



UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI

CORSO DI LAUREA IN INGEGNERIA MECCATRONICA

Tesi di Laurea

Development and Evaluation of Sensorless Model Predictive Force Control Schemes for Single Linear Actuators in Surgical Applications

Relatori:

Prof. Roberto Oboe

Prof. Tomoyuki Shimono

Candidato:

Pietro Toffanello

Matr. 2122058

Anno Accademico 2025/2026

Abstract

Ensuring safety in human-robot interaction is of paramount importance in surgical robotics applications, where excessive forces and velocities could cause irreversible damage to patients or handled tissues. Traditional controllers often fail to natively handle these types of constraints, thus risking tissue perforation.

This thesis presents the development and evaluation of various Model Predictive Control (MPC) architectures for the force control of linear actuators, specifically designed to guarantee patient safety. A distinctive feature of the proposed architectures is the use of a Kalman Filter configured as an Unaccessible Input Observer (UIO), acting as a state observer, which allows for a real-time estimation of the force exerted by the motor on the interacting environment.

Specifically, three different approaches were analyzed:

- **Linear MPC (L-MPC):** based on a linear tissue model, represented by its average stiffness.
- **Non-linear MPC (NL-MPC):** which integrates a non-linear model of tissue stiffness.
- **Adaptive MPC (AMPC):** which utilizes a Recursive Least Squares (RLS) algorithm to perform online estimation of environment characteristics, enabling the system to autonomously adapt to different tissue types.

Experimental results demonstrate that while the linear and non-linear architectures achieve good performance on modeled tissues, the AMPC provides greater flexibility and high performance even in unmodeled scenarios. Notably, all three proposed architectures achieved faster convergence, lower overshoot, and fewer oscillations compared to a more conventional Proportional controller with a Disturbance Observer. Furthermore, the transition between free motion and contact was also managed by investigating an effective contact detection strategy.

Contents

1	Introduction	3
1.1	Evolution of robotic surgery: from state of the art to open challenges	3
1.2	Robotic surgery and safety in human-robot interaction	5
1.3	Sensorless Force Estimation	7
1.4	Transition between Free-Motion and Contact State	7
1.5	Thesis Outline	8
2	System modeling	9
2.1	Linear Actuator's dynamics	9
2.2	Coulomb friction compensation	10
2.3	State-Space Representation	13
3	Kalman Filter as State Observer	15
3.1	State Observer	15
3.1.1	Open-Loop Observer	16
3.1.2	Closed-Loop Estimator	17
3.1.3	Observability and Detectability	18
3.1.4	The Separation Principle	19
3.2	Unaccessible Input Observer - UIO	21
3.3	Kalman Filter	23
3.3.1	Overview	23
3.3.2	Kalman Filter Environment-Agnostic Model and Tuning .	33
3.3.3	Force Estimation Validation	41
4	Model Predictive Control	43
4.1	Theoretical Background	44
4.1.1	State Prediction	44
4.1.2	Optimization Theory: Quadratic Programming	46
4.2	Control Architectures for Tissue Interaction	55
4.2.1	Common Control Scheme and Cost Function	55

4.2.2	Linear MPC (L-MPC)	59
4.2.3	Nonlinear MPC (NL-MPC)	60
4.2.4	Adaptive MPC (AMPC)	61
5	Handling Transitions: Free-Motion to Contact	64
5.1	Contact Detection Algorithm	65
5.2	Weights scheduling Algorithm	67
5.3	Dynamic Reference Algorithm	68
5.4	AMPC Model for Free Motion to Contact Transition	71
6	Experimental Implementation and Results	73
6.1	Experimental setup	74
6.2	Evolution and Comparative Analysis of Control Schemes	77
6.2.1	Linear MPC and Parameter Tuning	77
6.2.2	Nonlinear MPC Implementation	79
6.2.3	Adaptive MPC and Overall Comparison	82
6.3	Safety and Robustness Stress Test	86
6.4	Free-Motion to Contact Transition	88
6.5	Dynamic Trajectory Tracking in Continuous Contact	90
6.6	Experimental Validation on Complex Biological Tissues	91
7	Conclusions and Future Work	95
7.1	Summary of Contributions	95
7.2	Synthesis of Experimental Findings	96
7.3	System Limitations	97
7.4	Future Work	98
	Bibliography	99

Chapter 1

Introduction

1.1 Evolution of robotic surgery: from state of the art to open challenges

Robotic surgery is a discipline born from the synergy between engineering and medicine, with the aim of overcoming the limitations of traditional laparoscopic techniques. These limitations, as described by Rivero-Moreno et al. in [7] and Freschi et al. in [3], include:

- **Limited Degrees of Freedom (Dexterity):** traditional laparoscopic instruments offer only 4 degrees of freedom (DOFs), which include the positioning of the instrument tip and its actuation. Robotic systems such as the da Vinci (Intuitive Surgical), thanks to internal joints, allow for 7 degrees of freedom, also offering the possibility of performing complex movements such as pronation and supination of the instrument inside the patient.
- **Physiological tremor and precision:** due to their length, laparoscopic instruments tend to amplify the surgeon's natural hand tremor. Robotic systems allow for active filtering of these tremors and also offer the possibility of motion scaling, transforming large hand movements into precise micro-movements of the instrument tip.
- **The "fulcrum" effect:** in manual laparoscopy, the length of the instruments causes them to act as a lever against the abdominal wall, forcing the surgeon into counter-intuitive maneuvers. The robotic system eliminates this problem by translating the surgeon's hand movements into corresponding and intuitive movements of the instrument tip.

- **Ergonomics and fatigue:** manual surgery often forces the surgeon to remain in uncomfortable and tiring positions for prolonged periods. With robotic surgery, the surgeon operates comfortably seated at a console with ergonomic arm supports, significantly reducing physical fatigue during the procedure.
- **Two-dimensional vision and depth perception:** traditional laparoscopy uses 2D monitors that reduce depth perception, impairing hand-eye coordination. Robotic surgery, through the use of stereoscopic cameras, provides the surgeon with a three-dimensional view, restoring depth perception.



Figure 1.1: Da Vinci 5, patient side (<https://www.intuitive.com>)

Systems such as the Da Vinci [3] [7], shown in figures 1.1 and 1.2, have revolutionized the minimally invasive surgical approach, ensuring enormous advantages in terms of reduced tissue trauma, recovery times, and precision.



Figure 1.2: Da Vinci 5 setup (<https://www.intuitive.com>)

1.2 Robotic surgery and safety in human-robot interaction

Despite the significant advantages described above, many current systems still present a significant technological limitation: the lack of *tactile or haptic feedback*. Currently, the surgeon "sees" the tissue deformation but does not "feel" the force being exerted [3].

This lack of physical perception shifts the entire safety burden onto the operator's visual capabilities. It therefore becomes fundamental to develop control systems that not only estimate the force exerted on the tissues but are also capable of intervening autonomously to ensure that no unintentional damage is caused to the handled tissues. This is one of the themes addressed in this thesis, in which the use of **Model Predictive Control (MPC)** is proposed to automate safety management. Clinical studies [2] indicate that forces exceeding 10 N applied to abdominal organs represent a concrete risk of permanent damage, making rigorous control of force and velocity indispensable. Consequently, ensuring compliance with safety limits is not merely a performance requirement but a fundamental necessity.

To investigate this safety management problem, in this thesis the force interac-

tion dynamics is decoupled from the broader kinematic complexity of the robotic arm and the scope of the project is deliberately narrowed to the one-dimensional case. By focusing on a single linear actuator, which effectively models the insertion and penetration axis of a surgical tool, it becomes possible to rigorously isolate the performance of the force observer and the controller. This 1-DoF reduction allows for a deep analysis of the tissue impact dynamics and the safety constraints enforcement, providing a solid theoretical and experimental foundation that could, in future works, be scaled to multi-DoF architectures.

Traditional control techniques often lack native mechanisms to manage these limits, increasing the risk of tissue perforation. Furthermore, standard control approaches often struggle to balance force reference tracking accuracy with the need for smooth and safe trajectories. Indeed, as highlighted in the work of Y. Wang et al. [9], conventional *Proportional-Integral-Derivative* (*PID*) controllers often exhibit oscillatory behavior upon contact with tissue, making it difficult to apply a constant and precise force. In contrast, MPC, thanks to its predictive capabilities, is able to reach and maintain the force reference without incurring in dangerous fluctuations, unlike a classic reactive controller. In fact, the proposed systems, by using a model aware of the tissue contact, are able to anticipate the effect of the interaction rather than simply reacting to tracking errors, ensuring a more robust solution. In the end, the introduction of an algorithm to estimate the characteristics of the interacting tissue online, as proposed by F.A. Almasalmah et al. [1] in the case of bilateral control, allows for good tracking performance and stability in various interaction scenarios.

Specifically, in this thesis, three different MPC architectures are investigated:

- **Linear MPC (L-MPC):** represents the baseline approach, in which the controller relies on an average estimated environment stiffness. While it is the most computationally efficient solution, it lacks the ability to adapt to the nonlinear variations typical of biological tissues.
- **Non-linear MPC (NL-MPC):** integrates a polynomial formulation within its prediction model to describe the environment stiffness. This scheme allows the controller to manage the non-linear characteristics of tissues.
- **Adaptive MPC (AMPC):** constitutes the most advanced approach, where environmental parameters are identified online using a Recursive Least Squares (RLS) algorithm. This architecture provides maximum flexibility, making the system capable of operating on materials with significantly different mechanical properties without the need for manual retuning.

1.3 Sensorless Force Estimation

Another fundamental element for implementing force control in the surgical field is the availability of a precise and responsive measurement of the interaction force. However, integrating physical force sensors (such as load cells or strain gages) onto the tips of surgical instruments presents significant technical challenges related to miniaturization, sterilization requirements, and high costs.

To overcome these limitations, this work proposes adopting a *sensorless* approach, where the contact force is not measured but mathematically estimated. The core of this strategy is a **Kalman Filter (KF)** used as a *state observer* and configured as an *Unaccessible Input Observer* (UIO).

The fundamental idea consists of treating the environmental force not as a known parameter, but as an external disturbance acting on the dynamics of the linear motor. Specifically, the interaction is modeled as a *matched* disturbance, meaning it acts directly on the system input. The only measured variable is the motor position, obtained via a linear optical encoder with a resolution of $0.1\ \mu\text{m}$, which ensures the precision necessary for a responsive and reliable real-time force estimation.

1.4 Transition between Free-Motion and Contact State

The transition between the free-motion and contact states is a critical phase in surgical robotics, characterized by a significant discontinuity in the system's dynamics as the actuator initiates interaction with the environment. This phase is typically prone to instability or oscillatory behaviors.

This is particularly true in the case of Model Predictive Control, because the control law relies heavily on an accurate internal model of the system's dynamics. If the controller is not aware of the switch from free space to contact, the mismatch between the predicted and actual dynamics can lead to excessive impact forces or instability.

To ensure a stable and smooth transition, the proposed control architecture implements a "multi-layered" strategy:

- **Contact Detection:** a contact detection function based on different parallel thresholds to detect the impact point.
- **Dynamic Reference Function:** for making the motor approach the tissue

and then apply the desired force after the impact.

- **Gain-Scheduling Cost Function:** the weights within the MPC cost function are adjusted in real-time based on the contact state.
- **Real-time Model Update:** the internal model used by the MPC controller is updated dynamically based on the contact condition.

1.5 Thesis Outline

The remainder of this thesis is structured as follows to systematically present the development, implementation, and evaluation of the proposed surgical force control framework:

- **Chapter 2** describes the mathematical modeling of the linear actuator, the identification and feedforward compensation of friction forces.
- **Chapter 3** introduces the theoretical formulation and implementation of the Kalman Filter for sensorless force estimation, detailing the state-space representation of the system and the design of the Unaccessible Input Observer.
- **Chapter 4** presents the theory behind Model Predictive Control and the implementation of the three proposed MPC architectures (L-MPC, NL-MPC, and AMPC), including the formulation of the optimization problem.
- **Chapter 5** presents supervision algorithms used to manage the transition between free-motion and contact states, ensuring a stable and smooth interaction with the environment.
- **Chapter 6** showcases the experimental setup and presents the results obtained from testing the proposed control strategies on a linear actuator in contact with different materials, analyzing the performance in terms of force tracking accuracy, stability, and responsiveness.

Chapter 2

System modeling

2.1 Linear Actuator's dynamics

The considered system is composed of a single linear actuator of mass M and a viscous damping coefficient B . Its dynamic can be described as follows:

$$M\ddot{x}(t) + B\dot{x}(t) + F_{coulomb}(t) = F_{cmd}(t) \quad (2.1)$$

Where $\ddot{x}$ and $\dot{x}$ represent, respectively, the linear motor's acceleration and speed. $F_{coulomb}$ is the Coulomb dry friction component, while F_{cmd} is the control force. When the motor is in contact with the environment, it is also subject to a reaction force F_{env} . The tissue can be described using the Kelvin-Voigt visco-elastic model; thus, its reaction force can be modeled as the sum of an elastic and a viscous component, as shown in Fig. 2.1 and in equation (2.2), where K_{env} and B_{env} represent the stiffness and viscosity of the environment, and x represents the position of the motor, which is also the only quantity measured by the optical encoder.

$$F_{env}(t) = K_{env}x(t) + B_{env}\dot{x}(t) \quad (2.2)$$

Therefore, under contact conditions, the system's dynamic equation becomes:

$$M\ddot{x}(t) + B\dot{x}(t) + F_{coulomb}(t) = F_{cmd}(t) - F_{env}(t) \quad (2.3)$$

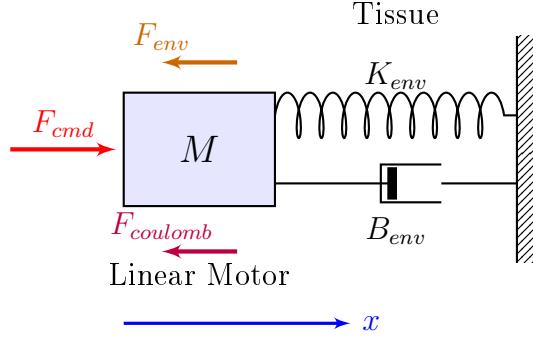


Figure 2.1: Schematic representation of the interaction between the motor, represented by the mass M , and the tissue, represented by a Kelvin-Voigt visco-elastic model with a spring and damper in parallel.

2.2 Coulomb friction compensation

To manage the non-linear nature of Coulomb friction, its effect is compensated in feedforward using a *soft* compensation strategy. Specifically, the compensation force F_{comp} is implemented using a hyperbolic tangent as an activation function to ensure a smoother zero-crossing transition:

$$F_{comp}(t) = F_c \cdot \tanh(\sigma \cdot \dot{x}(t)) \quad (2.4)$$

Where F_c is the estimated friction value and σ is a parameter governing the slope of the transition.

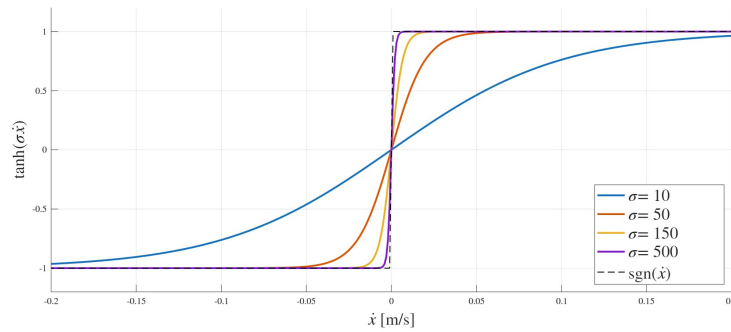


Figure 2.2: Activation function $\tanh$ with different slopes σ

This was implemented to prevent the feedforward compensation from causing instantaneous jumps, given the noisy nature of the velocity estimate.

The value of F_c was obtained through a series of experiments in which the motor was moved at various constant speeds, both positive and negative; these experiments also served to identify the motor's viscous damping coefficient B .

For each velocity value, the force generated by the motor was recorded. Since the velocity was constant, this force served solely to overcome the contributions of Coulomb and viscous friction , as can be inferred by substituting the equations into the system dynamics:

$$M\ddot{x}(t) + B\dot{x}(t) + F_{coulomb}(t) = F_{cmd}(t) \quad (2.5)$$

$$F_{coulomb}(t) = F_c \cdot \text{sgn}(\dot{x}(t)) \quad (2.6)$$

$$F_{fric}(t) = F_{coulomb}(t) + B \cdot \dot{x}(t) \quad (2.7)$$

$$\dot{x}(t) = \text{const} \implies \ddot{x}(t) = 0 \quad (2.8)$$

$$F_{cmd}(t) = F_{fric}(t) \quad (2.9)$$

For these force values, a linear fitting was performed to identify the value of B , corresponding to the slope of the lines, and the positive and negative values of F_c , corresponding to the y-axis intercepts. The results of this procedure are presented in Fig. 2.3 and Table 2.1.

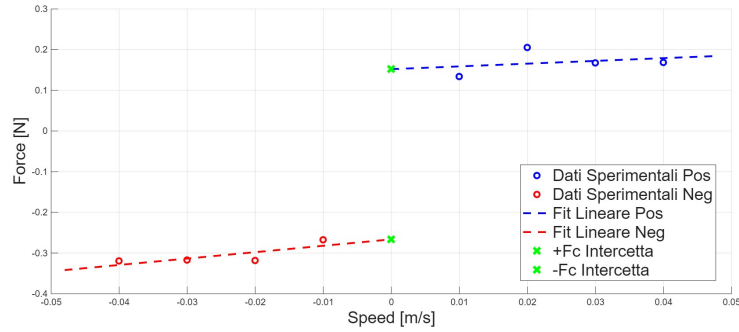


Figure 2.3: Linear fitting for the identification of B and F_c

Table 2.1: Results of the friction parameter identification procedure for the linear motor.

Parameter	Simbol	Direction (+)	Direction (-)	Average Value
Viscosity [Ns/m]	B	0.6761	1.5726	1.1244
Coulomb Friction [N]	F_c	0.1520	0.2664	0.2092

As observed, the results are asymmetrical. Consequently, the average value $B_n = 1.1244 \text{ Ns/m}$ was used for the system parameter definition , while the Coulomb friction compensation was managed as follows:

Algorithm 1: Asymmetric soft compensation algorithm

Funzione `Soft_Compensation( $\hat{x}$ )`:**Input:** $\hat{x}$: Estimated speed by the Kalman Filter**Output:** F_{comp} : Friction compensation force $F_{c,pos} \leftarrow 0.1520$ $F_{c,neg} \leftarrow 0.2664$ $\sigma \leftarrow 150$

// Transition slope

 $\alpha \leftarrow \tanh(\sigma \cdot \hat{x})$

// Activation function value

 $w \leftarrow 0.5 \cdot (1 + \alpha)$

// Interpolation weight

 $F_{c,curr} \leftarrow F_{c,pos} \cdot w + F_{c,neg} \cdot (1 - w)$ $F_{comp} \leftarrow F_{c,curr} \cdot \alpha$ **return** F_{comp}

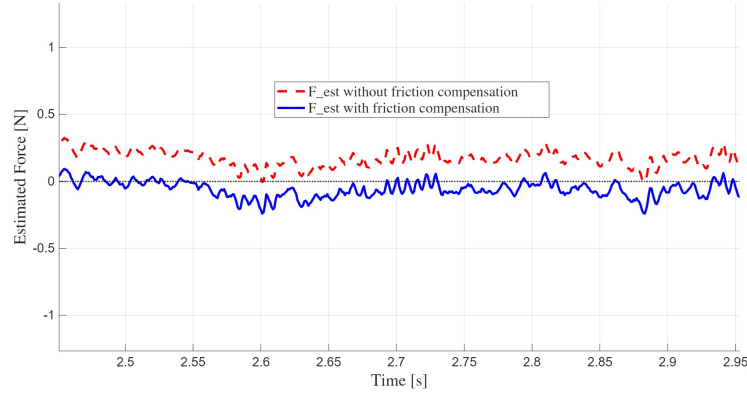


Figure 2.4: Friction compensation effect on the motor's force output. The red line represents the force without compensation, while the blue line shows the force with the implemented feedforward compensation.

Fig. 2.4 shows the results of an experiment conducted at a constant free-motion velocity. It illustrates the effect of the implemented feedforward compensation on the estimated interaction force. The red dashed line represents the reconstructed force estimation without compensation, while the solid blue line shows the actual estimation with the feedforward term active. It can be observed that the compensation effectively isolates the inherent friction dynamics from the total estimated force. This decoupling allows the Kalman Filter, described in the next chapter, to provide a clean and accurate estimation of the environmental interaction force.

2.3 State-Space Representation

Once the system dynamics and the friction compensation strategy have been defined, it is possible to derive a state-space model. This representation is required for both the design of the MPC controller and the implementation of the state observer.

