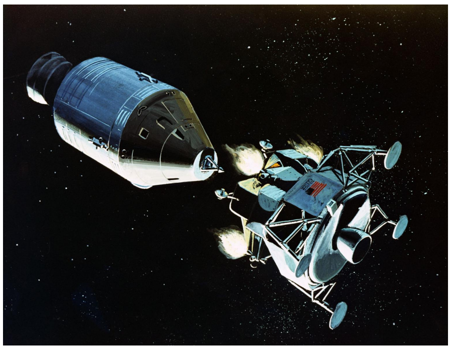
TÉCNICO
LISBOA



Guidance Navigation and Control System for a Rendezvous and Docking Mission

Control and Navigation 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

May 30th, 2025

Contents

Chapter 1

Introduction

This report presents the design, implementation and validation of the *Control* and *Navigation* subsystems of the Guidance, Navigation, and Control (GNC) system for an autonomous rendezvous and docking (RVD) manoeuvrer between two spacecrafts. The mission scenario and GNC architecture are initially defined in the Design Report, where a simplified 2D in-orbit docking problem is formulated. The present document builds on that foundation by addressing the design, implementation, and performance evaluation of the control algorithms and the state estimation framework.

The scope of this report is limited to the close-range RVD manoeuvrer, assuming that prior orbital manoeuvrers have already placed the chaser spacecraft in proximity to the target. The Guidance component, responsible for generating high-level reference trajectories and mode switching, is assumed to be predefined and is not the focus of this but of future work. Instead, this phase concentrates on the lower-level control loops and navigation algorithms required to ensure accurate trajectory tracking and safe docking.

For control, Linear Quadratic Regulator (LQR) methods are implemented separately for translational and rotational dynamics. The manoeuvrer is divided into three operational phases: attitude alignment, approach, and final docking. Each phase uses independently tuned LQR gains to meet phase-specific objectives such as minimizing attitude error, avoiding overshoot, and adhering to velocity constraints.

The navigation subsystem employs a Kalman Filter to estimate the full system state, including position, velocity, attitude, and angular velocity, by fusing noisy measurements from a simulated radar, gyroscope, and star tracker. The estimator is designed to operate in a realistic asynchronous sensor environment and aims to deliver robust performance despite initial estimation errors and measurement noise.

The validation of the system is performed through simulation in MATLAB Simulink. Two scenarios are analysed in detail: a best-case scenario with near-nominal initial conditions and a worst-case scenario involving large deviations in position and attitude. Additionally, a Monte Carlo simulation is performed testing 500 randomly created initial conditions within the best-case and worst-case scenario in order to confirm the robustness of the controller. Performance is evaluated based on state convergence, control effort, constraint adherence, and estimation accuracy.

The structure of the report is as follows: Chapter 2 outlines the theoretical methodology for the control and estimation design. Chapter 3 describes the software implementation. Chapter 4 presents the simulation scenarios and results. Chapter 5 discusses the findings, and Chapter 6 concludes the report with final observations and future recommendations.

Chapter 2

Methodology

2.1 State Space Model

2.1.1 System Model

As described in the design report, the following state space model is used to represent the system of the docking scenario:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{2.1}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{2.2}$$

with

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 3n^2 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -2n & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m_c} & 0 & 0 \\ 0 & \frac{1}{m_c} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_c} \end{bmatrix} \tag{2.3}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.4}$$

and:

$$\mathbf{u}(t) = \begin{bmatrix} F_x(t) \\ F_y(t) \\ \tau(t) \end{bmatrix} \tag{2.5}$$

Because it was seen in the design that the translational and rotational dynamics are not coupled, two independent controllers are developed in later sections. For this, the state space model of the entire system is split into a position and attitude part in the following.

2.1.2 Attitude Model

The state vector composes of the angle and the angular velocity

$$\mathbf{x}_{at} = \begin{bmatrix} \theta \; \dot{\theta} \end{bmatrix}^T, \quad \dot{\mathbf{x}}_{at} = \begin{bmatrix} \dot{\theta} \; \ddot{\theta} \end{bmatrix}^T \tag{2.6}$$

leading to the following state space matrices:

$$\mathbf{A}_{at} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B}_{at} = \begin{bmatrix} 0 \\ \frac{1}{I_c} \end{bmatrix} \tag{2.7}$$

$$\mathbf{C}_{at} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{D}_{at} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.8}$$

and:

$$\mathbf{u}_{at}(t) = \tau(t) \tag{2.9}$$

2.1.3 Position Model

The state vector composes of the position and velocity in both directions

$$\mathbf{x}_{pos} = \begin{bmatrix} x \; y \; \dot{x} \; \dot{y} \end{bmatrix}^T, \quad \dot{\mathbf{x}}_{pos} = \begin{bmatrix} \dot{x} \; \dot{y} \; \ddot{x} \; \ddot{y} \end{bmatrix}^T \tag{2.10}$$

leading to the following state space matrices:

$$\mathbf{A}_{pos} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & -2n \\ 0 & 0 & -2n & 0 \end{bmatrix}, \quad \mathbf{B}_{pos} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m_c} & 0 \\ 0 & \frac{1}{m_c} \end{bmatrix} \tag{2.11}$$

$$\mathbf{C}_{pos} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{D}_{pos} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{2.12}$$

and:

$$\mathbf{u}_{pos}(t) = \begin{bmatrix} F_x(t) \\ F_y(t) \end{bmatrix} \tag{2.13}$$

2.2 Controller

2.2.1 Control Phases

For this work, the rendezvous and docking manoeuvrer is constrained to the close-approach phase which is divided into three control phases. These are described in detail in the following after the initial state is explained.

Initial State

Within this project, the negative v-bar approach for docking the spacecrafts is simulated meaning that the chaser approaches the target spacecraft from behind. Thus, it is assumed that in advance the chaser performs orbital manoeuvres in order to arrive behind the target approximately in the same orbit. Typically, after arriving in the same orbit as the target, the chaser performs manoeuvres in so called far-approaching phases to get closer to the target in a safe manner. These are considered to be performed successfully in advance which typically bring the chaser to an initial position of $p_0 = [0\,\mathrm{m}, -200\,\mathrm{m}]$ [1–4]. It is assumed that the previous control phases achieve this position with an error of $\leq 20\,\mathrm{m}$. The initial attitude mainly depends on the geometry of the spacecraft and where the docking port is placed at the chaser. Therefore, in the worst case the chaser has to be rotated by $180\,\mathrm{deg}$ to align the docking port towards the target. Furthermore, the chaser is assumed to be approximately stationary at that position meaning that all initial translational and rotational velocity components are zero up to an error of $\leq 0.1\,\mathrm{m\,s^{-1}}$ and $\leq 0.5\,\mathrm{deg/s}$. In Fig. 2.1, the initial state is illustrated.

Docking Alignment Phase

The close-approach phase of the rendezvous and docking phase of this work starts with the alignment of the docking port meaning that an initial attitude offset θ_0 is driven to a reference stationary attitude with $\theta = 0\,\mathrm{deg}$ and $\dot{\theta} = 0\,\mathrm{deg/s}$ (see Fig. 2.1). During this phase, the position is not controlled as the translational and rotational motion is decoupled in the assumed state-space model.

Rendezvous Phase

After successfully controlling the attitude to an angle $|\theta| \leq \epsilon_\theta$ with $|\dot{\theta}| \leq \epsilon_{\dot{\theta}}$, the close-approach rendezvous phase begins. The attitude controller will continue to stabilize the attitude at $\theta = 0\,\mathrm{deg}$. As the final docking of the two spacecrafts has to be performed very carefully, first, the chaser will be placed close to the target only having an offset in the negative y-direction: $p_{ref} = [0\,\mathrm{m}, -d]$ (see Fig. 2.2). The offset d lies typically between $5\,\mathrm{m}$ to $20\,\mathrm{m}$ [1–4]. For this work, a value of $d = 10\,\mathrm{m}$ is chosen. The position controller will be used to drive the chaser from its initial position p_0 to this reference position p_{ref} and stabilize it there, meaning $\dot{x} = \dot{y} = 0\,\mathrm{m\,s^{-1}}$.

