



POLITECNICO DI TORINO

Master Degree course in Mechatronic Engineering

Master Degree Thesis

**MPC for Free-Floating
Satellite–Manipulator Systems: LPV
Prediction and Tube-Based Robust
Control**

Supervisor

prof. Mauro Mancini

Candidate

Ernesto Di Rienzo

Co-Supervisor

prof. Fabio Faliero

ACADEMIC YEAR 2025–2026

Abstract

Space manipulators are receiving increasing attention in on-orbit servicing, debris-removal, and capture missions, where manipulator motion and spacecraft attitude are dynamically coupled through internal momentum exchange. In these systems, actuator limits, nonlinear dynamics, and post-capture inertial changes make constrained control a challenging problem.

This thesis develops and validates a Model Predictive Control (MPC) framework for a planar free-floating spacecraft equipped with a two-link manipulator. The control design is based on a reduced-order model in which the spacecraft base twist is eliminated through momentum conservation, while the nonlinear simulation still propagates the spacecraft attitude quaternion for kinematic consistency and diagnostic checks. This separation keeps the online optimization problem compact while preserving the dominant base–arm coupling effects in the plant.

Starting from the reduced nonlinear model, discrete-time linear predictors are constructed around selected operating conditions. Two nominal formulations are considered: a Linear Time-Invariant (LTI) MPC based on a single linear predictor, and a Linear Parameter-Varying (LPV) MPC obtained from multiple local linearizations and online interpolation. A tube-based robust MPC formulation is then introduced to handle bounded mismatch and payload uncertainty.

The framework is implemented in MATLAB/Simulink and assessed over nominal regulation, end-effector tracking, post-capture regulation with payload mismatch, online payload-estimate updates, and retained stressed post-capture scenarios. The results show that the single-predictor formulation already provides a very strong baseline for final-residual minimization in local and moderately stressed regulation tasks. The LPV predictor captures local operating-point dependence more explicitly, but in the reported dataset it does not provide a systematic final-accuracy advantage over the LTI baseline. The robust formulation is useful when lower torque peaks and more conservative transients are preferred under bounded mismatch, although these gains come with stronger conservatism, stricter target semantics, and a substantially larger computational burden.

Overall, the thesis provides a control-oriented benchmark in which predictor choice, robustness mechanisms, torque demand, and model-mismatch effects can be compared on the same reduced plant and under shared evaluation assumptions.

Contents

1	Introduction	1
1.1	Motivation and Background	1
1.2	On-Orbit Manipulation and Servicing Missions	2
1.3	Control Approach: LPV Model Predictive Control and Robust Tubes	3
1.4	Case Studies and Evaluation Criteria	4
1.5	Reproducibility Notes	5
1.6	Research Objectives, Gaps, and Thesis Organization	5
2	Mathematical Model of the System	8
2.1	Free-Floating Spacecraft–Manipulator System	8
2.2	Kinematics and Mass–Inertia Properties	12
2.3	Energy-Based Modeling and Equations of Motion	13
2.4	Reduced Model for Control Design (Schur Complement Form)	14
2.5	Controller Variables, Outputs, and Constraints	15
2.6	Scheduling Variables	16
3	Dynamics Model	18
3.1	Inertia Blocks using Symbolic Composite Rigid-Body Modeling	18
3.2	Free-Floating Reduction and Reduced Joint-Space Dynamics .	20
3.3	Numerical Safeguards and Runtime Diagnostics	22
3.4	Attempted Full-Dynamics Prototype and Observed Limitations	24
3.5	Plant Structure and Equation Traceability	28
4	Linearization and LPV Prediction Model Construction	30
4.1	Why an LPV Surrogate: LTI vs LTV vs LPV	31
4.2	Continuous-Time Linearization and Discretization	32
4.3	Prediction Mismatch and Corner-Page Structure	35
4.4	Simplex-Based Scheduling and Online Weight Computation . .	36
4.5	Output Model, Bias Handling, and Consistency Checks	38

5	MPC Design	40
5.1	Common Formulation	40
5.2	Nominal MPC with a Single Predictor	42
5.3	Nominal MPC with an LPV Predictor	43
5.4	Tube-Based Robust MPC	44
5.5	Controller Interface and Practical Remarks	46
6	Simulation Setup and Metrics	48
6.1	Closed-Loop Simulation Architecture	48
6.2	Simulation Cases and Setup	50
6.3	Performance Metrics and Validation	51
7	Comparative Simulation Results	57
7.1	Reading Guide	57
7.2	Case 1: Nominal Regulation	58
7.3	Case 2: Pre-Capture Tracking and End-Effector Performance .	59
7.4	Case 3: Post-Capture Regulation Under Payload Mismatch . .	61
	7.4.1 Case 3A: Fixed Payload Mismatch	62
	7.4.2 Case 3B: Online Payload-Estimate Update	63
7.5	Case 4: Stressed Post-Capture Regulation Under Payload Uncertainty and Additive Mismatch	64
7.6	Cross-Case Synthesis	66
8	Conclusion and Future Work	67
8.1	Summary of Contributions	67
8.2	Future Research Directions	68

List of Figures

2.1	Schematic representation of the planar free-floating satellite–manipulator benchmark considered in this thesis. The figure shows the inertial frame $\{I\}$, the satellite body frame $\{S\}$, the two-link revolute arm, the joint locations, the link centers of mass, and the end-effector frame $\{EE\}$. The offset d_0 identifies the fixed distance between the spacecraft center of mass and the first joint along the positive x_S direction.	9
3.1	Structure of the reduced nonlinear plant used in the simulation campaign. The main blocks correspond to inertia construction, quaternion propagation and normalization, Coriolis/centrifugal terms, joint-acceleration computation, and momentum-based velocity composition. This organization provides a direct link between the nonlinear equations developed in this chapter and the plant used later for simulation, linearization, and controller validation.	28
6.1	Closed-loop MATLAB/Simulink architecture used in the comparative campaign. The reference block provides the current sample and preview information, the controller block solves the selected optimization problem and outputs the commanded torque, and the nonlinear plant propagates the reduced free-floating dynamics. The output error e is computed from the reference and the measured controlled output and is used for monitoring and KPI extraction.	49
7.1	Representative Case 1 regulation run. The top row shows the joint-position error, whereas the bottom row shows the zoomed torque histories. The nominal LTI controller achieves the smallest final residual, the LPV controller remains close in transient behavior, and the robust controller reduces the torque peaks at the price of slower convergence.	59

7.2	Representative Case 2 end-effector tracking run for the more demanding trajectory. The top row compares the output trajectories against their references, whereas the bottom row shows the output-error norm. The robust controller yields a smoother transient and lower integrated tracking error, while the nominal LTI controller preserves the smallest terminal residual.	61
7.3	Representative Case 3A run for the heavier fixed payload mismatch. The top row shows the joint-position error, whereas the bottom row shows the zoomed torque histories. The nominal controllers preserve smaller residual errors, while the robust controller limits the torque peak more effectively at the price of a much larger solve-time burden.	62
7.4	Representative Case 3B run for an upward payload-estimate correction. The top row shows the joint-position error, whereas the bottom row shows the zoomed torque histories. The correction improves the final residual with respect to the corresponding fixed-mismatch case, while the overall controller ordering remains qualitatively unchanged.	64
7.5	Representative Case 4 stressed run for the more demanding reported scenario. The top row shows the joint-position error, whereas the bottom row shows the zoomed torque histories. The nominal LTI controller preserves the smallest final residual, while the robust controller markedly reduces the torque peaks at the price of a much larger computational burden.	65

List of Tables

6.1	Common simulation settings used throughout the reported campaign. Only the benchmark scenario, output mode, reference type, and case-dependent mismatch settings vary across cases.	54
6.2	Overview of the reported deterministic case studies.	55
6.3	KPI families used throughout the comparative discussion. . . .	56
7.1	Case 1 aggregated KPIs, averaged over the five tested initial conditions.	58
7.2	Case 2 aggregated KPIs, averaged over the two retained trajectory-tracking scenarios.	60
7.3	Case 2 predictor-validity diagnostics, averaged over the two retained trajectory-tracking scenarios.	60
7.4	Case 3A aggregated KPIs, averaged over the retained fixed-mismatch scenarios.	62
7.5	Case 3B aggregated KPIs, averaged over the retained payload-update scenarios.	63
7.6	Case 4 aggregated KPIs, averaged over the retained stressed scenarios.	65

Chapter 1

Introduction

1.1 Motivation and Background

On-orbit servicing (OOS) and in-space assembly/manufacturing missions are increasingly relying on robotic manipulation. In these scenarios, a servicer spacecraft must interact with a target satellite or with orbital infrastructure while preserving safety, respecting operational constraints, and maintaining sufficient motion accuracy. When a robotic arm is mounted on a spacecraft, however, the manipulation task cannot be treated as an isolated arm-control problem: joint motion induces a reaction on the base, and this base motion in turn affects the end-effector behavior. Under free-floating conditions, where external actuation is intentionally not used during the maneuver, this dynamic coupling becomes one of the central control challenges [1]–[4].

From an operational viewpoint, space manipulator systems are commonly discussed in terms of different actuation modes. In *free-floating* operation, no external actuation is used and the motion is governed by momentum conservation. In *partial free-floating* operation, some attitude support may still be available, while translational motion remains unactuated [2]. The benchmark studied in this thesis focuses on the fully free-floating case, which represents the most strongly coupled and control-demanding regime among those considered.

The system considered here is a planar spacecraft–manipulator benchmark composed of an unactuated base and a two-link revolute arm. The simulations start from zero total momentum, so that the base motion is generated only by the manipulator dynamics and can be reconstructed from the joint evolution through momentum conservation [5], [6]. This enables a reduced-order control formulation in joint space, while the nonlinear simulation retains the spacecraft attitude evolution for physical consistency.

A second key aspect of the problem is that prediction mismatch cannot be neglected. Even with a carefully derived reduced model, the controller must operate in the presence of discrepancies caused by model reduction, local linear approximation, discretization, and payload uncertainty. In this thesis, these effects are treated in a control-oriented way as bounded mismatch, so that they can be incorporated consistently into the robust design developed later.

Overall, the thesis addresses a constrained manipulation-control problem under free-floating conditions. The objective is not to solve the most general spacecraft-control problem, but to build a coherent benchmark in which reduced-state prediction, constraint handling, and robustness to bounded mismatch can be studied within a common comparative framework.

1.2 On-Orbit Manipulation and Servicing Missions

Several space missions have highlighted the relevance of robotic manipulation for future orbital operations. Early demonstrations such as ETS-VII showed clearly that manipulator motion can induce significant spacecraft reactions, making the control of the coupled base–arm system a practical engineering issue rather than only a theoretical one [1], [2], [7], [8].

Robotic systems on the International Space Station, including Canadarm2 and the European Robotic Arm, have further demonstrated the operational value of orbital manipulation, even though they do not operate in free-floating conditions [2], [9], [10]. Technology demonstrations such as the Robotic Refueling Mission series have also shown that orbital manipulation is shaped by strict operational constraints, uncertainty, and demanding task sequencing [11]. More recent studies have emphasized the importance of coordinated spacecraft–arm control during close-proximity and capture operations, where the dynamic interaction between the base and the manipulator becomes even more critical [12].

The benchmark adopted in this thesis is intentionally simplified, but it is designed to retain the essential coupled-control problem. The system is planar, free-floating, and equipped with a two-link revolute arm. External forces and torques are neglected over the maneuver window, and the simulations start from zero total momentum. Within this setting, the nonlinear plant propagates the spacecraft attitude for consistency, whereas the controller operates on a reduced joint-space model obtained by eliminating the base twist through momentum conservation [5], [6].

This choice defines the scope of the thesis. The aim is not to reproduce all aspects of a complete mission scenario, but to isolate the coupled manipulation problem in a setting that is physically meaningful and suitable for a systematic comparison across multiple control formulations.

1.3 Control Approach: LPV Model Predictive Control and Robust Tubes

This thesis adopts a Model Predictive Control (MPC) framework built on a reduced-order model of the free-floating spacecraft–manipulator system. At each sampling instant, the controller computes joint torques by solving a constrained optimization problem that explicitly accounts for actuator saturation, joint-position limits, joint-velocity bounds, and, when required, end-effector-related output constraints. The prediction model is based on the reduced joint-space dynamics, while the nonlinear simulation is used to assess the closed-loop behavior of the coupled system.

The control problem studied here is therefore not a full spacecraft-attitude MPC problem. Instead, it is a constrained manipulation-control problem in which the dominant base–arm coupling is represented through the reduced free-floating dynamics. This allows the benchmark to remain focused on the interaction between prediction quality, constraint handling, and robustness, without moving to a fully nonlinear online optimization framework.

To capture the main dependence of the dynamics on the operating condition while preserving a tractable quadratic-program structure online, the thesis employs a scheduled linear prediction framework. In particular, the reduced nonlinear model is approximated through a family of local linear predictors defined over a relevant operating region, leading to an LPV-type surrogate [13], [14]. This makes it possible to compare different prediction strategies without abandoning the computational structure of linear MPC.

Three controller formulations are considered within a common framework. The first is a nominal MPC with a single linear predictor and serves as an LTI baseline. The second is a nominal MPC with a scheduled LPV predictor, introduced to improve prediction fidelity over a wider operating region. The third augments the nominal framework with a tube-based robust layer designed to handle bounded mismatch and payload uncertainty [15], [16].

For regulation and tracking tasks, the formulations adopt a constrained-target strategy so that the optimizer can handle aggressive or partially incompatible references in a controlled way, rather than enforcing them rigidly

under all conditions [17], [18]. In this way, feasibility and target consistency can be addressed within a common predictive-control structure.

Post-capture conditions are represented through a payload model at the end-effector, with a distinction between the true payload used in the nonlinear plant and the payload estimate used in the predictor. This allows the effect of parametric mismatch to be studied in a controlled way and supports a progressive comparison between nominal and robust controllers under increasingly demanding operating conditions.

The complete framework is implemented in MATLAB/Simulink and is used here as a comparative simulation environment for controller assessment under shared assumptions.

1.4 Case Studies and Evaluation Criteria

The simulation campaign is organized into four case studies:

- (i) *Case 1* (nominal regulation), used to verify basic closed-loop consistency and constraint handling under free-floating conditions without payload uncertainty;
- (ii) *Case 2* (pre-capture tracking/deployment), used to assess tracking performance and output-constrained end-effector behavior;
- (iii) *Case 3* (post-capture regulation with payload), used to quantify the effect of payload uncertainty and compare nominal versus robust MPC;
- (iv) *Case 4* (payload plus additive disturbance), used as a stress test for feasibility and constraint satisfaction under bounded mismatch.

In post-capture conditions, an additional step change in the payload estimate is introduced to emulate an onboard update of the prediction model and to observe the controller response to an abrupt change in the assumed parameters.

The compared controllers are evaluated through a common set of indicators. These include tracking measures, constraint-related quantities, computational metrics, and selected physics-based diagnostics used to verify the numerical consistency of the free-floating implementation. Unless otherwise stated, the controllers are compared under shared initialization, solver, and main weighting assumptions, so that the observed differences can be interpreted mainly as consequences of the prediction model and robustness layer rather than of case-specific retuning choices.

1.5 Reproducibility Notes

The reported case studies are generated within a shared simulation setup. Physical parameters, prediction models, controller settings, and scenario-dependent options are built from a common initialization procedure, and the same overall MATLAB/Simulink architecture is used throughout the campaign. This is important for the comparison, because it limits artificial differences caused by inconsistent parameter handling across cases.

All simulations store the signals needed to recompute the reported metrics offline, including tracking errors, state/input/output constraint indicators, QP solve times, and the diagnostics later used to interpret LPV locality and robust-controller behavior. These signals are post-processed into KPI tables and representative plots, while the raw time histories are preserved for traceability.

Within this shared setup, the developments specific to this thesis concern the reduced spacecraft–manipulator model integration, the generation and management of the scheduled predictors, the payload-dependent predictor updates used in the post-capture studies, and the robust-control components adopted in the final case studies.

1.6 Research Objectives, Gaps, and Thesis Organization

The main ingredients considered in this thesis are individually well known: reduced free-floating modeling, linear predictive control, LPV scheduling, and robust MPC. The specific gap addressed here is their joint use within the same control-oriented benchmark for a free-floating spacecraft–manipulator system. In particular, the thesis focuses on whether changes in predictor structure and robustness logic produce practically relevant differences when the plant, the main constraints, and the simulation assumptions are kept aligned.

For this reason, the work is not organized around a single controller claim. Instead, it studies three predictive-control formulations on the same reduced benchmark: a nominal MPC with a single predictor, a nominal MPC with scheduled local predictors, and a tube-based robust MPC formulation for bounded mismatch conditions. This setting makes it possible to compare final residuals, transient behavior, torque demand, feasibility margins, and computational cost without mixing these effects with unrelated changes in plant definition or case-dependent controller architecture.

The workflow of the thesis therefore connects five elements: reduced non-linear modeling of the free-floating plant, local operating-point linearization, construction of scheduled predictors, nominal and robust MPC design, and closed-loop validation over representative maneuver scenarios. The resulting comparison is intended to answer not only whether a more elaborate controller can be built, but also when that additional complexity is actually useful in the present benchmark.

The work is guided by the following research questions:

1. How can free-floating spacecraft–manipulator dynamics be approximated by a reduced-order LPV surrogate that remains sufficiently accurate over the relevant operating region while preserving a tractable online MPC problem?
2. How can an MPC-based controller coordinate manipulation objectives under tight state, input, and output constraints while accounting for the dominant base–arm coupling effects?
3. How can a tube-based robust layer be added on top of the reduced prediction model to improve behavior under bounded mismatch and payload uncertainty while retaining a practically usable optimization structure?
4. What trade-offs arise between nominal and robust MPC formulations in representative on-orbit scenarios, in terms of final accuracy, transient response, torque peaks, constraint handling, and computational effort?

The main contributions of this thesis can be summarized as follows:

- A reduced modeling and simulation benchmark for a planar free-floating spacecraft–manipulator system, built for constrained-control studies and consistent closed-loop comparison.
- An LPV prediction framework generated from multiple local linearizations over a relevant operating region, with online scheduling based on convex interpolation.
- A unified MPC structure including a nominal single-predictor formulation, a nominal scheduled-predictor formulation, and a tube-based robust extension under shared benchmark assumptions.
- A comparative simulation campaign over representative regulation, tracking, and post-capture scenarios, reporting tracking, constraint-related, diagnostic, and computational metrics.