The effective control input $u(t)$ is defined as the force delivered by the motor minus the effect of friction compensation:

$$u(t) = F_{cmd}(t) - F_{comp}(t) \quad (2.10)$$

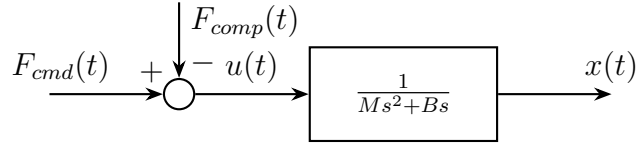


Figure 2.5: Feedforward Coulomb friction compensation scheme

Considering the motor in contact state with the environment and assuming the friction compensation is effective, the dynamic equation simplifies to the nominal model of the linear motor:

$$M_n \ddot{x}(t) + B_n \dot{x}(t) = u(t) - F_{env}(t) \quad (2.11)$$

where M_n and B_n are the nominal mass and viscous friction coefficient, respectively.

By choosing the state vector $\mathbf{x}(t) = [x(t), \dot{x}(t)]^T$, the system can be described in standard form:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}(u(t) - F_{env}(t)) \quad (2.12)$$

However, to allow the MPC controller to be aware of the interaction with the biological tissue, the Kelvin-Voigt model is integrated directly into the state matrix, resulting in an *Environment-Aware* model:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K_{env}}{M_n} & -\frac{B_n + B_{env}}{M_n} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M_n} \end{bmatrix} u \quad (2.13)$$

This formulation allows the optimizer used in the MPC to predict the system's evolution based not only on the motor's own dynamics but also on the elastic and viscous characteristics of the environment it is interacting with. The model can be further expanded by including the effect of a matched disturbance $d(t)$ acting

at the system input, described by a zero-order model ($\dot{d} = 0$), to account for possible disturbances and parametric errors:

$$M_n \ddot{x}(t) + (B_n + B_{env}) \dot{x}(t) + K_{env} x(t) = u(t) - d(t) \quad (2.14)$$

The *Environment-Aware* state-space model thus becomes:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{d} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ -\frac{K_{env}}{M_n} & -\frac{B_{env}+B_n}{M_n} & -\frac{1}{M_n} \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x \\ \dot{x} \\ d \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{M_n} \\ 0 \end{bmatrix}}_{\mathbf{B}} u(t) \quad (2.15)$$

$$\begin{bmatrix} F_{env} \\ \dot{x} \end{bmatrix} = \underbrace{\begin{bmatrix} K_{env} & B_{env} & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} x \\ \dot{x} \\ d \end{bmatrix}, \quad \mathbf{J} = 0 \quad (2.16)$$

The variables F_{env} and $\dot{x}$ were selected as outputs to allow the MPC controller to directly impose safety constraints, as discussed in the introduction, on the system outputs.

Since the control strategy is implemented in a digital computer with a sampling frequency of 1 *kHz*, the continuous-time state-space model obtained in (4.42) must be converted into a discrete-time model ($\mathbf{F}$, $\mathbf{G}$, $\mathbf{H}$, $\mathbf{J}$). Thus, assuming a Zero-Order Hold (ZOH) on the system input and denoting the sampling period as $T_s = 0.001$ s, the discretized system takes the following form:

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}u(k) \quad (2.17)$$

Where $k \in \mathbb{N}$ and:

$$\mathbf{F} = e^{\mathbf{A}T_s}, \quad \mathbf{G} = \int_0^{T_s} e^{\mathbf{A}\tau} \mathbf{B} d\tau \quad (2.18)$$

The matrices $\mathbf{H}$ and $\mathbf{J}$ remain unchanged, as the output equation is static.

Chapter 3

Kalman Filter as State Observer

The *Model Predictive Control (MPC)* strategy described in the following chapters relies on an explicit state-space model to predict the future behavior of the system. Consequently, the calculation of the optimal control action $u(k)$ at each time step requires full knowledge of the current state vector $\mathbf{x}(k)$. However, the state is rarely fully measured, and this is particularly true in surgical applications, where direct measurement of all state variables, in particular the interaction force, is often impractical due to the challenges of integrating physical sensors into miniaturized surgical instruments. In the considered system, the only measured variable is the position of the motor, which is measured with an optical linear encoder with a resolution of $0.1 \mu m$.

In literature, there are several types of state observers; however, in this project, a *Kalman Filter*, theorized by Rudolph Emil Kalman [5], was chosen as a state observer, and it was implemented as an *Unaccessible Input Observer (UIO)*, in order to obtain a sensorless estimation of the interaction force, in a similar manner as done in the work of K. Ohnishi et al. [8] with the Disturbance Observer (DOB).

In the following sections, the theoretical and mathematical formulation of the State Observer and the Kalman Filter will be detailed, including its stochastic tuning through the *Bartlett test*.

3.1 State Observer

As stated previously, in the design of advanced control systems, the availability of the full state vector $x(k)$ is a fundamental requirement. However, in many practical engineering applications measuring every state variable is either technically impossible or too expensive. This leads to the problem of *state estimation*, where internal variables must be reconstructed starting from a limited set of

measurements and the known system model.

Consider a linear time-invariant (LTI) system described by the following state-space equations:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) \end{cases} \quad (3.1)$$

where $\mathbf{x}(k) \in \mathbb{R}^n$ is the state vector, $\mathbf{u}(k) \in \mathbb{R}^m$ is the input, $\mathbf{y}(k) \in \mathbb{R}^p$ is the measured output, $\mathbf{F} \in \mathbb{R}^{n \times n}$, $\mathbf{G} \in \mathbb{R}^{n \times m}$, and $\mathbf{H} \in \mathbb{R}^{p \times n}$. The objective is to obtain an estimate $\hat{\mathbf{x}}(k)$ such that the estimation error $\mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k)$ converges to zero as $k \rightarrow \infty$. To do that, there are two possible approaches: *Open-Loop Observer* and *Closed-Loop Observer*.

3.1.1 Open-Loop Observer

This represents the "naive" approach, where a "copy" of the system dynamics is used to simulate the state evolution in open-loop.

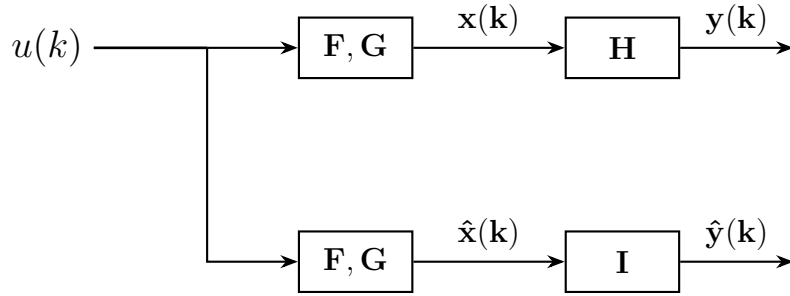


Figure 3.1: Block diagram of an open-loop state estimator.

As can be seen in Fig. 3.1, the input u is fed in parallel to both the actual system and the numerical model $(\mathbf{F}, \mathbf{G}, \mathbf{H})$ to reconstruct the state $\hat{\mathbf{x}}$ without any feedback correction.

$$\hat{\mathbf{x}}(k+1) = \mathbf{F}\hat{\mathbf{x}}(k) + \mathbf{G}\mathbf{u}(k) \quad (3.2)$$

$$\hat{\mathbf{y}}(k) = \mathbf{H}\hat{\mathbf{x}}(k) \quad (3.3)$$

This open-loop estimator, however, presents some critical flaws. In fact, the development of the estimation error can be described as follows:

$$\mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k) \quad (3.4)$$

$$\mathbf{e}(k+1) = \mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1) \quad (3.5)$$

$$= \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{u}(k) - \mathbf{F}\hat{\mathbf{x}}(k) - \mathbf{G}\mathbf{u}(k) \quad (3.6)$$

$$= \mathbf{F}\mathbf{x}(k) - \mathbf{F}\hat{\mathbf{x}}(k) \quad (3.7)$$

$$= \mathbf{F}\mathbf{e}(k) \quad (3.8)$$

So, if there is an initial condition mismatch between the estimator and the real system, and the matrix F is unstable:

$$\mathbf{e}(k) \xrightarrow{k \rightarrow \infty} \infty \quad (3.9)$$

The same goes for the presence of modeling errors and unmeasured disturbances.

3.1.2 Closed-Loop Estimator

To overcome the issues of the *Open-Loop Observer*, feedback can be added to the estimator to "correct" the prediction using the real measured output $\mathbf{y}(t)$. This configuration is called *Closed-Loop Observer*, also known as asymptotic to the state or Luenberger Observer, and it can be described as shown in Fig. 3.2 and in the following equations.

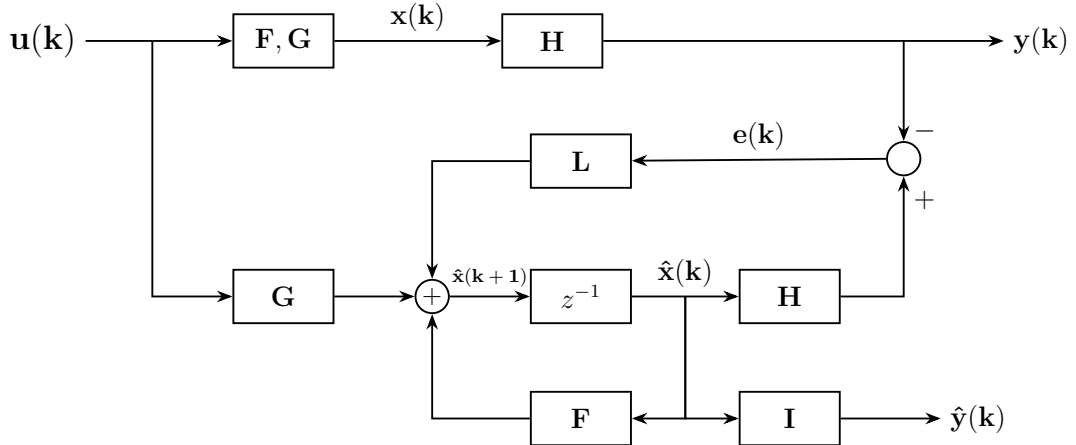


Figure 3.2: Block diagram of a closed-loop state observer

$$\hat{\mathbf{x}}(k+1) = \mathbf{F}\hat{\mathbf{x}}(k) + \mathbf{G}\mathbf{u}(k) + \mathbf{L}(\mathbf{H}\hat{\mathbf{x}}(k) - \mathbf{y}(k)) \quad (3.10)$$

$$\hat{\mathbf{y}}(k) = \hat{\mathbf{x}}(k) \quad (3.11)$$

$$\text{where } \mathbf{L} \in \mathbb{R}^{n \times p} \text{ is the observer gain} \quad (3.12)$$

The dynamics of the estimation error $\mathbf{e}(k)$ can then be derived as follows:

$$\mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k) \quad (3.13)$$

$$\mathbf{e}(k+1) = \mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1) \quad (3.14)$$

$$= \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{u}(k) - \mathbf{F}\hat{\mathbf{x}}(k) - \mathbf{G}\mathbf{u}(k) - \mathbf{L}(\mathbf{H}\hat{\mathbf{x}}(k) - \mathbf{H}\mathbf{x}(k)) \quad (3.15)$$

$$= \mathbf{F}\mathbf{x}(k) - \mathbf{F}\hat{\mathbf{x}}(k) + \mathbf{L}\mathbf{H}\mathbf{x}(k) - \mathbf{L}\mathbf{H}\hat{\mathbf{x}}(k) \quad (3.16)$$

$$= (\mathbf{F} + \mathbf{L}\mathbf{H})(\mathbf{x}(k) - \hat{\mathbf{x}}(k)) \quad (3.17)$$

$$= (\mathbf{F} + \mathbf{L}\mathbf{H})\mathbf{e}(k) \quad (3.18)$$

So, if the matrix $(\mathbf{F} - \mathbf{L}\mathbf{H})$ is Hurwitz (all eigenvalues have absolute values smaller than 1), the estimation error will converge to zero.

$$\mathbf{e}(k) \xrightarrow{k \rightarrow \infty} 0 \quad (3.19)$$

Then the speed of convergence depends on the placement of these eigenvalues, which is determined by the choice of the gain $\mathbf{L}$.

3.1.3 Observability and Detectability

The ability to reconstruct the state from the output depends on the property of *Observability*. A system is said to be observable if any initial state $\mathbf{x}(0)$ can be uniquely determined from the knowledge of the output $\mathbf{y}(t)$ and input $\mathbf{u}(t)$ over a finite time interval. Mathematically, this is true if the observability matrix $\mathcal{O}$ has full rank:

$$\mathcal{O} = \begin{bmatrix} \mathbf{H} \\ \mathbf{H}\mathbf{F} \\ \vdots \\ \mathbf{H}\mathbf{F}^{n-1} \end{bmatrix}, \quad \text{rank}(\mathcal{O}) = n \quad (3.20)$$

If a system is not fully observable, it may still be *Detectable* if all unobservable modes are naturally stable. This means that it still exists a closed loop state observer with an estimation error that asymptotically converges to zero.

3.1.4 The Separation Principle

The *Separation Principle* is a fundamental result in control theory that states that for a linear system, the design of the state observer and the design of the state-feedback controller (such as the MPC in this case) can be carried out independently. Considering the regulated system shown in Fig. 3.3, where "regulator" is intended as the combination of a state observer and a controller, identified with the matrix $\mathbf{K}$, the *Separation Principle* can be demonstrated as follows:

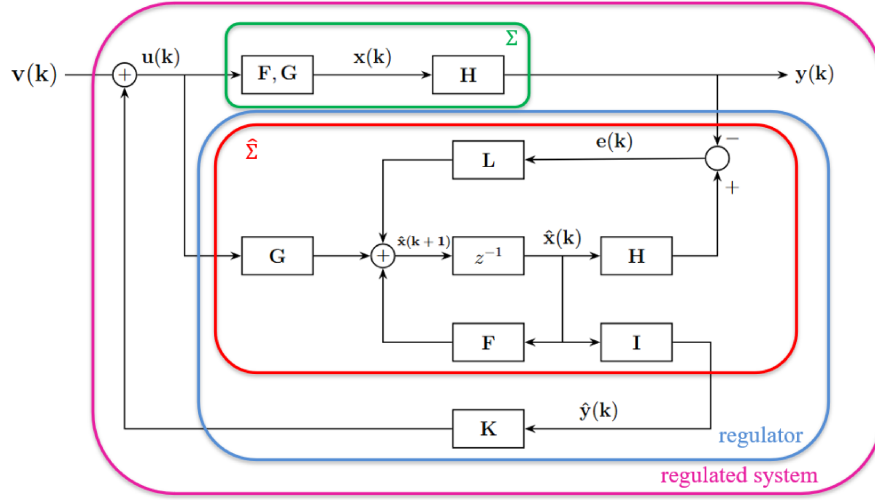


Figure 3.3: Block diagram of a regulated system

- **System Σ :**

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{u}(k) \quad (3.21)$$

$$\mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) \quad (3.22)$$

- **Controller:**

$$\mathbf{u}(k) = \mathbf{v}(k) + \mathbf{K}\hat{\mathbf{x}}(k) \quad (3.23)$$

- **State Observer:**

$$\hat{\mathbf{x}}(k+1) = \mathbf{F}\hat{\mathbf{x}}(k) + \mathbf{G}\mathbf{u}(k) - \mathbf{L}(\mathbf{y}(k) - \mathbf{H}\hat{\mathbf{x}}(k)) \quad (3.24)$$

Substituting the expressions:

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{v}(k) + \mathbf{G}\mathbf{K}\hat{\mathbf{x}}(k) \quad (3.25)$$

$$\hat{\mathbf{x}}(k+1) = \mathbf{F}\hat{\mathbf{x}}(k) + \mathbf{G}\mathbf{v}(k) + \mathbf{G}\mathbf{K}\hat{\mathbf{x}}(k) - \mathbf{L}(\mathbf{H}\mathbf{x}(k) - \mathbf{H}\hat{\mathbf{x}}(k)) \quad (3.26)$$

Then the system in augmented state-space form is:

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \hat{\mathbf{x}}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{F} & \mathbf{GK} \\ -\mathbf{LH} & \mathbf{F} + \mathbf{GK} + \mathbf{LH} \end{bmatrix}}_{\mathbf{F}_{reg} \in R^{2n \times 2n}} \begin{bmatrix} \mathbf{x}(k) \\ \hat{\mathbf{x}}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{G} \\ \mathbf{G} \end{bmatrix}}_{\mathbf{G}_{reg} \in R^{2n \times m}} \mathbf{v}(k) \quad (3.27)$$

$$\mathbf{y}(k) = \underbrace{\begin{bmatrix} \mathbf{H} & \mathbf{0} \end{bmatrix}}_{\mathbf{H}_{reg} \in R^{p \times 2n}} \begin{bmatrix} \mathbf{x}(k) \\ \hat{\mathbf{x}}(k) \end{bmatrix} \quad (3.28)$$

By applying the following change of basis:

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix} \in R^{2n \times 2n}, \quad \mathbf{T}^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix} \quad (3.29)$$

New state vector $\mathbf{z}(k)$:

$$\mathbf{z}(k) = \mathbf{T}^{-1} \mathbf{x}_{reg}(k) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \hat{\mathbf{x}}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}(k) - \hat{\mathbf{x}}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{e}(k) \end{bmatrix} \quad (3.30)$$

Matrix transformation:

$$\mathbf{F}'_{reg} = \mathbf{T}^{-1} \mathbf{F}_{reg} \mathbf{T} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{F} & \mathbf{GK} \\ -\mathbf{LH} & \mathbf{F} + \mathbf{GK} + \mathbf{LH} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix} \quad (3.31)$$

$$= \begin{bmatrix} \mathbf{F} & \mathbf{GK} \\ \mathbf{F} + \mathbf{LH} & -\mathbf{F} - \mathbf{LH} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{F} + \mathbf{GK} & -\mathbf{GK} \\ \mathbf{0} & \mathbf{F} + \mathbf{LH} \end{bmatrix} \quad (3.32)$$

$$\mathbf{G}'_{reg} = \mathbf{T}^{-1} \mathbf{G}_{reg} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{G} \\ \mathbf{G} \end{bmatrix} = \begin{bmatrix} \mathbf{G} \\ \mathbf{0} \end{bmatrix} \quad (3.33)$$

$$\mathbf{H}'_{reg} = \mathbf{H}_{reg} \mathbf{T} = \begin{bmatrix} \mathbf{H} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{H} & \mathbf{0} \end{bmatrix} \quad (3.34)$$

Final transformed system:

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{e}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{F} + \mathbf{GK} & -\mathbf{GK} \\ \mathbf{0} & \mathbf{F} + \mathbf{LH} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{e}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{G} \\ \mathbf{0} \end{bmatrix} \mathbf{v}(k) \quad (3.35)$$

$$\mathbf{y}(k) = \begin{bmatrix} \mathbf{H} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{e}(k) \end{bmatrix} \quad (3.36)$$

As it can be seen in (3.35), the eigenvalues of the overall regulated system are simply the union of the eigenvalues of $(\mathbf{F} + \mathbf{G}\mathbf{K})$ and $(\mathbf{F} + \mathbf{L}\mathbf{H})$. This confirms that the synthesis of the controller and the state observer can be performed independently, as the design of the gain matrix $\mathbf{K}$ does not affect the stability or the poles of the observer, and vice versa.

3.2 Unaccessible Input Observer - UIO

As stated before, the Kalman Filter in this project is configured as an *Unaccessible Input Observer - UIO*. This is a particular configuration of a state observer that permits reconstructing possible unmeasured interactions acting on the system, such as for the interaction force F_{env} in this case.

The core idea of an *UIO* is to treat the unknown disturbance not as a simple error, but as an additional state variable to be estimated. This is achieved by augmenting the state-space model with a disturbance model.

To estimate the unknown force, it is usually assumed that the disturbance $d(k)$ is a matched disturbance¹ that follows a zero-order Markov process (also known as a random walk model). This is a simplification, and it implies that the disturbance is assumed to be constant between sampling instants:

$$d(k+1) = d(k) \quad (3.37)$$

But since the observer is designed as a closed-loop estimator and, if designed correctly, it exhibits asymptotic behavior, the dynamics of the estimated disturbance $\hat{\mathbf{d}}(k)$ is as shown in Fig. 3.4.

¹matched disturbance: a disturbance that enters the system through the same channel as the control input

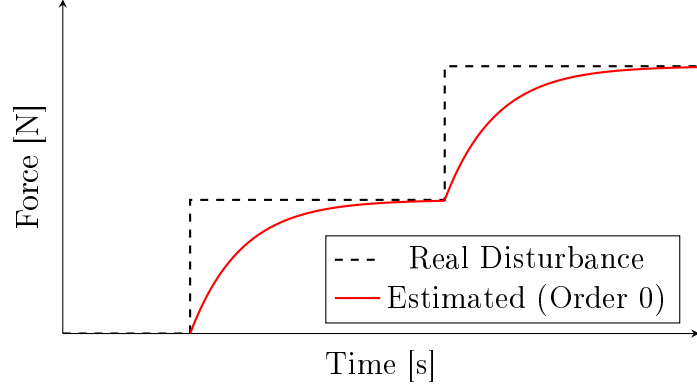


Figure 3.4: Step response of a zero-order disturbance observer.