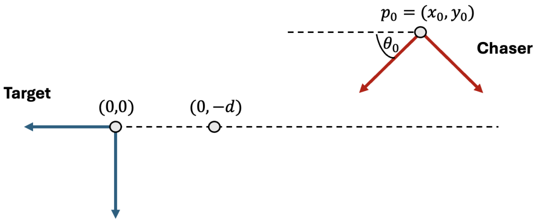


Figure 2.1: Phase 1

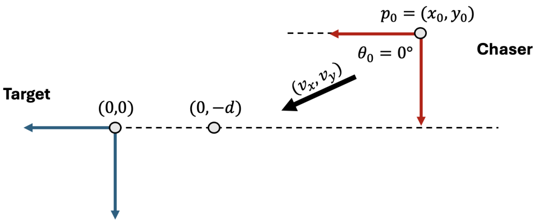


Figure 2.2: Phase 2

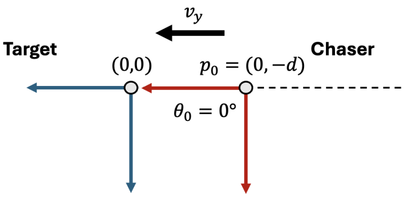


Figure 2.3: Phase 3

Final Docking Phase

When the chaser successfully arrives at the reference state of the rendezvous phase, meaning $|x| \leq \epsilon_{x_{ref}}$, $|y - d| \leq \epsilon_{y_{ref}}$, $|\dot{x}| \leq \epsilon_{\dot{x}_{ref}}$, $|\dot{y}| \leq \epsilon_{\dot{y}_{ref}}$, $|\theta| \leq \epsilon_{\theta}$ and $|\dot{\theta}| \leq \epsilon_{\dot{\theta}}$, the final docking phase begins. During this phase, the attitude controller continues to stabilize the spacecraft around $\theta = 0\,\mathrm{deg}$. The position controller drives the position of the chaser to the target's docking port, meaning regulating the chaser's position to $x = y = 0\,\mathrm{m}$ with final velocity components $\dot{x} = \dot{y} = 0\,\mathrm{m\,s^{-1}}$ (see Fig. 2.3). When the final docking conditions are met, namely $|x| \leq \epsilon_{x_{final}}$, $|y| \leq \epsilon_{y_{final}}$, $|\dot{x}| \leq \epsilon_{\dot{x}_{final}}$, $|\dot{y}| \leq \epsilon_{\dot{y}_{final}}$, $|\theta| \leq \epsilon_{\theta}$ and $|\dot{\theta}| \leq \epsilon_{\dot{\theta}}$, the docking is considered to be successful.

2.2.2 Control Method

Linear Quadratic Regulator

For both attitude and position control a Linear Quadratic Regulator (LQR) is implemented. The LQR is an optimal control strategy used to regulate the state of a linear dynamic system to the origin while minimizing a quadratic cost function. It is widely used in control engineering due to its balance between performance and control effort. Given a linear time-invariant (LTI) system described by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t),$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector and $\mathbf{u}(t) \in \mathbb{R}^m$ is the control input, the goal of LQR is to find a control law $\mathbf{x}(t) = -\mathbf{K}\mathbf{x}(t)$ that minimizes the cost function

$$J = \int_0^{\infty} \left(\mathbf{x}(t)^T\mathbf{Q}\mathbf{x}(t) + \mathbf{u}(t)^T\mathbf{R}\mathbf{u}(t)\right) dt,$$

where $\mathbf{Q} \succeq 0$ and $\mathbf{R} \succ 0$ are weighting matrices that penalize state deviations and control effort, respectively. The optimal gain matrix $\mathbf{K}$ is obtained by solving the continuous-time algebraic Riccati equation. LQR provides a feedback controller that ensures stability and optimality for the given system and cost criteria.

Error-Based LQR

In many practical control problems, the objective is to make the system follow a desired time-varying trajectory rather than simply stabilize it at the origin. To achieve this, the LQR can be applied to the error dynamics between the actual state and the reference state. Let $\mathbf{x}(t) \in \mathbb{R}^n$ denote the actual system state and $\mathbf{x}_r(t) \in \mathbb{R}^n$ the desired reference trajectory. The error is defined as

$$\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{x}_r(t).$$

Assuming the system dynamics are linear,

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t),$$

the goal is to design a control law based on the error:

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{e}(t) = -\mathbf{K}(\mathbf{x}(t) - \mathbf{x}_r(t)).$$

The gain matrix $\mathbf{K}$ is obtained by solving the standard LQR problem for the linear system. The controller acts to minimize the error between the system and the reference, while also balancing control effort via the quadratic cost function:

$$J = \int_0^\infty \left(\mathbf{e}(t)^T\mathbf{Q}\mathbf{e}(t) + \mathbf{u}(t)^T\mathbf{R}\mathbf{u}(t)\right) dt,$$

where $\mathbf{Q} \succeq 0$ and $\mathbf{R} \succ 0$ are the state and control weighting matrices, respectively. Usually a feed-forward term $\mathbf{u}_r(t)$ needs to be considered to be able to stabilize the system at the reference state. However, as for this project no disturbance are considered and no other external forces act on the system, $\mathbf{u}_r(t)$ can be neglected as no control input is needed to stabilize the system at a certain reference state.

Controllability Analysis

A requirement for being able to successfully use a LQR, one has to make sure that the system is fully controllable. To verify controllability, each subsystem is analysed independently by constructing its controllability matrix

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & \dots \end{bmatrix}$$

and computing its rank.

- **Attitude subsystem:** Given the state matrices $\mathbf{A}_{at}$ and $\mathbf{B}_{at}$ the controllability matrix is given as:

  $$\mathcal{C}_{at} = \begin{bmatrix} \mathbf{B}_{at} & \mathbf{A}_{at}\mathbf{B}_{at} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{I_c} \\ \frac{1}{I_c} & 0 \end{bmatrix}$$

  Since rank$(\mathcal{C}_{at}) = 2 = \dim(\mathbf{x}_{at})$, the attitude subsystem is fully controllable.

- **Position subsystem:** Given the state matrices $\mathbf{A}_{pos}$ and $\mathbf{B}_{pos}$ the controllability matrix is given as:

$$
\begin{aligned}
\mathcal{C}_{pos} &= \begin{bmatrix} \mathbf{B}_{pos} & \mathbf{A}_{pos}\mathbf{B}_{pos} & \mathbf{A}_{pos}^2\mathbf{B}_{pos} & \mathbf{A}_{pos}^3\mathbf{B}_{pos} \end{bmatrix} \\
&= \begin{bmatrix}
0 & 0 & \frac{1}{m_c} & 0 & 0 & \frac{1}{m_c} & \frac{1}{m_c} & 0 \\
0 & 0 & 0 & \frac{1}{m_c} & \frac{1}{m_c} & 0 & 0 & \frac{1}{m_c} \\
\frac{1}{m_c} & 0 & 0 & -\frac{2n}{m_c} & -\frac{6n^2}{m_c} & 0 & 0 & \frac{12n^3}{m_c} \\
0 & \frac{1}{m_c} & \frac{2n}{m_c} & 0 & 0 & -\frac{6n^2}{m_c} & -\frac{12n^3}{m_c} & 0
\end{bmatrix}
\end{aligned}
$$

  Since rank$(\mathcal{C}_{pos}) = 4 = \dim(\mathbf{x}_{pos})$, the position subsystem is fully controllable.

Attitude Controller

The attitude controller ensures that the chaser spacecraft rotates to match the target's orientation for successful docking. An LQR is used for this purpose. The desired orientation and angular velocity are both zero during all phases, meaning the chaser should stay perfectly aligned with the target and should not be rotating. Because of this, using a standard LQR or an error-based LQR will give the same result in all phases. However, tuning of the controller varies by mission phase. Early on, conserving power and fuel is prioritized, so higher weight is placed on the control input ($\mathbf{R}$), making the response slower but more efficient. Near docking, precision becomes critical, so the weight on the state error ($\mathbf{Q}$) is increased. This ensures the state error remains within the tolerances ϵ_θ and $\epsilon_{\dot{\theta}}$, as required for docking.

To ensure a realistic behaviour of the controller and the system's response, physical limitations of the spacecraft are considered. For instance, the maximum allowable angular velocity of the chaser is set to $\dot{\theta}_{\text{max}} = 10\,\text{deg/s}$, and the torque actuator is limited to a maximum output τ_{max}. This ensures that the simulated performance reflects what can be achieved with the actual spacecraft dynamics and hardware.