The remainder of this thesis is organized as follows. Chapter 2 introduces the system definition, reference frames, and modeling conventions. Chapter 3 develops the reduced nonlinear plant and discusses the numerical safeguards

adopted in simulation. Chapter 4 presents the linearization procedure and the construction of the scheduled prediction model. Chapter 5 formulates the nominal and robust MPC controllers. Chapter 6 describes the simulation architecture and the benchmark setup. Chapter 7 reports the comparative case studies and discusses the main observed trade-offs. Chapter 8 concludes the thesis and outlines possible research extensions.

Chapter 2

Mathematical Model of the System

2.1 Free-Floating Spacecraft–Manipulator System

On-orbit servicing missions often employ a robotic manipulator mounted on a free-flying spacecraft bus. In free-floating configurations, the base attitude and the manipulator motion are dynamically coupled through conservation of total momentum, and no external actuation is used to regulate the composite system motion. This chapter introduces the system architecture, reference frames, attitude representation, and the kinematic and mass–inertia notation adopted throughout the thesis. The energy-based equations of motion and the reduced model used for control design are then presented.

The system considered in this thesis consists of a rigid service spacecraft (“chaser”) carrying a serial robotic arm. The satellite body frame $\{S\}$ is attached to a reference point $CM0$ on the bus; by convention, the $+Sx$ axis points toward the location of the first joint of the robotic arm. The arm comprises n revolute joints and n rigid links L_i , $i = 1, \dots, n$, connected in an open kinematic chain. In the simulations presented in this thesis, a two-joint arm ($n = 2$) is considered in order to keep the reduced dynamics and the resulting MPC formulation sufficiently compact while still capturing the essential features of base–arm coupling.

Although the notation is kept generic for an n -link serial chain, the benchmark considered here is intentionally specialized to the planar two-joint case with homogeneous links. This simplification is adopted to keep the symbolic expressions, the numerical model, and the repeated linearization/MPC workflow tractable, while still retaining the essential free-floating base–arm

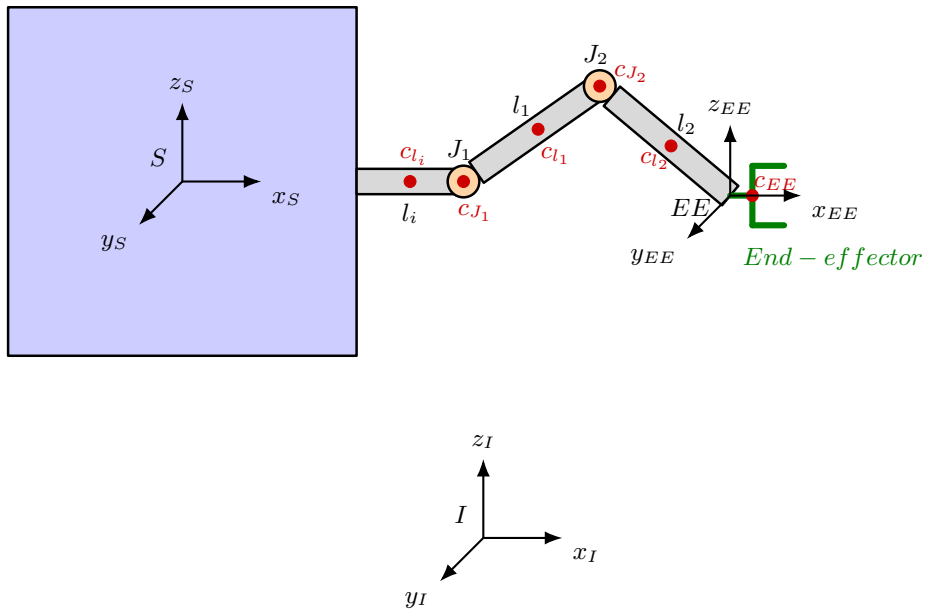


Figure 2.1: Schematic representation of the planar free-floating satellite-manipulator benchmark considered in this thesis. The figure shows the inertial frame $\{I\}$, the satellite body frame $\{S\}$, the two-link revolute arm, the joint locations, the link centers of mass, and the end-effector frame $\{EE\}$. The offset d_0 identifies the fixed distance between the spacecraft center of mass and the first joint along the positive x_S direction.

coupling mechanisms relevant to the control problem addressed in this thesis.

In the planar benchmark, the effective base attitude reduces to a rotation about the out-of-plane axis. The quaternion is nevertheless retained in the nonlinear plant so that the attitude representation remains compatible with the general 3D derivation.

The manipulator is connected to the spacecraft bus through a fixed offset of length d_0 , defined as the vector from the spacecraft CoM point CM0 to joint 1 along the $+Sx$ direction. In practice, d_0 is the equivalent constant offset used in the model and can be obtained from CAD by composing the fixed bus-to-mount and mount-to-joint distances along $+Sx$. This preserves the momentum-based formulation about CM0 while remaining consistent with the adopted geometric description.

Each link L_i is modeled as a rigid body with a body-fixed frame $\{L_i\}$ located at its center of mass (CoM), and with the link axis aligned with $+L_ix$. Joint positions and rates are collected in the vectors

$$\theta = [\theta_1, \dots, \theta_n]^\top, \quad \dot{\theta} = [\dot{\theta}_1, \dots, \dot{\theta}_n]^\top,$$

and the joint input torques are denoted by $\tau \in \mathbb{R}^n$.

To model grasped-object effects in a compact form, an effective payload mass m_p can be included in the inertia properties. In the numerical model, its contribution can be scaled by a coupling factor $\kappa \in [0, 1]$, where $\kappa = 0$ corresponds to no payload effect and $\kappa = 1$ to a fully coupled payload. This parametrization is convenient for uncertainty and scenario studies.

In this benchmark, the payload is modeled as an effective point mass attached at the end-effector. Therefore, only its translational contribution to the kinetic energy is retained, while payload rotational inertia is neglected. The coupling factor κ is used to scale the resulting effect continuously from the uncoupled case to the fully coupled case.

An inertial frame $\{I\}$ is introduced as a Newtonian reference frame fixed in space. All momentum quantities are ultimately referred to $\{I\}$, which simplifies the statement of conservation laws in free-floating mode [1]. Most kinematic quantities, such as positions and joint axes, are expressed in the satellite frame $\{S\}$.

The following notation is adopted throughout the chapter:

- $R_{IS}(\mathbf{q}) \in \text{SO}(3)$ denotes the rotation matrix from the satellite frame $\{S\}$ to the inertial frame $\{I\}$. Thus, for any vector Sx expressed in $\{S\}$, its representation in $\{I\}$ is ${}^Ix = R_{IS}(\mathbf{q}) {}^Sx$.
- $R_{SB}(\theta) \in \text{SO}(3)$ denotes the rotation matrix from a generic body-fixed frame $\{B\}$ (bus or link frame) to the satellite frame $\{S\}$. It is used to

express body inertias and directions consistently in $\{S\}$ before mapping them to $\{I\}$ through $R_{IS}(\mathbf{q})$.

- Positions are denoted by ${}^A p_B \in \mathbb{R}^3$, meaning the vector from the origin of frame $\{A\}$ to point B , expressed in $\{A\}$. In particular, the bus center-of-mass point CM0 is located at the origin of $\{S\}$, hence ${}^S p_{\text{CM0}} = 0$.
- The position of the link- i CoM in $\{S\}$ is denoted by ${}^S p_{L_i}(\theta)$, and the composite system CoM is denoted by ${}^S p_{\text{CoM}}(\theta)$.

The attitude of the satellite body frame $\{S\}$ with respect to $\{I\}$ is represented by a scalar-first unit quaternion

$$\mathbf{q} = [q_0 \ q_1 \ q_2 \ q_3]^\top, \quad \|\mathbf{q}\| = 1,$$

with vector part $\mathbf{q}_v = [q_1 \ q_2 \ q_3]^\top$. Throughout this thesis, $[a]_\times b \triangleq a \times b$ for any $a, b \in \mathbb{R}^3$. The corresponding rotation matrix $R_{IS}(\mathbf{q})$ is given by

$$R_{IS}(\mathbf{q}) = (q_0^2 - \mathbf{q}_v^\top \mathbf{q}_v) I_3 + 2 \mathbf{q}_v \mathbf{q}_v^\top + 2 q_0 [\mathbf{q}_v]_\times.$$

For later use, the base twist at CM0 is defined in both frames:

$$\nu_0^S = \begin{bmatrix} {}^S v_0 \\ {}^S \omega_0 \end{bmatrix} \in \mathbb{R}^6, \quad \dot{x}_0^I = \begin{bmatrix} {}^I v_0 \\ {}^I \omega_0 \end{bmatrix} \in \mathbb{R}^6.$$

They are related by

$$\dot{x}_0^I = \text{blkdiag}(R_{IS}(\mathbf{q}), R_{IS}(\mathbf{q})) \nu_0^S.$$

The symbol \dot{x}_0^I denotes the inertial-frame base twist stacked at CM0, not the time derivative of a minimal pose parametrization. This notation is retained for consistency with the momentum mapping used later in the model.

Kinematic Jacobians are expressed in $\{S\}$, while momentum and energy quantities are computed in $\{I\}$ using \dot{x}_0^I ; the two representations are mapped through $R_{IS}(\mathbf{q})$ as in the relation above. In the numerical model adopted in this thesis, \dot{x}_0^I is used to compute momentum quantities and to propagate the quaternion.

In the formulation adopted here, the quaternion is propagated using the inertial-frame angular velocity ${}^I \omega_0$:

$$\dot{\mathbf{q}} = \frac{1}{2} \Omega_L({}^I \omega_0) \mathbf{q}, \quad \Omega_L(\omega) = \begin{bmatrix} 0 & -\omega^\top \\ \omega & -[\omega]_\times \end{bmatrix}. \quad (2.1)$$

Equivalent forms can be written using the body-rate ${}^S \omega_0$ and a right-product matrix; the present convention is selected to remain consistent with the adopted modeling framework.

The focus of this thesis is on the *free-floating* operating mode, in which no external control forces or torques are applied to the spacecraft bus:

- No thrusters or translational actuators are used. The inertial CoM velocity is constant; if the initial linear momentum is zero, the inertial CoM remains fixed [2].
- Momentum-exchange devices (reaction wheels, CMGs) are kept inactive in the baseline model; only joint torques τ act as control inputs.
- All bodies are rigid, with fixed mass and inertia tensors; flexibility and elastic joints are neglected.
- External disturbances (drag, gravity gradient, SRP, magnetic torques) are neglected in the nominal model and treated later as bounded disturbances in the robust MPC formulation.

Under these assumptions, the total linear and angular momentum of the composite system are conserved in $\{I\}$. These invariants provide consistency checks for the numerical implementation and form the foundation for the reduced model used for control design [1].

2.2 Kinematics and Mass–Inertia Properties

This section summarizes the kinematic relations and mass–inertia quantities needed to build the generalized inertia matrix and the composite center of mass.

Manipulator kinematics are formulated in the satellite frame $\{S\}$. Let P be an arbitrary point rigidly attached to the manipulator (end-effector, link CoM, etc.), and let ${}^S p_P(\theta) \in \mathbb{R}^3$ be its position in $\{S\}$. Its combined velocity expressed in $\{S\}$ can be written as

$$\begin{bmatrix} {}^S v_P \\ {}^S \omega_P \end{bmatrix} = J_0^P(\theta) \nu_0^S + J_m^P(\theta) \dot{\theta}, \quad (2.2)$$

where $J_0^P(\theta) \in \mathbb{R}^{6 \times 6}$ maps the base twist to the twist at P , and $J_m^P(\theta) \in \mathbb{R}^{6 \times n}$ is the manipulator geometric Jacobian at P . The base contribution follows from rigid-body kinematics:

$$J_0^P(\theta) = \begin{bmatrix} I_3 & -[{}^S p_P(\theta)]_{\times} \\ 0_{3 \times 3} & I_3 \end{bmatrix}. \quad (2.3)$$

Let $B \in \{S, L_1, \dots, L_n\}$ denote a rigid body of the system. Its parameters are mass $m_B > 0$, CoM inertia tensor ${}^B I_B^C \in \mathbb{R}^{3 \times 3}$ expressed in the body-fixed CoM frame, and CoM position ${}^S p_B(\theta)$ in $\{S\}$. The inertia about the origin of $\{S\}$, expressed in $\{S\}$, is obtained using rotation and parallel-axis shift:

$${}^S I_B(\theta) = R_{SB}(\theta) {}^B I_B^C R_{SB}^{\top}(\theta) + m_B [{}^S p_B(\theta)]_{\times} [{}^S p_B(\theta)]_{\times}^{\top}. \quad (2.4)$$

For the satellite bus at CM0, $s_{p_S} = 0$ and $s_{I_S} \equiv s_{I_S^C}$.

The explicit dependence of the inertia blocks on the quaternion follows from expressing the generalized base twist through the inertial-frame quantity \dot{x}_0^I . An equivalent formulation based on body-frame angular rates could reduce this dependence, but the present convention is retained to remain consistent with the chosen quaternion propagation.

Let m_S be the satellite mass and m_{L_i} the link masses. Since the origin of $\{S\}$ is placed at CM0, the satellite contribution to the CoM numerator is zero. The total mass can include an effective payload term κm_p . Thus,

$$M_{\text{tot}} = m_S + \sum_{i=1}^n m_{L_i} + \kappa m_p, \quad s_{p_{\text{CoM}}}(\theta) = \frac{\sum_{i=1}^n m_{L_i} s_{p_{L_i}}(\theta) + \kappa m_p s_{p_p}(\theta)}{M_{\text{tot}}}, \quad (2.5)$$

where $s_{p_p}(\theta)$ is the payload attachment point, typically coincident with the end-effector position.

Collect the generalized velocity vector as

$$\xi \triangleq \begin{bmatrix} \dot{x}_0^I \\ \dot{\theta} \end{bmatrix} \in \mathbb{R}^{6+n}. \quad (2.6)$$

The kinetic energy can be written as

$$T(\mathbf{q}, \theta, \xi) = \frac{1}{2} \xi^\top H(\mathbf{q}, \theta) \xi, \quad (2.7)$$

where the generalized inertia matrix admits the partition

$$H(\mathbf{q}, \theta) = \begin{bmatrix} H_0(\mathbf{q}, \theta) & H_{0m}(\mathbf{q}, \theta) \\ H_{0m}^\top(\mathbf{q}, \theta) & H_m(\mathbf{q}, \theta) \end{bmatrix}. \quad (2.8)$$

This partition matches the structure used in the numerical model and in generalized Jacobian formulations for floating-base systems [1].

2.3 Energy-Based Modeling and Equations of Motion

Under the modeling assumptions above, the Lagrangian coincides with the kinetic energy. The equations of motion can be written in the standard robot-dynamics form:

$$H(\mathbf{q}, \theta) \dot{\xi} + C(\mathbf{q}, \theta, \xi) \xi = \begin{bmatrix} 0_{6 \times 1} \\ \tau \end{bmatrix}, \quad (2.9)$$

where $C(\mathbf{q}, \theta, \xi) \in \mathbb{R}^{(6+n) \times (6+n)}$ collects Coriolis and centripetal terms. The matrix C can be chosen such that $\dot{H} - 2C$ is skew-symmetric, ensuring the usual power-balance property.

Define the inertial-frame momentum vector associated with the base coordinates by stacking the total linear momentum and the total angular momentum about CM0, namely

$$h \triangleq \begin{bmatrix} P_I \\ L_{\text{CM0}} \end{bmatrix} = H_0(\mathbf{q}, \theta) \dot{x}_0^I + H_{0m}(\mathbf{q}, \theta) \dot{\theta} \equiv M_0. \quad (2.10)$$

In free-floating mode, this quantity is conserved:

$$\dot{M}_0 = 0. \quad (2.11)$$

This is the key structural property exploited to derive a reduced model suitable for control design [1], [2].

2.4 Reduced Model for Control Design (Schur Complement Form)

The numerical model adopted in this thesis exploits momentum conservation to eliminate the base twist from the dynamics, yielding a compact joint-space model complemented by quaternion kinematics.

From (2.10), the base twist can be expressed as

$$\dot{x}_0^I = H_0^{-1}(\mathbf{q}, \theta) (M_0 - H_{0m}(\mathbf{q}, \theta) \dot{\theta}). \quad (2.12)$$

In the simulation scenarios used for controller design, the initial momentum is set to zero, $M_0 = 0$. Therefore,

$$\dot{x}_0^I = -K(\mathbf{q}, \theta) \dot{\theta}, \quad K(\mathbf{q}, \theta) \triangleq H_0^{-1}(\mathbf{q}, \theta) H_{0m}(\mathbf{q}, \theta), \quad (2.13)$$

which matches the generalized Jacobian framework commonly used for floating-base spacecraft–manipulator systems [1], [5], [6].

The base angular velocity ${}^I\omega_0$ extracted from \dot{x}_0^I is then used in (2.1) to propagate the quaternion.

Partitioning (2.9) and eliminating \ddot{x}_0^I through the Schur complement yields the reduced inertia matrix

$$H^*(\mathbf{q}, \theta) = H_m(\mathbf{q}, \theta) - H_{0m}^\top(\mathbf{q}, \theta) H_0^{-1}(\mathbf{q}, \theta) H_{0m}(\mathbf{q}, \theta), \quad (2.14)$$

which is symmetric positive definite under standard inertia assumptions. The reduced joint dynamics can be written as

$$H^*(\mathbf{q}, \theta) \ddot{\theta} = \tau - C^*(\mathbf{q}, \theta, \dot{\theta}) \dot{\theta} + d_\theta(t), \quad (2.15)$$

where $C^*\dot{\theta}$ collects velocity-dependent terms and d_θ represents unmodeled effects and disturbances. Reduced joint-space formulations of this type are commonly used in the literature for zero-momentum spacecraft–manipulator systems [2].