So, assuming the system is detectable and the gain matrix $\mathbf{L}$ is properly tuned, as the number of discrete time steps k increases, the estimator effectively "closes" the gap between the true and estimated disturbance

By integrating the disturbance model into the nominal system dynamics, the augmented state vector $\mathbf{x}_a(k) \in R^{n+1}$ is obtained:

$$\mathbf{x}_a(k) = \begin{bmatrix} \mathbf{x}(k) & d(k) \end{bmatrix} \quad (3.38)$$

The augmented system is then represented as:

$$\begin{aligned} \begin{bmatrix} \mathbf{x}(k+1) \\ d(k+1) \end{bmatrix} &= \underbrace{\begin{bmatrix} \mathbf{F} & \mathbf{G}_d \\ \mathbf{0} & 1 \end{bmatrix}}_{\mathbf{F}_a} \begin{bmatrix} \mathbf{x}(k) \\ d(k) \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{G} \\ 0 \end{bmatrix}}_{\mathbf{G}_a} \mathbf{u}(k) \\ \mathbf{y}(k) &= \underbrace{\begin{bmatrix} \mathbf{H} & 0 \end{bmatrix}}_{\mathbf{H}_a} \begin{bmatrix} \mathbf{x}(k) \\ d(k) \end{bmatrix} \end{aligned}$$

where $\mathbf{G}_d$ represents the direction through which the disturbance enters the system. In this work, the interaction force is modeled as a matched disturbance, meaning it acts directly on the same input channel as the motor force ($\mathbf{G}_d = \mathbf{G}$).

However, since the primary objective of this project is to leverage the *UIO* for a sensorless estimation of the interaction force F_{env} , a first-order disturbance model ($\ddot{\mathbf{d}} = 0$) was adopted. This choice ensures a faster estimation and more accurately captures the ramp-like behavior of the force during contact. Indeed, a zero-order model would introduce a structural phase lag when dealing with time-varying disturbances.

The augmented state-space matrices for the first-order UIO are formulated as follows:

$$\begin{aligned} \begin{bmatrix} \mathbf{x}(k+1) \\ d(k+1) \\ \dot{d}(k+1) \end{bmatrix} &= \underbrace{\begin{bmatrix} \mathbf{F} & \mathbf{G}_d & \mathbf{0} \\ \mathbf{0} & 1 & T_s \\ \mathbf{0} & 0 & 1 \end{bmatrix}}_{\mathbf{F}_a} \begin{bmatrix} \mathbf{x}(k) \\ d(k) \\ \dot{d}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{G} \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{G}_a} \mathbf{u}(k) \\ \mathbf{y}(k) &= \underbrace{\begin{bmatrix} \mathbf{H} & 0 & 0 \end{bmatrix}}_{\mathbf{H}_a} \begin{bmatrix} \mathbf{x}(k) \\ d(k) \\ \dot{d}(k) \end{bmatrix} \end{aligned}$$

3.3 Kalman Filter

3.3.1 Overview

The *Kalman Filter (KF)* represents the stochastic evolution of the state estimation problem. While the Luenberger observer operates in a deterministic framework, the Kalman Filter is defined as an optimal recursive estimator for linear systems subject to stochastic disturbances. At its core, the *Kalman Filter* is a *Minimum Mean Square Error (MMSE)* estimator designed for linear dynamic systems subject to Gaussian noise. It addresses the fundamental Measurement Problem in engineering: physical sensors are inherently noisy, and mathematical models are imperfect representations of reality. The filter's operating principle is based on modeling two distinct sources of uncertainty:

- **Process Noise ($\mathbf{w}$):** Represents the intrinsic uncertainty of the dynamic model, due to mathematical simplifications or unmodeled external disturbances.
- **Measurement Noise ($\mathbf{v}$):** Models the inaccuracy of real sensors, such as encoder quantization noise or electromagnetic interference in feedback signals.

Which act on the system as follows:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{w}(t) \quad (3.39)$$

$$\mathbf{y}(t) = \mathbf{H}\mathbf{x}(t) + \mathbf{v}(t) \quad (3.40)$$

Specifically, $\mathbf{w}(t)$ and $\mathbf{v}(t)$ are assumed to be mutually independent, zero-mean, stationary² white noise stochastic processes³ with Gaussian distributions. The KF acts as an "intelligent mediator" between these two uncertainties by performing an optimal fusion of available information.

Bayesian Interpretation and MMSE

As stated earlier, the Kalman Filter belongs to the class of Minimum Mean Square Error (MMSE) estimators. To understand this concept, it is important to recall the Bayesian interpretation of the estimation problem. The Bayesian approach incorporates the intrinsic uncertainty of the system through probability theory.

The mathematical pillar of this discussion is the concept of Gaussian (or Normal) distribution. In this approach, the state $\mathbf{x}$ is not described as a deterministic vector, but through its *Probability Density Function*. The Gaussian distribution is fully characterized by its mean μ and covariance σ , which represent the best estimate of the state and the associated uncertainty, respectively.

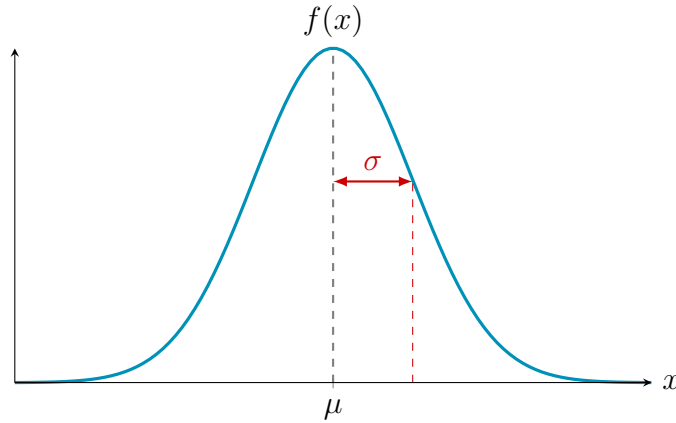


Figure 3.5: Gaussian distribution

$$\mathcal{N}(\mu, \sigma^2) \rightarrow f(\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\mathbf{x}-\mu)^2}{2\sigma^2}} \quad (3.41)$$

As stated before, the noise terms $\mathbf{w}(t)$ and $\mathbf{v}(t)$ are modeled as white Gaussian processes $\mathcal{N}(0, \Sigma_{\mathbf{w}})$ and $\mathcal{N}(0, \Sigma_{\mathbf{v}})$, where $\Sigma_{\mathbf{w}}$ and $\Sigma_{\mathbf{v}}$ are the process and measurement noise covariance matrices, respectively, and since we considered the noise processes to be stationary, these matrices are constant over time. This

²A stochastic process is said to be stationary in the strict sense if its joint probability distribution does not change when shifted in time.

³Stochastic process: a collection of stochastic variables characterized by the parameter t (time)

assumption is not merely a mathematical convenience, but it is justified by the *Central Limit Theorem*, which states that the sum of a large number of independent random variables tends to be normally distributed, regardless of their original distribution. In practical terms, this means that even if the underlying noise sources are not Gaussian, their aggregate effect can often be approximated as such, especially in complex systems with multiple noise contributions.

Unlike the Fisherian approach, where the parameter to be estimated is an unknown constant, the Bayesian approach treats the state x as a random variable.

The goal of Bayesian estimation is to compute the *posterior* distribution $p(x|y)$, which is the probability of the state given the collected measurements. This process allows for the integration of:

1. **Prior Knowledge:** Information about the system before a measurement is taken (based on the dynamic model).
2. **Likelihood:** The information provided by the new sensor measurement.

Within the Bayesian framework, it is possible to develop the idea of *Minimum Mean Square Error (MMSE) criterion*, which aims to minimize the variance of the estimation error. Formally, the MMSE estimator seeks the value $\hat{x}$ that minimizes the expected value of the squared norm of the error:

$$\hat{\mathbf{x}}_{MMSE} = \arg \min_{\mathbf{x} \in \mathcal{X}} E[\|\mathbf{x} - \hat{\mathbf{x}}\|^2] \quad (3.42)$$

$$MSE = E[\|\mathbf{x} - \hat{\mathbf{x}}\|^2] \quad (3.43)$$

A fundamental result in estimation theory states that for linear systems subject to Gaussian noise, the optimal MMSE estimation coincides with the *conditional mean* of the state given the observations:

$$\hat{x}_{MMSE} = E[x \mid y] \quad (3.44)$$

Then, considering the data modelled as follows, where the observation $\mathbf{y}$ is a linear transformation of the state $\mathbf{x}$ corrupted by additive noise:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}, \quad \mathbf{v} \sim \mathcal{N}(0, \mathbf{\Sigma}_{\mathbf{v}}), \quad x \sim \mathcal{N}(x_r, \mathbf{\Sigma}_{\mathbf{x}}) \quad (3.45)$$

Under these assumptions, the joint distribution of $\mathbf{x}$ and $\mathbf{y}$ is multivariate

Gaussian. Consequently, the posterior distribution $f(\mathbf{x}|\mathbf{y})$ is also Gaussian:

$$f(\mathbf{x}|\mathbf{y}) \sim \mathcal{N}(\mathbf{m}_{x|y}, \Sigma_{x|y}) \quad (3.46)$$

The parameters of this distribution can be computed by applying the Bayes' rule⁴ and completing the square in the exponent. The resulting posterior covariance $\Sigma_{x|y}$ and posterior mean $\mathbf{m}_{x|y}$ are given by:

$$\Sigma_{x|y} = (\mathbf{H}^\top \Sigma_v^{-1} \mathbf{H} + \Sigma_x^{-1})^{-1} \quad (3.47)$$

$$\mathbf{m}_{x|y} = \Sigma_{x|y} (\mathbf{H}^\top \Sigma_v^{-1} \mathbf{y} + \Sigma_x^{-1} \mathbf{x}_r) \quad (3.48)$$

As previously stated, for Gaussian distributions, the MMSE estimator $\hat{\mathbf{x}}_{MMSE}$ coincides with the conditional mean. Therefore, the optimal estimate is:

$$\hat{\mathbf{x}}_{MMSE} = (\mathbf{H}^\top \Sigma_v^{-1} \mathbf{H} + \Sigma_x^{-1})^{-1} (\mathbf{H}^\top \Sigma_v^{-1} \mathbf{y} + \Sigma_x^{-1} \mathbf{x}_r) \quad (3.49)$$

To evaluate the impact of the measurement on the estimation quality, we analyze the difference between the prior covariance Σ_x and the posterior covariance $\Sigma_{x|y}$. By applying the *Matrix Inversion Lemma*, we obtain:

$$\Sigma_x - \Sigma_{x|y} = \Sigma_x \mathbf{H}^\top (\Sigma_v + \mathbf{H} \Sigma_x \mathbf{H}^\top)^{-1} \mathbf{H} \Sigma_x \quad (3.50)$$

Since the resulting matrix is positive semi-definite (sdp), it follows that $\Sigma_{x|y} \leq \Sigma_x$. This analytically proves that, on average, the measurement process always reduces the uncertainty of the state estimate.

By rearranging the terms, it is possible to express the MMSE estimator in a form that highlights the correction applied to the prior estimate $\mathbf{x}_r$ based on the innovation (the difference between the actual measurement and the expected measurement):

$$\Sigma_{x|y}^{-1} \hat{\mathbf{x}}_{MMSE} = \mathbf{H}^\top \Sigma_v^{-1} \mathbf{y} + \Sigma_x^{-1} \mathbf{x}_r \quad (3.51)$$

$$= \mathbf{H}^\top \Sigma_v^{-1} \mathbf{y} + \Sigma_x^{-1} \mathbf{x}_r + \mathbf{H}^\top \Sigma_v \mathbf{H} \mathbf{x}_r - \mathbf{H}^\top \Sigma_v \mathbf{H} \mathbf{x}_r \quad (3.52)$$

$$\Rightarrow \hat{\mathbf{x}}_{MMSE} = \mathbf{x}_r + \Sigma_{x|y} \mathbf{H}^\top \Sigma_v^{-1} \underbrace{(\mathbf{y} - \mathbf{H} \mathbf{x}_r)}_{\text{innovation}} \quad (3.53)$$

⁴Bayes Theorem: $f(\mathbf{x}|\mathbf{y}) = \frac{f(\mathbf{y}|\mathbf{x})f(\mathbf{x})}{f(\mathbf{y})}$ where $f(\mathbf{y}|\mathbf{x})$ is the likelihood and $f(\mathbf{x})$ is the prior

⁴Matrix Inversion Lemma: $(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$

The static Bayesian framework can be extended to a sequential estimation problem, where measurements are collected over time. Let $\hat{\mathbf{x}}(t)$ and $\Sigma_{\hat{\mathbf{x}}}(t)$ be the estimate and its variance at time t . When a new measurement $\mathbf{y}_{t+1}$ becomes available, the update equations follow a recursive structure:

$$\Sigma_{\hat{\mathbf{x}}}(k+1) = (\mathbf{H}^\top \Sigma_v^{-1} \mathbf{H} + \Sigma_{\hat{\mathbf{x}}}(k)^{-1})^{-1} \quad (3.54)$$

$$\hat{\mathbf{x}}(k+1) = \hat{\mathbf{x}}(k) + \Sigma_{\hat{\mathbf{x}}}(k+1) \mathbf{H}^\top \Sigma_v^{-1} (\mathbf{y}(k+1) - \mathbf{H} \hat{\mathbf{x}}(k)) \quad (3.55)$$

This recursive formulation is computationally efficient as it avoids reprocessing all past data, requiring only the most recent estimate and the current measurement.

Kalman Filter Formulation

Now it is possible to reconsider the dynamic system with process and measurement noise, as described earlier:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{w}(t) \quad (3.56)$$

$$\mathbf{y}(t) = \mathbf{H}\mathbf{x}(t) + \mathbf{v}(t) \quad (3.57)$$

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (3.58)$$

Where:

- $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{y} \in \mathbb{R}^p$
- $\mathbf{w}(t) \sim \mathcal{N}(0, \Sigma_{\mathbf{w}})$ and $\mathbf{v}(t) \sim \mathcal{N}(0, \Sigma_{\mathbf{v}})$ are the process and measurement noise, respectively, and are stationary, white and independent stochastic processes.
- $\mathbf{x}(0) \sim \mathcal{N}(\mathbf{x}_0, \Sigma_0)$ is the initial state, which is also assumed to be Gaussian.

It is possible to derive the Kalman Filter equations redefining the state estimation problem as a recursive application of the MMSE estimator:

- State estimation based on $\mathbf{y}^T = [\mathbf{y}(0), \mathbf{y}(1), \dots, \mathbf{y}(k)]$.

$$\tilde{\mathbf{x}}(k+1|k) = \hat{\mathbf{x}}(k+1|k) - \mathbf{x}(k+1) \Leftarrow \text{estimation error at } k+1 \quad (3.59)$$

$$\mathbf{P}(k+1|k) = E[\tilde{\mathbf{x}}(k+1|k)\tilde{\mathbf{x}}(k+1|k)^T] \Leftarrow \text{error covariance} \quad (3.60)$$

Then in the absence of measurements and based on the model, the best estimate of the state at time $k + 1$ is given by:

$$\hat{\mathbf{x}}(k + 1|k) = \mathbf{F}\hat{\mathbf{x}}(k|k) + \mathbf{G}\mathbf{u}(k) \Leftarrow \text{prediction update} \quad (3.61)$$

Where $\mathbf{F}$ and $\mathbf{G}$ are the discretized state transition and input matrices obtained as shown in chapter 2.3.

$$\tilde{\mathbf{x}}(k + 1|k) = \mathbf{F}\hat{\mathbf{x}}(k|k) - \mathbf{F}\mathbf{x}(k) - \mathbf{w}(k) = \mathbf{F}\tilde{\mathbf{x}}(k|k) - \mathbf{w}(k) \quad (3.62)$$

$$\mathbf{P}(k + 1|k) = E[(\mathbf{F}\tilde{\mathbf{x}}(k|k) - \mathbf{w}(k))(\mathbf{F}\tilde{\mathbf{x}}(k|k) - \mathbf{w}(k))^T] \quad (3.63)$$

$$= E[\mathbf{F}\tilde{\mathbf{x}}(k|k)\tilde{\mathbf{x}}(k|k)^T\mathbf{F}^T - \mathbf{F}\tilde{\mathbf{x}}(k|k)\mathbf{w}^T(k) - \mathbf{w}(k)\tilde{\mathbf{x}}^T(k|k)\mathbf{F}^T + \mathbf{w}(k)\mathbf{w}^T(k)] \quad (3.64)$$

$$= E[\mathbf{F}\tilde{\mathbf{x}}(k|k)\tilde{\mathbf{x}}(k|k)^T\mathbf{F}^T + \mathbf{w}(k)\mathbf{w}^T(k)] \quad (3.65)$$

$$\mathbf{P}(t + 1|t) = \mathbf{F}\mathbf{P}(t|t)\mathbf{F}^T + \Sigma_{\mathbf{w}} \Leftarrow \text{prediction covariance update} \quad (3.66)$$

- State estimation given $\mathbf{y}_{k+1} = \mathbf{y}(k + 1)$

$$\tilde{\mathbf{x}}(k + 1|k + 1) = \hat{\mathbf{x}}(k + 1|k + 1) - \mathbf{x}(k + 1) \Leftarrow \text{estimation error at } k + 1 \quad (3.67)$$

$$\mathbf{P}(k + 1|k + 1) = E[\tilde{\mathbf{x}}(k + 1|k + 1)\tilde{\mathbf{x}}(k + 1|k + 1)^T] \Leftarrow \text{error covariance} \quad (3.68)$$

Then, using the new measurement and the MMSE recursive estimation:

$$\mathbf{P}(k + 1|k + 1) = (\mathbf{H}^T \Sigma_{\mathbf{v}}^{-1} \mathbf{H} + \mathbf{P}(k + 1|k)^{-1})^{-1} \quad (3.69)$$

$$\begin{aligned} \hat{\mathbf{x}}(k + 1|k + 1) &= \hat{\mathbf{x}}(k + 1|k) + \mathbf{P}(k + 1|k + 1)\mathbf{H}^T \Sigma_{\mathbf{v}}^{-1}(\mathbf{y}(k + 1) \\ &\quad - \mathbf{H}\hat{\mathbf{x}}(k + 1|k)) \end{aligned} \quad (3.70)$$

Finally it is possible to define the Kalman Filter formulation as follows:

Prior knowledge:

$$\mathbf{P}(k+1|k) = \mathbf{F}\mathbf{P}(k|k)\mathbf{F}^T + \Sigma_{\mathbf{w}} \quad (3.71)$$

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{F}\hat{\mathbf{x}}(k|k) + \mathbf{G}\mathbf{u}(k) \quad (3.72)$$

Posterior knowledge:

$$\mathbf{P}(k+1|k+1) = (\mathbf{H}^T \Sigma_{\mathbf{v}}^{-1} \mathbf{H} + \mathbf{P}(k+1|k)^{-1})^{-1} \quad (3.73)$$

$$\begin{aligned} \hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) \\ &+ \mathbf{P}(k+1|k+1)\mathbf{H}^T \Sigma_{\mathbf{v}}^{-1}(\mathbf{y}(k+1) - \mathbf{H}\hat{\mathbf{x}}(k+1|k)) \end{aligned} \quad (3.74)$$

Kalman Filter interpretation as a closed loop state estimator

From the Kalman Filter equations derived previously from a purely probabilistic perspective, it is possible to appreciate its *predictor-corrector* architecture in a closed-loop configuration. To highlight this, it is necessary to first apply the *Matrix Inversion Lemma* to the posterior covariance equation. This allows to rewrite $\mathbf{P}(k+1|k+1)$ without requiring the direct inversion of the prior covariance $\mathbf{P}(k+1|k)^{-1}$:

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{P}(k+1|k)\mathbf{H}^T (\mathbf{H}\mathbf{P}(k+1|k)\mathbf{H}^T + \Sigma_{\mathbf{v}})^{-1} \mathbf{H}\mathbf{P}(k+1|k) \quad (3.75)$$

By defining the *Innovation Covariance* matrix as $\mathbf{S}(k+1) = \mathbf{H}\mathbf{P}(k+1|k)\mathbf{H}^T + \Sigma_{\mathbf{v}}$, the expression simplifies to:

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{P}(k+1|k)\mathbf{H}^T \mathbf{S}(k+1)^{-1} \mathbf{H}\mathbf{P}(k+1|k) \quad (3.76)$$

This algebraic manipulation is essential to define the Kalman Gain. At first it is possible to define the intermediate gain matrix $\mathbf{L}'(k+1)$ as follows:

$$\mathbf{L}'(k+1) = \mathbf{P}(k+1|k+1)\mathbf{H}^T \Sigma_{\mathbf{v}}^{-1} \quad (3.77)$$

By substituting the expression for $\mathbf{P}(k+1|k+1)$ obtained via the Matrix Inversion Lemma into the equation above, $\mathbf{L}'(k+1)$ can be expressed as:

$$\begin{aligned}
\mathbf{L}'(k+1) &= [\mathbf{P}(k+1|k) - \mathbf{P}(k+1|k)\mathbf{H}^\top \mathbf{S}(k+1)^{-1} \mathbf{H} \mathbf{P}(k+1|k)] \mathbf{H}^\top \Sigma_{\mathbf{v}}^{-1} \\
&= \mathbf{P}(k+1|k)\mathbf{H}^\top \Sigma_{\mathbf{v}}^{-1} - \mathbf{P}(k+1|k)\mathbf{H}^\top \mathbf{S}(k+1)^{-1} \mathbf{H} \mathbf{P}(k+1|k)\mathbf{H}^\top \Sigma_{\mathbf{v}}^{-1} \\
&= \mathbf{P}(k+1|k)\mathbf{H}^\top \mathbf{S}(k+1)^{-1} (\mathbf{S}(k+1)\Sigma_{\mathbf{v}}^{-1} - \mathbf{H} \mathbf{P}(k+1|k)\mathbf{H}^\top \Sigma_{\mathbf{v}}^{-1})
\end{aligned} \tag{3.78}$$

To simplify the term inside the parentheses, it is possible to recall the definition of the innovation covariance $\mathbf{S}(k+1) = \mathbf{H} \mathbf{P}(k+1|k)\mathbf{H}^\top + \Sigma_{\mathbf{v}}$.