Position Controller

The position controller is based on a LQR approach as well and is designed to adapt to the different stages of the docking manoeuvrer. For the first phase, the chaser is allowed to drift freely under its initial conditions, with no active position control. After the attitude was controlled successfully, the second phase begins and a non-zero reference state for the position state is introduced. As soon as this happens, an error-based LQR controller is activated. This controller is tuned to prevent overshooting in the y-direction, which is critical to avoid unintentional collisions with the target. In the final docking phase, the reference state becomes zero, and a standard LQR controller is applied. Here, tuning focuses on limiting the approach velocity to below $0.061\,\mathrm{m\,s^{-1}}$ [5] to satisfy docking safety constraints, while still ensuring tight control in the y-axis to prevent overshoot. This progressive control strategy ensures both efficiency and safety throughout the manoeuvrer.

2.3 Navigation

Navigation is the process of estimating the current state, in this case the position and attitude as well as the linear and rotational velocity, of a vehicle. This involves managing the vehicle's course, orientation, and speed. For this purpose, the vehicle, in this case the spacecraft, integrates different sensors which measure physical quantities related to the spacecraft's state. Fusing these measurements by applying state estimation algorithms yields an estimate of the current state. In the following, first the integrated sensors for the proposed scenario are introduced. Afterwards, the algorithm to combine these measurements, namely the Kalman filter, is described.

2.3.1 Sensors

The introduced state vector of the chaser spacecraft includes the relative position, attitude, linear velocity and rotational velocity with respect to the target. Thus, for this mission sensors are required whose measurements cover those quantities.

Radar

A radar is a system that uses radio waves to detect and determine the distance, speed, and direction of objects. It works by transmitting electromagnetic signals, which reflect off objects and return to the receiver, allowing the system to calculate the object's position and motion relative to the radar source.

For the project, the radar is used to measure the relative position from the target to the chaser, as well as the relative linear velocity of the chaser with respect to the target. It provides measurements in both x- and y-direction, enabling accurate estimation of translational motion needed for position and velocity control during docking.

Gyroscope

Gyroscopes are instruments that detect rotational movement in three-dimensional space. They detect how fast the spacecraft is rotating providing real-time information about orientation changes. In this project, the gyroscope specifically measures the angular velocity around the z-axis, which corresponds to the rotational motion affecting the θ angle of the introduced state vector. This measurement, denoted as $\dot{\theta}$, is critical for controlling the spacecraft's attitude and ensuring smooth alignment during the docking process. When combined with star tracker data, gyroscopes help maintain accurate and continuous attitude estimation, even during brief periods when optical measurements may be unavailable [6].

Star Tracker

Star trackers are optical sensors used on spacecrafts to determine their attitude. They work by capturing images of stars in their field of view and comparing the observed star patterns with an onboard star catalogue. By identifying known stars and their positions, the star tracker can calculate how the spacecraft is oriented relative to a fixed celestial reference frame, typically the International Celestial Reference Frame (ICRF) [6]. In the context of the project, the star tracker determines the attitude of the chaser spacecraft with respect to the stars, meaning it provides an absolute orientation in the ICRF. Based on the current orbit parameters of the chaser and the known orbit of the target, the relative attitude of the chaser with respect to the target can then be estimated. This allows to compute the attitude error, denoted as θ, which is essential for precise docking alignment.

2.3.2 Observability Analysis

To be able to successfully estimate the whole state, it is required that the system is fully observable. To validate this, the rank of the observability matrix of the full state space model is computed:

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix}$$

where $A \in \mathbb{R}^{6\times 6}$ is the system matrix, $C \in \mathbb{R}^{6\times 6}$ is the output matrix, and $n = 6$ is the number of states. As seen before the output matrix is the identity matrix $\mathbf{I}_{6\times 6}$. This means that all state variables are directly measured in the output $\mathbf{y}(t)$, and no information is hidden from the observer. Therefore, the observability matrix simplifies to:

$$\mathcal{O} = \begin{bmatrix} \mathbf{I}_{6\times 6} \\ \mathbf{A} \\ \mathbf{A}^2 \\ \vdots \\ \mathbf{A}^5 \end{bmatrix}$$

It holds rank$(\mathcal{O}) = 6$ and thus the state is fully observable.

2.3.3 State Estimation using Kalman Filter

This section describes the Kalman Filter implemented to estimate the full state of the presented dynamic system, including translational and rotational motion. Thus the full state space model presented in 2.1.1. The filter fuses data from a radar, a gyroscope, and a star tracker to provide optimal estimates of position, velocity, orientation, and angular velocity.

System Model

The system is modeled as a discrete-time linear state-space system. The full state vector is:

$$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} & \theta & \dot{\theta} \end{bmatrix}^T$$

The discrete-time dynamics are determined based on the Euler approximation and are defined by:

$$\mathbf{x}_{k+1} = \mathbf{A}_d\mathbf{x}_k + \mathbf{B}_d\mathbf{u}_k + \mathbf{w}_k$$

where:

- $\mathbf{A}_d = \mathbf{A}\Delta t + \mathbf{I}_{6\times 6} \in \mathbb{R}^{6\times 6}$ is the discretized state transition matrix,
- $\mathbf{B}_d = \mathbf{B}\Delta t \in \mathbb{R}^{6\times 3}$ is the input matrix,
- $\mathbf{u}_k = [F_x, F_y, \tau]^T$ is the control input vector containing force and torque inputs,
- $\mathbf{w}_k$ is the zero-mean Gaussian process noise with covariance $\mathbf{Q}$.

Measurement Models

Three different sensors are used for measurement updates:

- **Radar:** Provides measurements of position and velocity in the x and y directions. Measurement vector:

  $$\mathbf{z}_{radar} = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^T + \mathbf{v}_{radar}, \quad \mathbf{v}_{radar} \sim \mathcal{N}(0, \mathbf{R}^{nav}_{radar})$$

  Measurement matrix:

  $$\mathbf{H}_{radar} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

- **Gyroscope:** Measures angular velocity:

  $$z_{gyro} = \dot{\theta} + v_{gyro}, \quad v_{gyro} \sim \mathcal{N}(0, \mathbf{R}^{nav}_{gyro})$$

  Measurement matrix:

  $$\mathbf{H}_{gyro} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- **Star Tracker:** Measures orientation angle:

  $$z_{star} = \theta + v_{star}, \quad v_{star} \sim \mathcal{N}(0, \mathbf{R}^{nav}_{star})$$

  Measurement matrix:

  $$\mathbf{H}_{star} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Kalman Filter Algorithm

At each time step, the Kalman Filter performs the following steps:

- **Prediction Step:** Based on the commanded control input to the system which is provided by the controller, the previous state estimate and the discrete system model, a prediction of the next state is determined.

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_d \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_d \mathbf{u}_{k-1}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_d \mathbf{P}_{k-1|k-1} \mathbf{A}_d^T + \mathbf{Q}$$

- **Update Step:** For each available sensor, a measurement update is performed if new sensor data is available meaning each sensor is incorporated asynchronously as they have typically different update rates:

$$\begin{aligned}
\mathbf{y}_k &= \mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1} \\
\mathbf{S}_k &= \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R} \\
\mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}^T\mathbf{S}_k^{-1} \\
\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{y}_k \\
\mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_{k|k-1}
\end{aligned}$$

Chapter 3

Implementation

3.1 Software

To simulate the docking of the two spacecrafts and to analyse the performance of the controller and the state estimator, a Matlab Simulink simulation is implemented. A screenshot of the overall structure is given in Fig. 3.1. The simulation is split into five main components, namely the Spacecraft Dynamics, Navigation, Guidance, Control and Actuators. How each component is implemented is briefly described in the following sections.

3.2 Spacecraft Dynamics

In this block the dynamics of the system is simulated by simply implementing the attitude and the position state space model. It takes the acting forces and torque generated by the actuators as input and provides the output vectors of both subsystems.