The term $d_\theta(t)$ is introduced only as a control-oriented lumped residual. It collects effects that are present in the nonlinear plant but are not retained explicitly in the compact prediction model, including attitude-dependent terms neglected online, linearization and discretization residuals, and parametric mismatch. In the discrete-time MPC formulation, this continuous-time residual is later represented by an equivalent bounded additive mismatch $w_k \in W$. This choice preserves the reduced free-floating structure while keeping the predictor compact enough for LPV page generation and QP-based MPC.

The simulation state used in subsequent chapters is

$$x_{\text{sim}} = \begin{bmatrix} \mathbf{q} \\ \theta \\ \dot{\theta} \end{bmatrix},$$

where the quaternion is propagated through (2.1) and the base twist is recovered algebraically from (2.13). For control design, this same residual is later represented in the discrete-time predictor as an equivalent bounded additive mismatch $w_k \in W$, so that the reduced model remains compact and directly usable for LPV prediction and MPC design.

A further approximation is introduced only at predictor-construction level: in Chapter 4, the quaternion entering the inertia packing is frozen to the normalized initial attitude q_0 , and the resulting residual is absorbed into the bounded mismatch description.

This reduced model is the basis for the controller state, the operating-point linearizations, the LPV predictor pages, and the robust mismatch description used in the following chapters. Consequently, the controller comparisons must be interpreted with respect to this reduced joint-space model rather than with respect to a full six-degree-of-freedom spacecraft state representation [1], [5], [6].

2.5 Controller Variables, Outputs, and Constraints

For control purposes, the prediction model used by the MPC is expressed in the reduced state

$$x = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \in \mathbb{R}^{2n}, \quad u \equiv \tau \in \mathbb{R}^n,$$

while the plant simulation additionally propagates the spacecraft attitude quaternion \mathbf{q} using inertial-rate kinematics consistent with the free-floating momentum composition.

Two output configurations are considered:

- (i) *state-output mode*, where $y = x$;
- (ii) *end-effector-output mode*, where the constrained output collects planar end-effector quantities, $y = [x_{ee}, y_{ee}, v_x, v_y]^\top$.

In end-effector-output mode, the output is evaluated through a local linearization around the operating point,

$$y = C_0 x + y_{\text{bias}},$$

where y_{bias} enforces consistency between the nonlinear output and its linear approximation at the operating point. In the planar case considered here, the bias affects only the position components. For constraint handling in the MPC, this bias is absorbed into shifted references and bounds, so that the constrained output map remains linear in the optimization variable.

The end-effector-output mode should therefore be interpreted as a local constraint/tracking surrogate around the selected operating point, not as a globally exact kinematic output map over the full workspace.

Hard bounds are imposed on joint positions and rates, $\theta \in [\theta_{\min}, \theta_{\max}]$, $\dot{\theta} \in [\dot{\theta}_{\min}, \dot{\theta}_{\max}]$, and on input torques $u \in [u_{\min}, u_{\max}]$. In end-effector-output mode, the same shift is applied to the output bounds.

Accordingly, the plant and the controller do not use the same state and output description. The nonlinear simulation propagates the quaternion and reconstructs base motion from momentum composition, whereas the MPC layer uses only the reduced joint-space state and, when required, a local affine end-effector output map. This separation keeps the prediction model compact while preserving the dominant base reaction in the plant. In end-effector-output mode, any loss of tracking quality far from the selected operating point should therefore be interpreted as output-model mismatch rather than as an inconsistency of the constrained MPC formulation itself.

2.6 Scheduling Variables

For later use in the LPV predictor, the scheduling vector is chosen as

$$s \triangleq \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \in \mathbb{R}^{n_s}, \quad n_s = 2n.$$

The operating region is defined as a hyper-rectangle in s . The interpolation and simplex-partition strategy used for online LPV model assembly are introduced in Chapter 4.

In the predictor adopted here, the geometric scheduling coordinates are the joint positions and rates collected in s . Payload-mass estimates and coupling-factor estimates are not treated as additional simplex coordinates; instead, they are handled through separate page blocks generated offline and selected or interpolated only at block level. This separation prevents an unnecessary growth of the simplex dimension and keeps the local scheduling geometry tied only to the state-dependent variables $(\theta, \dot{\theta})$.

Chapter 3

Dynamics Model

This chapter derives the nonlinear plant used in simulation and in the linearization pipeline. The adopted formulation is free-floating and control-oriented: the spacecraft base is unactuated, base motion is induced by internal joint torques, and the resulting dynamics are governed by inertial-frame momentum conservation. All conventions (frames, scalar-first quaternions, and the inertial base twist \dot{x}_0^I) follow Chapter 2. The final part of the chapter documents the correspondence between the main equations and the structure of the nonlinear plant used in the numerical study.

Throughout this chapter, external forces and torques acting on the spacecraft–manipulator system are assumed negligible over the maneuver time window. Under this hypothesis, total linear and angular momentum are conserved, and these quantities are used both as structural properties of the model and as consistency checks for the implemented plant.

3.1 Inertia Blocks using Symbolic Composite Rigid-Body Modeling

The nonlinear plant uses the same free-floating spacecraft–manipulator benchmark introduced in Chapter 2: a rigid unactuated bus, a planar n -DoF serial arm (with $n = 2$ in this thesis), and joint torques $\tau \in \mathbb{R}^n$ as the only control inputs. Payload effects are represented through an effective point mass m_p attached at the end-effector and scaled by a coupling factor $\kappa \in [0, 1]$.

The payload is modeled as a point mass at the end-effector; therefore, only its translational kinetic-energy contribution is retained, while payload rotational inertia is neglected.

The generalized velocity used by the plant is

$$\begin{aligned}\xi &\triangleq \begin{bmatrix} \dot{x}_0^I \\ \dot{\theta} \end{bmatrix} \in \mathbb{R}^{6+n}, \\ \dot{x}_0^I &= \begin{bmatrix} I v_0 \\ I \omega_0 \end{bmatrix}.\end{aligned}\tag{3.1}$$

The kinetic energy is expressed in quadratic form as

$$T(\mathbf{q}, \theta, \xi) = \frac{1}{2} \xi^\top H(\mathbf{q}, \theta) \xi,\tag{3.2}$$

where $H(\mathbf{q}, \theta)$ is symmetric positive definite and partitioned as

$$H(\mathbf{q}, \theta) = \begin{bmatrix} H_0(\mathbf{q}, \theta) & H_{0m}(\mathbf{q}, \theta) \\ H_{0m}^\top(\mathbf{q}, \theta) & H_m(\mathbf{q}, \theta) \end{bmatrix},\tag{3.3}$$

with $H_0 \in \mathbb{R}^{6 \times 6}$, $H_{0m} \in \mathbb{R}^{6 \times n}$, and $H_m \in \mathbb{R}^{n \times n}$.

The inertia blocks are generated from composite rigid-body kinetic-energy contributions (bus, links, and payload point mass). To keep the symbolic expressions compact, the adopted formulation uses a homogeneous-link parametrization:

- identical link inertias: a single $J_l \in \mathbb{R}^{3 \times 3}$ is reused for all links;
- identical link masses: a single m_l is reused for all links;
- identical link length: a single scalar l_{arm} is reused for all links;
- the payload contribution is controlled by κm_p (with $\kappa = 0$ disabling payload coupling).

These assumptions keep the symbolic expressions manageable and are sufficient for the uncertainty and stress scenarios considered later. They should be interpreted as benchmark-oriented modeling choices rather than as general properties of spacecraft manipulators. They are adopted to keep the symbolic generation, numerical evaluation, and repeated linearization steps computationally tractable while preserving the dominant base–arm coupling mechanisms relevant to the control study.

To avoid online symbolic assembly, analytic expressions for $\{H_0, H_{0m}, H_m\}$ and their joint derivatives are generated offline and exported as fixed-dimension routines:

$$\begin{aligned}(H_0, H_{0m}, H_m) &= \mathcal{F}_H(\pi), \\ \left(\frac{\partial H_0}{\partial \theta}, \frac{\partial H_{0m}}{\partial \theta}, \frac{\partial H_m}{\partial \theta}\right) &= \mathcal{F}_{dH}(\pi),\end{aligned}\tag{3.4}$$

where the parameter vector π follows the ordering used consistently in the plant:

$$\pi = \begin{bmatrix} \mathbf{q} \\ \theta \\ m_s \\ m_l \\ m_p \\ \kappa \\ d_0 \\ l_{\text{arm}} \\ \text{vec}(J_l) \\ \text{vec}(J_s) \end{bmatrix}. \quad (3.5)$$

Here d_0 is the $+Sx$ offset from the bus CoM (CM0) to joint 1. In the numerical model, n is kept as a fixed configuration parameter so that matrix dimensions remain constant throughout the plant.

3.2 Free-Floating Reduction and Reduced Joint-Space Dynamics

The free-floating equations of motion can be written in standard manipulator form:

$$H(\mathbf{q}, \theta) \dot{\xi} + C(\mathbf{q}, \theta, \xi) \xi = \begin{bmatrix} 0_{6 \times 1} \\ \tau \end{bmatrix}, \quad (3.6)$$

where the upper six generalized forces are zero because the base is unactuated.

A key structural quantity is the inertial-frame momentum mapping

$$h \triangleq \begin{bmatrix} P_I \\ L_{\text{CM0}} \end{bmatrix} = H_0(\mathbf{q}, \theta) \dot{x}_0^I + H_{0m}(\mathbf{q}, \theta) \dot{\theta}, \quad (3.7)$$

which is conserved in the absence of external wrenches.

The stacked momentum mapping $h = [P_I^\top \ L_{\text{CM0}}^\top]^\top$ corresponds to the conserved momentum vector introduced in Chapter 2, written here with an explicit separation between linear and angular components.

All simulation cases used for controller validation start from a zero total momentum condition $h(0) = 0$. Therefore, the base twist is algebraically determined by

$$\dot{x}_0^I = -K(\mathbf{q}, \theta) \dot{\theta}, \quad K(\mathbf{q}, \theta) \triangleq H_0^{-1}(\mathbf{q}, \theta) H_{0m}(\mathbf{q}, \theta). \quad (3.8)$$

This relation is used directly in the plant to recover the base twist from the joint rates under zero-momentum conditions. The resulting momentum-based

reduction and Schur-complement formulation are standard in free-floating spacecraft–manipulator modeling and are adopted here in a control-oriented form [1], [2].

In this benchmark, this reduction is the natural consequence of the chosen operating regime: free-floating motion over finite horizons, negligible external wrenches, and zero initial momentum in all controller-validation scenarios.

Eliminating \dot{x}_0^I yields a reduced joint-space inertia

$$H^*(\mathbf{q}, \theta) \triangleq H_m(\mathbf{q}, \theta) - H_{0m}^\top(\mathbf{q}, \theta) H_0^{-1}(\mathbf{q}, \theta) H_{0m}(\mathbf{q}, \theta), \quad (3.9)$$

which is symmetric positive definite, as a Schur complement, and is the matrix used online to compute $\ddot{\theta}$.

The reduced joint dynamics are written in the classical second-order form

$$H^*(\mathbf{q}, \theta) \ddot{\theta} + C^*(\mathbf{q}, \theta, \dot{\theta}) \dot{\theta} = \tau + d_\theta(t), \quad (3.10)$$

where $d_\theta(t)$ collects unmodeled effects and numerical residuals.

Rather than differentiating (3.9) directly, the implementation computes $\partial H^*/\partial \theta_k$ from the derivatives of $\{H_0, H_{0m}, H_m\}$ generated offline. With $K \triangleq H_0^{-1} H_{0m}$, a convenient identity is

$$\frac{\partial H^*}{\partial \theta_k} = \frac{\partial H_m}{\partial \theta_k} - \left(\frac{\partial H_{0m}}{\partial \theta_k} \right)^\top K - H_{0m}^\top H_0^{-1} \left(\frac{\partial H_{0m}}{\partial \theta_k} \right) + H_{0m}^\top H_0^{-1} \left(\frac{\partial H_0}{\partial \theta_k} \right) K. \quad (3.11)$$

In the plant, (3.11) is evaluated through stable linear solves and explicit symmetrization to reduce sensitivity to floating-point asymmetries.

Given $\partial H^*/\partial \theta_k$, the Coriolis matrix is formed through Christoffel symbols:

$$C_{ij}^*(\mathbf{q}, \theta, \dot{\theta}) = \sum_{k=1}^n \frac{1}{2} \left(\frac{\partial H_{ij}^*}{\partial \theta_k} + \frac{\partial H_{ik}^*}{\partial \theta_j} - \frac{\partial H_{kj}^*}{\partial \theta_i} \right) \dot{\theta}_k, \quad (3.12)$$

and the velocity-dependent generalized force is computed as

$$C^* \dot{\theta} \triangleq C^*(\mathbf{q}, \theta, \dot{\theta}) \dot{\theta}. \quad (3.13)$$

This is the velocity-dependent term used by the reduced nonlinear plant.

The inertia blocks depend on the base attitude through \mathbf{q} because body-frame inertias are rotated into the inertial convention used by \dot{x}_0^I . However, the reduced Coriolis/centrifugal term is constructed using only partial derivatives with respect to joint angles, since the exported symbolic derivatives provide $\partial(\cdot)/\partial \theta$ only. As a consequence, contributions induced by the time variation of $H^*(\mathbf{q}, \theta)$ through $\dot{\mathbf{q}}$ are not represented explicitly in $C^*(\cdot)\dot{\theta}$. In this thesis,

these neglected terms are treated as part of a lumped mismatch $d_\theta(t)$ in (3.10). In the implemented plant, $d_\theta(t)$ is therefore not reconstructed as a separate signal. It represents the residual between the ideal reduced model and the adopted joint-space computation, and it is intentionally deferred to the mismatch description used later for prediction-error budgeting and robust MPC design. This choice keeps the online plant equations compact while making the neglected attitude-dependent contribution explicit at the modeling level.

3.3 Numerical Safeguards and Runtime Diagnostics

The plant is designed to remain numerically stable across wide operating envelopes and under aggressive inputs. Diagonal loading, explicit symmetrization, and quaternion fallback logic are numerical safeguards introduced to prevent ill-conditioning and NaNs; they do not represent additional physical effects.

Free-floating dynamics also provide conservation properties in ideal conditions and diagnostic quantities that are used as runtime consistency checks.

In this thesis, these quantities are not monitored only for post-processing purposes. They also act as acceptance indicators for the nonlinear plant, especially when the same model must be simulated repeatedly and then reused as the basis for local linearization and controller validation.

These diagnostics are not used as a substitute for formal proof, but as model-validation consistency checks. In a thesis centered on a reduced nonlinear plant and repeated controller calls, they provide an important engineering safeguard against hidden errors in frame conventions, inertia assembly, and quaternion propagation.

Since H_0 and H_m are theoretically symmetric, the plant enforces

$$H_0 \leftarrow \frac{1}{2}(H_0 + H_0^\top), \quad H_m \leftarrow \frac{1}{2}(H_m + H_m^\top), \quad H^* \leftarrow \frac{1}{2}(H^* + H^{*\top}), \quad (3.14)$$

to prevent drift due to floating-point noise, especially before matrix factorizations and linear solves.

All occurrences of H_0^{-1} are implemented through linear solves with diagonal loading:

$$K = (H_0 + \varepsilon_{H_0} I)^{-1} H_{0m}, \quad \varepsilon_{H_0} = 10^{-12} \max\left(1, \max|\text{diag}(H_0)|\right), \quad (3.15)$$

and similarly the reduced acceleration is obtained as

$$\ddot{\theta} = (H^* + \varepsilon_{H^*} I)^{-1} (\tau - C^* \dot{\theta}), \quad \varepsilon_{H^*} = 10^{-10}. \quad (3.16)$$

The diagonal-loading levels are selected empirically to prevent ill-conditioning and NaNs under aggressive inputs, while remaining negligible compared with typical diagonal magnitudes of H_0 and H^* over the operating envelope.

The quaternion is propagated using the inertial angular rate ${}^I\omega_0$ extracted from \dot{x}_0^I :

$$\dot{\mathbf{q}} = \frac{1}{2} \Omega_L({}^I\omega_0) \mathbf{q}, \quad \Omega_L(\omega) = \begin{bmatrix} 0 & -\omega^\top \\ \omega & -[\omega]_\times \end{bmatrix}, \quad (3.17)$$

Normalization is applied only when needed; if $\|\mathbf{q}\|$ becomes too small because of a numerical failure, the plant falls back to the identity quaternion $[1, 0, 0, 0]^\top$ in order to avoid NaNs.

Taken together, these safeguards are introduced to make the plant reliable under repeated closed-loop simulations, operating-point extraction, and subsequent linearization steps. Their role is numerical rather than physical: they reduce the probability of solver failure and non-finite signals without altering the intended free-floating structure of the model.

Using the inertia partition (3.3), the kinetic energy is evaluated as

$$E_K = \frac{1}{2} \dot{x}_0^{I\top} H_0 \dot{x}_0^I + \dot{x}_0^{I\top} H_{0m} \dot{\theta} + \frac{1}{2} \dot{\theta}^\top H_m \dot{\theta}. \quad (3.18)$$

Since joint torques can inject or remove mechanical power, E_K is not expected to remain constant in the actuated simulations; it is monitored to detect numerical instabilities and modeling inconsistencies, such as frame mismatches or integration artifacts.

The momentum vector (3.7) is computed as

$$\mathbf{h} = H_0 \dot{x}_0^I + H_{0m} \dot{\theta}, \quad P_I = h_{1:3}, \quad L_{CM0} = h_{4:6}. \quad (3.19)$$

In ideal free-floating operation (no external forces or torques), P_I and L_{CM0} remain constant. For the validation cases considered in this thesis, the zero-momentum initialization implies that $\|\mathbf{h}(t)\|$ is expected to remain close to zero, up to numerical tolerances.

Let the total mass be

$$m_{\text{tot}} = m_s + n m_l + \kappa m_p. \quad (3.20)$$

Under the adopted spatial inertia convention, the plant infers the skew matrix $[r_{0C}]_\times$ (the vector from CM0 to the system CoM) directly from the spatial inertia coupling:

$$[r_{0C}]_\times = -\frac{1}{m_{\text{tot}}} H_0(1:3, 4:6), \quad (3.21)$$

and then computes the angular momentum about the system CoM as

$$L_I = L_{CM0} - [r_{0C}]_\times P_I. \quad (3.22)$$

These signals are exported by the plant and are used as high-level consistency checks.