Substituting this into the parenthesis yields:

$$\begin{aligned}
\mathbf{S}(k+1)\Sigma_{\mathbf{v}}^{-1} - \mathbf{H} \mathbf{P}(k+1|k)\mathbf{H}^\top \Sigma_{\mathbf{v}}^{-1} &= (\Sigma_{\mathbf{v}} + \mathbf{H} \mathbf{P}(k+1|k)\mathbf{H}^\top) \Sigma_{\mathbf{v}}^{-1} - \mathbf{H} \mathbf{P}(k+1|k)\mathbf{H}^\top \Sigma_{\mathbf{v}}^{-1} \\
&= \Sigma_{\mathbf{v}} \Sigma_{\mathbf{v}}^{-1} + \mathbf{H} \mathbf{P}(k+1|k)\mathbf{H}^\top \Sigma_{\mathbf{v}}^{-1} - \mathbf{H} \mathbf{P}(k+1|k)\mathbf{H}^\top \Sigma_{\mathbf{v}}^{-1} \\
&= \mathbf{I}
\end{aligned} \tag{3.79}$$

Therefore, $L'(k+1)$ can be expressed in this form:

$$\mathbf{L}'(k+1) = \mathbf{P}(k+1|k)\mathbf{H}^\top \mathbf{S}(k+1)^{-1} \tag{3.80}$$

$$= \mathbf{P}(k+1|k)\mathbf{H}^\top (\mathbf{H} \mathbf{P}(k+1|k)\mathbf{H}^\top + \Sigma_{\mathbf{v}})^{-1} \tag{3.81}$$

$$\tag{3.82}$$

By substituting the correction gain $\mathbf{L}'(k+1)$ into the state update equation, the *a posteriori* state estimate can be expressed as:

$$\begin{aligned}
\hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) + \mathbf{P}(k+1|k+1)\mathbf{H}^\top \Sigma_{\mathbf{v}}^{-1} (\mathbf{y}(k+1) - \mathbf{H}\hat{\mathbf{x}}(k+1|k)) \\
&= \hat{\mathbf{x}}(k+1|k) + \mathbf{L}'(k+1) (\mathbf{y}(k+1) - \mathbf{H}\hat{\mathbf{x}}(k+1|k))
\end{aligned} \tag{3.83}$$

Then to highlight the closed-loop nature of the estimator, it is useful to consider the prediction for the next time step $k+1$:

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{F}\hat{\mathbf{x}}(k|k) + \mathbf{G}\mathbf{u}(k) \tag{3.84}$$

and substituting the expression for the filtered estimate at time k , $\hat{\mathbf{x}}(k|k)$:

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{L}'(k) (\mathbf{y}(k) - \mathbf{H}\hat{\mathbf{x}}(k|k-1)) \tag{3.85}$$

$\hat{\mathbf{x}}(k+1|k)$ becomes:

$$\begin{aligned}\hat{\mathbf{x}}(k+1|k) &= \mathbf{F} [\hat{\mathbf{x}}(k|k-1) + \mathbf{L}'(k) (\mathbf{y}(k) - \mathbf{H}\hat{\mathbf{x}}(k|k-1))] + \mathbf{G}\mathbf{u}(k) \\ &= \mathbf{F}\hat{\mathbf{x}}(k|k-1) + \mathbf{G}\mathbf{u}(k) + \mathbf{F}\mathbf{L}'(k) (\mathbf{y}(k) - \mathbf{H}\hat{\mathbf{x}}(k|k-1))\end{aligned}\quad (3.86)$$

By defining a new time-varying gain $\mathbf{L}(k) = \mathbf{F}\mathbf{L}'(k)$, the state predictor takes the form:

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{F}\hat{\mathbf{x}}(k|k-1) + \mathbf{G}\mathbf{u}(k) + \mathbf{L}(k) (\mathbf{y}(k) - \mathbf{H}\hat{\mathbf{x}}(k|k-1)) \quad (3.87)$$

where the Kalman Predictor Gain $\mathbf{L}(k)$ is given by:

$$\mathbf{L}(k) = \mathbf{F}\mathbf{P}(k|k-1)\mathbf{H}^\top (\boldsymbol{\Sigma}_v + \mathbf{H}\mathbf{P}(k|k-1)\mathbf{H}^\top)^{-1} \quad (3.88)$$

In a similar way it is possible to obtain a recursive expression for the evolution of the prediction error covariance $\mathbf{P}(k+1|k)$. This allows to describe how the uncertainty of the estimate propagates through the system dynamics and the measurement updates in a single step. Starting from the prediction covariance equation at time $k+1$:

$$\mathbf{P}(k+1|k) = \mathbf{F}\mathbf{P}(k|k)\mathbf{F}^\top + \boldsymbol{\Sigma}_w \quad (3.89)$$

by substituting the expression for the *a posteriori* covariance at time k , $\mathbf{P}(k|k)$, using the form previously derived through the Matrix Inversion Lemma:

$$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{P}(k|k-1)\mathbf{H}^\top \mathbf{S}(k)^{-1} \mathbf{H}\mathbf{P}(k|k-1) \quad (3.90)$$

Then by substituting $\mathbf{S}(k)$ into the prediction equation, the evolution of the covariance can be expressed as:

$$\begin{aligned}\mathbf{P}(k+1|k) &= \mathbf{F} [\mathbf{P}(k|k-1) - \mathbf{P}(k|k-1)\mathbf{H}^\top \mathbf{S}(k)^{-1} \mathbf{H}\mathbf{P}(k|k-1)] \mathbf{F}^\top + \boldsymbol{\Sigma}_w \\ &= \mathbf{F}\mathbf{P}(k|k-1)\mathbf{F}^\top - \mathbf{F}\mathbf{P}(k|k-1)\mathbf{H}^\top \mathbf{S}(k)^{-1} \mathbf{H}\mathbf{P}(k|k-1)\mathbf{F}^\top + \boldsymbol{\Sigma}_w\end{aligned}\quad (3.91)$$

This result corresponds to a Discrete-time Riccati Difference Equation (DRDE). It governs the evolution of the estimation uncertainty over time.

From the equations (3.91) and (3.88) it is possible to appreciate several prop-

erties of the Kalman Filter:

- **Data Independence and Offline Computation:** A crucial observation regarding the DRDE is that the evolution of the prediction covariance $\mathbf{P}(k+1|k)$ does not depend on the actual measurement vector $\mathbf{y}(k)$. The update depends exclusively on the system matrices $\mathbf{F}$ and $\mathbf{H}$, and the noise covariance matrices $\Sigma_{\mathbf{w}}$ and $\Sigma_{\mathbf{v}}$.
- **Independence of the Kalman Gain:** Since $\mathbf{P}(k+1|k)$ is independent of the measurements, the Kalman Predictor Gain $\mathbf{L}(k)$ is also independent of the data $\mathbf{y}(k)$, as it depends only on the prediction covariance and system parameters.
- **Predictability:** This independence implies that, provided the system parameters and if the noise characteristics are known a priori, the entire sequence of error covariances and optimal gains can be pre-computed offline before the filter is deployed on real-time data.
- **Uncertainty Balance:** The DRDE analytically describes the balance between two competing effects on estimation certainty. The term $\mathbf{F}\mathbf{P}(k|k-1)\mathbf{F}^\top + \Sigma_{\mathbf{w}}$ represents the increase in uncertainty caused by the system dynamics and the injection of process noise. In contrast, the negative term $-\mathbf{F}\mathbf{P}(k|k-1)\mathbf{H}^\top \mathbf{S}(k)^{-1} \mathbf{H}\mathbf{P}(k|k-1)\mathbf{F}^\top$ represents the information gain provided by the new measurement, which serves to reduce the estimation error variance.
- **Steady-State Convergence:** If the system is observable and stabilizable, the prediction covariance $\mathbf{P}(k+1|k)$ converges to a constant positive semi-definite matrix $\mathbf{P}_\infty$ as $k \rightarrow \infty$. In this scenario, the DRDE becomes the Discrete Algebraic Riccati Equation (DARE), and the gain $\mathbf{L}(k)$ converges to a steady-state value $\mathbf{L}_\infty$. This leads to the *Steady-State Kalman Filter*.

In the application described in this thesis, the system parameters are constant and the filter operates over a long time horizon. So, under the assumptions of detectability and stabilizability of the system, the conditions for the application of the steady-state Kalman Filter are met, and the prediction error covariance matrix $\mathbf{P}(k+1|k)$ converges to a unique positive semi-definite steady-state matrix $\mathbf{P}$ as $k \rightarrow \infty$. Then, in this regime, the Riccati Difference Equation (RDE) simplifies into the Discrete Algebraic Riccati Equation (DARE):

$$\mathbf{P} = \mathbf{F}\mathbf{P}\mathbf{F}^\top - \mathbf{F}\mathbf{P}\mathbf{H}^\top (\boldsymbol{\Sigma}_v + \mathbf{H}\mathbf{P}\mathbf{H}^\top)^{-1} \mathbf{H}\mathbf{P}\mathbf{F}^\top + \boldsymbol{\Sigma}_w \quad (3.92)$$

Consequently, the time-varying gain $\mathbf{L}(k)$ also converges to a constant Steady-State Kalman Gain $\mathbf{L}$, defined as:

$$\mathbf{L} = \mathbf{F}\mathbf{P}\mathbf{H}^\top (\boldsymbol{\Sigma}_v + \mathbf{H}\mathbf{P}\mathbf{H}^\top)^{-1} \quad (3.93)$$

Substituting these constant values into the predictor structure, the final recursive equation for the state estimate becomes:

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{F}\hat{\mathbf{x}}(k|k-1) + \mathbf{G}\mathbf{u}(k) + \mathbf{L}(\mathbf{y}(k) - \mathbf{H}\hat{\mathbf{x}}(k|k-1)) \quad (3.94)$$

The nature of the Kalman Filter's gain $\mathbf{L}$ can be explained intuitively as follows: if the measurement noise covariance $\boldsymbol{\Sigma}_v$ is large, the filter places less trust in the measurements, resulting in a smaller gain $\mathbf{L}$, and thus relying more on the model prediction. Conversely, if the process noise covariance $\boldsymbol{\Sigma}_w$ is large, the filter places more trust in the measurements, leading to a larger gain $\mathbf{L}$, and thus relying more on the measurement updates to correct the state estimate.

3.3.2 Kalman Filter Environment-Agnostic Model and Tuning

Environment-Agnostic Model

As stated previously, to obtain a fast estimation of the interaction force, the Kalman Filter was implemented with an Environment-Agnostic Model, in order to consider the interaction force as an external disturbance and use the filter as an Unaccessible Input Observer with the external disturbance described with a first-order disturbance model, $\ddot{d} = 0$. The considered model can then be represented as follows:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{d} \\ \ddot{d} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{B_n}{M_n} & -\frac{1}{M_n} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{A_{kf}} \begin{bmatrix} x \\ \dot{x} \\ d \\ \dot{d} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{M_n} \\ 0 \\ 0 \end{bmatrix}}_{B_{kf}} F_{cmd} \quad (3.95)$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}}_{H_{kf}} \begin{bmatrix} x \\ \dot{x} \\ d \\ \dot{d} \end{bmatrix} \quad (3.96)$$

The observability of the system can be verified by computing the observability matrix $\mathcal{O}$:

$$\mathcal{O} = \begin{bmatrix} H_{kf} \\ H_{kf}A_{kf} \\ H_{kf}A_{kf}^2 \\ H_{kf}A_{kf}^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\frac{B_n}{M_n} & -\frac{1}{M_n} & 0 \\ 0 & \frac{B_n^2}{M_n^2} & \frac{B_n}{M_n^2} & -\frac{1}{M_n} \end{bmatrix} \quad (3.97)$$

Since the observability matrix $\mathcal{O}$ has full rank, the system is observable, and thus the Kalman Filter can be designed.

Kalman Filter Definition and Noise Characterization

As described in the Kalman Filter Overview section, to formulate the filter it is necessary to define the process noise covariance $\Sigma_{\mathbf{w}}$ and the measurement noise covariance $\Sigma_{\mathbf{v}}$, related respectively to the process noise $\mathbf{w}(t)$ and the measurement noise $\mathbf{v}(t)$.

Which act on the system as follows:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{w}(t) \quad (3.98)$$

$$\mathbf{y}(t) = \mathbf{H}\mathbf{x}(t) + \mathbf{v}(t) \quad (3.99)$$

and are assumed to be mutually independent, zero-mean, stationary white noise stochastic processes with Gaussian distributions $\mathbf{w}(t) \in \mathbb{R}^4 \sim \mathcal{N}(0, \Sigma_{\mathbf{w}})$ and $\mathbf{v}(t) \in \mathbb{R} \sim \mathcal{N}(0, \Sigma_{\mathbf{v}})$, respectively.

Now, it is possible to consider the discretized system:

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{u}(k) + \mathbf{w}_d(k) \quad (3.100)$$

$$\mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{v}(k) \quad (3.101)$$

Where $\mathbf{w}_d(k)$ is the discretized process noise and is calculated as follows:

$$\mathbf{w}_d(k) = \int_0^{T_s} e^{\mathbf{A}\tau} \mathbf{B} \mathbf{w}(\tau) d\tau \quad (3.102)$$

But what is needed is the covariance of $\mathbf{w}_d(k)$, which can be computed as:

$$\Sigma_{\mathbf{w}_d} = E[\mathbf{w}_d(k) \mathbf{w}_d(k)^T] \quad (3.103)$$

$$= E \left[\int_0^{T_s} \int_0^{T_s} e^{\mathbf{A}\tau} \mathbf{B} \mathbf{w}(\tau) \mathbf{w}(\eta)^T \mathbf{B}^T e^{\mathbf{A}^T \eta} d\tau d\eta \right] \quad (3.104)$$

$$= \int_0^{T_s} \int_0^{T_s} e^{\mathbf{A}\tau} \mathbf{B} E[\mathbf{w}(\tau) \mathbf{w}(\eta)^T] \mathbf{B}^T e^{\mathbf{A}^T \eta} d\tau d\eta \quad (3.105)$$

Since $\mathbf{w}(\tau)$ is a stochastic process, it is composed of independent stochastic variables at different time instants. Therefore:

$$E[\mathbf{w}(\tau) \mathbf{w}(\eta)^T] = \Sigma_{\mathbf{w}} \delta(\tau - \eta) \quad (3.106)$$

By substituting this into the previous equation, it becomes:

$$\Sigma_{\mathbf{w}_d} = \Sigma_{\mathbf{w}} \underbrace{\int_0^{T_s} e^{\mathbf{A}\tau} \mathbf{B} \mathbf{B}^T e^{\mathbf{A}^T \tau} d\tau}_{W_{0 \rightarrow T_s}} \quad (3.107)$$

Where $W_{0 \rightarrow T_s}$ corresponds to the system's reachability Gramian in one step, that describes how the process noise propagates through the system states in the time interval $[0, T_s]$, and $\Sigma_{\mathbf{w}_d}$ should be symmetric and at least positive semi-definite.

Now, it is possible to express the filter's matrices as follows:

$$\mathbf{A}_{kf} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \alpha & -\beta & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B}_{kf} = \begin{bmatrix} 0 \\ \beta \\ 0 \\ 0 \end{bmatrix} \quad (3.108)$$

Where $\alpha = -\frac{B_n}{M_n}$ and $\beta = \frac{1}{M_n}$ for simplicity. Then, by approximating the

matrix exponential with a first-order Taylor expansion $e^{\mathbf{A}_{kf}T_s} \approx \mathbf{I} + \mathbf{A}_{kf}T_s$, the Gramian $W_{0 \rightarrow T_s}$ can be calculated as follows:

$$e^{A_{kf}t} \approx \begin{bmatrix} 1 & t & 0 & 0 \\ 0 & 1 + \alpha t & -\beta t & 0 \\ 0 & 0 & 1 & t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.109)$$

$$e^{A_{kf}t} B_{kf} = \begin{bmatrix} 1 & t & 0 & 0 \\ 0 & 1 + \alpha t & -\beta t & 0 \\ 0 & 0 & 1 & t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \beta \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \beta t \\ \beta + \alpha \beta t \\ 0 \\ 0 \end{bmatrix} \quad (3.110)$$

$$(3.111)$$

$$(e^{A_{kf}t} B_{kf})(e^{A_{kf}t} B_{kf})^T = \begin{bmatrix} \beta t \\ \beta + \alpha \beta t \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \beta t & \beta + \alpha \beta t & 0 & 0 \end{bmatrix} \quad (3.112)$$

$$(3.113)$$

$$W_{0 \rightarrow T_s} = \int_0^{T_s} \begin{bmatrix} \beta^2 t^2 & \beta^2 t + \alpha \beta^2 t^2 & 0 & 0 \\ \beta^2 t + \alpha \beta^2 t^2 & \beta^2 + 2\alpha \beta^2 t + \alpha^2 \beta^2 t^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} dt \quad (3.114)$$

$$(3.115)$$

$$W_{0 \rightarrow T_s} = \begin{bmatrix} \beta^2 \frac{T_s^3}{3} & \beta^2 \frac{T_s^2}{2} + \alpha \beta^2 \frac{T_s^3}{3} & 0 & 0 \\ \beta^2 \frac{T_s^2}{2} + \alpha \beta^2 \frac{T_s^3}{3} & \beta^2 T_s + \alpha \beta^2 T_s^2 + \alpha^2 \beta^2 \frac{T_s^3}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.116)$$

As it can be observed from the expression (3.116), the reachability Gramian $W_{0 \rightarrow T_s}$ is not positive definite, as its last two rows and columns are zero. This

could lead to a solution of the DARE that generates a gain $\mathbf{L}$ that is not stabilizing, and therefore to an unstable filter.

To avoid this issue, it is possible to add the effect of small process noises on the last two states, which corresponds to the disturbance and its derivative.

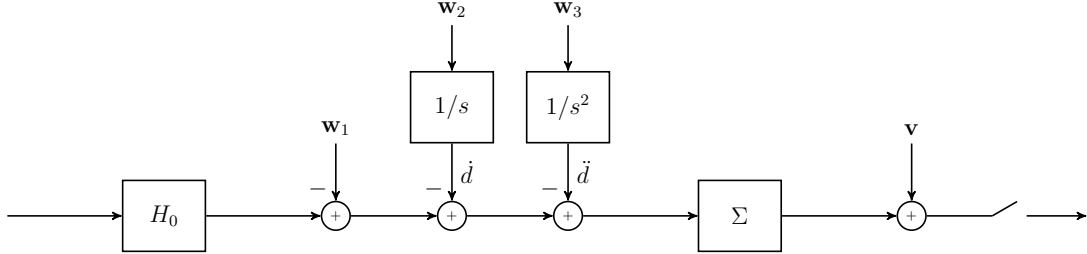


Figure 3.6: Block diagram of the system with the effect of the process noises acting on the disturbance states.

Fig. 3.6 shows the block diagram of the system with the effect of the added process noises acting on the disturbance states, where $\mathbf{w}_1 \in \mathbb{R}^2$ is the process noise acting on the system dynamics, while $\mathbf{w}_2 \in \mathbb{R}$ and $\mathbf{w}_3 \in \mathbb{R}$ are the process noises acting on the disturbance and its derivative, respectively. The measurement noise $\mathbf{v} \in \mathbb{R}$ is also shown as an input to the measurement summation node, while H_0 represents the effect of the zero-order hold on the system input.