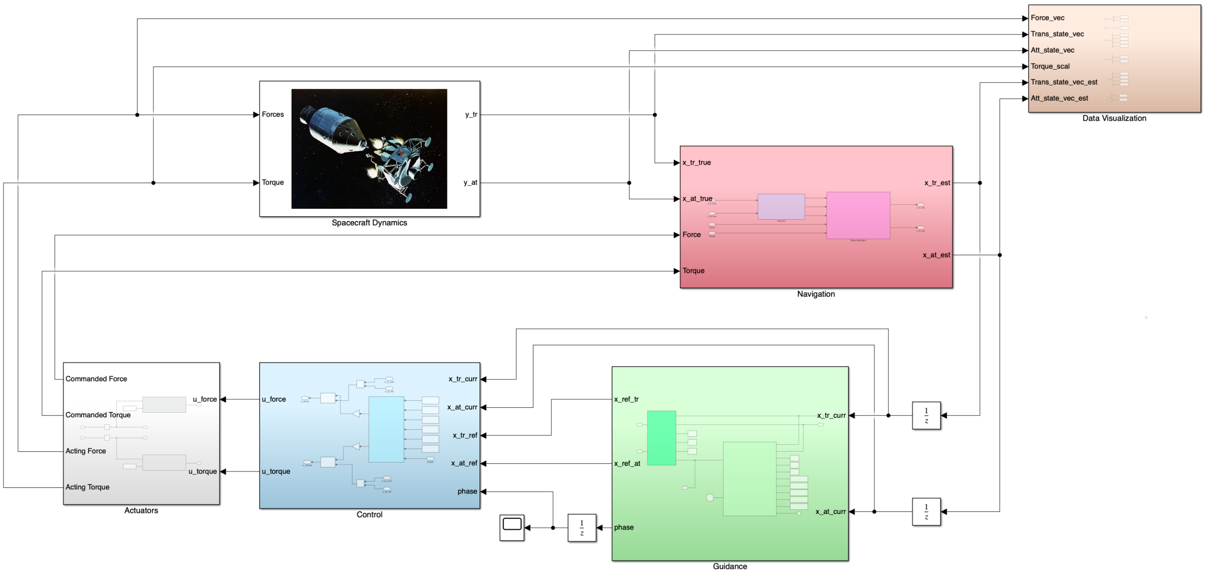


Figure 3.1: Simulink model of the simulation.

3.3 Navigation

This block simulates the navigation subsystem of the chaser spacecraft (see Fig. 3.2). It composes of two blocks: Sensors and State Estimation. The Sensors block (see Fig. 3.3) simulates the sensor measurements based on the true system's output vectors by adding a random white gaussian noise given a certain variance for each sensor. Important to mention is that no bias or delay of the measurements is simulated. Those measurements are then fed to the State Estimation block (see Fig. 3.4) together with the commanded control input forces and torque. Additionally, the initial estimate, the covariance matrices for the system, for the process noise and for the measurement noise of each sensor as well as the discretized state space model are provided. The Kalman Filter block simulates the state estimation based on the presented Kalman filter algorithm and outputs an estimate for the attitude and position state vectors.

The choice of the variances of the sensors' noise is based on the accuracy of real sensors for comparable missions. The SAG-SS by FIBERPRO is chosen as a reference for the gyroscope. Used for space applications, the device presents a random angle walk of $0.005\,\mathrm{deg/h}$ [7]. In radians per second, it corresponds to an accuracy of $\sigma_{\dot{\theta}} = 1.45 \times 10^{-6}\,\mathrm{rad\,s^{-1}}$. For the star tracker, the ST-16RT2 by Rocket Lab is chosen. It presents an absolute accuracy of 55 arcseconds around boresight [8]. That's an accuracy of $\sigma_{\theta} = 2.666 \times 10^{-4}\,\mathrm{rad}$. For the radar, no data sheets can be found that specify exact accuracies for the position and velocity measurements. Based on other missions, the following accuracies are used: $\sigma_{pos} = 0.01\,\mathrm{m}$ and $\sigma_{vel} = 0.01\,\mathrm{m\,s^{-1}}$.

The covariance matrices for the measurement noise $(\mathbf{R}^{nav}_{radar}, \mathbf{R}^{nav}_{star}, \mathbf{R}^{nav}_{gyro})$ are computed based on those variances. The resulting values used in the simulation are summarized in Tab. 3.1. Furthermore, the update rates for the radar, the star tracker and the gyro are also chosen based on real sensors and are set in the simulation to $5\,\mathrm{Hz}$, $1\,\mathrm{Hz}$ and $100\,\mathrm{Hz}$ respectively. The covariance matrix for the process noise $\mathbf{Q}^{nav}_{proc}$

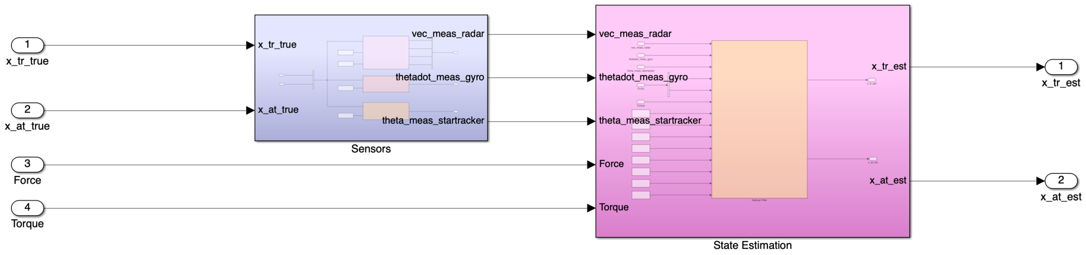
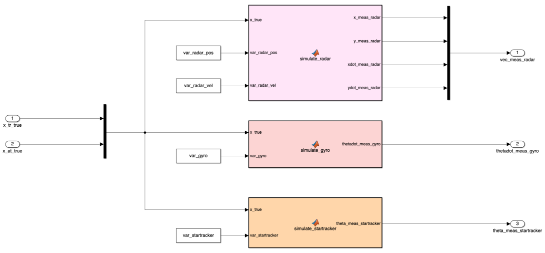


Figure 3.2: Simulink navigation block



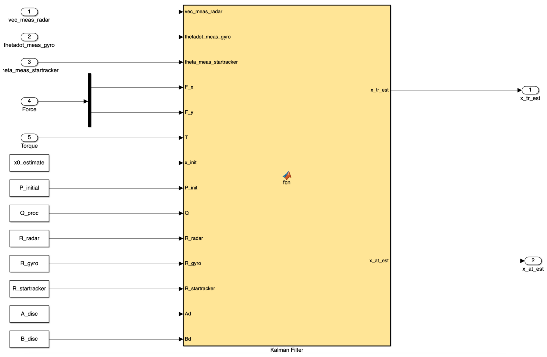


Figure 3.3: Simulink sensors block.

Figure 3.4: Simulink state estimation block

	Variance	$\mathbf{R}^{nav}_{sensor}$
Radar	$\sigma^2_{radar-pos} = 0.01^2$ $\sigma^2_{radar-vel} = 0.01^2$	$\mathbf{R}^{nav}_{radar} = \mathrm{diag}(\sigma^2_{radar-pos}, \sigma^2_{radar-pos}, \sigma^2_{radar-vel}, \sigma^2_{radar-vel})$
Star Tracker	$\sigma^2_{star} = (2.666\mathrm{e}{-4})^2$	$\mathbf{R}^{nav}_{star} = \mathrm{diag}(\sigma^2_{star})$
Gyroscope	$\sigma^2_{gyro} = (8.33\frac{2\pi}{360}\mathrm{e}{-5})^2$	$\mathbf{R}^{nav}_{gyro} = \mathrm{diag}(\sigma^2_{gyro})$

Table 3.1: Overview of the sensor's variances used in the simulation.

is tuned by analysing the Root Mean Squared Error (RMSE) between estimated and true state to ensure that the estimation is not too noisy while not responding too slow. The iteratively found final value is $\mathbf{Q}^{nav}_{proc} = \mathrm{diag}(1\mathrm{e}{-3}; 1\mathrm{e}{-3}; 1\mathrm{e}{-8}; 1\mathrm{e}{-8}; 1\mathrm{e}{-2}\frac{2\pi}{360}; 1\mathrm{e}{-5}\frac{2\pi}{360})^2$.

3.4 Guidance

The focus of this stage of the project is the control and navigation subsystem of the chaser. However, to simulate the rendezvous and docking scenario a simple guidance has to be implemented. It consists only of two steps (see Fig. 3.5). First, based on the current state estimate, it is checked whether the current reference state is reached within certain tolerances. If that is the case, the phase switches. If the last reference state is reached, the simulation stops. Second, based on the current phase, the correct reference state is selected. The used thresholds and reference states are given in Tab. 3.2. The thresholds are chosen based on [9].