3.4 Attempted Full-Dynamics Prototype and Observed Limitations

This section documents a full-dynamics verification branch explored during the project and clarifies why it was not retained as the plant for the final comparative campaign. Its purpose is methodological: to record the attempted extension beyond the reduced free-floating formulation and to explain, based on invariant-based consistency checks, why the reduced model was selected as the reference plant for linearization, LPV predictor construction, and closed-loop MPC validation.

Alongside the reduced plant adopted in this thesis, a full-dynamics prototype was implemented as an internal verification environment. The goal was twofold: first, to simulate the coupled spacecraft–manipulator system without enforcing the zero-momentum velocity composition algebraically; second, to assess the implementation through invariant-based tests in unforced conditions. The section summarizes the adopted formulation, the main differences with the reduced plant, the observed invariant drift, and the resulting model-selection choice.

The full prototype uses generalized coordinates including spacecraft translation, attitude, and joint motion:

$$z \triangleq \begin{bmatrix} {}^I p_0 \\ \mathbf{q} \\ \theta \end{bmatrix}, \quad (3.23)$$

where ${}^I p_0 \in \mathbb{R}^3$ is the spacecraft reference-point position in the inertial frame, \mathbf{q} is a scalar-first quaternion, and $\theta \in \mathbb{R}^n$ are the joint angles. The corresponding generalized velocity is

$$\nu \triangleq \begin{bmatrix} {}^I v_0 \\ {}^I \omega_0 \\ \dot{\theta} \end{bmatrix} \in \mathbb{R}^{6+n}, \quad (3.24)$$

with ${}^I v_0$ and ${}^I \omega_0$ denoting the inertial-frame base linear and angular velocities at CM0.

The kinematics close the state evolution:

$${}^I \dot{p}_0 = {}^I v_0, \quad \dot{\mathbf{q}} = \frac{1}{2} \Omega_L({}^I \omega_0) \mathbf{q}, \quad (3.25)$$

where $\Omega_L(\cdot)$ is defined as in (3.17).

The full rigid-body dynamics are written in manipulator-like form:

$$M(z) \dot{\nu} + C(z, \nu) \nu = \tau_{\text{gen}}, \quad (3.26)$$

where $M(z) \in \mathbb{R}^{(6+n) \times (6+n)}$ is the generalized inertia matrix of the composite system, $C(z, \nu) \in \mathbb{R}^{(6+n) \times (6+n)}$ collects velocity-dependent terms, and τ_{gen} is the generalized input. In free-floating mode, the base is unactuated:

$$\tau_{\text{gen}} = \begin{bmatrix} 0_{6 \times 1} \\ \tau \end{bmatrix}. \quad (3.27)$$

In this section, $M(\cdot)$ denotes the full generalized inertia matrix associated with the coordinates $({}^L p_0, \mathbf{q}, \theta)$ and the velocity ν . This should be distinguished from the inertia blocks H_0 , H_{0m} , and H_m used in the reduced formulation to build the momentum map and the Schur complement inertia.

A central verification condition for an Euler–Lagrange rigid-body implementation is the power-balance property. In the absence of potential energy and external work, a consistent choice of $C(\cdot)$ satisfies

$$\dot{M}(z, \dot{z}) - 2C(z, \nu) \text{ is skew-symmetric,} \quad (3.28)$$

which implies

$$\frac{d}{dt} \left(\frac{1}{2} \nu^\top M(z) \nu \right) = \nu^\top \tau_{\text{gen}}. \quad (3.29)$$

Hence, for $\tau_{\text{gen}} = 0$, the kinetic energy should remain constant up to numerical integration errors. In this section, this property is used as a practical verification criterion for the internal full-model prototype.

By contrast, the reduced plant adopted throughout the thesis eliminates the base twist algebraically through momentum conservation [1], [2]. Using the partition introduced in Chapter 2 and in (3.3), the inertial momentum mapping is

$$h \triangleq H_0(\mathbf{q}, \theta) \dot{x}_0^I + H_{0m}(\mathbf{q}, \theta) \dot{\theta}. \quad (3.30)$$

All controller validation scenarios considered in this thesis start from $h(0) = 0$, so that the base twist is given by

$$\dot{x}_0^I = -H_0^{-1}(\mathbf{q}, \theta) H_{0m}(\mathbf{q}, \theta) \dot{\theta}. \quad (3.31)$$

Substituting (3.31) into the full dynamics yields the compact reduced model governed by the Schur complement inertia

$$H^*(\mathbf{q}, \theta) = H_m(\mathbf{q}, \theta) - H_{0m}^\top(\mathbf{q}, \theta) H_0^{-1}(\mathbf{q}, \theta) H_{0m}(\mathbf{q}, \theta), \quad (3.32)$$

with reduced second-order dynamics of the form

$$H^*(\mathbf{q}, \theta) \ddot{\theta} + C^*(\mathbf{q}, \theta, \dot{\theta}) \dot{\theta} = \tau + d_\theta(t). \quad (3.33)$$

In this reduced setting, momentum consistency is enforced by construction through (3.31), while quaternion propagation remains available for kinematic consistency and diagnostics.

The main differences between the two formulations are the following:

- **State dimension and integration.** The full prototype integrates spacecraft translation, attitude, and base twist through (3.26)–(3.25). The reduced plant integrates only $(\mathbf{q}, \theta, \dot{\theta})$ and reconstructs the base twist algebraically.
- **Momentum treatment.** In the reduced plant with $h(0) = 0$, momentum conservation is embedded in the velocity composition. In the full prototype, it must emerge from a fully consistent assembly of kinematics, inertia, and velocity-dependent terms.
- **Control-oriented suitability.** The reduced formulation directly matches the controller state and supports repeated linearization and LPV scheduling with a compact joint-space model. The full formulation requires a larger and more delicate multibody assembly.

The full-dynamics prototype was evaluated through standard unforced tests. With $\tau_{\text{gen}} = 0$ and no external wrenches, the following quantities should remain constant, up to numerical tolerance:

- **Total linear and angular momentum** of the composite system, expressed in $\{I\}$.
- **Kinetic energy** $E_K = \frac{1}{2} \nu^\top M(z) \nu$.
- **Power-balance residual** $\nu^\top (\dot{M} - 2C) \nu$, which should remain close to zero when (3.28) is satisfied.

These checks are more informative than a purely qualitative trajectory inspection, because hidden inconsistencies in frame conventions or velocity-dependent terms can remain visually unnoticed while still violating momentum or energy conservation.

The drift was assessed by monitoring $\|h(t) - h(0)\|$, the relative kinetic-energy error

$$\frac{|E_K(t) - E_K(0)|}{\max(1, E_K(0))},$$

and the power-balance residual over the simulation horizon. In the implemented prototype, these quantities did not remain close to their expected constant values in unforced simulations. Both momentum and kinetic energy exhibited a noticeable drift, and the power-balance residual was not consistently near zero. As a consequence, the trajectories obtained with the full prototype did not remain consistent with those produced by the reduced plant under the same initial conditions.

For the purposes of this thesis, this evidence is sufficient to conclude that the implemented full prototype did not yet provide the internal consistency required for a reliable simulation-to-linearization workflow. The issue is therefore methodological rather than conceptual: the full model is physically

meaningful in principle, but its adopted implementation state did not satisfy the invariant-based verification criteria used in this work.

The most likely source of the observed behavior is an incomplete consistency between attitude kinematics, frame-dependent inertia variations, and the construction of the velocity-dependent terms. In a quaternion-based multibody implementation, the dependence of M on orientation implies that \dot{M} contains contributions induced by $\dot{\mathbf{q}} = \frac{1}{2}\Omega_L(\omega)\mathbf{q}$. If these terms are not represented consistently in $C(\cdot)$, or in an equivalent correction preserving (3.28), spurious energy or momentum drift can appear even in unforced conditions. Additional numerical effects, such as quaternion renormalization and integration-step choices, can further perturb the invariant behavior, but they do not change the main conclusion drawn here.

Given the observed invariant drift and the resulting mismatch with respect to the reduced model, the full-dynamics prototype was not used in the final simulation campaign. The reduced free-floating formulation was retained because it is:

- **control-oriented**, since the controller state is naturally $(\theta, \dot{\theta})$ while base motion is represented through momentum composition and quaternion propagation;
- **numerically robust**, since momentum-based velocity composition and Schur complement reduction lead to well-structured computations that can be stabilized locally;
- **aligned with the adopted scenarios**, since all retained case studies start from $h(0) = 0$ and therefore match the intended operating envelope of the reduced formulation;
- **suited to the workflow of the thesis**, since it provides a reliable basis for repeated linearization, LPV scheduling, and comparative closed-loop MPC assessment.

The full-dynamics branch remains useful as a future extension and as an independent verification path. However, extending the benchmark in that direction requires a fully consistent multibody implementation preserving the power-balance condition,

$$\dot{M} - 2C \text{ skew-symmetric,}$$

or, equivalently, a formulation whose invariant properties are guaranteed by construction. Within the scope of this thesis, the reduced free-floating plant was therefore retained not only because it is lighter computationally, but because it is the formulation that remained sufficiently consistent for repeated linearization, controller integration, and closed-loop interpretation under the adopted validation workflow.

3.5 Plant Structure and Equation Traceability

The block-level organization of the nonlinear plant is shown in Figure 3.1. The figure is intended as a traceability map showing how the main mathematical steps are organized inside the plant used throughout the thesis.

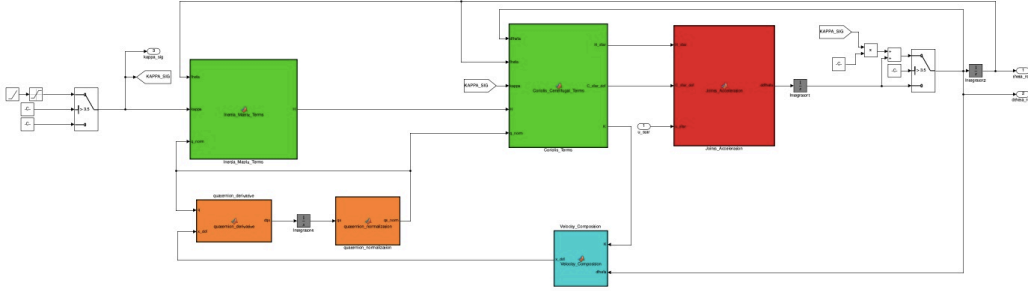


Figure 3.1: Structure of the reduced nonlinear plant used in the simulation campaign. The main blocks correspond to inertia construction, quaternion propagation and normalization, Coriolis/centrifugal terms, joint-acceleration computation, and momentum-based velocity composition. This organization provides a direct link between the nonlinear equations developed in this chapter and the plant used later for simulation, linearization, and controller validation.

For traceability and validation, each plant block corresponds to a well-defined mathematical step:

- the inertia-construction block assembles H in (3.3) from the parameter vector π in (3.5), then enforces symmetry as in (3.14);
- the reduced Coriolis/centrifugal block extracts (H_0, H_{0m}, H_m) , computes K and H^* using (3.8)–(3.9) with loading (3.15), builds $\partial H^*/\partial \theta_k$ using (3.11), and evaluates $C^*\dot{\theta}$ using (3.12)–(3.13);
- the velocity-composition block computes the base twist from the zero-momentum relation (3.8);
- the joint-acceleration block solves the reduced dynamics (3.10) for $\ddot{\theta}$ using (3.16);
- the quaternion-propagation and normalization blocks implement (3.17) together with the associated safeguards;
- the diagnostics block computes the invariants (3.18)–(3.22) used for consistency checks.

The parameter vector π follows the same fixed ordering used by the exported symbolic routines and by the plant implementation.

This explicit equation-to-plant traceability makes it easier to validate the model numerically, isolate modeling errors, and keep the linearization and LPV pipeline developed in the next chapter consistent with the nonlinear simulation model. It also ensures that the quantities linearized later are generated by a plant whose numerical structure remains directly connected to the governing equations stated here.

Chapter 4

Linearization and LPV Prediction Model Construction

This chapter presents the linearization and discretization pipeline used to build the prediction models for MPC. The starting point is the reduced free-floating nonlinear model derived in Chapter 3, where the spacecraft base is eliminated through momentum-based reduction and the controller state is

$$x = [\theta^\top \ \dot{\theta}^\top]^\top.$$

The prediction model is built under the same assumptions adopted for the reduced plant, namely free-floating operation and negligible external wrenches over the maneuver window. The quaternion is propagated only in the nonlinear simulation model and is not part of the MPC state. During predictor construction, the quaternion entering the inertia packing is frozen to the normalized initial attitude q_0 ; the resulting residual attitude-dependent error is later treated as part of the bounded modeling mismatch in the robust design (Chapter 5).

In the adopted numerical workflow, the continuous-time linearization is generated by a dedicated routine, the discrete-time pages are assembled offline, and the scheduling simplices are generated through an offline region-construction step. Online barycentric scheduling is then handled within the LPV controller structure.

From a control-design viewpoint, this is a standard LPV compromise: the nonlinear dynamics are embedded into a measurable scheduled family of linear models, so that constraint handling remains compatible with QP-based MPC while the dominant operating-point dependence is retained [13].

The LPV model developed in this chapter should be interpreted as a local control-oriented surrogate of the reduced nonlinear plant rather than as an exact global embedding. Its purpose is to retain the dominant dependence on

configuration and joint rates while preserving a predictor structure compatible with repeated QP-based MPC. The resulting approximation is therefore meaningful only inside the selected operating region and under monitored interpolation activity.

4.1 Why an LPV Surrogate: LTI vs LTV vs LPV

A single LTI model obtained at one operating point is accurate only in a local neighborhood and is therefore mainly suited to regulation around that point. An LTV description is more appropriate when the relevant variations follow a known time profile. In the present problem, however, the dominant variations are driven by the manipulator configuration and rates, which are directly measurable online. For this reason, an LPV predictor is adopted.

The scheduling vector is chosen as

$$s := \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \in \mathbb{R}^{n_s}, \quad n_s = 2n,$$

which is directly available online from the controller state.

To keep the online complexity low, the global operating set is described through a corner-page polytope defined by the box corners in s , while online interpolation is performed over simplices so that at most $n_s + 1$ pages are mixed at any sampling instant.

In this thesis, the LPV predictor is used as a structured compromise between model fidelity and online tractability. The operating region is described by a bounded scheduling box, this box is triangulated offline, and only the pages associated with the active simplex are mixed online. As a result, the online predictor assembly remains lightweight and compatible with the repeated QP solution required by the considered MPC setup.

In this benchmark, LPV is preferred over a purely single-page LTI approximation because the operating region includes non-negligible configuration dependence, while it is preferred over a generic online LTV relinearization strategy because the predictor must remain compatible with repeated QP solution inside a lightweight MATLAB/Simulink workflow. The adopted choice should therefore be interpreted as a control-oriented compromise between predictor fidelity and online tractability [13], [14].

4.2 Continuous-Time Linearization and Discretization

Let $x = [\theta^\top \ \dot{\theta}^\top]^\top$ and $u \equiv \tau$. Using the reduced model of Chapter 3, the joint acceleration is written as

$$\ddot{\theta} = H^*(q_0, \theta)^{-1} \left(\tau - C^*(q_0, \theta, \dot{\theta}) \dot{\theta} \right), \quad (4.1)$$

where H^* is the Schur-complement inertia and $C^*\dot{\theta}$ is the reduced Coriolis/centrifugal generalized force. The corresponding first-order state model is

$$\dot{x} = f(x, u) = \begin{bmatrix} \dot{\theta} \\ H^*(q_0, \theta)^{-1} \left(\tau - C^*(q_0, \theta, \dot{\theta}) \dot{\theta} \right) \end{bmatrix}. \quad (4.2)$$

Linearization of (4.2) about a point (x_j, u_j) yields

$$\Delta \dot{x} = A_c(x_j) \Delta x + B_c \Delta u, \quad A_c = \begin{bmatrix} 0 & I \\ A_{21} & A_{22} \end{bmatrix}. \quad (4.3)$$

In the adopted linearization routine, the input Jacobian is kept constant across all pages and is computed at the controller operating point:

$$B_c \triangleq \begin{bmatrix} 0_{n \times n} \\ H^*(q_0, \theta_{\text{op}})^{-1} \end{bmatrix}. \quad (4.4)$$

In practice, B_c is computed once at the operating point $(\theta_{\text{op}}, \dot{\theta}_{\text{op}}, \tau_{\text{op}})$, with τ_{op} typically set to zero, and at the current payload estimate used to initialize the model parameters. The resulting $B_d = T_s B_c$ is then replicated across all pages.

This is the LPV-in- A approximation adopted to reduce online assembly cost. Any residual dependence of B on configuration and payload parameters is later treated as bounded mismatch in the robust formulation (Chapter 5).

This choice should be interpreted as an engineering approximation around the controller operating point, not as a structural property of the reduced dynamics. It is adopted to keep the online predictor compact and to avoid page-dependent input matrices in the repeated MPC assembly.

