was effectively shifted forward in time.

Once this behaviour was observed, the corresponding initial reference for Phase 3 was manually adjusted to account for the earlier transition. This manual correction was introduced in order to prevent unnecessary actuator actions that could result from a mismatch between the expected and actual starting conditions of Phase 3. However, a more systematic way to avoid such adjustments would be to generate a single continuous polynomial trajectory that covers both Phase 2 and Phase 3 simultaneously. In this way, the full guidance profile would be smooth and time-consistent, inherently removing phase boundary issues and further reducing unnecessary control activity.

This behaviour can also be observed in the curve representing the force applied by the thrusters in the y direction, as shown in Fig. 4.1. Specifically, the curve exhibits two distinct peaks. The first peak is expected, as it corresponds to the initial thrust applied at the beginning of Phase 2. The second peak, however, is not anticipated and results from the same mechanism described earlier, this time involving both position and velocity deviations. While the timespan for Phase 2 was fixed, the presence of tolerances for the states introduced a mismatch between the expected state at the moment of phase switching and the actual state achieved by the system. This discrepancy leads to unnecessary activations of the actuators in both positive and negative y directions, as the controller attempts to correct the state in order to meet the switching conditions. A similar, though less pronounced, effect can also be observed in the x direction, where the magnitude of these corrective actions is smaller.

Another discrepancy that could be observed was the significant error in position and velocity around $t = 100\,\mathrm{s}$, as shown in Fig. 4.2 and Fig. 4.3, can be attributed to the system configuration during Phase 1. In this phase, only attitude correction is performed, while the chaser starts with a small initial velocity. The attitude correction takes approximately $100\,\mathrm{s}$ in this scenario, during which the spacecraft drifts, leading to a positional offset. Additionally, due to the dynamics described by the 2D Hill's equations, the relative velocity changes even without active position control. As a result, when Phase 2 begins, the system state differs from the intended reference trajectory, prompting the controller to apply corrective inputs. This deviation is also visible in Fig. 4.5, where the actual trajectory diverges from the desired path. This issue could be mitigated by implementing a velocity correction phase before the attitude control, ensuring the correct initial state for the start of Phase 2.

Nevertheless, the inclusion of the guidance resulted in a nearly five-fold decrease in operating time. As seen in Tab. 4.2, the manoeuvrer was completed in $533.2\,\mathrm{s}$, instead of $2600\,\mathrm{s}$ without guidance. This was achieved with an 18% increase of Δv, from $1.0833\,\mathrm{m\,s^{-1}}$ to $1.2839\,\mathrm{m\,s^{-1}}$. This is made possible by the smooth, precomputed fifth-order polynomial trajectories, which eliminate the unnecessary stop-and-go behaviour and allow for continuous motion between mission phases. The system eliminates excessive corrective actions while maintaining a precise path, demonstrating that effective guidance can enhance performance while maintaining resource efficiency.

Chapter 6

Outlook

While the implemented guidance subsystem successfully generates dynamically feasible trajectories that allow the spacecraft to perform precise rendezvous and docking, several potential improvements can be identified for future work. These enhancements could further increase the performance, robustness, and realism of the guidance strategy.

First, although the current fifth-order polynomial formulation already produces minimum jerk trajectories by satisfying boundary conditions on position, velocity, and acceleration, the total trajectory duration remains a manually tuned parameter. Future work could explore optimizing the trajectory duration to balance between fuel efficiency, docking time, and control effort, while still preserving the minimum jerk property.

Secondly, while the current LQR controller demonstrates satisfactory tracking performance for the generated trajectories, several enhancements could be considered to improve both precision and overall performance. On one hand, more advanced tracking control methods could be introduced, such as the integration of feedforward terms that directly compensate for the known derivatives of the reference trajectory (particularly accelerations), or the implementation of time-varying LQR (TVLQR) and Model Predictive Control (MPC) approaches, which can better handle time-varying references and system constraints. On the other hand, the tuning of the LQR weighting matrices was carried out manually through iterative simulations and empirical adjustments. Future work could explore the use of automated tuning algorithms, such as genetic algorithms, particle swarm optimization, or Bayesian optimization, to systematically search for optimal $\mathbf{Q}$ and $\mathbf{R}$ matrices. Such methods would allow for objective minimization of performance criteria like tracking error, control effort, docking time, or fuel consumption, while ensuring compliance with actuator limitations and mission constraints. The adoption of these advanced control design and tuning strategies could further enhance the robustness, efficiency, and reproducibility of the closed-loop system.

Another important extension would be the development of online trajectory replanning capabilities. Currently, the trajectory is computed offline and loaded into the simulation prior to execution. Embedding the trajectory generation directly into the Simulink model would allow the system to generate new trajectories at runtime, reacting to deviations from the nominal path due to unforeseen disturbances, sensor errors, or estimation inaccuracies. This capability would enhance system autonomy and robustness during the docking phase.

Moreover, future guidance improvements could include the implementation of constraint-aware planning, allowing the spacecraft to dynamically avoid potential keep-out zones or respect safety corridors around the target spacecraft. This would be particularly relevant for more realistic operational scenarios and for cooperative or multi-agent docking situations.

Finally, additional extensions may involve the inclusion of energy-optimal profiles, such as bang-coast-bang or Lambert-based solutions, which could provide globally optimal solutions for propellant usage while maintaining safe and feasible docking approaches.

These extensions would not only enhance the efficiency of the mission but also bring the guidance system closer to the complexity and capability required for operational autonomous space docking missions.

Chapter 7

Conclusion

This part of the project successfully establishes a robust framework for the GNC system necessary for close-range autonomous docking operations. The implemented system demonstrated reliable tracking of time-varying trajectories using an LQR controller, with performance verified through simulations under challenging initial conditions. The system was tested under both simple and more realistic starting conditions, showing that the guidance and control worked well and stayed within the required accuracy limits.

The use of fifth-order polynomial trajectories ensured smooth motion profiles that reduced actuator strain and avoided unnecessary stop-go behavior. Although the controller did not explicitly optimize for fuel consumption, gain tuning minimized excessive maneuvers and kept control effort within realistic bounds. The total Δv remained reasonable, supporting the practical feasibility of the system.

However, certain limitations remain. No external disturbances were modeled, meaning the system's robustness under real-world uncertainties (e.g., unmodeled forces or sensor noise) has yet to be fully validated. In addition, trajectory planning focused solely on feasibility and smoothness, not fuel efficiency. These aspects represent key areas for future improvement.

Despite these constraints, the current solution provides a solid foundation for further development. The insights and architecture developed here pave the way for extending the GNC system to full orbital rendezvous scenarios, with future efforts focused on integrating disturbance modeling, fuel-efficient guidance, and potentially online trajectory replanning for greater autonomy and resilience.