3.5 Controller

As stated out before, the docking manoeuvrer is divided into three distinct phases, each using separate LQR controllers with differently tuned control gains. Based on the current phase p, which is provided by the guidance, the controller block in the simulation selects the current gain matrix $\mathbf{K}^{con}_{att,p}$ and $\mathbf{K}^{con}_{pos,p}$. Subsequently, these matrices, the current estimated state and the current reference state are used to compute the commanded control input for the position and attitude subsystem based on the presented error-based LQR control law (see Fig. 3.6).

To tune the control gains accordingly, the $\mathbf{Q}^{con}$ and $\mathbf{R}^{con}$ matrices are selected to reflect the priorities of each phase. Tuning is done through an iterative, trial-and-error approach, initial values are chosen based on physical intuition, then adjusted based on the system's response. In most cases, either $\mathbf{Q}^{con}$ or $\mathbf{R}^{con}$ was initially set to the identity matrix, and the other is scaled to emphasize desired penalties. When both state and input has to be penalized, only selected elements are scaled while others remained at 1.

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

Table 3.2: 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 $[\mathrm{deg/s}]$.

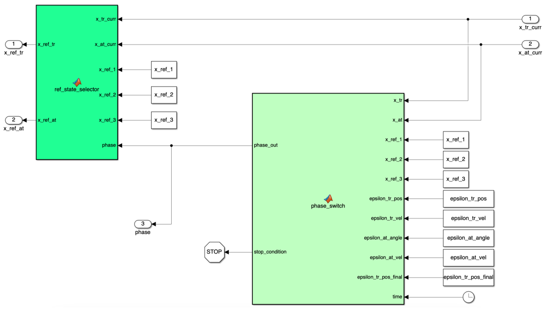


Figure 3.5: Simulink guidance block.

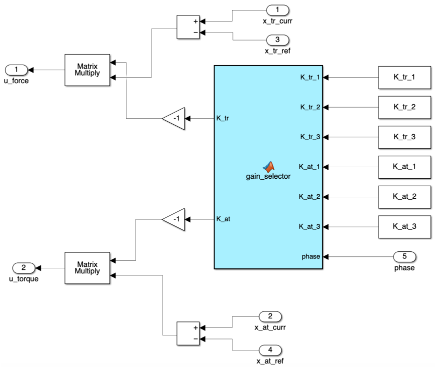


Figure 3.6: Simulink controller block.

Phase 1 – Attitude Alignment

During the initial phase, only the attitude controller is active. The focus is on aligning the spacecraft's orientation while allowing it to drift under its initial translational velocity. Since the spacecraft's orientation may be significantly misaligned with the target's docking port, the commanded torque can get high. However, as time is not a limiting factor, the control input is penalized to reduce fuel consumption and actuator power usage.

$$
\begin{aligned}
\mathbf{Q}^{con}_{att,1} &= \mathrm{diag}(1,\ 1)\\
\mathbf{R}^{con}_{att,1} &= 10\\
\mathbf{K}^{con}_{att,1} &= \begin{bmatrix} 0.3162 & 5.3972 \end{bmatrix}
\end{aligned}
$$

Phase 2 – Approach and Alignment

In the second phase, both position and attitude controller are active. The focus is shifted to align the spacecraft's position with the docking port at 10 meters away. High penalties are applied to velocity and thrust in y-direction to avoid overshooting or crashing into the target. The x-position is also penalized to ensure lateral alignment. Attitude control in this phase emphasizes accurate state tracking, allowing for greater control effort to maintain orientation as attitude control accuracy in the later docking phases becomes crucial.

$$
\begin{aligned}
\mathbf{Q}^{con}_{pos,2} &= \mathrm{diag}(10,\ 1,\ 1,\ 10^5)\\
\mathbf{R}^{con}_{pos,2} &= \mathrm{diag}(1,\ 10)\\
\mathbf{K}^{con}_{pos,2} &= \begin{bmatrix} 3.1672 & -0.0622 & 172.5890 & -28.6566\\ 0.0622 & 0.3156 & -2.8657 & 113.5537 \end{bmatrix}\\
\mathbf{Q}^{con}_{att,2} &= \mathrm{diag}(10,\ 10)\\
\mathbf{R}^{con}_{att,2} &= 1\\
\mathbf{K}^{con}_{att,2} &= \begin{bmatrix} 3.1623 & 17.3291 \end{bmatrix}
\end{aligned}
$$

Phase 3 – Final Docking

In the final phase, the controller prioritizes a slow and controlled approach. The largest penalties are placed on translational velocities to ensure docking speed stays within tolerance limits.

$$
\begin{aligned}
\mathbf{Q}^{con}_{pos,3} &= \mathrm{diag}(1,\ 1,\ 10^3,\ 10^4) \\
\mathbf{R}^{con}_{pos,3} &= \mathrm{diag}(1,\ 1) \\
\mathbf{K}^{con}_{pos,3} &= \begin{bmatrix} 1.0111 & 0.0004 & 102.2348 & -8.2529 \\ -0.0004 & 1.0000 & -8.2529 & 139.1015 \end{bmatrix} \\
\mathbf{Q}^{con}_{att,3} &= \mathrm{diag}(10,\ 10) \\
\mathbf{R}^{con}_{att,3} &= 1 \\
\mathbf{K}^{con}_{att,3} &= \begin{bmatrix} 3.1623 & 17.3291 \end{bmatrix}
\end{aligned}
$$

This phase completes the docking manoeuvrer with an emphasis on precision and safety. The overall tuning strategy allows each phase to meet its specific requirements while keeping the behaviour physically reasonable, even without optimizing for fuel or power consumption.

3.6 Actuators

To convert the control signal computed by the controller block into forces and torque, a simple actuator block composing of two main steps is implemented to simulate their behaviour (see Fig. 3.7). First, the signal gets saturated based on realistic limits for maximum forces and torque that could be provided by real actuators. Second, if the control signals are not equal to zero, meaning that control forces or torque needs be applied, a random white Gaussian noise is applied based on a given variance. This is intended to simulate a realistic, non-ideal actuator.

As this work is based on the Rendezvous and Docking of the Apollo 11 mission, a similar RCS configuration to the Apollo 11 Lunar Module ascent stage is simulated for the chaser to generate those control forces. Based on this, the maximum force achievable in any linear direction is estimated in order to model the actuators in the simulation. The Apollo LM ascent stage utilized 16 RCS thrusters arranged in four "quads", located symmetrically around the spacecraft. Each quad consisted of four thrusters oriented along the $\pm x$, $\pm y$, and $\pm z$ body axes. Each RCS thruster was capable of providing approximately $445\,\mathrm{N}$ of thrust [10]. To achieve pure linear translation without inducing rotational torque, pairs of thrusters positioned symmetrically about the spacecraft's center of mass are fired simultaneously. For example, to translate forward along the $+y$ axis, two thrusters (one from each rear quad) are fired in the $+y$ direction. This configuration ensures that the torques generated by the individual thrusters cancel out, resulting in net linear force only:

$$F_{max} = 2 \cdot F_{thruster} = 2 \cdot 445\,\mathrm{N} = 890\,\mathrm{N}.$$

To decouple also the translational and rotational actuators for the spacecraft, besides the mentioned RCS

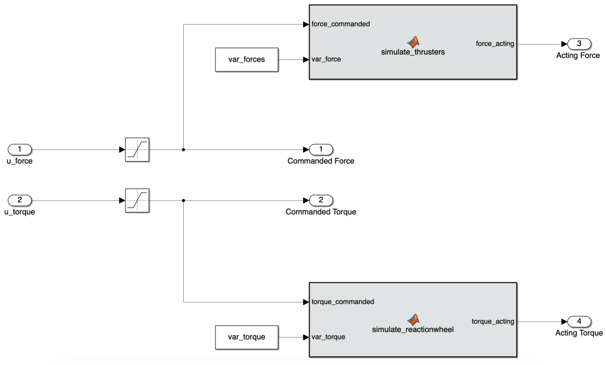


Figure 3.7: Simulink actuators block.

configuration, reaction wheels are integrated for attitude control. As no reaction wheel was used in the Apollo 11 Lunar Module ascent stage, a realistic modern actuator is chosen to approximate the maximum available control torque. For this, the Honeywell HR16 reaction wheels are selected due to their reliability, proven performance in space missions, and suitability for large spacecraft requiring high precision with moderate torque demands making it ideal for a spacecraft of the chaser's class. With a configuration of four HR16 reaction wheels, the spacecraft can generate a maximum torque around each axis (roll, pitch, yaw) of $\tau_{max} = 0.4\,\mathrm{N\,m}$ [11]. To model the noise of the actuators, variances of $\sigma^2_{thrust} = 0.1$ and $\sigma^2_{torque} = 0.00001$ are used.