The obtained system can be described with the following state-space representation:

$$\begin{aligned}
 \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{d} \\ \ddot{d} \end{bmatrix} &= \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{B_n}{M_n} & -\frac{1}{M_n} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{A_{kf}} \begin{bmatrix} x \\ \dot{x} \\ d \\ \dot{d} \end{bmatrix} \\
 &+ \underbrace{\begin{bmatrix} 0 \\ \frac{1}{M_n} \\ 0 \\ 0 \end{bmatrix}}_{B_{kf1}} (F_{cmd} + \mathbf{w}_1) + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{B_{kf2}} \mathbf{w}_2 + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{B_{kf3}} \mathbf{w}_3
 \end{aligned} \tag{3.117}$$

Then it is possible to calculate the reachability Gramians and covariances for the three process noises separately:

$$W_{1_0 \rightarrow T_s} = \int_0^{T_s} e^{\mathbf{A}_{kf}\tau} \mathbf{B}_{1_{kf}} \mathbf{B}_{1_{kf}}^T e^{\mathbf{A}_{kf}^T \tau} d\tau \quad (3.118)$$

$$= \begin{bmatrix} \beta^2 \frac{T_s^3}{3} & \beta^2 \frac{T_s^2}{2} + \alpha \beta^2 \frac{T_s^3}{3} & 0 & 0 \\ \beta^2 \frac{T_s^2}{2} + \alpha \beta^2 \frac{T_s^3}{3} & \beta^2 T_s + \alpha \beta^2 T_s^2 + \alpha^2 \beta^2 \frac{T_s^3}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.119)$$

$$\Sigma_{\mathbf{w}_{1_d}} = \Sigma_{\mathbf{w}_1} W_{1_0 \rightarrow T_s} \quad (3.120)$$

$$W_{2_0 \rightarrow T_s} = \int_0^{T_s} e^{\mathbf{A}_{kf}\tau} \mathbf{B}_{2_{kf}} \mathbf{B}_{2_{kf}}^T e^{\mathbf{A}_{kf}^T \tau} d\tau \quad (3.121)$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \beta \frac{T_s^3}{3} & -\beta \frac{T_s^2}{2} & 0 \\ 0 & -\beta \frac{T_s^2}{2} & T_s & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.122)$$

$$\Sigma_{\mathbf{w}_{2_d}} = \Sigma_{\mathbf{w}_2} W_{2_0 \rightarrow T_s} \quad (3.123)$$

$$W_{3_0 \rightarrow T_s} = \int_0^{T_s} e^{\mathbf{A}_{kf}\tau} \mathbf{B}_{3_{kf}} \mathbf{B}_{3_{kf}}^T e^{\mathbf{A}_{kf}^T \tau} d\tau \quad (3.124)$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{T_s^3}{3} & \frac{T_s^2}{2} \\ 0 & 0 & \frac{T_s^2}{2} & T_s \end{bmatrix} \quad (3.125)$$

$$\Sigma_{\mathbf{w}_{3_d}} = \Sigma_{\mathbf{w}_3} W_{3_0 \rightarrow T_s} \quad (3.126)$$

Then, since $\mathbf{w}_1$, $\mathbf{w}_2$, and $\mathbf{w}_3$ are independent, the total process noise covariance $\Sigma_{\mathbf{w}_d}$ is given by the sum of the three covariances, which in this case is positive definite, and therefore the DARE will have a stabilizing solution:

$$\Sigma_{\mathbf{w}_d} = \Sigma_{\mathbf{w}_{1_d}} + \Sigma_{\mathbf{w}_{2_d}} + \Sigma_{\mathbf{w}_{3_d}} \quad (3.127)$$

$$= \Sigma_{\mathbf{w}_1} W_{1_0 \rightarrow T_s} + \Sigma_{\mathbf{w}_2} W_{2_0 \rightarrow T_s} + \Sigma_{\mathbf{w}_3} W_{3_0 \rightarrow T_s} \quad (3.128)$$

Since the output is the position measurement and it is a static equation, for

the measurement noise $\mathbf{v}$ it is only required to define the variance $\Sigma_{\mathbf{v}}$, which is a scalar in this case.

Kalman Filter Tuning

The tuning of the Kalman Filter consists in the selection of the process noise covariances $\Sigma_{\mathbf{w}_{d_1}}$, $\Sigma_{\mathbf{w}_{d_2}}$, and $\Sigma_{\mathbf{w}_{d_3}}$, and the measurement noise covariance $\Sigma_{\mathbf{v}}$.

A possible approach could be to estimate the noise covariances from experimental data, by collecting samples of the process noise and measurement noise, and then calculating their sample covariance matrices. However, in practice, it is often very difficult to obtain accurate estimates of the noise covariances, especially for the process noise, which is not directly measurable.

Therefore, a common approach is to treat the noise covariances as tuning parameters, and adjust them based on the *Bartlett Test*.

The general idea behind the *Bartlett Test* is that in a well-tuned Kalman Filter, the sequence of innovations should be uncorrelated and then the innovations should be equal to a white noise process.

The test is conducted by giving a random input to the system in open-loop, and then collecting the sequence of innovations generated by the Kalman Filter. Then, the cumulative periodogram of the innovations is calculated, as shown in Fig. 3.7, and compared to the expected cumulative periodogram of a white noise process.

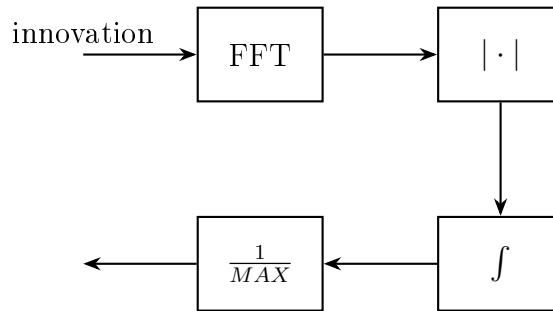


Figure 3.7: Example of cumulative periodogram of the innovations compared to the expected cumulative periodogram of a white noise process.

In Fig. 3.8 are shown three examples of cumulative periodograms of the innovations, compared to the expected cumulative periodogram of a white noise process. In the ideal case (a), the cumulative periodogram of the innovations closely follows the expected curve, indicating that the innovations are uncorrelated and have a flat power spectral density, which is consistent with a white noise process. In case (b), the cumulative periodogram shows a significant deviation from the

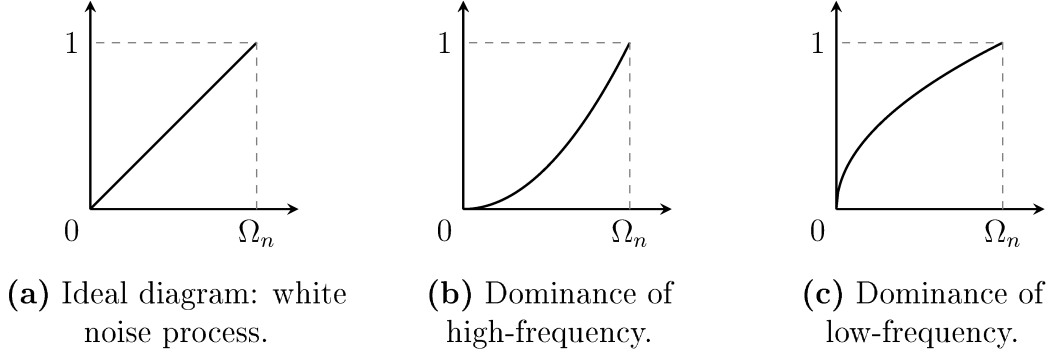


Figure 3.8: Analysis of cumulative periodograms for the Bartlett test.

expected curve at higher frequencies, indicating that the innovations are dominated by high-frequency components, which could be due to an underestimation of the measurement noise covariance Σ_v . In case (c), the cumulative periodogram shows a significant deviation from the expected curve at lower frequencies, indicating that the innovations are dominated by low-frequency components, which could be due to an underestimation of the process noise covariance Σ_{w_d} .

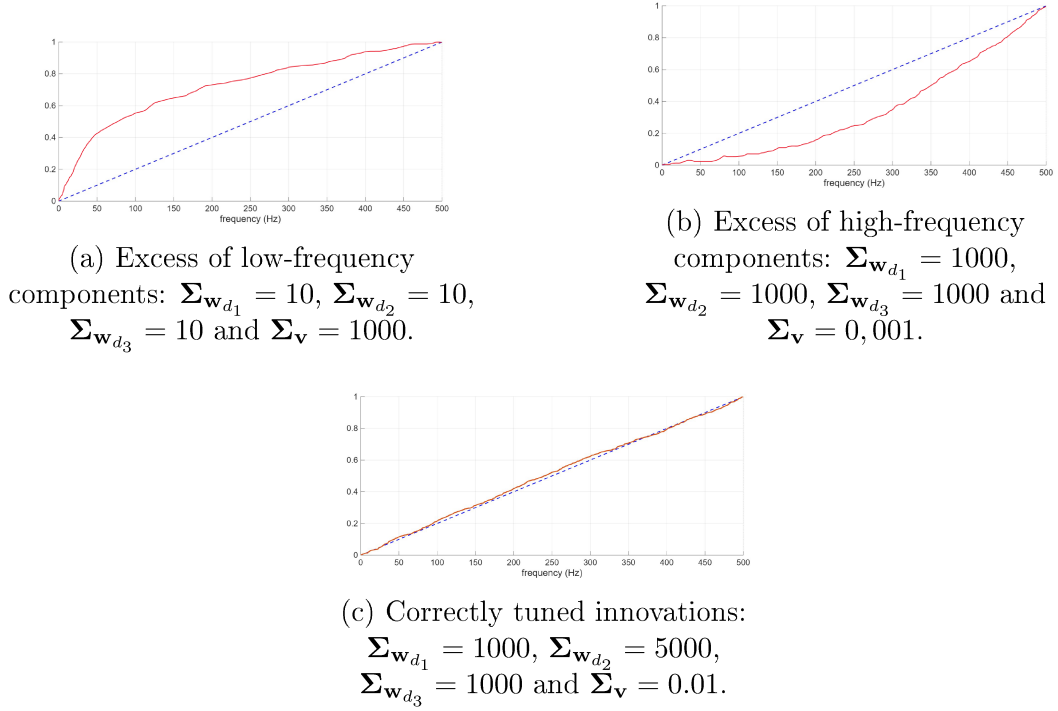


Figure 3.9: Analysis of the Bartlett Test cumulative periodograms for different tuning of the Kalman Filter.

Fig. 3.9 shows the obtained Bartlett test results on physical system. The input signal was a PRBS (Pseudo-Random Binary Sequence) signal generated to have zero-mean property. In the first case (a), the process noise covariances

were set to low values, while the measurement noise covariance was set to a high value, which led to an excess of low-frequency components in the innovations. In the second case (b), the process noise covariances were set to high values, while the measurement noise covariance was set to a low value, which led to an excess of high-frequency components in the innovations. Finally, in the third case (c), the process noise covariances and measurement noise covariance were tuned to achieve a good match between the cumulative periodogram of the innovations and the expected curve, indicating that the filter is well-tuned from the stochastic perspective.

However, often, the filter obtained with the Bartlett test tends to be very conservative and not particularly reactive. For this reason, the result obtained with the Bartlett test was used as a starting point for the tuning, and then the noise covariances were further adjusted to achieve a good trade-off between reactivity, in particular of the interaction force estimation, and noise sensitivity of the filter.

The selected values for the noise covariances are reported in table 3.1:

$\Sigma_{\mathbf{w}_{d_1}}$	$\Sigma_{\mathbf{w}_{d_2}}$	$\Sigma_{\mathbf{w}_{d_3}}$	$\Sigma_{\mathbf{v}}$
1000	50000	100	0.005

Table 3.1: Selected noise covariance values for the Kalman Filter tuning.

Finally, with the selected noise covariance values, the Kalman Filter gain $\mathbf{L}$ was calculated by solving the DARE of the equivalent transposed system on MATLAB by using the `dlqr` function:

$$\mathbf{L} = \text{dlqr}(\mathbf{A}_{kf}^T, \mathbf{H}_{kf}^T, \Sigma_{\mathbf{w}_d}, \Sigma_{\mathbf{v}}) \quad (3.129)$$

3.3.3 Force Estimation Validation

To validate the interaction force estimation, an experiment was conducted by applying a 3 N reference signal to the system. The Kalman Filter (KF) estimate was compared both to the ground truth provided by a force sensor and to the output of a traditional Disturbance Observer (DOB) with a cutoff frequency of 500 rad/s.

The measuring hardware consists of a KYOWA LMA-A-5N load cell (5 N capacity, 0.771 mV/V sensitivity) mounted on the back of the motor tip. To ensure an accurate measurement, the tip was suspended via a cable system; this setup was designed to decouple the assembly from the motor body, to eliminate

friction losses and ensuring that the entire interaction force was transferred to the load cell.

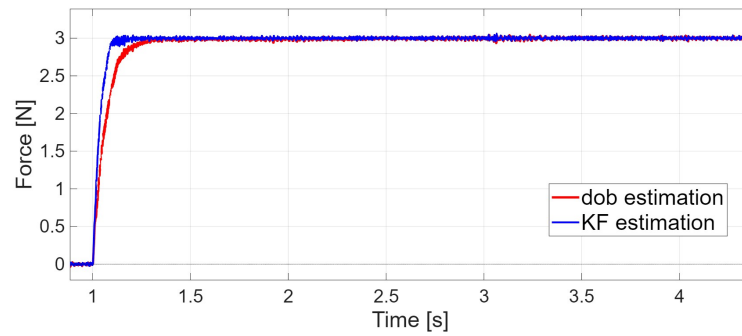


Figure 3.10: Comparison of the interaction force estimation obtained with the Kalman Filter and a Disturbance Observer (DOB) approach.



Figure 3.11: Actual force measured in steady-state with a force sensor located on the back of the motor tip.

As shown in Fig. 3.10, the KF estimation is coherent with the DOB output. Furthermore, the steady-state accuracy is confirmed by the force sensor readings (approximately 3 N), as shown in the picture in Fig. 3.11.

Chapter 4

Model Predictive Control

Model Predictive Control (MPC) is a sophisticated extension of predictive control that extends the prediction from a single step to a longer horizon, allowing for more informed decision-making. The control strategy is based on disposing of an explicit model of the system dynamics, which is used to predict the future behavior of the system over a specified time horizon. At each time step, MPC solves an optimization problem to find the optimal control inputs that minimize a cost function while satisfying constraints on the system's states and inputs.

Because of its high computational demand, MPC was initially used only in process with slow dynamics, such as chemical processes. However, with the advent of faster processors and more efficient algorithms, MPC has been successfully applied to a wide range of systems, including those with fast dynamics, such as robotic systems and autonomous vehicles, as shown in the work of F.A. Almasalmah et al. [1], and also in surgical applications, as can be seen in work of Y.Wang et al [9].

MPC is particularly well-suited for controlling systems with complex dynamics and constraints, making it an attractive choice for controlling robotic systems interacting with soft tissues. The ability of MPC to handle multi-variable systems and to optimize control inputs over a future horizon allows it to effectively manage the uncertainties and nonlinearities associated with soft tissue interactions and its ability to incorporate constraints makes it ideal for ensuring safe and effective control of robotic systems in surgical applications.

Other advantages of MPC, as described in [10], include:

- The ability to handel MIMO processes, which is common in robotic systems. In fact by using a model of the system dynamics, MPC "knows" all the interactions between the different variables, allowing it to optimize control

inputs in a way that takes into account the interdependencies between the variables.

- The ability to handle systems where the number of inputs is different from the number of outputs.
- MPC parameters have a clear physical meaning, which makes it easier to tune the controller and to understand its behavior.

4.1 Theoretical Background

As stated before, MPC obtains the optimal control inputs by solving an optimization problem at each time step. This optimization problem is solved by using the system model to predict the future behavior of the system over a specified time horizon and is typically formulated as the minimization of a quadratic cost function. To this end, this section first details how the state predictions are computed, followed by an introduction to the fundamental concepts of optimization theory.

4.1.1 State Prediction

Consider a generic discrete-time system described by the following state-space model:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) \end{cases} \quad (4.1)$$

where $\mathbf{x}(k) \in \mathbb{R}^n$ is the state vector, $\mathbf{u}(k) \in \mathbb{R}^m$ is the control input vector, $\mathbf{y}(k) \in \mathbb{R}^p$ is the output vector, and $\mathbf{F} \in \mathbb{R}^{n \times n}$, $\mathbf{G} \in \mathbb{R}^{n \times m}$, and $\mathbf{H} \in \mathbb{R}^{p \times n}$ are the system matrices.

Given an estimate of the current state $\hat{\mathbf{x}}(k|k)$, the MPC can easily predict the state evolution $\hat{\mathbf{x}}(k+i|k)$ for $i = 1, 2, \dots, N$ over a prediction horizon of length N using the system dynamics.

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{F}\hat{\mathbf{x}}(k|k) + \mathbf{G}\mathbf{u}(k) \quad (4.2)$$

$$\begin{aligned} \hat{\mathbf{x}}(k+2|k) &= \mathbf{F}\hat{\mathbf{x}}(k+1|k) + \mathbf{G}\mathbf{u}(k+1) \\ &= \mathbf{F}^2\hat{\mathbf{x}}(k|k) + \mathbf{F}\mathbf{G}\mathbf{u}(k) + \mathbf{G}\mathbf{u}(k+1) \end{aligned} \quad (4.3)$$

$$\begin{aligned} \hat{\mathbf{x}}(k+3|k) &= \mathbf{F}\hat{\mathbf{x}}(k+2|k) + \mathbf{G}\mathbf{u}(k+2) \\ &= \mathbf{F}^3\hat{\mathbf{x}}(k|k) + \mathbf{F}^2\mathbf{G}\mathbf{u}(k) + \mathbf{F}\mathbf{G}\mathbf{u}(k+1) + \mathbf{G}\mathbf{u}(k+2) \end{aligned} \quad (4.4)$$

$\vdots$

That for a generic N leads to the following compact form:

$$\hat{\mathbf{x}}(k+N|k) = \mathbf{F}^N\hat{\mathbf{x}}(k|k) + \sum_{i=1}^N \mathbf{F}^{N-i}\mathbf{G}\mathbf{u}(k+i-1) \quad (4.5)$$

Now it is possible to summarize all this by using matrix notation, defining the following matrices:

$$\mathbf{X} = \begin{bmatrix} \hat{\mathbf{x}}(k+1|k) \\ \hat{\mathbf{x}}(k+2|k) \\ \vdots \\ \hat{\mathbf{x}}(k+N|k) \end{bmatrix} \in \mathbb{R}^{Nn}, \quad \mathbf{U} = \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1) \\ \vdots \\ \mathbf{u}(k+N-1) \end{bmatrix} \in \mathbb{R}^{Nm} \quad (4.6)$$

So (4.5) can be rewritten as:

$$\underbrace{\begin{bmatrix} \hat{\mathbf{x}}(k+1|k) \\ \hat{\mathbf{x}}(k+2|k) \\ \vdots \\ \hat{\mathbf{x}}(k+N|k) \end{bmatrix}}_{\mathbf{\hat{X}} \in \mathbb{R}^{Nn}} = \underbrace{\begin{bmatrix} \mathbf{F} \\ \mathbf{F}^2 \\ \vdots \\ \mathbf{F}^N \end{bmatrix}}_{\mathbf{F} \in \mathbb{R}^{Nn \times n}} \hat{\mathbf{x}}(k|k) + \underbrace{\begin{bmatrix} \mathbf{G} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{F}\mathbf{G} & \mathbf{G} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{F}^{N-1}\mathbf{G} & \mathbf{F}^{N-2}\mathbf{G} & \mathbf{F}^{N-3}\mathbf{G} & \dots & \mathbf{G} \end{bmatrix}}_{\mathbf{G} \in \mathbb{R}^{Nn \times Nm}} \underbrace{\begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1) \\ \vdots \\ \mathbf{u}(k+N-1) \end{bmatrix}}_{\mathbf{U} \in \mathbb{R}^{Nm}} \quad (4.7)$$

$$\hat{\mathbf{X}} = \mathbf{F}\hat{\mathbf{x}}(k|k) + \mathbf{G}\mathbf{U}$$

Since:

$$\hat{\mathbf{y}}(k) = \mathbf{H}\mathbf{x}(k) \quad (4.8)$$

the output evolution can be computed as follows:

$$\hat{y}(k+N|k) = \mathbf{H}\mathbf{F}^N\hat{\mathbf{x}}(k|k) + \sum_{i=1}^N \mathbf{H}\mathbf{F}^{N-i}\mathbf{G}\mathbf{u}(k+i-1) \quad (4.9)$$

that in matrix form can be rewritten as:

$$\underbrace{\begin{bmatrix} \hat{\mathbf{y}}(k+1|k) \\ \hat{\mathbf{y}}(k+2|k) \\ \vdots \\ \hat{\mathbf{y}}(k+N|k) \end{bmatrix}}_{\hat{\mathbf{Y}} \in \mathbb{R}^{Nl}} = \underbrace{\begin{bmatrix} \mathbf{H} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{H} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{H} \end{bmatrix}}_{\mathcal{H} \in \mathbb{R}^{Np \times Nn}} \underbrace{\begin{bmatrix} \hat{\mathbf{x}}(k+1|k) \\ \hat{\mathbf{x}}(k+2|k) \\ \vdots \\ \hat{\mathbf{x}}(k+N|k) \end{bmatrix}}_{\hat{\mathbf{X}} \in \mathbb{R}^{Nn}}$$

$$\hat{\mathbf{Y}} = \mathcal{H}\mathbf{X} = \mathcal{H}\hat{\mathbf{x}}(k|k) + \mathcal{H}\mathbf{G}\mathbf{U} \quad (4.10)$$

4.1.2 Optimization Theory: Quadratic Programming

Consider $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{f} \in \mathbb{R}^n$ and $\mathbf{H} = \mathbf{H}^T \succ 0 \in \mathbb{R}^{n \times n}$ a symmetric-positive definite matrix. Now it is possible to consider the following optimization problem:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} & \left[\frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{f}^T \mathbf{x} \right] \\ \text{subject to } & \mathbf{A} \mathbf{x} \leq \mathbf{b} \end{aligned} \quad (4.11)$$

This corresponds to a convex optimization problem, since the cost function is convex (because $\mathbf{H}$ is positive definite) and the constraints are linear. This type of optimization problem is known as a *Quadratic Programming (QP)* problem and it has the following properties:

- It has no local minima, since the cost function is convex, so any local minimum is also a global minimum.
- The convergence to the optimal solution is guaranteed, since the cost function is strictly convex and the feasible region defined by the constraints is convex.
- It can be solved efficiently using various algorithms, such as interior-point methods, active-set methods, and gradient-based methods, which are implemented in many optimization libraries and software packages. For example, in MATLAB, the `quadprog` function can be used to solve QP problems.