The implementation exploits the structure induced by the Christoffel-based construction of C^* . In the reduced model, $C^*(\theta, \dot{\theta})$ is linear in $\dot{\theta}$, hence $C^*(\theta, \dot{\theta})\dot{\theta}$ is quadratic in $\dot{\theta}$. The adopted routine assembles a matrix $C_{\text{mat}}(\theta_j, \dot{\theta}_j)$ such that

$$C^*(\theta_j, \dot{\theta}_j)\dot{\theta}_j = C_{\text{mat}}(\theta_j, \dot{\theta}_j)\dot{\theta}_j. \quad (4.5)$$

With Christoffel symbols satisfying $\mathcal{C}_{ijk} = \mathcal{C}_{ikj}$, the derivative of the quadratic term with respect to $\dot{\theta}$ yields a factor 2, and the implementation uses the closed-form block

$$A_{22}(x_j) = -\frac{\partial}{\partial \dot{\theta}} \left(H^*(\theta)^{-1} C^*(\theta, \dot{\theta}) \dot{\theta} \right) \Big|_{x_j} = -2 H^*(\theta_j)^{-1} C_{\text{mat}}(\theta_j, \dot{\theta}_j). \quad (4.6)$$

The factor 2 follows from the quadratic dependence on $\dot{\theta}$. With the definition $C^*(\theta, \dot{\theta}) \dot{\theta} = C_{\text{mat}}(\theta, \dot{\theta}) \dot{\theta}$ and C_{mat} linear in $\dot{\theta}$, the derivative satisfies

$$\frac{\partial}{\partial \dot{\theta}} (C_{\text{mat}}(\theta, \dot{\theta}) \dot{\theta}) = 2 C_{\text{mat}}(\theta, \dot{\theta}).$$

The block $A_{21}(x_j) = \partial \ddot{\theta} / \partial \theta$ would require higher-order derivatives if computed in closed form. To avoid this cost and to improve numerical robustness, the implementation evaluates A_{21} using a complex-step perturbation on θ applied directly to the acceleration map (4.1):

$$A_{21}(:, k) \approx \frac{\Im \left\{ \ddot{\theta}(\theta_j + i h e_k, \dot{\theta}_j, \tau_j) \right\}}{h}, \quad h \ll 1. \quad (4.7)$$

This avoids the subtractive cancellation typical of finite differences and, in practice, provides very accurate directional derivatives when the acceleration map is evaluated in a complex-safe form.

From this point onward, the dependence on the frozen attitude q_0 is kept implicit in $H^*(\cdot)$ and in the reduced terms to simplify notation; all pages are generated with the same fixed quaternion q_0 as stated in the chapter assumptions.

The approximation in (4.7) requires the implemented acceleration map to be complex-safe with respect to the perturbed joint coordinates. In the present implementation, this condition is enforced by evaluating the reduced acceleration through algebraic operations and linear solves, while avoiding branch-sensitive operations on the complex-perturbed states.

For each linearization point, the continuous-time linearization routine also returns the drift $f(x_j, u_j)$, namely the vector field evaluated at the anchor point:

$$f(x_j, u_j) = \begin{bmatrix} \dot{\theta}_j \\ H^*(\theta_j)^{-1} \left(\tau_j - C^*(\theta_j, \dot{\theta}_j) \dot{\theta}_j \right) \end{bmatrix}, \quad (4.8)$$

which is later used to construct a discrete-time affine correction that removes one-step offsets at the linearization point.

With sampling time T_s , each continuous-time Jacobian page is discretized using Forward Euler:

$$A_{d,j} = I_{2n} + T_s A_c(x_j), \quad B_d = T_s B_c. \quad (4.9)$$

Forward Euler preserves the piecewise-affine dependence of A inside each simplex and keeps the discrete-time assembly cost low. This choice is consistent with the small sampling times used in the simulation campaign and with the goal of fast online QP solution. Higher-order discretization schemes are possible, but Forward Euler was selected here to preserve the intended piecewise-affine structure of the scheduled predictor while keeping both page construction and online assembly simple.

Among the available discretization options for nonlinear-model-based MPC, this choice is justified here by the small sampling time, the need to preserve a cheap affine page construction, and the priority given to online efficiency over high-order per-step integration accuracy [14].

The resulting discrete-time pages are therefore chosen for structural simplicity rather than for maximal per-step accuracy. The discretization residual is not ignored; it is one of the components later absorbed into the bounded prediction-mismatch description used by the robust controller.

A practical issue in linearization-based prediction is the appearance of offsets when the linear model is used in absolute coordinates. To enforce local consistency at each linearization point, the discrete-time page-construction routine builds a discrete affine term $f_{d,j}$ so that the discrete predictor matches the Forward Euler propagation of the nonlinear reduced dynamics at the anchor point.

Let $x_j = [\theta_j^\top \ \dot{\theta}_j^\top]^\top$ and u_j be the linearization point. Define the Euler-propagated next state using the nonlinear drift:

$$x_j^+ \triangleq x_j + T_s f(x_j, u_j). \quad (4.10)$$

Then the affine term is computed as

$$f_{d,j} \triangleq x_j^+ - A_{d,j} x_j - B_d u_j, \quad (4.11)$$

and the discrete-time page predictor is

$$x_{k+1} = A_{d,j} x_k + B_d u_k + f_{d,j}. \quad (4.12)$$

With (4.11), the one-step prediction is exact at (x_j, u_j) , which helps reduce systematic offsets in tracking tasks.

4.3 Prediction Mismatch and Corner-Page Structure

The LPV predictor is affected by several structured approximation errors: (i) Forward Euler truncation, (ii) freezing the quaternion to q_0 , (iii) the LPV-in- A approximation with constant B_d , (iv) loss of affinity when barycentric weights are clipped and renormalized outside the active simplex, and (v) gridded payload interpolation effects when $(\hat{m}_p, \hat{\kappa})$ are bracketed between offline grid points, including implicit clamping at the grid boundaries.

These effects are treated as bounded prediction mismatch in the robust MPC design and are monitored in the simulation campaign through one-step residual checks, simplex-coverage statistics, and closed-loop indicators such as tightening activity and slack usage.

The predictor is not intended to reproduce the nonlinear plant exactly over the full operating region. It is a structured approximation designed for constrained MPC under monitored and bounded mismatch. In the adopted workflow, what matters is that the approximation error remains structured, bounded in the intended operating region, and sufficiently visible through the logged scheduling and residual indicators.

For clarity, the predictor used in the MPC chapters combines four deliberate approximations: local linearization of the reduced dynamics, Forward Euler discretization, freezing of the quaternion to q_0 during page generation, and replication of a constant B_d across pages.

The LPV operating region is defined as a hyper-rectangle in $s = [\theta; \dot{\theta}]$. A set of corner pages is built at the 2^{n_s} box corners:

$$\mathcal{V}_{\text{corn}} = \{s^{(1)}, \dots, s^{(2^{n_s})}\}, \quad s^{(j)} \in \mathbb{R}^{n_s}. \quad (4.13)$$

Each corner $s^{(j)}$ is mapped to $(\theta_j, \dot{\theta}_j)$ and linearized through the continuous-time routine, producing $(A_{d,j}, f_{d,j})$. The matrices C and D are kept vertex-invariant, consistent with the controller formulation, and B_d is shared by all pages as in (4.4)–(4.9).

Keeping B_d constant across pages reduces memory usage and avoids recomputing input-dependent condensed matrices online. This is an approximation because $H^*(\theta)^{-1}$ varies with configuration and, in post-capture scenarios, with payload mass estimates. In nominal controllers, this mismatch is accepted as a modeling error. In the robust tube-based controller, the induced term $\Delta B u$ is conservatively bounded offline around θ_{op} and absorbed into the additive disturbance set (Chapter 5).

In practice, this bound is obtained offline by evaluating the variation of $H^*(\theta)^{-1}$ over the scheduling box, combining the induced input-channel

deviation with the known input constraint set $u \in U$, and storing the resulting contribution as part of the additive disturbance budget used for tube tightening. This preserves the fixed- B_d predictor structure adopted online while accounting conservatively for the neglected page dependence of the input channel.

This yields a conservative contribution to the disturbance budget used for tube tightening while preserving the fixed- B_d predictor structure adopted for online MPC.

When payload effects are enabled, the page-construction routine builds multiple blocks of corner pages corresponding to different payload masses m_p and coupling factors κ . The total number of discrete-time pages is therefore

$$N_v = n_{m_p} n_\kappa 2^{n_s},$$

where n_{m_p} and n_κ are the grid sizes used for payload-mass and coupling-factor blocks. In the adopted ordering, the corner index runs fastest, followed by the coupling-factor block and then by the payload-mass block.

To keep page 1 coherent with the current parameter estimates, the payload grids are reordered so that the first (m_p, κ) block matches the current \hat{m}_p and $\hat{\kappa}$ used to initialize the controller. This layout is exploited online to map local simplex weights into the correct page subset and to perform inexpensive interpolation across payload-mass blocks when enabled.

4.4 Simplex-Based Scheduling and Online Weight Computation

The scheduling box is partitioned into simplices using a Kuhn triangulation. The offline region-generation routine returns:

- the corner set $V_{\text{corn}} \in \mathbb{R}^{n_s \times 2^{n_s}}$, stored column-wise;
- a list of simplices $\{V_r\}_{r=1}^R$, each $V_r \in \mathbb{R}^{n_s \times (n_s+1)}$ with vertices chosen among the box corners;
- a simplex-to-corner index map that links each simplex vertex to the corresponding column of V_{corn} .

By default, the number of regions is

$$R = \text{factorial}(n_s),$$

corresponding to the full Kuhn triangulation. For $n = 2$ and therefore $n_s = 4$, this yields $R = 24$ simplices, each mixing exactly $n_s + 1 = 5$ corner pages.

The full Kuhn triangulation scales poorly with n_s . In this thesis, however, the planar benchmark keeps $n_s = 4$, so that the number of simplices remains small and scheduling remains inexpensive.

Given the current scheduling point $z = s(x_k) \in \mathbb{R}^{n_s}$, the LPV scheduler selects the simplex that best contains z by maximizing the minimum raw barycentric weight. For each simplex $V_r = [v_1, \dots, v_{n_s+1}]$, the raw weights are computed by solving the affine system

$$\begin{bmatrix} 1 & \cdots & 1 \\ v_1 & \cdots & v_{n_s+1} \end{bmatrix} w_{\text{raw}} = \begin{bmatrix} 1 \\ z \end{bmatrix}. \quad (4.14)$$

The selected region is

$$r^* = \arg \max_r \min_i (w_{\text{raw},i}^{(r)}). \quad (4.15)$$

Before simplex selection, the scheduling point is projected inside the predefined hyper-rectangle to avoid unstable region switching. In addition, a simple hysteresis/dwell logic is used: the current simplex is retained if its minimum raw weight remains within a small tolerance, and switching is allowed only if an alternative simplex provides a clearly better score and a minimum dwell time has elapsed.

These protection mechanisms are implementation-level safeguards rather than part of an ideal LPV interpolation law. Their role is to avoid numerically fragile simplex switching and harmful extrapolation near the scheduling-box boundaries, at the price of a predictor that is only piecewise affine under nominal coverage and no longer globally affine once clipping becomes active.

To avoid extrapolation outside the bounded convex hull used for MPC, the selected weights are then projected by non-negativity clipping and renormalization:

$$w \leftarrow \max(w_{\text{raw}}, 0), \quad w \leftarrow \frac{w}{\mathbf{1}^\top w}. \quad (4.16)$$

Inside a simplex, that is, when no clipping is required, $w(z)$ is affine in z ; with clipping, the map remains bounded but is no longer globally affine. The frequency of clipping is therefore used as a practical indicator of model-set coverage in the validation tests.

When multiple payload blocks are available, the online scheduler interpolates across payload mass m_p and coupling factor κ using a bracketing strategy on each axis. Let $(m_{p,L}, m_{p,U})$ be the mass bracket and define $\beta \in [0, 1]$ as the normalized position of \hat{m}_p inside that bracket. Similarly, let (κ_L, κ_U) be the coupling bracket and define $\eta \in [0, 1]$ as the normalized position of $\hat{\kappa}$.

The final page-weight vector is assembled by distributing the corner-weight vector to at most four neighboring blocks through bilinear interpolation:

$$w^{LL} = (1 - \beta)(1 - \eta)w_{\text{corn}}, \quad w^{UL} = \beta(1 - \eta)w_{\text{corn}},$$

$$w^{LU} = (1 - \beta)\eta w_{\text{corn}}, \quad w^{UU} = \beta\eta w_{\text{corn}}.$$

If only one grid point is used along one axis, the interpolation reduces to the corresponding one-dimensional bracketing case without increasing the number of mixed pages beyond the simplex mixture.

4.5 Output Model, Bias Handling, and Consistency Checks

Two output definitions are supported:

- *state-output mode*: $y = x = [\theta^\top \dot{\theta}^\top]^\top$, so $C = I$ and the output is directly constrained by the state bounds;
- *end-effector-output mode*: $y_{\text{phys}} = [x_{ee}, y_{ee}, v_x, v_y]^\top$, approximated locally around the operating point by

$$y_{\text{phys}} \approx C_0 x + \begin{bmatrix} b \\ 0 \end{bmatrix}, \quad (4.17)$$

where $b \in \mathbb{R}^2$ is the position bias.

Accordingly, the end-effector-output mode used inside the controller is a local affine surrogate of the physical end-effector map. It is suitable for constraint handling and tracking around the selected operating point, but it should not be interpreted as a globally exact output description over the full workspace.

The adopted workflow includes sanity checks directly tied to the predictor construction:

- **One-step residual check**: verify that (4.12) reproduces the Euler step at linearization points due to the definition (4.11);
- **Simplex coverage statistics**: monitor how often the raw weights in (4.14) fall outside the simplex and how frequently the projection (4.16) is used;
- **Conditioning**: track the conditioning of the barycentric system in (4.14) and the numerical robustness of the Schur-based computations in the linearization routine.

If projection and clipping are frequent, the scheduling box should be re-centered and/or resized so that typical closed-loop trajectories remain inside the triangulated region. This reduces loss of affinity, improves interpolation fidelity, and generally decreases bias and offset in tracking scenarios.

If these checks indicate frequent projection, poor simplex conditioning, or persistent one-step residuals outside the expected range, the predictor should

be regarded as operating close to the edge of its intended validity region. In that case, the scheduling box, the operating point, or the payload grid should be revised before drawing strong conclusions from the closed-loop comparison.

The final discrete-time LPV predictor used by the controller has the form

$$x_{k+1} = A_d(\theta_k, \dot{\theta}_k) x_k + B_d u_k + f_d(\theta_k, \dot{\theta}_k), \quad (4.18)$$

where A_d and f_d are obtained by convex mixing of precomputed corner pages using simplex-based barycentric weights, and B_d is kept constant to preserve online efficiency. This model is the basis for the nominal LPV-MPC and for the robust tube-based MPC developed in the subsequent chapter.

Chapter 5

MPC Design

This chapter presents the three model predictive control formulations adopted in the thesis for the reduced free-floating spacecraft–manipulator benchmark. The three controllers share the same basic predictive structure, state and input constraints, and cost-design rationale, while differing in predictor construction, robustness logic, and, in the robust case, target-anchoring semantics.

The first formulation is a nominal MPC based on a single affine linear predictor. The second replaces the single predictor with an LPV scheduled model built from local linearization pages. The third adds a tube-based robust layer around the scheduled center model in order to handle bounded prediction mismatch and payload-related uncertainty. The presentation below focuses on the control problems themselves rather than on software structure.

5.1 Common Formulation

The prediction state is the reduced joint-space vector

$$x \triangleq \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \in \mathbb{R}^{2n}, \quad u \equiv \tau \in \mathbb{R}^n. \quad (5.1)$$

For all controllers, the online predictor has the affine discrete-time form

$$x_{k+1} = A_k x_k + B_k u_k + f_k, \quad (5.2)$$

where the matrices depend on the selected formulation. In the single-model controller they are constant, in the LPV controller they are scheduled online, and in the robust controller they define the tube-center dynamics.

State and input constraints are enforced through the polyhedral set

$$\mathcal{Z} \triangleq \{(x, u) \mid Fx + Gu \leq \mathbf{1}, F_0x + G_0u \leq 0\}, \quad (5.3)$$

where the first block collects normalized inequalities and the second block collects zero-right-hand-side faces, which remain hard. This representation is used consistently across the three controllers.

When end-effector constraints are enabled, an additional output map is considered,

$$y_k = C_k x_k, \quad A_y y_k \leq b_y. \quad (5.4)$$

In the nominal formulations, these output inequalities are enforced directly. In the robust formulation, whenever a normalized output representation is available, the corresponding faces are included in the tightened inequality stack.

To unify regulation and tracking under constraints, the controllers optimize *artificial references* [17], [18]. In setpoint mode, the optimizer searches for an admissible artificial steady pair (x_a, u_a) satisfying

$$x_a = A_k x_a + B_k u_a + f_k. \quad (5.5)$$

In trajectory mode, the same logic is applied to an admissible artificial trajectory $\{x_{a,k}, u_{a,k}\}$,

$$x_{a,k+1} = A_k x_{a,k} + B_k u_{a,k} + f_k. \quad (5.6)$$

The role of the artificial target is to preserve feasibility when the requested reference cannot be matched exactly under the imposed constraints.

The artificial target is restricted to a target-admissible set

$$\mathcal{Z}_{\text{tar}} \triangleq \{(x, u) \mid Fx + Gu \leq \sigma \mathbf{1}, F_0 x + G_0 u \leq 0\}, \quad \sigma \in (0, 1). \quad (5.7)$$

This set is used only for artificial targets and terminal ingredients. It should not be confused with the tightened constraints introduced later for the robust tube-based formulation.

For the terminal ingredients it is convenient to introduce error coordinates,

$$\tilde{x}_k \triangleq x_k - x_{a,k}, \quad \tilde{u}_k \triangleq u_k - u_{a,k}. \quad (5.8)$$

The terminal condition is imposed in these coordinates through a terminal set

$$X_f \subseteq \{\tilde{x} \mid \tilde{x}^\top P \tilde{x} \leq \alpha_f\}, \quad (5.9)$$

and appears in the online problem as

$$x_N - x_{a,N} \in X_f. \quad (5.10)$$

A compact common form of the tracking problem is therefore

$$\begin{aligned}
\min \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k, x_{a,k}, u_{a,k}) + V_f(x_N, x_{a,N}) + o(r, \{x_{a,k}\}) \\
\text{s.t.} \quad & x_{k+1} = A_k x_k + B_k u_k + f_k, \\
& (x_k, u_k) \in \mathcal{Z}, \\
& (x_{a,k}, u_{a,k}) \in \mathcal{Z}_{\text{tar}}, \\
& x_{a,k+1} = A_k x_{a,k} + B_k u_{a,k} + f_k, \\
& x_N - x_{a,N} \in X_f, \\
& x_0 \text{ given.}
\end{aligned} \tag{5.11}$$

The three formulations presented below can all be interpreted as particular realizations of (5.11).

Remark on slack variables. The framework also supports slack-based relaxation of selected normalized inequalities in specific simulation settings. Since this mechanism is auxiliary and case-dependent, it is not treated here as a defining design ingredient of the controllers.