3.7 Simulation Parameters

For running the simulation, parameters related to the spacecraft as well as to the lunar orbit have to be specified. The mass of the chasing spacecraft is chosen based on the ascent stage of the Apollo lunar model. The total mass of the spacecraft sums up the dry mass ($2131.88\,\mathrm{kg}$), the propellant mass ($2358.68\,\mathrm{kg}$) and the astronaut mass ($2 \cdot 75\,\mathrm{kg}$) which yields to $m_{total} = 4640.56\,\mathrm{kg}$. The moment of inertia around the z-axis computed in the design phase of this project is $I_c = 45.9\,\mathrm{kg\,m^2}$. The altitude of the lunar orbit is chosen to be $100\,\mathrm{km}$ leading to a mean orbital motion of $n = 8.892\,18 \times 10^{-4}\,\mathrm{rad\,s^{-1}}$. The time step size of the simulation is set to $0.01\,\mathrm{s}$.

Chapter 4

Simulation

4.1 Scenarios

In order to validate the performance of the designed and implemented controller and state estimation algorithm, two different scenarios need to be considered tested. As described in Sec. 2.2.1, the chaser arrives behind the target spacecraft after orbital manoeuvrers ideally at a position of $p_0 = [0\,\mathrm{m}, -200\,\mathrm{m}]$, with an attitude of $\theta \approx 0\,\mathrm{deg}$, and with velocity components of $v_x = v_y \approx 0\,\mathrm{m\,s^{-1}}$, $\dot{\theta} \approx 0\,\mathrm{deg/s}$. This scenario represents what will hereafter be referred to as the *best-case scenario*.

However, as the orbital manoeuvres are not guaranteed to be performed perfectly, deviations of the initial state can occur. Specifically, deviations in the position of $20\,\mathrm{m}$, in the attitude of $180\,\mathrm{deg}$ and in the velocities of $0.1\,\mathrm{m\,s^{-1}}$ and $0.5\,\mathrm{deg/s}$ are assumed. This scenario will be further referred to as the *worst case scenario*.

The robustness of the controller is tested by simulating these two scenarios in detail as well as performing a Monte Carlo simulation with 500 randomly chosen initial states between the best- and worst-case scenario to evaluate if the controller can handle these deviations. The true initial states used in the simulation are listed in Tab. 4.1.

To evaluate the performance and robustness of the Kalman filter, an offset is added to the true initial state to obtain the estimated initial state (see Tab. 4.1). This offset and the initial covariance matrix for the state estimate are set to:

$$\sigma_{offset} = \begin{bmatrix} 3\,\mathrm{m} \\ 3\,\mathrm{m} \\ 0.05\,\mathrm{m\,s^{-1}} \\ 0.05\,\mathrm{m\,s^{-1}} \\ 10\,\mathrm{deg} \\ 0.05\,\mathrm{deg/s} \end{bmatrix}, \quad \mathbf{P}_{\text{offset}} = \mathrm{diag}\left(\sigma^2_{\text{offset}}\right)$$

4.2 Results

This section presents the results of the simulated rendezvous and docking scenarios. The results include time histories of state variables and control inputs, spacecraft trajectories in the target frame, and summary tables comparing key performance metrics.

Figures 4.1 and 4.2 display the time evolution of the state variables and control inputs for the best-case and

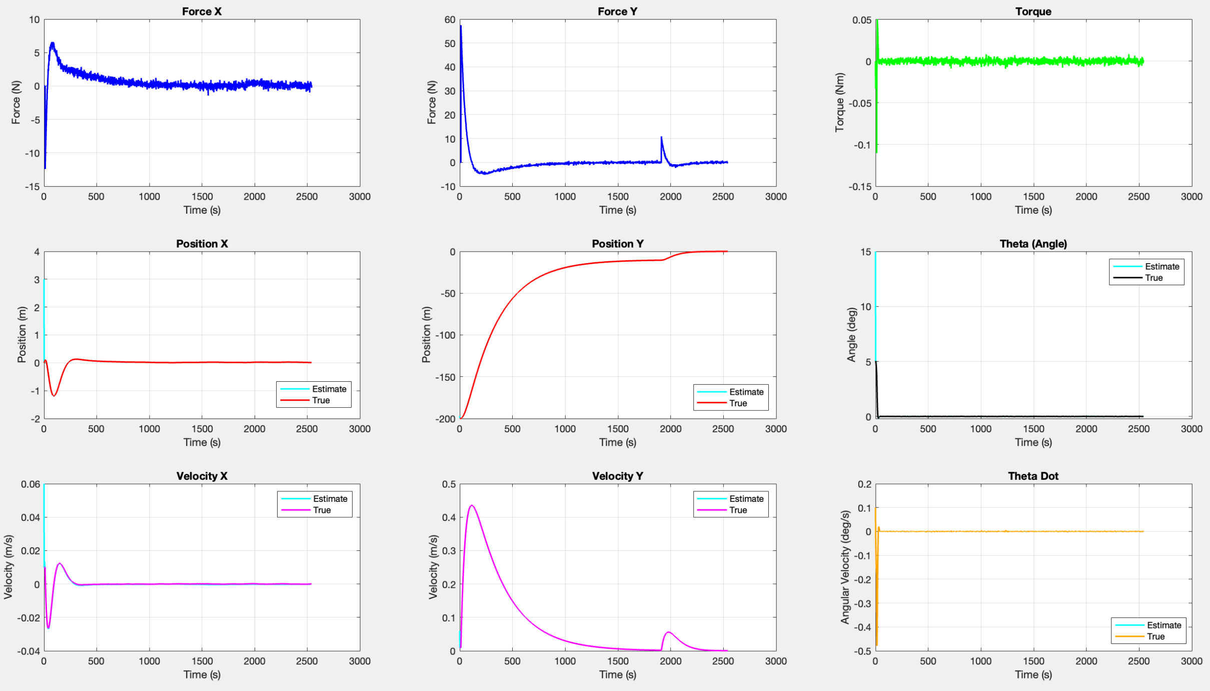


Figure 4.1: Results of the best case scenario: plot of the input and state variables over time.