Now, to formalize the MPC framework, reconsider the generic discrete-time system described by the following state-space model:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) \end{cases} \quad (4.12)$$

where $\mathbf{x}(k) \in \mathbb{R}^n$, $\mathbf{u}(k) \in \mathbb{R}^m$, $\mathbf{y}(k) \in \mathbb{R}^p$, and $\mathbf{F} \in \mathbb{R}^{n \times n}$, $\mathbf{G} \in \mathbb{R}^{n \times m}$, and $\mathbf{H} \in \mathbb{R}^{p \times n}$.

Then, it is possible to define the following generic quadratic cost function:

$$J = \sum_{i=0}^{N-1} [(\hat{\mathbf{x}}(k+i|k))^T \mathbf{Q}(\hat{\mathbf{x}}(k+i|k)) + \mathbf{u}(k+i)^T \mathbf{R}\mathbf{u}(k+i)] \quad (4.13)$$

where $\mathbf{Q} \succeq 0$ and $\mathbf{R} \succ 0$ are the state and input weighting matrices, respectively they weight the distance between the actual and the desired state (in this general case the desired state is the null vector $\mathbf{0}$) and the magnitude of the control input, N is the prediction horizon. The equation (4.13) is then subjected to the following linear constraints:

$$\begin{aligned} \mathbf{A}_{k+i}\mathbf{u}(k+i) &\leq f_{k+i} \\ i &= 0, 1, \dots, N-1 \end{aligned} \quad (4.14)$$

with $\mathbf{A}_{k+i} \in \mathbb{R}^{n_{c_{k+i}} \times m}$ and $f_{k+i} \in \mathbb{R}^{n_{c_{k+i}}}$, where $n_{c_{k+i}}$ is the number of constraints at time step $k+i$.

The generic linear MPC optimization problem previously introduced becomes:

$$\min_{\mathbf{u} \in \mathbb{R}^m} J \left(\{\hat{\mathbf{x}}(k+i+1|k), \mathbf{u}(k+i)\}_{i=0, \dots, N-1} \right) \quad (4.15)$$

where

$$\{\mathbf{u}(k+i)\}_{i=0, \dots, N-1}$$

$$\{\hat{\mathbf{x}}(k+i+1|k)\}_{i=0, \dots, N-1}$$

are subjected to

$$\mathbf{A}_{k+i}\mathbf{u}(k+i) \leq f_{k+i}$$

$$\hat{\mathbf{x}}(k+i+1|k) = \mathbf{F}\hat{\mathbf{x}}(k+i|k) + \mathbf{G}\mathbf{u}(k+i), \quad i = 0, 1, \dots, N-1$$

$$\hat{\mathbf{x}}(k|k) = \bar{x}$$

This problem can then be rewritten more compactly in matrix form. Starting from the cost function:

$$J = \sum_{i=0}^{N-1} [(\hat{\mathbf{x}}(k+i|k))^T \mathbf{Q}(\hat{\mathbf{x}}(k+i|k)) + \mathbf{u}(k+i)^T \mathbf{R}\mathbf{u}(k+i)] \quad (4.16)$$

Then by defining the following vectors and matrices:

$$\mathbf{U}(k) = \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1) \\ \vdots \\ \mathbf{u}(k+N-1) \end{bmatrix}, \quad \hat{\mathbf{X}}(k) = \begin{bmatrix} \hat{\mathbf{x}}(k|k) \\ \hat{\mathbf{x}}(k+1|k) \\ \vdots \\ \hat{\mathbf{x}}(k+N-1|k) \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} Q & 0 & \cdots & 0 \\ 0 & Q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R \end{bmatrix}$$

The cost function becomes:

$$J = \hat{\mathbf{X}}(k)^T \mathbf{Q} \hat{\mathbf{X}}(k) + \mathbf{U}(k)^T \mathbf{R} \mathbf{U}(k) \quad (4.17)$$

After that it is possible to consider the state evolution constraints imposed by the linear model:

$$\hat{\mathbf{X}}(k) = \underbrace{\begin{bmatrix} \mathbf{F} \\ \mathbf{F}^2 \\ \vdots \\ \mathbf{F}^N \end{bmatrix}}_{\mathcal{F}} \hat{\mathbf{x}}(k|k) + \underbrace{\begin{bmatrix} \mathbf{G} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{FG} & \mathbf{G} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{F}^{N-1}\mathbf{G} & \mathbf{F}^{N-2}\mathbf{G} & \mathbf{F}^{N-3}\mathbf{G} & \cdots & \mathbf{G} \end{bmatrix}}_{\mathcal{G}} \mathbf{U}(k) \quad (4.18)$$

and the linear constraints:

$$\begin{aligned}
& \underbrace{\begin{bmatrix} A_k & 0 & 0 & \cdots & 0 \\ 0 & A_{k+1} & 0 & \cdots & 0 \\ 0 & 0 & A_{k+2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & A_{k+N-1} \end{bmatrix}}_{\mathbf{A}} \mathbf{U}(k) \leq \underbrace{\begin{bmatrix} b_k \\ b_{k+1} \\ b_{k+2} \\ \vdots \\ b_{k+N-1} \end{bmatrix}}_{\mathbf{b}} \quad (4.19) \\
& \mathbf{AU}(k) \leq \mathbf{b}
\end{aligned}$$

Therefore, the MPC optimization problem can be rewritten in the following compact form:

$$\min_{\mathbf{U}(k) \in \mathbb{R}^{Nm}} \left(\begin{bmatrix} \hat{\mathbf{X}}(k)^T \\ \mathbf{U}(k)^T \end{bmatrix} \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}}(k) \\ \mathbf{U}(k) \end{bmatrix} \right) \quad (4.20)$$

where

$$\hat{\mathbf{X}}(k) = \mathcal{F}\hat{\mathbf{x}}(k|k) + \mathcal{G}\mathbf{U}(k)$$

and

$$\mathbf{AU}(k) \leq \mathbf{b}$$

Now this is an optimization problem in the form of a Quadratic Programming (QP) problem, since the cost function is quadratic in $\mathbf{U}(k)$ and the constraints are linear in $\mathbf{U}(k)$, and so it respects the definition done in (4.11). The number of optimization variables n_{opt} is $n_{opt} = Nn + Nm$ since we are solving for both $\hat{\mathbf{X}}(k)$ and $\mathbf{U}(k)$.

It is possible to further simplify the cost function, by using the system dynamics:

$$\begin{aligned}
J &= \hat{\mathbf{X}}(k)^T \mathbf{Q} \hat{\mathbf{X}}(k) + \mathbf{U}(k)^T \mathbf{R} \mathbf{U}(k) \\
&= (\mathcal{F}\hat{\mathbf{x}}(k|k) + \mathcal{G}\mathbf{U}(k))^T \mathbf{Q} (\mathcal{F}\hat{\mathbf{x}}(k|k) + \mathcal{G}\mathbf{U}(k)) + \mathbf{U}(k)^T \mathbf{R} \mathbf{U}(k) \\
&= \hat{\mathbf{x}}(k|k)^T \mathcal{F}^T \mathbf{Q} \mathcal{F} \hat{\mathbf{x}}(k|k) + \hat{\mathbf{x}}(k|k)^T \mathcal{F}^T \mathbf{Q} \mathcal{G} \mathbf{U}(k) + \mathbf{U}(k)^T \mathcal{G}^T \mathbf{Q} \mathcal{F} \hat{\mathbf{x}}(k|k) \\
&\quad + \mathbf{U}(k)^T (\mathcal{G}^T \mathbf{Q} \mathcal{G} + \mathbf{R}) \mathbf{U}(k) \quad (4.21)
\end{aligned}$$

The term $\hat{\mathbf{x}}(k|k)^T \mathcal{F}^T \mathbf{Q} \mathcal{F} \hat{\mathbf{x}}(k|k)$ is a constant with respect to $\mathbf{U}(k)$, so it can be discarded for the optimization problem. The term $\hat{\mathbf{x}}(k|k)^T \mathcal{F}^T \mathbf{Q} \mathcal{G} \mathbf{U}(k)$ is a

scalar, so it is equal to its transpose, which is $\mathbf{U}(k)^T \mathcal{G}^T \mathbf{Q} \mathcal{F} \hat{\mathbf{x}}(k|k)$, so the cost function can be redefined as follows:

$$\bar{J} = 2 \underbrace{\hat{\mathbf{x}}(k|k)^T \mathcal{F}^T \mathbf{Q} \mathcal{G}}_{\text{vector}} \mathbf{U}(k) + \mathbf{U}(k)^T \underbrace{(\mathcal{G}^T \mathbf{Q} \mathcal{G} + \mathbf{R})}_{\text{positive definite matrix}} \mathbf{U}(k) \quad (4.22)$$

So, summarizing, it is possible to rewrite the MPC optimization problem in the following compact form:

$$\min_{\mathbf{U}(k) \in \mathbb{R}^{Nm}} (2\hat{\mathbf{x}}(k|k)^T \mathcal{F}^T \mathbf{Q} \mathcal{G} \mathbf{U}(k) + \mathbf{U}(k)^T (\mathcal{G}^T \mathbf{Q} \mathcal{G} + \mathbf{R}) \mathbf{U}(k)) \quad (4.23)$$

where

$$\hat{\mathbf{X}}(k) = \mathcal{F} \hat{\mathbf{x}}(k|k) + \mathcal{G} \mathbf{U}(k)$$

and

$$\mathbf{A} \mathbf{U}(k) \leq \mathbf{b}$$

and written like this, it has the same structure as the equation (4.11), so it corresponds to a QP problem in $\mathbf{U}(k)$ with $n_{opt} = Nm$ optimization variables, since $\hat{\mathbf{X}}(k)$ is not an optimization variable anymore, but it is a function of $\mathbf{U}(k)$ and the initial state $\hat{\mathbf{x}}(k|k)$, which is a constant with respect to $\mathbf{U}(k)$. Usually $m \ll n$, so this is a very valuable simplification.

In addition to input limitations, the control problem can be expanded to account for constraints directly on the state variables:

$$\mathbf{B}_{k+1} \hat{\mathbf{x}}(k+1|k) \leq \mathbf{g}_{k+1} \quad (4.24)$$

In matrix form, this can be rewritten as:

$$\underbrace{\begin{bmatrix} B_{k+1} & 0 & 0 & \cdots & 0 \\ 0 & B_{k+2} & 0 & \cdots & 0 \\ 0 & 0 & B_{k+3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & B_{k+N} \end{bmatrix}}_{\mathbf{B}} \hat{\mathbf{X}}(k) \leq \underbrace{\begin{bmatrix} g_{k+1} \\ g_{k+2} \\ g_{k+3} \\ \vdots \\ g_{k+N} \end{bmatrix}}_{\mathbf{g}} \quad (4.25)$$

$$\mathbf{B} \hat{\mathbf{X}}(k) \leq \mathbf{g}$$

And by recalling that $\hat{\mathbf{X}}(k) = \mathcal{F}\hat{\mathbf{x}}(k|k) + \mathcal{G}\mathbf{U}(k)$, the state constraints can be rewritten as follows:

$$\mathbf{B}\mathbf{G}\mathbf{U}(k) \leq \mathbf{g} - \mathbf{B}\mathcal{F}\hat{\mathbf{x}}(k|k) \quad (4.26)$$

(4.26) corresponds to another set of linear constraints on $\mathbf{U}(k)$, so it can be easily added to the previous optimization problem.

In conclusion, the MPC optimization problem can be defined in the following compact form:

$$\min_{\mathbf{U}(k) \in \mathbb{R}^{N_m}} (2\hat{\mathbf{x}}(k|k)^T \mathcal{F}^T \mathbf{Q} \mathcal{G} \mathbf{U}(k) + \mathbf{U}(k)^T (\mathcal{G}^T \mathbf{Q} \mathcal{G} + \mathbf{R}) \mathbf{U}(k)) \quad (4.27)$$

subject to

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{B}\mathcal{G} \end{bmatrix} \mathbf{U}(k) \leq \begin{bmatrix} \mathbf{b} \\ \mathbf{g} - \mathbf{B}\mathcal{F}\hat{\mathbf{x}}(k|k) \end{bmatrix} \quad (4.28)$$

This remains a QP problem in $\mathbf{U}(k)$. However, there is a major difference with the previous case. In fact, if there are only input constraints, unless the set of possible inputs is empty, it is always possible to find a solution for the QP problem and thus an applicable control action. On the other hand, if there are also state constraints, it is possible that the state constraints might be impossible to be satisfied for any feasible input, and thus the QP problem might not have a solution. This is a major issue in MPC, since it can lead to situations where the controller is not able to find a control action that satisfies the constraints, which can lead to instability or unsafe behavior of the system. To address this issue, various techniques have been proposed in the literature, such as softening the constraints, as stated in the work of E.C. Kerrigan and J.M. Maciejowski [6].

The same issue can arise in the case of output constraints.

Reference Tracking

The following section introduces another general MPC problem aimed at tracking a reference output profile $\mathbf{y}_r(k)$ and accounting for a measurable disturbance $\mathbf{d}(k)$.

Consider the following linear discrete-time system:

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{u}(k) + \mathbf{M}\mathbf{d}(k) \quad (4.29)$$

$$\mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) \quad (4.30)$$

with $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{y} \in \mathbb{R}^p$, $\mathbf{d} \in \mathbb{R}^q$ and $\mathbf{F} \in \mathbb{R}^{n \times n}$, $\mathbf{G} \in \mathbb{R}^{n \times m}$, $\mathbf{H} \in \mathbb{R}^{p \times n}$, $\mathbf{M} \in \mathbb{R}^{n \times q}$, where q is the dimension of the disturbance vector $\mathbf{d}(k)$.

The cost function can be defined as follows:

$$\begin{aligned} J = & \sum_{i=0}^{N-1} [(\hat{\mathbf{y}}(k+i|k) - \mathbf{y}_r(k+i))^T \mathbf{Q}(\hat{\mathbf{y}}(k+i|k) - \mathbf{y}_r(k+i)) \\ & + \mathbf{u}(k+i)^T \mathbf{R}\mathbf{u}(k+i)] \end{aligned} \quad (4.31)$$

where $\mathbf{Q} \succeq 0$ and $\mathbf{R} \succ 0$ are the output tracking error and input weighting matrices, respectively.

And consider the following linear constraints on the inputs:

$$\begin{aligned} \mathbf{A}_{k+i}\mathbf{u}(k+i) & \leq f_{k+i} \\ i & = 0, 1, \dots, N-1 \end{aligned} \quad (4.32)$$

where $\mathbf{A}_{k+i} \in \mathbb{R}^{n_{c_{k+i}} \times m}$ and $f_{k+i} \in \mathbb{R}^{n_{c_{k+i}}}$, where $n_{c_{k+i}}$ is the number of constraints at time step $k+i$.

This again corresponds to a QP problem and it can be demonstrated in a similar manner as before. At first, we introduce the matrix representation:

$$\begin{aligned} \mathbf{U}(k) &= \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1) \\ \vdots \\ \mathbf{u}(k+N-1) \end{bmatrix}, \quad \hat{\mathbf{Y}}(k) = \begin{bmatrix} \hat{\mathbf{y}}(k|k) \\ \hat{\mathbf{y}}(k+1|k) \\ \vdots \\ \hat{\mathbf{y}}(k+N-1|k) \end{bmatrix}, \quad \mathbf{Y}_r(k) = \begin{bmatrix} \mathbf{y}_r(k) \\ \mathbf{y}_r(k+1) \\ \vdots \\ \mathbf{y}_r(k+N-1) \end{bmatrix} \\ \mathbf{Q} &= \begin{bmatrix} Q & 0 & \cdots & 0 \\ 0 & Q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R \end{bmatrix} \end{aligned}$$

With this notation, the cost function can be rewritten as follows:

$$J = (\hat{\mathbf{Y}}(k) - \mathbf{Y}_r(k))^T \mathbf{Q} (\hat{\mathbf{Y}}(k) - \mathbf{Y}_r(k)) + \mathbf{U}(k)^T \mathbf{R} \mathbf{U}(k) \quad (4.33)$$

Then, by updating the formula for the state prediction and the output prediction:

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{F}\hat{\mathbf{x}}(k|k) + \mathbf{G}\mathbf{u}(k) + \mathbf{M}\mathbf{d}(k) \quad (4.34)$$

$$\begin{aligned} \hat{\mathbf{x}}(k+2|k) &= \mathbf{F}\hat{\mathbf{x}}(k+1|k) + \mathbf{G}\mathbf{u}(k+1) + \mathbf{M}\mathbf{d}(k+1) \\ &= \mathbf{F}^2\hat{\mathbf{x}}(k|k) + \mathbf{F}\mathbf{G}\mathbf{u}(k) + \mathbf{G}\mathbf{u}(k+1) + \mathbf{F}\mathbf{M}\mathbf{d}(k) + \mathbf{M}\mathbf{d}(k+1) \end{aligned} \quad (4.35)$$

$$\begin{aligned} \hat{\mathbf{x}}(k+3|k) &= \mathbf{F}\hat{\mathbf{x}}(k+2|k) + \mathbf{G}\mathbf{u}(k+2) + \mathbf{M}\mathbf{d}(k+2) \\ &= \mathbf{F}^3\hat{\mathbf{x}}(k|k) + \mathbf{F}^2\mathbf{G}\mathbf{u}(k) + \mathbf{F}\mathbf{G}\mathbf{u}(k+1) + \mathbf{G}\mathbf{u}(k+2) \\ &\quad + \mathbf{F}^2\mathbf{M}\mathbf{d}(k) + \mathbf{F}\mathbf{M}\mathbf{d}(k+1) + \mathbf{M}\mathbf{d}(k+2) \end{aligned} \quad (4.36)$$

$\vdots$

$$\hat{\mathbf{x}}(k+N|k) = \mathbf{F}^N\hat{\mathbf{x}}(k|k) + \sum_{i=1}^N \mathbf{F}^{N-i}\mathbf{G}\mathbf{u}(k+i-1) + \sum_{j=1}^N \mathbf{F}^{N-j}\mathbf{M}\mathbf{d}(k+j-1) \quad (4.37)$$

That in matrix form is:

$$\begin{aligned} \hat{\mathbf{X}}(k) &= \begin{bmatrix} \hat{\mathbf{x}}(k+1|k) \\ \hat{\mathbf{x}}(k+2|k) \\ \vdots \\ \hat{\mathbf{x}}(k+N|k) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{F} \\ \mathbf{F}^2 \\ \vdots \\ \mathbf{F}^N \end{bmatrix}}_{\mathcal{F}} \hat{\mathbf{x}}(k|k) + \underbrace{\begin{bmatrix} \mathbf{G} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{F}\mathbf{G} & \mathbf{G} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{F}^2\mathbf{G} & \mathbf{F}\mathbf{G} & \mathbf{G} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{F}^{N-1}\mathbf{G} & \mathbf{F}^{N-2}\mathbf{G} & \mathbf{F}^{N-3}\mathbf{G} & \cdots & \mathbf{G} \end{bmatrix}}_{\mathcal{G}} \mathbf{U}(k) \\ &\quad + \underbrace{\begin{bmatrix} \mathbf{M} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{F}\mathbf{M} & \mathbf{M} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{F}^2\mathbf{M} & \mathbf{F}\mathbf{M} & \mathbf{M} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{F}^{N-1}\mathbf{M} & \mathbf{F}^{N-2}\mathbf{M} & \mathbf{F}^{N-3}\mathbf{M} & \cdots & \mathbf{M} \end{bmatrix}}_{\mathcal{M}} \underbrace{\begin{bmatrix} \mathbf{d}(k) \\ \mathbf{d}(k+1) \\ \mathbf{d}(k+2) \\ \vdots \\ \mathbf{d}(k+N-1) \end{bmatrix}}_{\mathbf{D}(k)} \end{aligned} \quad (4.38)$$