5.2 Nominal MPC with a Single Predictor

The first controller uses one affine linear predictor only. This corresponds to a fixed local model, discretized once and then reused over the full horizon. The center dynamics are

$$x_{k+1} = A_1 x_k + B_1 u_k + f_{d,1}, \tag{5.12}$$

where $f_{d,1}$ is the affine correction associated with the chosen operating point. The affine term is kept explicitly because it removes the one-step offset that would otherwise appear if the predictor were written as a purely homogeneous model around a nonzero anchor.

In setpoint mode, the artificial target is a steady pair (x_a, u_a) satisfying

$$x_a = A_1 x_a + B_1 u_a + f_{d,1}, \quad (x_a, u_a) \in \mathcal{Z}_{\text{tar}}. \tag{5.13}$$

In trajectory mode, the same structure is extended to an admissible artificial trajectory generated by the same predictor. In both cases, the requested reference is handled through an offset penalty on the artificial output rather than through hard equalities, so that feasibility is preserved even when the requested target is not exactly reachable.

The resulting finite-horizon problem is

$$\begin{aligned}
\min_{\{x_k, u_k\}, \{x_{a,k}, u_{a,k}\}} & \sum_{k=0}^{N-1} (\|x_k - x_{a,k}\|_Q^2 + \|u_k - u_{a,k}\|_R^2) + \|x_N - x_{a,N}\|_P^2 + o_{\text{LTI}}(r, \{x_{a,k}\}) \\
\text{s.t.} & x_{k+1} = A_1 x_k + B_1 u_k + f_{d,1}, \\
& (x_k, u_k) \in \mathcal{Z}, \\
& (x_{a,k}, u_{a,k}) \in \mathcal{Z}_{\text{tar}}, \\
& x_{a,k+1} = A_1 x_{a,k} + B_1 u_{a,k} + f_{d,1}, \\
& x_N - x_{a,N} \in X_f, \\
& x_0 \text{ given.}
\end{aligned} \tag{5.14}$$

In setpoint mode, the artificial trajectory reduces to a constant pair and the offset term penalizes the mismatch between the artificial output and the requested constant reference. In trajectory mode, the same term penalizes the stage-wise mismatch with the preview sequence.

This controller acts as the nominal baseline of the thesis. Its main advantage is simplicity: the online problem remains fully quadratic, numerically regular, and easy to interpret. Its main limitation is that prediction accuracy can degrade when the closed-loop motion moves away from the operating region in which the single page was generated.

5.3 Nominal MPC with an LPV Predictor

The second controller keeps the same optimization structure, but updates the predictor online through an LPV interpolation over precomputed local pages. The goal is to retain the dominant configuration dependence of the reduced nonlinear model while preserving a QP-based online problem.

The scheduled predictor is written as

$$A_k = \sum_{i=1}^{N_v} \alpha_i(z_k) A_i, \quad B_k = \bar{B}, \quad f_k = \sum_{i=1}^{N_v} \alpha_i(z_k) f_i, \tag{5.15}$$

where $z_k = [\theta_k^\top \dot{\theta}_k^\top]^\top$ is the scheduling point and $\alpha_i(\cdot)$ are barycentric weights associated with the active simplex. The input matrix is kept constant across pages in the current implementation in order to keep the online structure simpler and more regular. The mismatch induced by this approximation is later absorbed into the disturbance description used by the robust formulation.

When an LPV end-effector output model is enabled, the output map is updated consistently as

$$C_k = \sum_{j=1}^{N_c} \alpha_j^C(z_k) C_j. \quad (5.16)$$

The artificial reference is defined with respect to the scheduled center model,

$$x_{a,k+1} = A_k x_{a,k} + B_k u_{a,k} + f_k, \quad (x_{a,k}, u_{a,k}) \in \mathcal{Z}_{\text{tar}}. \quad (5.17)$$

This keeps the optimization problem quadratic and avoids introducing additional decision variables for the scheduling law itself.

The corresponding optimization problem is

$$\begin{aligned} \min_{\{x_k, u_k\}, \{x_{a,k}, u_{a,k}\}} \quad & \sum_{k=0}^{N-1} (\|x_k - x_{a,k}\|_Q^2 + \|u_k - u_{a,k}\|_R^2) + \|x_N - x_{a,N}\|_P^2 + o_{\text{LPV}}(r, \{x_{a,k}\}) \\ \text{s.t.} \quad & x_{k+1} = A_k x_k + B_k u_k + f_k, \\ & (x_k, u_k) \in \mathcal{Z}, \\ & (x_{a,k}, u_{a,k}) \in \mathcal{Z}_{\text{tar}}, \\ & x_{a,k+1} = A_k x_{a,k} + B_k u_{a,k} + f_k, \\ & x_N - x_{a,N} \in X_f^{\text{LPV}}, \\ & x_0 \text{ given.} \end{aligned} \quad (5.18)$$

The LPV controller should therefore be interpreted as a receding-horizon tracker of a locally scheduled center model. It improves the local representativeness of the predictor with respect to the single-model case, but its quality still depends on the operating-region coverage achieved by the offline page set.

A practical point deserves explicit mention because it is important for the later interpretation of the results. The scheduling weights are computed through simplex-based barycentric coordinates. If the scheduling point leaves the selected simplex, the raw weights are clipped and renormalized before entering the predictor. This avoids harmful extrapolation outside the convex hull used for control design. In the simulation campaign, the frequency of clipping is therefore monitored as a diagnostic of local-model coverage rather than being treated as a negligible numerical detail.

5.4 Tube-Based Robust MPC

The third controller augments the scheduled center model with an explicit robust tube. Conceptually, the optimizer predicts a nominal center trajectory

and simultaneously enforces a bound on the deviation between that center trajectory and the true plant state under bounded mismatch. Robustness is therefore achieved by combining an ancillary stabilizing feedback, a tube-size recursion, and tightened constraints.

The center dynamics are the same scheduled affine predictor used by the LPV controller,

$$x_{k+1} = A_k x_k + B_k u_k + f_k, \quad (5.19)$$

but the applied input is parameterized as

$$u_k = u_{a,k} + K(x_k - x_{a,k}) + c_k, \quad (5.20)$$

where K is an ancillary feedback gain and c_k is an optimized correction term. The role of K is to stabilize the error dynamics inside the tube, while c_k preserves enough freedom for the optimizer to move the tube center without relying only on the ancillary action.

The disturbance set \mathcal{W} is introduced to cover the dominant prediction mismatches of the benchmark. In this thesis it collects, in a control-oriented way, the residual effects of model reduction, local linearization, discretization, scheduled predictor approximation, and, when relevant, payload-related uncertainty. The disturbance budget is therefore not an abstract robustness placeholder, but an engineering description of the mismatch that the controller is designed to absorb.

If e_k denotes the deviation between the true plant state and the predicted center trajectory, the tube design is based on the conservative error model

$$e_{k+1} = (A_i + B_i K)e_k + w_k, \quad w_k \in \mathcal{W}, \quad (5.21)$$

evaluated on the vertices used in the robust design. The tube is represented through a scalable polyhedral family

$$\mathcal{E}(\alpha_k) \triangleq \{e \mid V_x e \leq \alpha_k\}, \quad \alpha_k \geq 0, \quad (5.22)$$

and its size is propagated through the recursion

$$\alpha_{k+1} \geq H_{\max} \alpha_k + \bar{w}, \quad \alpha_0 = 0. \quad (5.23)$$

The quantities entering this recursion are precomputed consistently with the chosen tube shape, ancillary gain, and disturbance budget.

The nominal inequalities are then tightened so that all trajectories inside the tube satisfy the original constraints. In compact form, the tightened condition reads

$$H_k \begin{bmatrix} x_k \\ u_k \end{bmatrix} \leq \mathbf{1} - T_{xu} \alpha_k, \quad (5.24)$$

where H_k collects the normalized inequalities active at the current stage and T_{xu} maps the tube size into a worst-case tightening margin. Zero-right-hand-side faces remain hard.

The robust online problem is therefore

$$\begin{aligned}
& \min_{\{x_k, u_k, c_k, \alpha_k\}, \{x_{a,k}, u_{a,k}\}} && \sum_{k=0}^{N-1} (\|x_k - x_{a,k}\|_Q^2 + \|u_k - u_{a,k}\|_R^2 + \|c_k\|_{R_c}^2) + \|x_N - x_{a,N}\|_P^2 \\
& \text{s.t.} && x_{k+1} = A_k x_k + B_k u_k + f_k, \\
& && u_k = u_{a,k} + K(x_k - x_{a,k}) + c_k, \\
& && H_k \begin{bmatrix} x_k \\ u_k \end{bmatrix} \leq \mathbf{1} - T_{xu} \alpha_k, \\
& && \alpha_{k+1} \geq H_{\max} \alpha_k + \bar{w}, \\
& && (x_{a,k}, u_{a,k}) \in \mathcal{Z}_{\text{tar}}, \\
& && x_{a,k+1} = A_k x_{a,k} + B_k u_{a,k} + f_k, \\
& && x_N - x_{a,N} \in X_f, \\
& && x_0 \text{ given.}
\end{aligned} \tag{5.25}$$

A point that is important for the interpretation of the comparative results must be stated explicitly. In this benchmark, the robust controller uses stricter target anchoring than the nominal formulations. In particular, when tracking is enabled, the artificial target is constrained more directly to the requested output, whereas the nominal controllers allow a deviation from the requested reference through an offset penalty. As a consequence, any observed reduction in feasibility margin cannot be attributed to tube tightening alone; it must be read as the combined effect of robustness and stricter target semantics.

Within the scope of this thesis, the robust controller is therefore interpreted as a practical tube-based MPC formulation tailored to the reduced free-floating benchmark. Its purpose is to make the robust ingredients explicit and operational in the simulation campaign: the mismatch budget is stated, the tube-size propagation is part of the online problem, and the effect of tightening is visible in the reported diagnostics. The formulation is theory-consistent, but it is not introduced here as a claim of the most general recursive-feasibility result available in the robust tracking MPC literature [15], [16].

5.5 Controller Interface and Practical Remarks

The three controllers share the same online interface and are called through a common controller architecture. The interface receives the current measured

state, the case-dependent reference, and the model data generated in the initialization workflow, and then calls the corresponding controller formulation.

The single-model controller uses one predictor page only. The LPV and robust formulations use the scheduling structures generated over the bounded operating region and update the active simplex and barycentric weights online. The robust controller additionally updates the tightened inequality margins and the tube-size variables according to the current disturbance description and case configuration.

The numerical values of the weights, horizons, payload parameters, robustness margins, and scenario-dependent settings are intentionally not detailed here, because they belong to the simulation setup rather than to the formulation itself. Those elements are described in Chapter 6 together with the validation workflow and the performance metrics used in the comparative campaign.

Taken together, the three formulations provide a common comparison framework for the thesis: the same reduced benchmark, the same constrained tracking structure, and three different levels of model fidelity and robustness, namely a single linear predictor, an LPV scheduled predictor, and a tube-based robust extension of the scheduled formulation.

Chapter 6

Simulation Setup and Metrics

This chapter describes the simulation framework used to evaluate the controllers introduced in Chapter 5. Its purpose is to define the closed-loop architecture, the case-dependent setup, and the performance indicators used throughout the comparative campaign. The corresponding results are discussed in Chapter 7.

The simulations must be interpreted consistently with the assumptions introduced in Chapters 2–5. In particular, the controller operates on the reduced joint-space model, the spacecraft attitude is propagated only in the nonlinear plant, the LPV predictor is scheduled within a bounded local region, and the robust formulation combines tube tightening with stricter target semantics than the nominal ones. As a result, the reported closed-loop behavior reflects the combined effect of prediction locality, mismatch assumptions, and constraint handling.

6.1 Closed-Loop Simulation Architecture

The closed-loop simulations are executed in MATLAB/Simulink using a nonlinear reduced plant and a controller block. The plant propagates the reduced joint coordinates and rates through the free-floating reduced dynamics, while the spacecraft attitude quaternion is integrated separately from the reconstructed base angular velocity obtained through momentum composition. At each sampling instant, the controller receives the measured reduced state, the active reference, and the scenario-dependent model parameters, solves the corresponding QP, and applies the commanded joint torque to the nonlinear plant.

All controller variants are evaluated through the same interface. The differences between them are limited to prediction-model construction and,

for the robust formulation, to the additional tightening and ancillary-feedback logic. The optimization problems are formulated in YALMIP and solved with OSQP, with warm-start based on the receding-horizon shift of the previous solution. This common structure is important for fairness, because the controllers share the same plant, the same simulation environment, and the same initialization logic.

The overall closed-loop organization is shown in Figure 6.1. The scheme highlights the separation between the reference interface, the controller block, the nonlinear plant, and the output-error path used for KPI extraction.

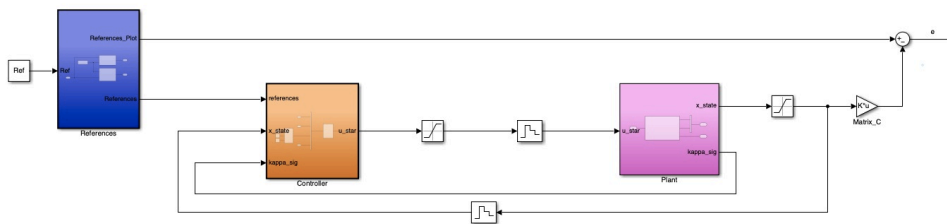


Figure 6.1: Closed-loop MATLAB/Simulink architecture used in the comparative campaign. The reference block provides the current sample and preview information, the controller block solves the selected optimization problem and outputs the commanded torque, and the nonlinear plant propagates the reduced free-floating dynamics. The output error e is computed from the reference and the measured controlled output and is used for monitoring and KPI extraction.

The reference block is treated as a separate interface layer between offline reference generation and the online controller call. This allows the same closed-loop structure to support both constant setpoints and preview-based trajectory references. In state-regulation mode, the controller receives either a constant setpoint or its repeated preview stack. In end-effector mode, the same interface provides the local output reference together with the affine shift terms required by the output model introduced in Chapter 4. This choice keeps the controller core independent of case-specific reference-generation logic and makes the reference semantics explicit at the simulation level.

All simulations log the signals needed to recompute the reported metrics offline, including tracking errors, state/input/output constraint indicators, QP solve times, and selected scheduler- or tube-related diagnostics. These signals are then post-processed into KPI summaries and representative figures. Raw time histories are preserved for traceability and figure generation, whereas the comparative discussion in Chapter 7 relies mainly on aggregated KPIs

and a limited number of representative plots. This keeps the results chapter focused while preserving the information needed for technical interpretation.

6.2 Simulation Cases and Setup

The simulation setup is generated from a common initialization procedure that builds the parameter structure shared by plant and controller. This structure contains geometry, inertial data, operating-point information, prediction-model settings, controller settings, constraint definitions, and case-dependent overrides. Plant-side quantities include the nonlinear reduced model, payload-dependent inertial quantities, quaternion propagation options, and diagnostic signals. Controller-side quantities include the reduced state definition, horizon length, weights, output mode, LPV region data, tightening data, and solver settings. Constraint definitions are kept consistent across nominal and robust formulations.

This common initialization flow is important because the comparative campaign reuses the same benchmark structure across all cases. The observed differences should therefore originate from the scenario definition and controller formulation rather than from parameter inconsistencies.

The campaign is organized into four case studies with different purposes. Case 1 is a nominal state-regulation benchmark used to verify local closed-loop consistency across multiple initial conditions. Case 2 is a pre-capture tracking scenario in end-effector mode and is used to assess trajectory-following behavior under a local affine output map. Case 3 collects post-capture regulation scenarios with payload mismatch and is divided into a fixed-mismatch subgroup and a payload-estimate-update subgroup. Case 4 is a stressed post-capture benchmark in which payload uncertainty is combined with additional bounded mismatch.

The controller output mode is selected consistently with the case definition. Cases 1, 3, and 4 are evaluated in state-regulation mode, whereas Case 2 is evaluated in end-effector mode. This distinction is essential because the interpretation of the tracking error depends on whether the controlled quantity is the reduced state or the local affine end-effector output.

In state-regulation mode, the references are setpoint values on the reduced state components. In end-effector mode, the online controller does not optimize the exact nonlinear end-effector map. Instead, it uses the local affine surrogate introduced in Chapter 4 and therefore requires a consistent shift of both references and bounds. For this reason, the end-effector references used in Case 2 are generated offline together with the corresponding affine bias terms. The online controller then receives quantities that are consistent with

the linear constrained OCP actually being solved. As a result, the reported end-effector error in Case 2 reflects both tracking quality and output-model locality.

The LPV predictor uses a local scheduling box selected for the considered case study. This region is triangulated offline, and online scheduling is performed by identifying the active simplex and computing the corresponding barycentric weights. To limit unnecessary switching between neighboring simplices, the implementation includes hysteresis and minimum-dwell logic. Projection and clipping are applied when the scheduling point approaches the boundary of the selected region. These mechanisms improve runtime robustness and also provide an indicator of predictor locality. Frequent clipping suggests that the closed-loop trajectory is exploring regions close to the boundary of the intended predictor domain.

Payload-related quantities, admissible mass ranges, coupling parameters, and mismatch proxies are modified consistently through the case definition before predictor construction and controller initialization. In post-capture cases, the controller uses an estimated payload mass \hat{m}_p , whereas the nonlinear plant uses the true payload value m_p . In Case 3B, the estimated payload mass is updated online through a step change that emulates a runtime model update after capture. The update is clamped to the offline parameter grid used to construct the LPV pages, so that the online predictor remains inside the convex hull assumed during model construction.

For clarity, the simulation setup is summarized at three complementary levels. Table 6.1 collects the common numerical and structural settings shared across the campaign. Table 6.2 summarizes the role of each reported case. Table 6.3 lists the KPI families used in the final comparison.

6.3 Performance Metrics and Validation

Tracking performance is assessed through RMS error, final tracking error, peak error, and settling-related indicators when meaningful. In state-regulation mode, these quantities are computed on the controlled state components. In end-effector mode, they are computed on the output components defined by the local affine map used by the controller. The settling indicator is used comparatively and must be interpreted with care. In the regulation cases, it is primarily influenced by the convergence of the joint rates and therefore does not necessarily imply the smallest residual position error. For this reason, final error and transient torque demand are given more interpretive weight than settling time alone.