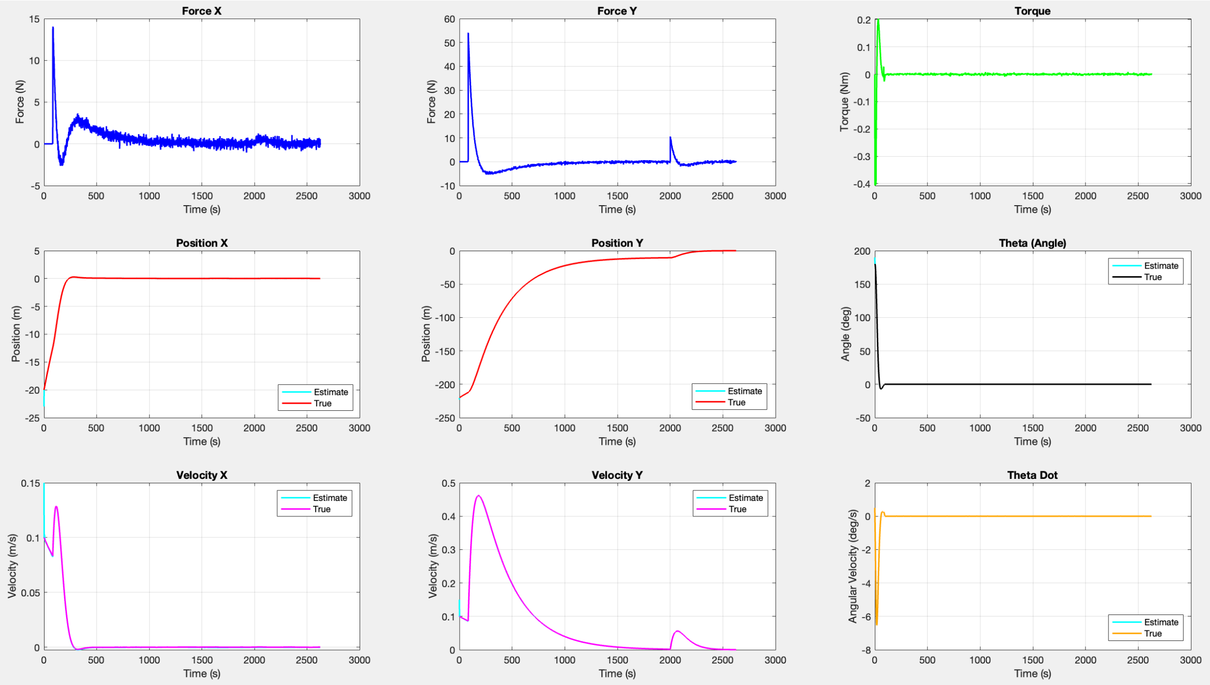


Figure 4.2: Results of the worst case scenario: plot of the input and state variables over time.

worst-case scenario, respectively. Both the true values and filter estimates for the state variables are plotted for all variables.

Figures 4.3 and 4.4 present the trajectories of the spacecraft in the target-relative frame for the best-case and worst-case simulations, respectively. The x and y positions are plotted, showing both the estimated and true trajectories.

Table 4.1 summarizes the main quantitative results for the best- and worst-case scenario as well as for the performed Monte Carlo simulation, including the initial state and estimate, the time taken to reach intermediate reference points, the final states at the end of each of the three docking phases, the total docking duration and the Root Mean Square Error (RMSE) between the estimated and true states.

Category	**Metric**	**Best Case**	**Worst Case**	**Monte Carlo (mean ± std)**
Initial State	X [m]	0.0000	-20.0000	10.0538 ± 5.7899
	Y [m]	-200.0000	-220.0000	-190.0796 ± 5.8109
	Vx [m/s]	0.0100	0.1000	0.0484 ± 0.0294
	Vy [m/s]	0.0100	0.1000	0.0521 ± 0.0284
	θ [deg]	5.0000	180.0000	88.6371 ± 54.3927
	$\dot{\theta}$ [deg/s]	0.1000	0.5000	0.2515 ± 0.1446
Initial Estimate	X [m]	3.0000	-23.0000	7.0538 ± 5.7899
	Y [m]	-197.0000	-223.0000	-193.0796 ± 5.8109
	Vx [m/s]	0.0600	0.1500	0.0984 ± 0.0294
	Vy [m/s]	0.0600	0.1500	0.1021 ± 0.0284
	θ [deg]	15.0000	190.0000	98.6372 ± 54.3927
	$\dot{\theta}$ [deg/s]	0.1500	0.5500	0.30154 ± 0.1446
Reference 1 reached after [s]		10.06	83.26	55.08 ± 18.94
Phase 1 Final State	X [m]	0.1079	-12.4109	12.6080 ± 6.2390
	Y [m]	-199.9061	-212.2491	-187.3362 ± 6.1794
	Vx [m/s]	0.0089	0.0822	0.0426 ± 0.0300
	Vy [m/s]	0.0085	0.0857	0.0489 ± 0.0288
	θ [deg]	3.9985	-1.9315	-1.7803 ± 1.7848
	$\dot{\theta}$ [deg/s]	-0.1791	0.1999	-0.0254 ± 0.1919
Reference 2 reached after [s]		1911.80	2001.80	1928.27 ± 21.57
Phase 2 Final State	X [m]	0.0104	0.0119	0.0140 ± 0.0041
	Y [m]	-10.4959	-10.4993	-10.4983 ± 0.0017
	Vx [m/s]	-0.0002	-0.0002	-0.00023 ± 0.00001
	Vy [m/s]	0.0016	0.0016	0.00162 ± 0.00001
	θ [deg]	-0.0181	0.0040	-0.0066 ± 0.0171
	$\dot{\theta}$ [deg/s]	-0.0000	0.0015	0.0002 ± 0.0013
Total Docking Time [s]		2541.40	2626.20	2557.44 ± 19.06
Phase 3 Final State	X [m]	0.0093	0.0072	0.0086 ± 0.0074
	Y [m]	-0.0092	-0.0091	-0.0078 ± 0.0016
	Vx [m/s]	-0.0001	-0.0001	-0.00016 ± 0.00001
	Vy [m/s]	0.0002	0.0002	0.00025 ± 0.00002
	θ [deg]	0.0093	0.0245	0.0014 ± 0.0141
	$\dot{\theta}$ [deg/s]	0.0004	0.0007	-0.0001 ± 0.0007
RMSE - Filter vs. True	Position X [m]	0.02640829	0.02589915	0.026239124 ± 0.000094587
	Position Y [m]	0.02640900	0.02590088	0.026239365 ± 0.000094043
	Velocity X [m/s]	0.00051685	0.00050751	0.000514408 ± 0.000001813
	Velocity Y [m/s]	0.00050427	0.00049384	0.000500916 ± 0.000002452
	θ [deg]	0.19796642	0.19486003	0.19739871 ± 0.000724535
	$\dot{\theta}$ [deg/s]	0.00006179	0.00006174	0.000061903 ± 0.000000358

Table 4.1: Comparison of best- and worst-case docking scenarios as well as Monte Carlo simulations (mean ± standard deviation over 500 runs).

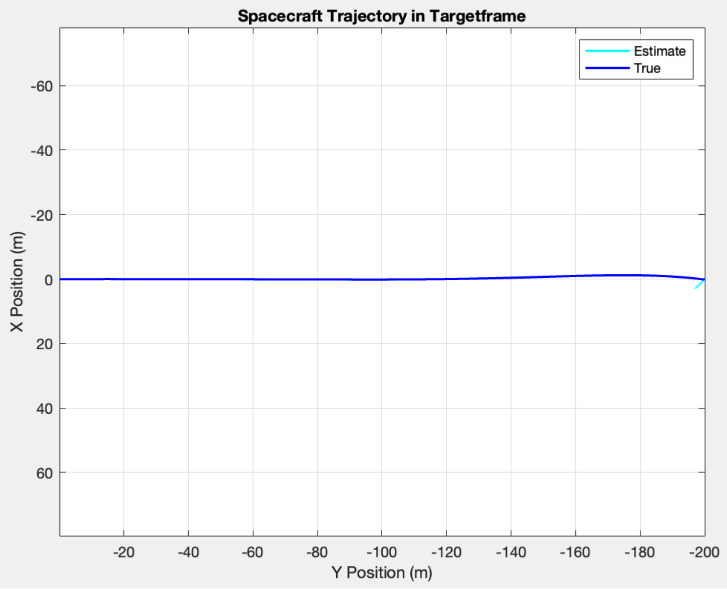


Figure 4.3: Best case scenario trajectory in target frame.

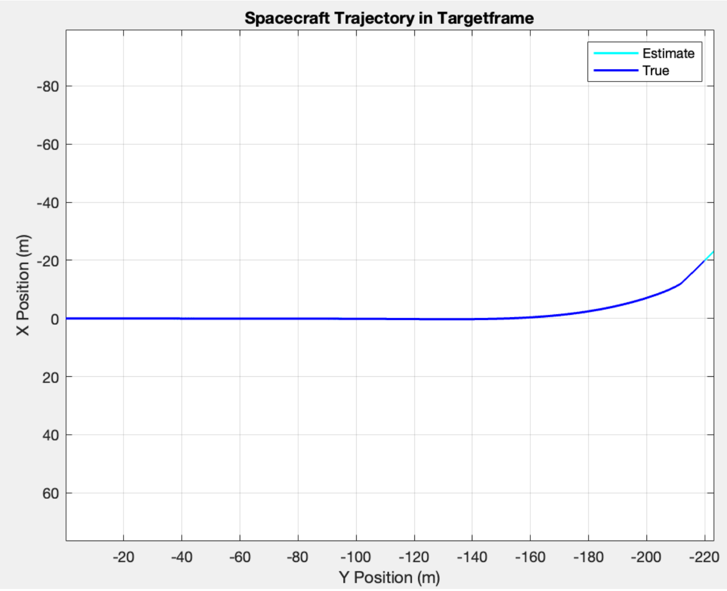


Figure 4.4: Worst case scenario trajectory in target frame.