And:

$$\begin{aligned}
\hat{\mathbf{Y}}(k) &= \mathcal{H}\hat{\mathbf{X}}(k) \\
&= \underbrace{\mathcal{HF}}_{\mathbf{F}_C} \hat{\mathbf{x}}(k|k) + \underbrace{\mathcal{HG}}_{\mathbf{G}_C} \mathbf{U}(k) + \underbrace{\mathcal{HM}}_{\mathbf{M}_C} \mathbf{D}(k) \\
&= \mathbf{F}_C \hat{\mathbf{x}}(k|k) + \mathbf{G}_C \mathbf{U}(k) + \mathbf{M}_C \mathbf{D}(k)
\end{aligned} \tag{4.39}$$

Now, by substituting (4.39) in the cost function (4.33), the cost function can be rewritten as follows:

$$\begin{aligned}
J &= (\mathbf{F}_C \hat{\mathbf{x}}(k|k) + \mathbf{G}_C \mathbf{U}(k) + \mathbf{M}_C \mathbf{D}(k) - \mathbf{Y}_r(k))^T \mathbf{Q} \\
&\quad (\mathbf{F}_C \hat{\mathbf{x}}(k|k) + \mathbf{G}_C \mathbf{U}(k) + \mathbf{M}_C \mathbf{D}(k) - \mathbf{Y}_r(k)) \\
&\quad + \mathbf{U}(k)^T \mathbf{R} \mathbf{U}(k) \\
&= \underbrace{(\mathbf{F}_C \hat{\mathbf{x}}(k|k) + \mathbf{M}_C \mathbf{D}(k) - \mathbf{Y}_r(k))^T \mathbf{Q} (\mathbf{F}_C \hat{\mathbf{x}}(k|k) + \mathbf{M}_C \mathbf{D}(k) - \mathbf{Y}_r(k))}_{\diamond} \\
&\quad + \underbrace{(\mathbf{F}_C \hat{\mathbf{x}}(k|k) + \mathbf{M}_C \mathbf{D}(k) - \mathbf{Y}_r(k))^T \mathbf{Q} (\mathbf{G}_C \mathbf{U}(k))}_{\bullet} \\
&\quad + \underbrace{(\mathbf{G}_C \mathbf{U}(k))^T \mathbf{Q} (\mathbf{F}_C \hat{\mathbf{x}}(k|k) + \mathbf{M}_C \mathbf{D}(k) - \mathbf{Y}_r(k))}_{*} \\
&\quad + \mathbf{U}(k)^T (\mathbf{G}_C^T \mathbf{Q} \mathbf{G}_C + \mathbf{R}) \mathbf{U}(k)
\end{aligned} \tag{4.40}$$

From (4.40) it is possible to note that $\diamond$ is constant with regard to the optimization variable $\mathbf{U}$, so it can be discarded, and $\bullet$ is a scalar, so $\bullet^T = \bullet$, but $* = \bullet^T$, therefore $\bullet = *$.

So it is possible to redefine the cost function as follows:

$$\begin{aligned}
\bar{J} &= 2 \left(\underbrace{(\mathbf{F}_C \hat{\mathbf{x}}(k|k) + \mathbf{M}_C \mathbf{D}(k) - \mathbf{Y}_r(k))^T \mathbf{Q} \mathbf{G}_C}_{\text{vector}} \right) \mathbf{U}(k) \\
&\quad + \mathbf{U}(k)^T \underbrace{(\mathbf{G}_C^T \mathbf{Q} \mathbf{G}_C + \mathbf{R})}_{\text{positive definite matrix}} \mathbf{U}(k)
\end{aligned} \tag{4.41}$$

That, again, has the same structure as (4.11), and so that demonstrate that also the MPC tracking problem is a QP problem in $\mathbf{U}(k)$, easily solvable with standard algorithms.

4.2 Control Architectures for Tissue Interaction

As evident from the mathematical formulation detailed in the previous section, the performance and stability of an MPC controller are intrinsically dependent upon the fidelity of its internal predictive model. While the electromechanical dynamics of the linear actuator are well defined and relatively time-invariant, the interaction dynamics introduce significant complexities, particularly when dealing with biological environments. Soft tissues are, in fact, inherently nonlinear, viscoelastic, and anisotropic.

Because the controller relies on forward predictions to optimize the control input sequence, any mismatch between the internal model and the actual tissue mechanics can lead to severe degradation in force tracking performance or, more critically, to dangerous violations of safety constraints.

Consequently, the selection of the control architecture and the dynamic model represents a critical design step, characterized by a fundamental trade-off between modeling accuracy and computational complexity.

For these reasons, three different control architectures, with an increasing level of complexity, have been investigated and compared in this work.

Specifically, the analysis begins with the standard **Linear MPC (L-MPC)**, which assumes a nominal, time-invariant approximation of the environment and serves as a baseline. Subsequently, the theoretical formulation of a **Nonlinear MPC (NL-MPC)** is discussed to account for the complex viscoelastic nature of the tissue. Finally, an **Adaptive MPC (AMPC)** is proposed as the optimal synthesis for this application, where MPC capabilities are complemented with the robustness of a real-time parameter estimation algorithm.

4.2.1 Common Control Scheme and Cost Function

All the three architectures under study share the same cost function structure and the same control scheme.

The general structure of the considered system dynamics is the one already explained in Chapter [2]:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{d} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ -\frac{K_{env}}{M_n} & -\frac{B_{env}+B_n}{M_n} & -\frac{1}{M_n} \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x \\ \dot{x} \\ d \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{M_n} \\ 0 \end{bmatrix}}_{\mathbf{B}} u(t) \quad (4.42)$$

$$\begin{bmatrix} F_{env} \\ \dot{x} \end{bmatrix} = \underbrace{\begin{bmatrix} K_{env} & B_{env} & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} x \\ \dot{x} \\ d \end{bmatrix}, \quad \mathbf{J} = 0 \quad (4.43)$$

The basic cost function is defined as follows:

$$J(k) = \underbrace{\sum_{i=1}^{H_p} w_y (F_{ref}(k) - F_{env}(k+i|k))^2}_{\text{Force Tracking}} + \underbrace{\sum_{j=1}^{H_c} w_{\Delta u} (\Delta F_{cmd}(k+j|k))^2}_{\text{Damping}} \quad (4.44)$$

The first term of the cost function minimizes the deviation of the predicted interreaction force from the reference F_{ref} .

The second term penalizes the variation of the control action, instead of its absolute value. This idea, as explained in the work of González et al. [4], permits to obtain an offset-free solution, working effectively as an implicit integral action. To do so, the system dynamics needs to be redefined with an equivalent representation in discrete time that uses $\Delta F_{cmd}(k)$ as input, by expanding the state:

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}F_{cmd}(k-1) + \mathbf{G}\Delta F_{cmd}(k) \quad (4.45)$$

$$F_{cmd}(k) = F_{cmd}(k-1) + \Delta F_{cmd}(k) \quad (4.46)$$

$$\mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) \quad (4.47)$$

Equivalently, by introducing the augmented state:

$$\mathbf{x}_a(k) = \begin{bmatrix} \mathbf{x}(k) \\ F_{cmd}(k-1) \end{bmatrix} \quad (4.48)$$

the augmented model becomes:

$$\begin{bmatrix} \mathbf{x}(k+1) \\ F_{cmd}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ F_{cmd}(k-1) \end{bmatrix} + \begin{bmatrix} \mathbf{G} \\ \mathbf{I} \end{bmatrix} \Delta F_{cmd}(k) \quad (4.49)$$

$$\mathbf{y}(k) = \begin{bmatrix} \mathbf{H} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ F_{cmd}(k-1) \end{bmatrix} \quad (4.50)$$

With this structure, the weight $w_{\Delta u}$ intuitively acts as a damping factor. It can be used to prevent aggressive maneuvers and high-frequency vibrations, which can

be the causes of unintended tissue damage [9]. By tuning the weighting factors w_y and $w_{\Delta u}$, it is possible to prioritize the safety and smoothness of the interaction over the convergence speed, tailoring the control to the specific requirements of surgical applications.

H_p is the prediction horizon and corresponds to the number of future sampling intervals over which the controller predicts the system's states, starting from the current measurements at time k . H_c is called the control horizon and corresponds to the number of future steps for which the optimization algorithm actively calculates changes in the control variables. Typically, to ensure a good trade-off between control performance and real-time computational feasibility, $H_c \ll H_p$, and the control action is assumed to be constant for all steps beyond the control horizon within the prediction window.

To ensure safe operation, the following physical constraints were imposed on the optimization problem:

$$\begin{cases} -66 \text{ N} \leq F_{cmd} \leq 66 \text{ N} & (\text{Actuator Saturation}) \\ -10 \text{ N} \leq F_{env} \leq 10 \text{ N} & (\text{Force Safety Constraints}) \\ -0.15 \text{ m/s} \leq v \leq 0.15 \text{ m/s} & (\text{Speed Safety Constraints}) \end{cases} \quad (4.51)$$

The force safety threshold was set to 10 N based on clinical literature; in fact, forces exceeding 10 N have been identified as potential causes of tissue trauma, including localized ischemia or micro-vascular damage in soft abdominal organs [2].

However, as explained aerlier, enforcing strict *hard constraints* on output variables (such as F_{env} and v) can lead to the infeasibility of the Quadratic Programming (QP) problem.

To guarantee recursive feasibility under all operating conditions, output constraints are implemented as *soft constraints*. This is achieved by introducing a non-negative slack variable $\epsilon \geq 0$, which dynamically relaxes the strict boundaries:

$$\begin{cases} -10 - \epsilon \leq F_{env} \leq 10 + \epsilon \\ -0.15 - \epsilon \leq v \leq 0.15 + \epsilon \end{cases} \quad (4.52)$$

To prevent the controller from violating the safety thresholds unnecessarily, the slack variable is heavily penalized. The final cost function adopted for the implementations is thus updated as follows:

$$\begin{aligned}
J(k) = & \underbrace{\sum_{i=1}^{H_p} w_y (F_{ref}(k) - F_{env}(k + i|k))^2}_{\text{Force Tracking}} \\
& + \underbrace{\sum_{j=1}^{H_c} w_{\Delta u} (\Delta F_{cmd}(k + j|k))^2}_{\text{Damping}} \\
& + \underbrace{w_\epsilon \epsilon^2}_{\text{Soft Constraints Penalty}}
\end{aligned} \tag{4.53}$$

To enforce an *exact penalty function* behavior, the weighting factor w_ϵ is chosen to be several orders of magnitude larger than w_y and $w_{\Delta u}$. This aggressive tuning ensures that the solver will always prioritize maintaining $\epsilon = 0$ to strictly respect the clinical bounds, accepting a positive slack variable only as a last resort to avoid the mathematical failure of the QP solver.

Finally, the optimization is then performed using the `mpcqp solver` provided by MATLAB 2025b, which was then converted into C code to ensure realtime execution compatibility with the target hardware.

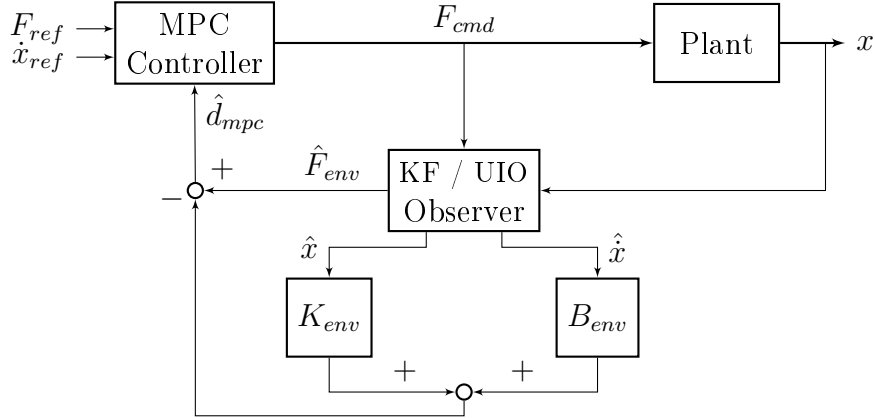


Figure 4.1: General proposed control scheme

The general block diagram of the regulated system is depicted in Fig. 4.1. A critical feature of this architecture is the compensation loop located below the Kalman Filter, which is specifically designed to prevent the phenomenon of *double counting*.

This issue arises because of the structural difference between the models implemented within the control loop. As detailed in Chapter [3], the Kalman Filter employs an *environment-agnostic* model, treating the interaction force $\hat{F}_{env}$ purely

as an external disturbance. Conversely, the MPC relies on an *environment-aware* model that explicitly incorporates the tissue dynamics ($\hat{K}_{env}$ and $\hat{B}_{env}$) within its prediction matrices.

Because the expected environmental reaction is already anticipated and managed by the MPC's internal predictions, feeding the total estimated force back to the controller would cause the algorithm to react to the same force twice. To avoid this, the nominal environmental contribution ($\hat{K}_{env}\hat{x} + \hat{B}_{env}\dot{\hat{x}}$) is continuously subtracted from the observer's output. The resulting signal, denoted as $\hat{d}_{mpc}$, represents solely the unmodeled disturbances and parameter mismatches, providing a clean and correct feedback signal to the predictive controller.

$$\hat{d}_{mpc}(k) = \hat{d}_{kf} - K_{env}\hat{x}(k) - B_{env}\hat{v}(k) \quad (4.54)$$

4.2.2 Linear MPC (L-MPC)

The Linear Model Predictive Control (L-MPC) framework represents the baseline architecture investigated in this work. In this configuration, the internal predictive model assumes that the interaction environment behaves as a Linear Time-Invariant (LTI) system. Consequently, the mechanical properties of the biological tissue are approximated using a classic Kelvin-Voigt formulation:

$$F_{env}(t) = K_{nom}x(t) + B_{nom}\dot{x}(t) \quad (4.55)$$

where K_{nom} and B_{nom} represent the constant, nominal stiffness and damping coefficients of the tissue, respectively, while $x(t)$ and $\dot{x}(t)$ denote the compression depth and velocity of the linear actuator shaft.

In this baseline formulation, the nominal parameter K_{nom} is derived by averaging the tissue stiffness values estimated across a series of experimental indentations at varying compression depths. The detailed identification procedure, along with the specific numerical values adopted for the controller tuning, will be discussed in Chapter [6].

Regarding the nominal damping B_{nom} , an experimental identification was bypassed due to the well-known challenges in characterising tissue viscosity. Instead, a realistic value was chosen by referencing the nominal material properties of the synthetic sponge used during the emulation.

The primary limitation of the L-MPC lies in its rigidity. Biological tissues are profoundly non-homogeneous and time-varying. If the actual tissue parameters deviation from the nominal values K_{nom} and B_{nom} is significant, the internal

predictions become inaccurate. This model mismatch can lead to suboptimal control actions, resulting in severe overshoot, oscillations, or critical violations of the safety force thresholds.

Therefore, while the L-MPC guarantees computational efficiency thanks to its simplicity, its performance under severe parameter variations is compromised. This brings the necessity of more sophisticated strategies, like the nonlinear MPC and the adaptive MPC that will be discussed in the next sections.

4.2.3 Nonlinear MPC (NL-MPC)

To overcome the modeling inaccuracies inherent to the linear baseline, a Nonlinear Model Predictive Control (NL-MPC) framework is introduced. In this architecture, the internal predictive model incorporates the nonlinear elastic characteristics of the biological tissue. Specifically, the nominal linear stiffness K_{nom} is replaced by a polynomial fitting function $\mathcal{P}(x)$, which continuously describes the environment's elastic reaction as a function of the compression depth. The viscous damping coefficient, conversely, is kept constant at the nominal value B_{nom} used in the linear case.

The resulting nonlinear interaction model is formulated as follows:

$$F_{env}(t) = \mathcal{P}(x(t)) + B_{nom}\dot{x}(t) \quad (4.56)$$

where $\mathcal{P}(x(t)) = p_n x^n(t) + p_{n-1} x^{n-1}(t) + \dots + p_1 x(t)$ represents the polynomial curve derived from the experimental compression tests.

By embedding the nonlinear equation (4.56) into the multi-step prediction framework, the forward state evolution can no longer be computed via standard linear matrix multiplications. Consequently, the optimization problem loses its quadratic structure, transitioning into a non-convex Nonlinear Programming (NLP) problem. Solving a NLP requires complex iterative algorithms which exhibit non-deterministic computational times and potential feasibility issues, for example the algorithms could converge on a local minima. This makes a pure NLP scheme physically unfeasible for the strict 1 kHz real-time and safety requirements of the considered application.

To bypass these feasibility and computational issues, it was decided to manage the nonlinearity outside of the prediction horizon through a Successive Linearization approach. At each sampling instant k , an equivalent linear stiffness $K_{eq}(k)$

is evaluated based on the current measured position $x(k)$:

$$K_{eq}(k) = \frac{\mathcal{P}(x(k))}{x(k)} \quad (4.57)$$

Once computed, this parameter is assumed constant over the entire prediction horizon H_p . By handling the model update outside the control horizon, the discrete-time state-space matrices $\mathbf{F}(k)$ and $\mathbf{G}(k)$ are updated at each step, but the optimization problem itself retains a purely quadratic and convex structure.

This design choice allows to still use the highly efficient `mpcqp solver` provided by MATLAB, ensuring optimal tracking performance without violating the real-time computational constraints.

Therefore, while this formulation successfully considers the tissue nonlinearity and maintains computational efficiency, it remains fundamentally constrained by the need for a rigorous offline preliminary identification. To use this controller, the exact mechanical properties of the tissue must be known a priori to generate the polynomial $\mathcal{P}(x)$. To improve the flexibility of the system and eliminate the dependency on a priori tissue knowledge, a more sophisticated strategy including an online identification process is required. This requirement motivates the development of the Adaptive MPC, which will be discussed in the next section.

4.2.4 Adaptive MPC (AMPC)

To overcome the main structural limitation of the L-MPC and the NL-MPC architectures, namely the reliance on a priori offline tissue identification, an Adaptive Model Predictive Control (AMPC) strategy is implemented. The AMPC represents the optimal synthesis for this application: it preserves the computational efficiency necessary for the high-frequency control loop while introducing the flexibility to interact with completely unknown, time-varying environments.

The main idea of the AMPC is to dynamically identify the mechanical properties of the tissue online, during the physical interaction, rather than relying on a pre-computed polynomial model. At each sampling instant k , the equivalent stiffness $\hat{K}_{env}(k)$ and viscous damping $\hat{B}_{env}(k)$ are estimated in real-time based on the instantaneous estimations of the contact force and the Kelvin-Voigt formulation.

Once the updated parameters are estimated, they are used to compute the discrete-time state-space matrices $\mathbf{F}(k)$ and $\mathbf{G}(k)$ for the current optimization step. Following the same Successive Linearization approach validated in the previous section, these matrices are held constant over the entire prediction horizon

H_p . By managing the parameter variation strictly outside the control horizon, the optimization solver continues to face a purely convex Quadratic Programming (QP) problem at each iteration.

This dynamic updating mechanism allows the controller to autonomously adapt to unmodeled environment dynamics, and makes it easier to handle transitions between free-motion and contact phases.

To enable this continuous real-time adaptation, the AMPC architecture requires a robust and computationally lightweight online estimator, which is realized in this work through a Recursive Least Squares (RLS) algorithm.

Recursive Least Squares Algorithm

The Recursive Least Squares (RLS) algorithm is a well-established methodology widely used in real-time parameter estimation due to its fast convergence and low computational overhead. In the context of the AMPC, the RLS is tasked with identifying the equivalent stiffness and damping of the environment at each time step k .

To formulate the estimation problem, the interaction force is modeled using a linear approximation, which is updated step-by-step to capture the underlying nonlinear dynamics. The environment force $F_{env}(k)$ can be expressed in a standard linear regression form:

$$y(k) = \phi^T(k)\theta(k) \quad (4.58)$$

where:

- $y(k) = F_{env}(k)$ is the estimated interaction force acting as the system output;
- $\phi(k) = \begin{bmatrix} x(k) & \dot{x}(k) \end{bmatrix}^T$ is the regression vector containing the current states of the actuator;
- $\theta(k) = \begin{bmatrix} \hat{K}_{env}(k) & \hat{B}_{env}(k) \end{bmatrix}^T$ is the vector of the unknown environment parameters to be estimated.