To characterize actuation demand, the reported results also distinguish

between peak and RMS control effort. This is useful when comparing impulsive torque demand with sustained actuation effort across the nominal and robust formulations.

Constraint handling is quantified through violation counts, maximum violation magnitude, and violation duration for state, input, and output bounds. When soft constraints are enabled, slack activation is also monitored to distinguish hard physical violations from optimizer-driven relaxation. Infeasibility is tracked separately through solver status and fail-rate statistics. These indicators are used both to validate numerical consistency and to interpret conservative behavior in the robust controller.

Computational effort is quantified through QP solve-time statistics, including mean solve time, 95th-percentile solve time, and maximum observed solve time. These quantities are used comparatively under a common software stack to rank the relative computational burden of the three controllers. They are not interpreted as a direct claim of hard real-time deployability.

Additional diagnostics are used to distinguish actual controller behavior from artifacts caused by frame inconsistencies, scheduling-region edge effects, or predictor-locality limitations. Within the reported campaign, two indicators are especially useful: the projection/clipping frequency, which measures how often the online scheduling point must be forced back into the intended region, and the maximum infinity-norm prediction residual, which provides a compact indicator of predictor mismatch. These diagnostics are reported explicitly only when they become central to the interpretation of a case, most notably in the end-effector tracking benchmark. Additional signals such as slack activation, tightening-related indicators, quaternion-norm drift, and physics-consistency checks were also inspected offline and are discussed only when case-defining.

Before comparing the controllers on the final case studies, deterministic checks are used to verify plant-controller interface consistency, output-map coherence, constraint handling, scheduler behavior, and numerical reliability of the reduced free-floating implementation. These checks are intended to ensure that the differences reported later in Chapter 7 reflect controller behavior rather than interface inconsistencies.

Unless explicitly stated otherwise, the compared controllers are evaluated under the same plant benchmark, the same main constraint definitions, the same sampling time, the same horizon length, and the same overall weighting structure. The purpose is to preserve a fair comparison in which the observed differences can be attributed mainly to predictor locality, scheduling, and robustness mechanisms rather than to case-by-case retuning. The only case-specific changes are those required by the scenario itself: operating-point selection, output mode, payload mismatch definition, and runtime payload-estimate updates in Case 3B.

The reported campaign is treated as a deterministic comparative benchmark. Monte Carlo and ablation studies remain relevant extensions for future work, but they are not part of the final set of results discussed in Chapter 7 unless explicitly stated. Only the final deterministic subset is discussed in detail in the manuscript, mainly for readability and case-to-case comparability. Cases 1–4 are therefore treated as the reference benchmark of the thesis. Compared with the regulation cases, Case 2 relies more strongly on output-space metrics and predictor-validity diagnostics because the controlled variable is the local affine end-effector output introduced in Chapter 4 rather than the reduced joint state itself. Case 3 is split into two complementary subgroups. Case 3A collects the fixed-mismatch scenarios, whereas Case 3B collects the corresponding runtime parameter-update scenarios. This distinction is maintained throughout the reported results because it separates static model mismatch from sensitivity to online payload-estimate correction.

Table 6.1: Common simulation settings used throughout the reported campaign. Only the benchmark scenario, output mode, reference type, and case-dependent mismatch settings vary across cases.

Setting category	Adopted configuration
Simulation environment	MATLAB/Simulink closed-loop architecture with nonlinear reduced plant and controller block
Manipulator structure	Planar free-floating spacecraft with a two-link revolute arm ($n = 2$)
Prediction horizon	$N = 50$ samples
Control horizon	$N_c = 50$ samples
Sampling time	$T_s = 0.01$ s
Reference table duration	$T_{\text{ref}} = 15$ s
Nominal simulation stop time	$T_{\text{stop}} = 15$ s for regulation cases; extended in the tracking case to allow post-reference settling
Optimization stack	YALMIP formulation with OSQP as QP solver and warm-start through receding-horizon shift
Controller state	Reduced joint-space state $x = [\theta^\top \dot{\theta}^\top]^\top$
Prediction-model family	Single linear predictor, local LPV scheduled predictor, and tube-based robust predictor built on the same reduced-state structure
Output modes	State regulation for Cases 1, 3, and 4; end-effector tracking for Case 2
Reference modes	Setpoint references for Cases 1, 3, and 4; trajectory reference for Case 2
Spacecraft geometry	Satellite side length $l_s = 1.3$ m; joint offset $d_0 = 0.84$ m
Arm geometry	Equal link lengths $l_1 = l_2 = 0.8$ m
Satellite mass	$m_s = 1700$ kg
Link mass	$m_l = 20$ kg per link
Satellite inertia	$J_s = \text{diag}(1434, 1434, 1735)$ kg m ²
Link inertia	$J_l = \text{diag}(0.01, 1.0667, 1.0667)$ kg m ²
Base attitude representation	Quaternion propagated in the nonlinear plant with scalar-first convention
Scheduling logic	Local LPV region with simplex-based interpolation, hysteresis, and projection/clipping diagnostics
Performance reporting	Tracking residuals, peak and RMS torque demand, settling-related indicators, solve-time statistics, and selected predictor-validity diagnostics

Table 6.2: Overview of the reported deterministic case studies.

Case	Output mode	Main scenario feature	Main objective
Case 1	State regulation	Nominal free-floating benchmark	Verify local closed-loop consistency and nominal constraint handling
Case 2	End-effector tracking	Pre-capture trajectory tracking	Assess output-tracking behavior under the local affine output model
Case 3A	State regulation	Post-capture fixed payload mismatch	Evaluate controller behavior under structured parametric mismatch
Case 3B	State regulation	Online payload-estimate update	Assess sensitivity to runtime parameter correction after capture
Case 4	State regulation	Payload uncertainty plus additive mismatch	Stress-test the benchmark under stronger mismatch conditions

Table 6.3: KPI families used throughout the comparative discussion.

KPI family	Reported quantities
Tracking	e_{rms} , e_{final} , e_{peak} , and settling-related indicators when meaningful
Actuation effort	Peak and RMS control effort, used to distinguish impulsive torque demand from sustained actuation effort
Constraint handling	Violation counts, maximum violation magnitude, violation duration, and slack activation when applicable
Computation	Mean, 95th-percentile, and maximum QP solve time; observed fail-rate if present
Predictor validity	Scheduler activity, projection/clipping frequency, and residual consistency indicators such as the maximum infinity-norm prediction residual
Physical consistency	Quaternion-norm deviation and selected momentum-/energy-related diagnostic trends
Robustness-related diagnostics	Tightening activity, slack signals, and auxiliary indicators used to interpret conservative behavior

Chapter 7

Comparative Simulation Results

This chapter reports the final closed-loop results of the comparative simulation campaign. The discussion is organized by case study and follows a common structure: scenario definition, KPI summary, representative plots, and technical interpretation. The metrics and diagnostics used throughout the chapter are those introduced in Chapter 6.

The results must be interpreted within the assumptions of the benchmark. In particular, the nominal LTI controller uses a single reduced predictor, the nominal LPV controller uses a locally scheduled predictor, and the robust controller combines tube tightening with stricter target anchoring than the nominal formulations. Consequently, differences in tracking accuracy, actuation demand, and computational burden should be read as the combined effect of prediction structure, locality, and robustness mechanism.

7.1 Reading Guide

Across the reported cases, nominal performance is assessed primarily through RMS error, final error, and transient torque demand. For this reason, the aggregate tables report both peak and RMS torque, so that impulsive actuation and sustained control effort can be distinguished. Robustness-oriented behavior is instead interpreted through the combination of residual error, torque demand, solve time, and, when relevant, predictor-validity diagnostics.

The controller ranking is therefore case-dependent. In local regulation conditions, a simple predictor may already be sufficient. In post-capture mismatch scenarios, lower torque peaks may be obtained at the price of higher conservatism and longer solve times. In the end-effector tracking case, the comparison must also be interpreted through projection frequency and prediction-residual diagnostics, because the controlled output is a local affine

surrogate rather than the exact nonlinear end-effector map.

7.2 Case 1: Nominal Regulation

Case 1 is the nominal free-floating regulation benchmark. Its role is to verify closed-loop consistency and comparative controller behavior in the absence of payload uncertainty and additive mismatch. Five initial conditions are considered, spanning different combinations of initial joint configuration and joint-rate level. The representative case retained here is the one with the largest transient actuation demand.

All three controllers converge to the target in the reported runs, with no state, output, or input violations and no solver failure. The aggregate results in Table 7.1 show that the nominal LTI controller provides the smallest mean final error. The LPV controller remains close in RMS behavior but does not improve the final residual. The robust controller substantially reduces peak and RMS actuation effort, but this benefit is obtained at the price of slower convergence and higher computational cost.

Table 7.1: Case 1 aggregated KPIs, averaged over the five tested initial conditions.

Controller	\bar{e}_{rms}	\bar{e}_f	\bar{u}_{pk} [N m]	\bar{u}_{rms} [N m]	\bar{t}_{set} [s]	\bar{t}_{sol} [s]	$\bar{t}_{\text{sol},95}$ [s]
Nominal LTI	2.92×10^{-2}	1.26×10^{-4}	85.1	4.64	0.84	0.033	0.047
Nominal LPV	2.95×10^{-2}	3.76×10^{-3}	85.2	4.64	0.81	0.042	0.059
Robust	3.87×10^{-2}	8.22×10^{-3}	47.9	4.07	1.73	0.118	0.146

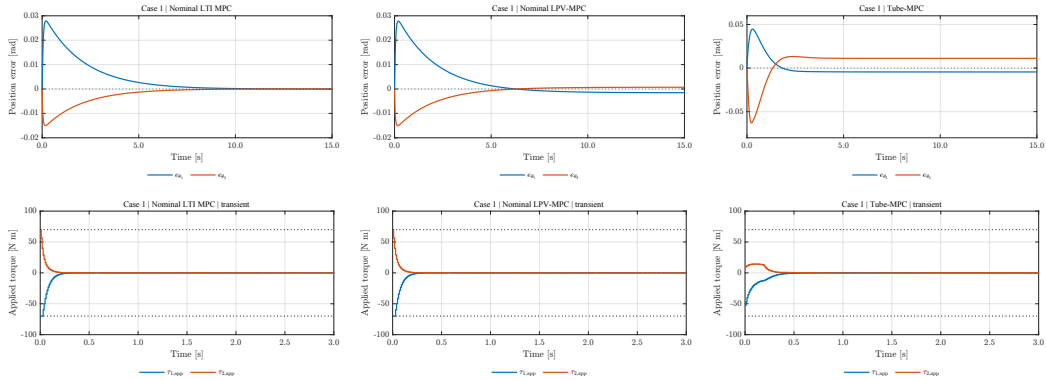


Figure 7.1: Representative Case 1 regulation run. The top row shows the joint-position error, whereas the bottom row shows the zoomed torque histories. The nominal LTI controller achieves the smallest final residual, the LPV controller remains close in transient behavior, and the robust controller reduces the torque peaks at the price of slower convergence.

Predictor-validity diagnostics remain benign in this case: the mean projection/clipping frequency stays below 8×10^{-3} and the maximum infinity-norm prediction residual remains on the order of 10^{-3} . Case 1 therefore confirms that the benchmark is strongly local from the predictor viewpoint.

Overall, Case 1 shows that the single linear predictor is already highly effective for local regulation, whereas the robust formulation makes the expected trade-off between lower actuation aggressiveness and higher conservatism visible even in the nominal benchmark.

7.3 Case 2: Pre-Capture Tracking and End-Effector Performance

Case 2 is the pre-capture tracking benchmark and is evaluated in end-effector mode. The controlled output is the local affine end-effector map introduced in Chapter 4, so the reported tracking error must be interpreted in output space rather than in joint space.

Two trajectory-tracking scenarios are retained in the reported benchmark. The more demanding one in output space is used as the representative case.

Across both scenarios, all three controllers complete the task without state, output, or input violations and without solver failure. Table 7.2 shows a different trade-off from the regulation cases: the nominal LTI controller gives the smallest final tracking residual, whereas the robust controller yields the smallest mean RMS tracking error together with substantially lower peak

and RMS torque demand. The LPV controller remains close to the LTI formulation in RMS tracking error, but does not improve the final residual and is computationally heavier.

Table 7.2: Case 2 aggregated KPIs, averaged over the two retained trajectory-tracking scenarios.

Controller	\bar{e}_{rms}	\bar{e}_f	\bar{u}_{pk} [N m]	\bar{u}_{rms} [N m]	\bar{t}_{set} [s]	\bar{t}_{sol} [s]	$\bar{t}_{\text{sol},95}$ [s]
Nominal LTI	8.27×10^{-2}	7.89×10^{-5}	98.99	12.08	6.27	0.0588	0.1048
Nominal LPV	8.23×10^{-2}	6.75×10^{-4}	98.99	12.08	6.27	0.1272	0.2641
Robust	7.98×10^{-2}	1.27×10^{-3}	64.57	6.80	6.62	2.2311	3.3587

Table 7.3: Case 2 predictor-validity diagnostics, averaged over the two retained trajectory-tracking scenarios.

Controller	Projection/clipping frequency	Max. prediction residual
Nominal LTI	7.61×10^{-1}	3.10×10^{-1}
Nominal LPV	7.61×10^{-1}	3.10×10^{-1}
Robust	7.65×10^{-1}	9.07×10^{-2}

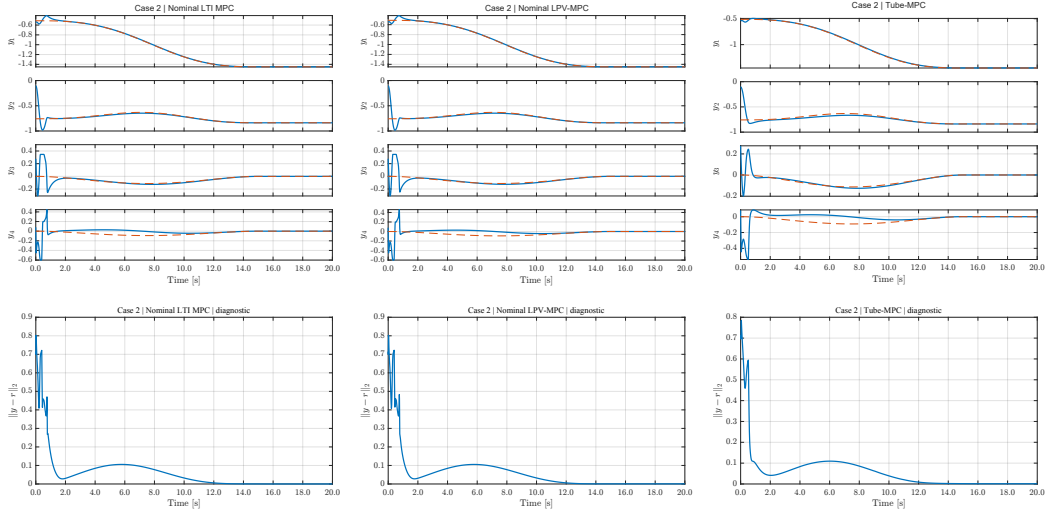


Figure 7.2: Representative Case 2 end-effector tracking run for the more demanding trajectory. The top row compares the output trajectories against their references, whereas the bottom row shows the output-error norm. The robust controller yields a smoother transient and lower integrated tracking error, while the nominal LTI controller preserves the smallest terminal residual.

A key feature of this case is the high projection frequency of the scheduled predictor. Table 7.3 shows a mean projection/clipping frequency close to 0.76 for all three controllers, much larger than in the regulation cases. At the same time, the nominal predictors exhibit a maximum prediction residual on the order of 3×10^{-1} , whereas the robust controller reduces this indicator below 10^{-1} .

Case 2 should be read primarily as a locality-limited end-effector tracking benchmark, rather than as a pure terminal-accuracy ranking test, because predictor-validity indicators become large for all controller formulations.

7.4 Case 3: Post-Capture Regulation Under Payload Mismatch

Case 3 evaluates the controllers in post-capture regulation scenarios affected by payload-related model mismatch. The case is divided into two subgroups: Case 3A, with fixed payload mismatch, and Case 3B, with an online update of the payload estimate used by the controller.

Across the reported scenarios, all three controllers complete the regulation task and remain comparable on a common deterministic benchmark. The

relevant differences therefore concern final accuracy, transient aggressiveness, settling behavior, and computational burden rather than hard feasibility loss.

7.4.1 Case 3A: Fixed Payload Mismatch

Case 3A collects the fixed-mismatch post-capture scenarios. The representative case used below corresponds to the heavier payload mismatch combined with a demanding initial condition.

Table 7.4 shows that the nominal LTI controller provides the smallest mean final regulation error. The LPV controller yields the largest final residual across the subgroup. The robust controller does not minimize final error, but it reduces the average peak torque and often produces a smoother transient. This interpretation is refined by \bar{u}_{rms} : the robust controller lowers the peak demand, but does not reduce the sustained RMS effort.

Table 7.4: Case 3A aggregated KPIs, averaged over the retained fixed-mismatch scenarios.