Chapter 5

Discussion

This chapter provides an analysis of the docking simulation results for the best- and worst-case initial conditions as well as for the Monte Carlo simulation. The performance is evaluated with respect to the mission tolerances, docking duration, velocity constraints, and the accuracy of the state estimation.

It can be seen in Fig. 4.1 and Fig. 4.2 that the controller successfully handles the desired phases and approaches the different reference states accordingly. As specified in Tab. 3.2, each docking phase has clearly defined tolerances for position, velocity, attitude angle, and angular velocity. The final state values for both the best-case and worst-case scenarios, as well as for the Monte Carlo simulation runs (see Tab. 4.1), demonstrate that the spacecraft successfully meets all required tolerances by the end of each phase, and specifically at the final docking. This indicates that the docking controller and guidance logic succeed in bringing the spacecraft into compliance with the required terminal conditions, even in the presence of large initial deviations in the worst-case scenario demonstrating the robustness of the controller.

Additionally, in Fig. 4.1 and Fig. 4.2 it can be observed that the position controller successfully avoids overshoot in y-direction. This is a crucial requirement which needs to be met as otherwise a collision between chaser and target could occur. Furthermore, for the x-position as well as for the attitude θ, almost no overshoot results, which confirms well-tuned parameters of the controller.

The total expected docking time is $2557.44\,\mathrm{s} \pm 19.06\,\mathrm{s}$. The difference between best- and worst-case scenario of around $85\,\mathrm{s}$ reflects the increased control effort and manoeuvring required to correct the larger errors in the worst-case initial state. All scenarios require less than $45\,\mathrm{min}$ for full docking, which is reasonable for a proximity operations phase of this nature. As desired, the system exhibits a rapid correction in early stages (as seen in Fig. 4.1 and Fig. 4.2 in the steep state transitions in Phase 1 and Phase 2), followed by more gradual fine-tuning during Phase 3 for a safe and accurate docking.

In the final docking phase, a strict constraint was imposed on the linear velocity, limiting it to $0.061\,\mathrm{m\,s^{-1}}$. According to the plots in Fig. 4.1 and Fig. 4.2, both best and worst-case scenarios maintain velocity magnitudes in Phase 3 well below this threshold (with maximum observed values around $0.055\,\mathrm{m\,s^{-1}}$). This indicates successful adherence to safety margins designed to ensure soft contact with the target.

Similarly, the angular velocity remained far below the $10\,\mathrm{deg/s}$ limit during the entire docking manoeuvrer. Even for the worst-case scenario with an initial angle of $\theta = 180\,\mathrm{deg}$, the maximum $\dot{\theta}$ is with about $6.5\,\mathrm{deg/s}$

significantly below the threshold. These results confirm that the control system effectively limits motion rates as required.

The estimation accuracy is evaluated using the RMSE between the true states and filter estimates. As shown in the lower section of Tab. 4.1, the RMSEs for position and velocity are on the order of $0.025\,\mathrm{m}$ to $0.027\,\mathrm{m}$ and $0.0005\,\mathrm{m\,s^{-1}}$, respectively, across all cases. The RMSEs in angular variables are similarly small, with a maximum error in angle estimation of approximately $0.197\,\mathrm{deg}$ and in angular velocity of around $0.000\,06\,\mathrm{deg/s}$. These results suggest that the state estimator performed robustly. It can be seen that each estimated state variable successfully converges quickly to the true values early in the manoeuvrer and maintains close tracking throughout the docking process. In Fig. 4.3 and Fig. 4.4 it can also be seen that besides the initial estimation offset, the true and the estimated trajectory perfectly overlay for the best- and worst-case scenario which visually confirms the RMSE results.

Chapter 6

Conclusion

This work presented a simulation-based validation of a controller and navigation system for a closed-range rendezvous and docking mission under contrasting scenarios: 502 simulations are performed with initial states between a best-case with near-nominal initial conditions, and a worst-case with significant initial deviations in position and attitude. The outcomes demonstrate the effectiveness and robustness of the implemented control architecture and state estimator across both cases.

The controller is able to guide the chaser spacecraft through all three docking phases, consistently bringing the system within the specified tolerances for position, velocity, orientation, and angular velocity. Docking is achieved in under 45 minutes for all scenarios, with the worst-case requiring slightly more time due to the increased correction effort.

Position and velocity constraints in the final phase are strictly adhered to throughout the simulations. This is critical for safe contact with the target and validates the constraint-handling capability of the controller.

The state estimator maintains low RMSEs across all state variables and shows fast convergence. This highlights the reliability of the estimation framework for use in the closed-loop control. However, a potential vulnerability is identified: if one of the sensors fails, especially the relative position or attitude sensors, the state estimation process could degrade. In the current configuration, only the star tracker state may be partially recoverable through gyro integration, which is insufficient for long-term robustness. This points to the need for estimator redundancy or fault-tolerant sensor fusion in future designs.

From a control performance perspective, no saturation is observed in the translational thrusters. However, the torque control channel exhibits saturation during Phase 1 of the worst-case scenario. This indicates that while the controller can manage large attitude errors, it is not yet optimized for control effort. Similarly, the current guidance strategy is not fuel-optimal meaning that manoeuvrers are computed based on convergence rather than fuel efficiency. Implementing a trajectory planner or fuel-optimized guidance law would be a natural next step.

It should also be noted that the current simulations do not include external disturbances (e.g., gravitational perturbations, solar radiation pressure). To further validate the robustness of the control system, such disturbance models should be integrated into future simulation campaigns.

In summary, the simulation results successfully validate the baseline controller and state estimator for autonomous docking. The system demonstrates robustness to varying initial conditions, constraint compliance, and strong estimation performance. However, further improvements in robustness, efficiency, and fault tolerance are necessary to ensure suitability for deployment in operational space missions.

Bibliography

[1] S. Dascanio. *Design of Guidance and Algorithms for Space Rendezvous Maneuvers*. PhD thesis, Politecnico di Torino, 2021.

[2] F. Stesina. Tracking model predictive control for docking maneuvers of a cubesat with a big spacecraft. *Aerospace*, 8(8):197, 2021.

[3] Y. Ariba, D. Arzelier, L. S. Urbina, and C. Louembet. V-bar and r-bar glideslope guidance algorithms for fixed-time rendezvous: A linear programming approach. *IFAC-PapersOnLine*, 49(17):385–390, 2016.

[4] R. Zanetti. Optimal glideslope guidance for spacecraft rendezvous. *Journal of guidance, control, and dynamics*, 34(5):1593–1597, 2011.

[5] Wikipedia contributors. Space rendezvous, 2025. URL `https://en.wikipedia.org/wiki/Space_rendezvous`. [Online; accessed 28-May-2025].

[6] Wikipedia contributors. Spacecraft attitude determination and control, 2025. URL `https://en.wikipedia.org/wiki/Spacecraft_attitude_determination_and_control`. [Online; accessed 28-May-2025].

[7] FIBERPRO. Single axis fiber optic gyroscope for space grade. `https://fiberpro.com/product_upload/SAG-SS_Single%20Axis%20Fiber%20Optic%20Gyroscope%20for%20space%20grade.pdf`. Accessed: 2025-05-08.

[8] R. Lab. St-16rt2 data sheet. `https://www.rocketlabusa.com/assets/Uploads/ST-16RT2-Datasheet-v4.1.pdf`. Accessed: 2025-05-08.

[9] International Space Station Multilateral Control Board. International docking system standard (idss) interface definition document (idd). Technical Report Revision F, NASA, July 2022. URL `https://www.internationaldockingstandard.com/download/IDSS_IDD_Revision_F.pdf`. Approved for Public Release.

[10] NASA. "Apollo 11 Mission Report". Technical report, NASA, 1969. URL `https://sma.nasa.gov/SignificantIncidents/assets/a11_missionreport.pdf`. Accessed: 2025-03-06.

[11] H. Aerospace. Hr16-100 reaction wheel. `https://www.satcatalog.com/component/hr16-100/`, 2020. Accessed: 2025-05-08.