Controller	\bar{e}_{rms}	\bar{e}_{f}	\bar{u}_{pk} [N m]	\bar{u}_{rms} [N m]	\bar{t}_{set} [s]	\bar{t}_{sol} [s]	$\bar{t}_{\text{sol},95}$ [s]
Nominal LTI	7.09×10^{-2}	6.36×10^{-3}	94.3	10.83	3.55	0.022	0.034
Nominal LPV	7.56×10^{-2}	3.35×10^{-2}	94.4	10.83	3.69	0.288	0.391
Robust	8.13×10^{-2}	8.81×10^{-3}	76.3	12.28	2.45	1.329	1.787

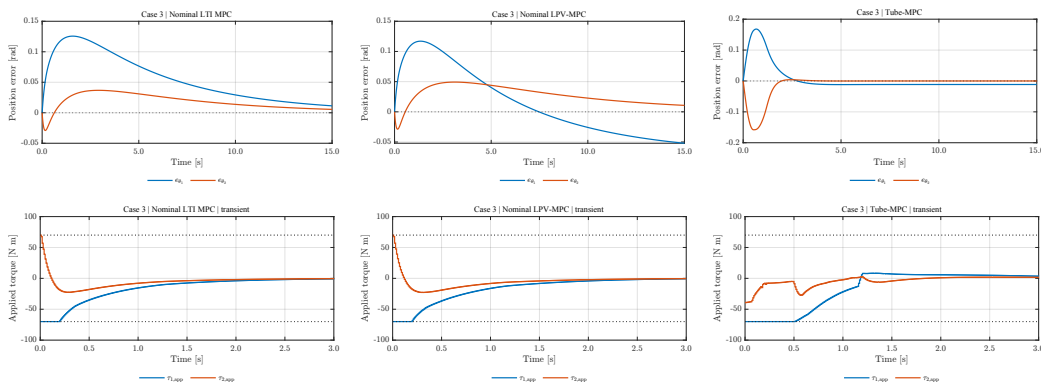


Figure 7.3: Representative Case 3A run for the heavier fixed payload mismatch. The top row shows the joint-position error, whereas the bottom row shows the zoomed torque histories. The nominal controllers preserve smaller residual errors, while the robust controller limits the torque peak more effectively at the price of a much larger solve-time burden.

Predictor-validity diagnostics remain moderate in this subgroup: the mean projection/clipping frequency is about 1.5×10^{-2} for the nominal controllers and about 3.5×10^{-2} for the robust formulation, while the maximum prediction residual remains on the order of 10^{-3} for all three controllers. The subgroup therefore remains largely inside the intended predictor region.

Overall, Case 3A shows that under fixed payload mismatch the nominal LTI formulation remains highly competitive in final regulation accuracy, whereas the robust formulation trades nominal accuracy for reduced peak actuation and smoother transients.

7.4.2 Case 3B: Online Payload-Estimate Update

Case 3B investigates the effect of an online update of the payload estimate used by the controller. The representative case used below corresponds to an upward correction of the payload estimate from a lighter to a heavier value.

The KPI values in Table 7.5 show that the controller ordering remains qualitatively similar to Case 3A: the nominal LTI controller retains the smallest mean final error, the LPV controller does not provide a systematic final-error advantage, and the robust controller remains the least aggressive in peak torque demand while also being the most computationally demanding.

Table 7.5: Case 3B aggregated KPIs, averaged over the retained payload-update scenarios.

Controller	\bar{e}_{rms}	\bar{e}_{f}	\bar{u}_{pk} [N m]	\bar{u}_{rms} [N m]	\bar{t}_{set} [s]	\bar{t}_{sol} [s]	$\bar{t}_{\text{sol},95}$ [s]
Nominal LTI	7.43×10^{-2}	4.83×10^{-3}	93.4	10.92	3.24	0.023	0.035
Nominal LPV	7.86×10^{-2}	2.64×10^{-2}	93.4	10.92	3.75	0.291	0.379
Robust	8.38×10^{-2}	7.75×10^{-3}	76.2	13.55	2.32	1.218	1.416

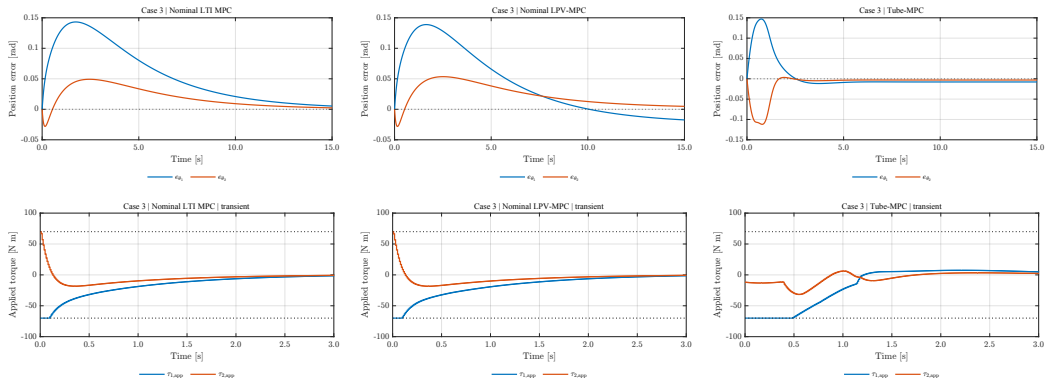


Figure 7.4: Representative Case 3B run for an upward payload-estimate correction. The top row shows the joint-position error, whereas the bottom row shows the zoomed torque histories. The correction improves the final residual with respect to the corresponding fixed-mismatch case, while the overall controller ordering remains qualitatively unchanged.

In the retained scenarios, upward corrections toward the heavier-payload cases improve the final residual more clearly, whereas downward corrections do not show the same benefit and may even degrade the final error. This indicates that the effect of runtime payload updates depends on both the direction and the magnitude of the correction.

Compared with Case 3A, the predictor-validity diagnostics become more informative. The mean projection/clipping frequency remains moderate, but the maximum prediction residual rises more markedly in the nominal formulations than in the robust one. Case 3B is therefore best interpreted as a controlled runtime parameter-update study rather than as a fully adaptive benchmark.

7.5 Case 4: Stressed Post-Capture Regulation Under Payload Uncertainty and Additive Mismatch

Case 4 is the stressed post-capture scenario. It combines payload uncertainty with additional bounded mismatch and is intended to probe controller behavior when the regulation task remains feasible but the closed-loop demand becomes more severe. The more demanding reported scenario is used as the representative case.

No solver failure or infeasibility was observed in the retained scenarios. Case 4 is therefore best interpreted as a severe regulation benchmark in

which the main differences emerge in final residual, torque aggressiveness, and computational burden rather than in loss of feasibility.

Table 7.6: Case 4 aggregated KPIs, averaged over the retained stressed scenarios.

Controller	\bar{e}_{rms}	\bar{e}_{f}	\bar{u}_{pk} [N m]	\bar{u}_{rms} [N m]	\bar{t}_{set} [s]	\bar{t}_{sol} [s]	$\bar{t}_{\text{sol},95}$ [s]
Nominal LTI	3.00×10^{-2}	2.81×10^{-5}	87.6	4.87	1.34	0.031	0.044
Nominal LPV	3.01×10^{-2}	3.40×10^{-3}	87.7	4.86	1.64	0.134	0.158
Robust	4.20×10^{-2}	5.52×10^{-4}	51.1	4.52	1.58	1.932	2.460

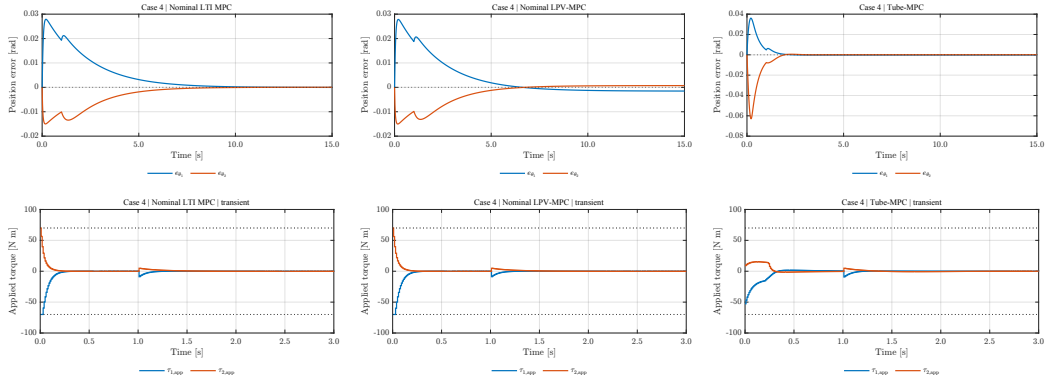


Figure 7.5: Representative Case 4 stressed run for the more demanding reported scenario. The top row shows the joint-position error, whereas the bottom row shows the zoomed torque histories. The nominal LTI controller preserves the smallest final residual, while the robust controller markedly reduces the torque peaks at the price of a much larger computational burden.

The additional diagnostics clarify the nature of this case. The mean projection/clipping frequency remains low, below 9×10^{-3} for all controllers, whereas the maximum prediction residual rises to about 5×10^{-2} for all three formulations. Case 4 is therefore not dominated by scheduling-region excursion, but by a stronger mismatch level within still-local trajectories.

Overall, Case 4 reinforces the same trade-off already visible in the post-capture scenarios: the robust formulation reduces torque aggressiveness, but at the price of stronger conservatism and much larger solve times.

7.6 Cross-Case Synthesis

Three recurring trends emerge from the reported benchmark.

First, in local and moderately stressed regulation scenarios, the single linear predictor is already highly effective. It repeatedly provides the smallest final regulation residual, as shown by Case 1, by the mean results in Case 3A and Case 3B, and by the retained stressed Case 4 scenario.

Second, in the present implementation, the LPV predictor does not provide a systematic final-error improvement over the LTI baseline in the retained regulation cases. Its role is instead to represent the local dependence of the predictor on the operating condition more explicitly. This becomes useful when predictor locality and output-space interpretation matter, but in the reported dataset it does not change the overall ranking in terminal accuracy.

Third, the robust controller changes the transient more than the final ranking. Across the post-capture cases, it repeatedly lowers peak torque and often produces less aggressive behavior under bounded mismatch. These benefits are obtained at the price of stronger conservatism and a much larger computational burden, and they do not imply that the robust formulation is the best one for final-residual minimization.

The predictor-validity diagnostics help interpret these results. In Case 1 they remain very small, confirming that the benchmark is strongly local. In Case 2 they become large, which shows that the tracking results must be read together with the locality of the affine end-effector output model. In Case 3B the payload-estimate update perturbs the nominal predictors more clearly than the robust one. In Case 4 the projection frequency remains small while the prediction residual increases, indicating a mismatch-dominated stressed condition rather than a lack of scheduling coverage.

Taken together, the cases show a clear engineering trade-off. A minimal predictor is already sufficient in the local regulation scenarios considered here, LPV scheduling makes the predictor more sensitive to operating-condition changes without yet outperforming the LTI baseline in terminal accuracy, and the robust layer is mainly useful when reduced torque aggressiveness and safer conservative behavior are preferred over nominal sharpness.

Chapter 8

Conclusion and Future Work

8.1 Summary of Contributions

This thesis developed an MPC framework for a planar free-floating spacecraft–manipulator benchmark under bounded prediction mismatch and payload uncertainty. The work combines reduced nonlinear modeling, local linear prediction, nominal MPC design, tube-based robustness, and comparative simulation-based validation within the same control-oriented study.

From the modeling side, the thesis established a reduced free-floating plant based on momentum conservation and Schur-complement reduction. The nonlinear simulation still propagates the spacecraft attitude quaternion for kinematic consistency and diagnostic checks, while the controller operates on a reduced joint-space state. This choice preserves the dominant base–arm coupling effects while keeping the online optimization problem tractable.

From the prediction side, the thesis constructed a local discrete-time predictor family from multiple operating-point linearizations, affine correction terms, and lightweight online interpolation. The resulting LPV predictor is not intended as a globally exact model of the nonlinear plant. Its purpose is to improve local predictor representation while preserving a QP structure suitable for online MPC.

From the control side, three formulations were developed and compared under the same benchmark assumptions: a nominal LTI-MPC, a nominal LPV-MPC, and a tube-based robust MPC. The validation campaign covered nominal regulation, end-effector tracking, post-capture payload mismatch, online payload-estimate updates, and retained stressed post-capture conditions.

The reported results support three main conclusions. First, the nominal LTI controller based on a single linear predictor is already a very strong baseline for local and moderately stressed regulation, and it repeatedly provides

the smallest final residual in the retained regulation cases. Second, the LPV predictor captures local operating-point dependence more explicitly, but in the present dataset it does not provide a systematic final-accuracy advantage over the LTI baseline. Third, the robust controller is useful when lower torque peaks and more conservative transients are required under bounded mismatch, but these gains come with stronger conservatism, stricter target semantics, and much larger solve times.

These conclusions must be read within the scope of the benchmark adopted in the thesis: planar free-floating motion, reduced joint-space prediction, local LPV scheduling, constant-Bd predictor structure, and bounded mismatch budgeting. Within this scope, the main contribution of the work is that the effect of predictor choice, robustness mechanisms, and model mismatch can be evaluated on the same reduced plant and under the same main comparison assumptions, instead of being inferred from disconnected case-specific designs.

8.2 Future Research Directions

Several extensions follow naturally from the limitations identified in the present work.

A first direction concerns prediction fidelity. The current LPV surrogate relies on local linearization, Forward Euler discretization, frozen-quaternion page generation, and a constant- B_d page structure. Future work may investigate more accurate discretization and page-construction strategies while preserving tractable online QP assembly.

A second direction concerns the LPV architecture itself. In the current benchmark, LPV scheduling remains local and can require substantial projection activity in end-effector tracking conditions. A natural extension is to improve region design, page coverage, and interpolation logic so that the scheduled predictor remains informative over a wider set of trajectories without introducing excessive online complexity.

A third direction concerns the robust controller. In the present formulation, robustness and target semantics are coupled, because the tube center is anchored more strictly to the requested reference than in the nominal formulations. A more homogeneous robust-tracking formulation should combine tube-based robustness with artificial-reference optimization, so that robustness effects and target-definition effects can be separated more cleanly.

A fourth direction concerns payload adaptation. In this thesis, runtime payload-estimate updates are injected externally and clamped to a precomputed predictor family. A more advanced extension would combine online payload estimation with principled uncertainty-set management, so that pa-

parameter adaptation can improve closed-loop behavior without violating the assumptions used to construct the scheduled predictor and the tightening layer.

Finally, the benchmark could be extended beyond the present planar free-floating setting toward more general three-dimensional dynamics, richer post-capture inertial models, or architectures in which manipulator motion, spacecraft attitude, and guidance generation are optimized in a more integrated way. These extensions would allow the benchmark to cover richer dynamics, broader uncertainty descriptions, and tighter integration between estimation and control.

Bibliography

- [1] M. Wilde, S. Kwok Choon, A. Grompone, and M. Romano, “Equations of motion of free-floating spacecraft–manipulator systems: An engineer’s tutorial,” *Frontiers in Robotics and AI*, vol. 5, p. 41, 2018. DOI: 10.3389/frobt.2018.00041.
- [2] E. Papadopoulos, F. Aghili, O. Ma, and R. Lampariello, “Robotic manipulation and capture in space: A survey,” *Frontiers in Robotics and AI*, vol. 8, p. 686723, 2021. DOI: 10.3389/frobt.2021.686723.
- [3] M. Oda, “Motion control of the satellite mounted robot arm which assures satellite attitude stability,” *Acta Astronautica*, vol. 41, no. 11, pp. 739–750, 1997, PII: S0094-5765(97)00214-2.
- [4] J. J. Korczyk, D. Posada, A. Malik, and T. Henderson, *Modeling of an on-orbit maintenance robotic arm test-bed*, arXiv preprint, Preprint (AAS 21-760), 2022. arXiv: 2203.01403 [cs.R0].
- [5] E. Papadopoulos and S. Dubowsky, “Kinematics, dynamics, and control of free-flying and free-floating space robotic systems,” *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 531–543, 1993. DOI: 10.1109/70.258046.
- [6] Y. Xu and T. Kanade, *Space Robotics: Dynamics and Control*. Springer, 1993. DOI: 10.1007/978-1-4615-3588-1.
- [7] K. Yoshida, “Engineering test satellite vii flight experiments for space robot dynamics and control: Theories on laboratory test beds ten years ago, now in orbit,” *The International Journal of Robotics Research*, vol. 22, no. 5, pp. 321–335, 2003. DOI: 10.1177/0278364903022005003.
- [8] K. Yoshida, “Ets-VII flight experiments for space robot dynamics and control,” in *Experimental Robotics VII*, Springer, 2007. DOI: 10.1007/3-540-45118-8_22.

- [9] Canadian Space Agency. “About canadarm2.” Official ISS Canadarm2 overview; last modified 16 July 2024, Canadian Space Agency. (2024), [Online]. Available: <https://www.asc-csa.gc.ca/eng/iss/canadarm2/about.asp>.
- [10] European Space Agency. “European robotic arm.” Overview and facts & figures for the ISS Russian segment, European Space Agency. (2021), [Online]. Available: https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/International_Space_Station/European_Robotic_Arm.
- [11] S. R. Breon, R. F. Boyle, M. B. Francom, *et al.*, “Robotic refueling mission-3—an overview,” in *IOP Conference Series: Materials Science and Engineering*, vol. 755, 2020, p. 012002. DOI: 10.1088/1757-899X/755/1/012002.
- [12] F. Basana, Z. Pavanello, F. Branz, *et al.*, “Satellite and robotic arm combined control for spacecraft close-proximity operations,” *CEAS Space Journal*, vol. 17, pp. 309–335, 2025. DOI: 10.1007/s12567-024-00560-0.
- [13] T. Besselmann, J. Löfberg, and M. Morari, “Explicit mpc for lpv systems: Stability and optimality,” *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2322–2332, 2012. DOI: 10.1109/TAC.2012.2187400.
- [14] G. Sánchez, M. Murillo, L. Genzelis, N. N. Deniz, and L. Giovanini, “Mpc for nonlinear systems: A comparative review of discretization methods,” in *XVII Workshop on Information Processing and Control (RPIC)*, 2017, pp. 1–6. DOI: 10.23919/RPIC.2017.8214333.
- [15] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000. DOI: 10.1016/S0005-1098(99)00214-9.
- [16] T. Peschke and D. Görges, “Robust adaptive tube tracking model predictive control for piece-wise constant reference signals,” *International Journal of Robust and Nonlinear Control*, vol. 33, no. 14, pp. 8158–8182, 2023. DOI: 10.1002/rnc.6814.
- [17] A. Ferramosca, D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho, “Mpc for tracking with optimal closed-loop performance,” *Automatica*, vol. 45, no. 8, pp. 1975–1978, 2009. DOI: 10.1016/j.automatica.2009.04.007.

- [18] P. Krupa, J. Köhler, A. Ferramosca, I. Alvarado, M. N. Zeilinger, T. Alamo, and D. Limon, “Model predictive control for tracking using artificial references: Fundamentals, recent results and practical implementation,” in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, 2024. DOI: 10.1109/CDC56724.2024.10886854.