Since biological tissues are highly time-varying systems (e.g., their apparent stiffness changes drastically depending on the indentation depth), a standard RLS algorithm would eventually lose its tracking capability as historical data overwhelms recent measurements. To prevent this, a *forgetting factor* $\lambda \in (0, 1]$ is introduced. The forgetting factor exponentially discounts older data, allowing the algorithm to prioritize recent dynamic changes.

The optimization objective of the RLS at step k is to minimize the weighted sum of squared estimation errors:

$$J_{RLS}(\boldsymbol{\theta}) = \sum_{i=1}^k \lambda^{k-i} (y(i) - \boldsymbol{\phi}^T(i)\boldsymbol{\theta})^2 \quad (4.59)$$

The real-time recursive update laws for the parameter vector $\hat{\boldsymbol{\theta}}(k)$ and the covariance matrix $\mathbf{P}(k)$ are executed at each sampling interval according to the following sequence:

$$\mathbf{L}(k) = \frac{\mathbf{P}(k-1)\boldsymbol{\phi}(k)}{\lambda + \boldsymbol{\phi}^T(k)\mathbf{P}(k-1)\boldsymbol{\phi}(k)} \quad (4.60)$$

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \mathbf{L}(k) \left(y(k) - \boldsymbol{\phi}^T(k)\hat{\boldsymbol{\theta}}(k-1) \right) \quad (4.61)$$

$$\mathbf{P}(k) = \frac{1}{\lambda} (\mathbf{I} - \mathbf{L}(k)\boldsymbol{\phi}^T(k)) \mathbf{P}(k-1) \quad (4.62)$$

where $\mathbf{L}(k)$ represents the estimator gain vector, which dictates the magnitude of the parameter correction based on the current prediction error, and $\mathbf{P}(k)$ is the inverse correlation matrix of the input data.

The choice of the parameter λ dictates the fundamental trade-off of the estimator: a lower λ increases the tracking speed of fast parameter variations but makes the estimation more sensitive to measurement noise. Conversely, a λ closer to 1 provides smoother estimates at the cost of a slower adaptation rate.

At the conclusion of the RLS execution at step k , the newly extracted parameters $\hat{K}_{env}(k)$ and $\hat{B}_{env}(k)$ are directly fed into double counting branches and the AMPC, closing the adaptive control loop and enabling the formulation of the updated QP problem for the linear actuator.

Chapter 5

Handling Transitions: Free-Motion to Contact

In surgical applications, a robotic actuator does not always maintain constant contact with the biological tissues. Typically, the operation involves an approach phase (free-motion), where the tool moves through the air or a fluid towards the target organ, followed by an interaction phase (contact), where a specific force must be applied.

The transition between these two states could be problematic for several reasons:

- **Discontinuity in Dynamics:** The system dynamics change abruptly from a simple second-order linear motor model to a coupled motor-tissue model.
- **Impact Oscillations:** Sudden contact often triggers high-frequency vibrations or a "bouncing" effect, which can lead to tissue trauma or instability in the controller.
- **Model Mismatch:** this is particularly true in the case of Model Predictive Control, since the control law relies heavily on a system model. So it is important to find a strategy to switch the model when the contact is detected.

To address these issues, a supervisory logic was developed to manage the transition through three coordinated mechanisms: contact detection, dynamic reference generation, and weight scheduling.

The fundamental concept of the supervision logic is illustrated in Fig. 5.1, and the specific implemented algorithms will be detailed in the following sections.

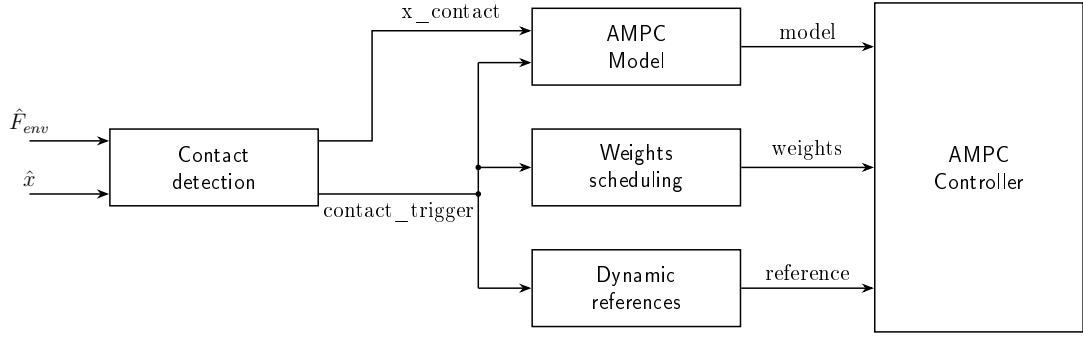


Figure 5.1: Overall logical architecture for the Free-Motion to Contact transition.

5.1 Contact Detection Algorithm

Algorithm 2: Contact Detection algorithm

Contact detection logic:

```

if contact_state == FREE_MOTION then
    // Contact conditions
    if ( $|F_{meas}| > F_{th}$  or  $|\dot{F}_{filt}| > \dot{F}_{th}$ ) then
        contact_state  $\leftarrow$  CONTACT
        x_cont  $\leftarrow$  x // contact point storing
        P  $\leftarrow$  P_init // RLS covariance matrix reset
        Contact_trigger  $\leftarrow$  1
    else
        x_cont  $\leftarrow$  x // In the air, x_cont = x
        Contact_trigger  $\leftarrow$  0
else if contact_state == CONTACT then
    if  $|F_{meas}| < F_{release}$  then
        contact_state  $\leftarrow$  FREE_MOTION
        x_cont  $\leftarrow$  0
return x_cont, Contact_trigger

```

The foundation of the transition logic relies on a robust and highly reactive Contact Detection algorithm. As detailed in Algorithm 2, the supervisor acts as a Finite State Machine (FSM) governed by two primary states: *FREE_MOTION* and *CONTACT*. The main objective of this algorithm is to continuously monitor the interaction variables and dictate exactly when the control architecture must switch from the approach phase to the active tissue indentation phase.

During the *FREE_MOTION* state, the actuator moves towards the target tissue without any physical interaction. In this phase, the algorithm continuously updates the potential contact coordinate ($\mathbf{x}_{cont}$) with the current shaft position x .

As a result, the computed penetration depth remains zero, and the environment-aware features of the controller are kept dormant (`Contact_trigger` = 0).

The detection of a physical impact is governed by a dual-condition trigger. The state machine switches to **CONTACT** if either the absolute measured force exceeds a predefined safety threshold ($|F_{meas}| > F_{th}$) or if the rate of change of the filtered force exceeds its respective limit ($|\dot{F}_{filt}| > \dot{F}_{th}$). The inclusion of the force derivative condition is a crucial design choice: it allows the algorithm to detect the sudden stiffening of the environment immediately upon impact, often fractions of a second before the absolute force reaches the steady F_{th} threshold. This is important because reacting as early as possible to the contact reduces the kinetic energy of the impact, improving the safety of the interaction, and also helps the RLS algorithm to estimate the tissue parameters more precisely and with less delay.

Upon the fulfillment of the contact condition, three fundamental actions are executed simultaneously:

1. **Contact Point Freezing:** The variable `x_cont` is locked at the exact spatial coordinate where the impact occurred. This value becomes the reference zero for computing the actual tissue penetration depth (Δx) in the subsequent control stages.
2. **Control Triggering:** The boolean flag `Contact_trigger` is activated, signaling the downstream algorithms (Weight Scheduling and Dynamic Reference) to adapt the MPC configuration for the tissue interaction.
3. **RLS Covariance Reset:** The covariance matrix of the Recursive Least Squares estimator is forcibly reset to its initial high-confidence state ($P \leftarrow P_{init}$). During the free-motion phase, the estimator processes zero-force and zero-stiffness measurements. Resetting the covariance effectively erases this "air" data history, granting the algorithm high initial gain. This ensures that the RLS is extremely reactive and capable of identifying the true tissue parameters ($\hat{K}_{env}, \hat{B}_{env}$) within the very first milliseconds of contact.

Finally, to manage the detachment from the tissue, the algorithm includes a release logic. If the state is **CONTACT** and the force drops below a specific release threshold ($|F_{meas}| < F_{release}$), the system transitions back to **FREE_MOTION**. It is important to note that $F_{release}$ is intentionally set to a lower value compared to F_{th} . This asymmetric thresholding implements a logical hysteresis, which prevents high-frequency chattering between the two states caused by sensor noise when the force fluctuates near the nominal threshold.

5.2 Weights scheduling Algorithm

Algorithm 3: Weights Scheduling

```

Function MPC_Supervisor(Contact_trigger,  $F_{meas}$ ,  $F_{release}$ ):
  Input: RLS_trigger (Binary),  $F_{meas}$  (Estimated Force)
  Output:  $\mathbf{w}_y$  (Output weights),  $w_{\Delta u}$  (Damping weight),  $state_{curr}$ 

  // -- 1. WEIGHTS CONFIGURATION --
   $w_{F,app} \leftarrow 0.0$ ,  $w_{F,impact} \leftarrow *$ 
   $w_{V,app} \leftarrow *$ ,  $w_{V,stab} \leftarrow 0.0$ 
   $w_{du,nom} \leftarrow \bullet$ ,  $w_{du,impact} \leftarrow \circ$ 

  // -- 3. TRANSITION LOGIC --
  switch state do
    case APPROACH do
      if Contact_trigger == 1 then
        state  $\leftarrow$  CONTACT
    case CONTACT do
      if  $|F_{meas}| < F_{release}$  then
        state  $\leftarrow$  APPROACH

  // -- 4. WEIGHTS GENERATION --
  switch state do
    case APPROACH do
       $\mathbf{w}_y \leftarrow [0, w_{V,app}]^T$ 
       $w_{\Delta u} \leftarrow w_{du,nom}$ 
    case CONTACT do
       $\mathbf{w}_y \leftarrow [w_{F,impact}, w_{V,stab}]^T$ 
       $w_{\Delta u} \leftarrow w_{du,impact}$  // Enhanced damping during contact

   $state_{curr} \leftarrow state$ 
  return  $\mathbf{w}_y$ ,  $w_{\Delta u}$ ,  $state_{curr}$ 

```

In a multi-objective Model Predictive Control formulation, the output weight matrix directly determines how the optimization solver prioritizes competing tracking goals. As the linear actuator transitions from free motion to tissue indentation, maintaining a static set of weights would lead to severe performance degradation. For instance, prioritizing velocity tracking during the contact phase would conflict with the force regulation task, potentially causing instability. To handle this paradigm shift, a state-dependent Weight Scheduling logic was implemented, as outlined in Algorithm 3.

Operating synchronously with the Contact Detection and Dynamic Reference modules, this supervisor adjusts the output tracking weight vector $\mathbf{w}_y$ and the

control increment penalty $w_{\Delta u}$ dynamically based on the active operational state.

- **Approach Phase (APPROACH):** During free-space motion, the controller must strictly track the safe approach velocity V_{safe} while ignoring any noise-induced force errors. Therefore, the weight vector is configured as $\mathbf{w}_y = [0.0, w_{V,app}]^T$. The force tracking weight ($w_{F,app}$) is deliberately set to 0.0, making the controller completely blind to the force reference. Conversely, a high weight is assigned to the velocity tracking. The control variation penalty $w_{\Delta u}$ is set to its nominal value $w_{du,nom}$ to ensure a smooth but responsive kinematic motion.
- **Contact Phase (CONTACT):** The instant the `Contact_trigger` goes high, the control objective shifts entirely to interaction stability and force regulation. The output weight vector is inverted to $\mathbf{w}_y = [w_{F,impact}, 0.0]^T$. The force tracking weight is activated ($w_{F,impact}$), commanding the solver to minimize the error between F_{meas} and the target force. Critically, the velocity tracking weight ($w_{V,stab}$) is reduced to 0.0. This prevents the solver from aggressively penalizing the natural velocity variations necessary to compress the tissue and reach the dynamic equilibrium. Additionally, the architecture allows for the scheduling of the damping weight $w_{\Delta u}$ ($w_{du,impact}$). Maintaining a high penalty on the control increment Δu at the exact moment of impact acts as an artificial damping mechanism, effectively suppressing high-frequency control spikes that could cause the actuator to bounce on the tissue surface.

To ensure operational continuity and cyclic recovery, the algorithm continuously evaluates the measured force F_{meas} even during the **CONTACT** state. If the force drops below the $F_{release}$ hysteresis threshold—indicating that the actuator has been retracted and contact is lost—the internal state resets to **APPROACH**.

By dynamically orchestrating the cost function parameters, the Weight Scheduling algorithm ensures that the Quadratic Programming (QP) solver always receives a well-posed optimization problem, perfectly tailored to the physical reality of the specific surgical phase.

5.3 Dynamic Reference Algorithm

In standard Model Predictive Control formulations, the reference vector is typically static or predefined along a known trajectory. However, the surgical linear

Algorithm 4: Dynamic Reference Generator

Function

Dynamic_Reference_Generator($Contact_trigger, F_{meas}, F_{target}, V_{safe}, F_{release}$):

Input: $Contact_trigger$ (Binary), F_{meas} (Force), F_{target} (Target Force), V_{safe} (Safe Velocity), $F_{release}$ (Release threshold)

Output: $\mathbf{r}_{vec}$ (Reference vector $[F_{env}, v]^T$), $mode_{curr}$ (Current mode)

// -- 1. PERSISTENT STATE MANAGEMENT --

static $state \leftarrow APPROACH$ // 0: APPROACH, 1: CONTACT

// -- 2. SWITCHING LOGIC --

if $state == APPROACH$ **then**

if $Contact_trigger$ **then**

$state \leftarrow CONTACT$ // Impact detected

else if $state == CONTACT$ **then**

if $|F_{meas}| < F_{release}$ **then**

$state \leftarrow APPROACH$ // Contact lost

// -- 3. REFERENCE VECTOR GENERATION $[F_{env}; v]$ --

if $state == APPROACH$ **then**

 // Approach phase: target safe velocity, force is ignored

$\mathbf{r}_{vec} \leftarrow [0, V_{safe}]^T$

else

 // Contact phase: target desired force, velocity set to 0 for stability

$\mathbf{r}_{vec} \leftarrow [F_{target}, 0]^T$

$mode_{curr} \leftarrow state$

return $\mathbf{r}_{vec}, mode_{curr}$

actuator must perform a hybrid task: transitioning from a purely kinematic motion in free space to a dynamic force regulation task upon tissue contact. To manage this dual objective seamlessly, a Dynamic Reference Generator was developed, as described in Algorithm 4.

This module acts in strict synchronization with the Contact Detection algorithm, receiving its binary output (`Contact_trigger`) to switch the internal persistent state between `APPROACH` and `CONTACT`. Depending on the active state, the algorithm dynamically reshapes the reference vector $\mathbf{r}_{vec} = [F_{ref}, v_{ref}]^T$ supplied to the MPC cost function.

- **Approach Phase (APPROACH):** As long as the system operates in free space, the primary control objective is to reach the tissue in a controlled and safe manner. Therefore, the reference vector is forced to $\mathbf{r}_{vec} = [0, V_{safe}]^T$. The force reference is kept at zero, as no environmental reaction is expected, while the velocity reference is set to a predefined constant V_{safe} . This constant velocity approach is critical: it bounds the kinetic energy of the actuator shaft prior to the impact, ensuring that the initial collision with the tissue occurs at a moderate, predictable speed, thus minimizing the risk of mechanical trauma or high-frequency oscillations.
- **Contact Phase (CONTACT):** As soon as an impact is detected (`Contact_trigger` becomes high), the state immediately transitions to the contact phase. The control paradigm flips: the priority becomes the accurate application of the surgical force. The reference vector is instantly updated to $\mathbf{r}_{vec} = [F_{target}, 0]^T$. The force reference switches to the desired steady-state indentation force.

Similarly to the Weights Scheduling algorithm, the transition logic then incorporates a $F_{release}$ threshold to guarantee a safe and robust reset to the approach configuration if the actuator breaks contact with the tissue.

Through this dual-state switching logic, the Dynamic Reference Generator continuously provides the downstream AMPC with a mathematically coherent and physically safe setpoint, perfectly aligned with the instantaneous operational phase of the surgical tool.

5.4 AMPC Model for Free Motion to Contact Transition

To effectively control the interaction force during the transition phase, the internal predictive model of the Adaptive MPC must perfectly reflect the physical reality of the indentation process. A standard state-space formulation utilizing the absolute actuator position $x(t)$ would be inadequate for a discontinuous environment, as the biological tissue does not exert a reaction force starting from the global coordinate origin, but rather from the specific point of impact.

To resolve this discrepancy, the system dynamics are reformulated using a relative coordinate system based on the penetration depth. Utilizing the contact point coordinate ($x_{contact}$) stored by the Contact Detection algorithm at the exact moment of impact, the penetration depth $\Delta x(t)$ is defined as:

$$\Delta x(t) = x(t) - x_{contact} \quad (5.1)$$

Since $x_{contact}$ is a spatial constant locked during the entire contact phase, the kinematic derivatives remain unaffected:

$$\Delta \dot{x}(t) = \dot{x}(t), \quad \Delta \ddot{x}(t) = \ddot{x}(t) \quad (5.2)$$

By shifting the state vector from absolute to relative kinematics, the continuous-time system dynamics described in Chapter [2] are updated as follows:

$$\begin{bmatrix} \Delta \dot{x} \\ \Delta \ddot{x} \\ \dot{d} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ -\frac{\hat{K}_{env}}{M_n} & -\frac{\hat{B}_{env} + B_n}{M_n} & -\frac{1}{M_n} \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}_c} \begin{bmatrix} \Delta x \\ \Delta \dot{x} \\ d \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{M_n} \\ 0 \end{bmatrix}}_{\mathbf{B}_c} u(t) \quad (5.3)$$

Consequently, the output equation for the predicted environmental force becomes directly proportional to the actual tissue compression:

$$F_{env}(t) = \hat{K}_{env} \Delta x(t) + \hat{B}_{env} \Delta \dot{x}(t) \quad (5.4)$$

This relative formulation is intrinsically linked to the supervisor logic. During the **APPROACH** phase, Δx evaluates to zero and the effect of B_{env} is not considered, effectively decoupling the tissue dynamics from the motor model and allowing the MPC to act as a pure motion controller.

When the **Contact_trigger** activates the **CONTACT** phase, the Relative State-

Space model (5.3) is fed to the optimization solver. Simultaneously, the Recursive Least Squares (RLS) estimator begins utilizing the relative states (Δx and $\Delta \dot{x}$) to compute the online updates for $\hat{K}_{env}$ and $\hat{B}_{env}$. This architectural synchronization guarantees that the AMPC predicts the interaction forces accurately, starting exactly from the tissue boundary, thereby ensuring a stable and bumpless transition from free-space tracking to active force regulation.

Chapter 6

Experimental Implementation and Results

While the previous chapters established the mathematical frameworks, the optimization architectures, and the transition logic, this chapter presents the practical deployment and the empirical validation of the proposed control strategies. The primary objective is to demonstrate that the advanced Model Predictive Control (MPC) techniques, particularly the Adaptive MPC (AMPC) coupled with the supervisory logic, not only meet the strict real-time and safety requirements of surgical applications but also significantly outperform classical reactive control architectures.

All the proposed algorithms, initially validated through simulations, were deployed on a real system featuring a high-precision linear actuator interacting with varying synthetic tissue models.

6.1 Experimental setup

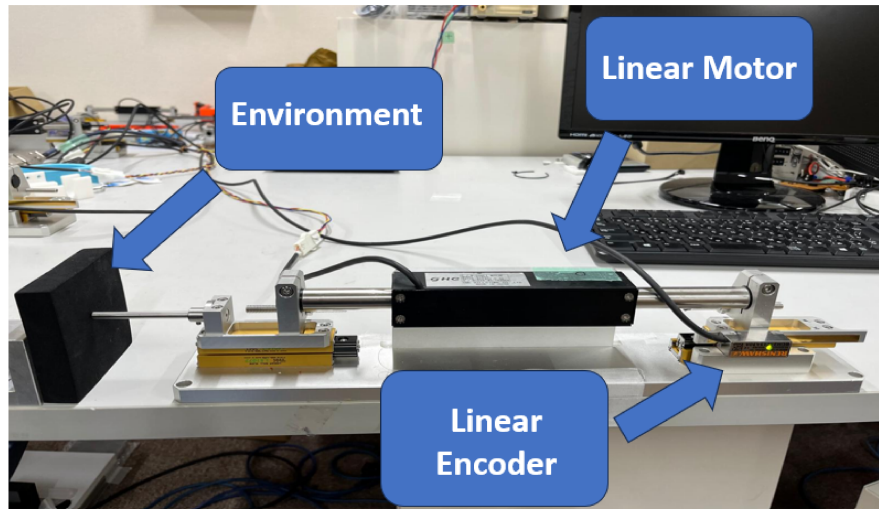


Figure 6.1: Experimental

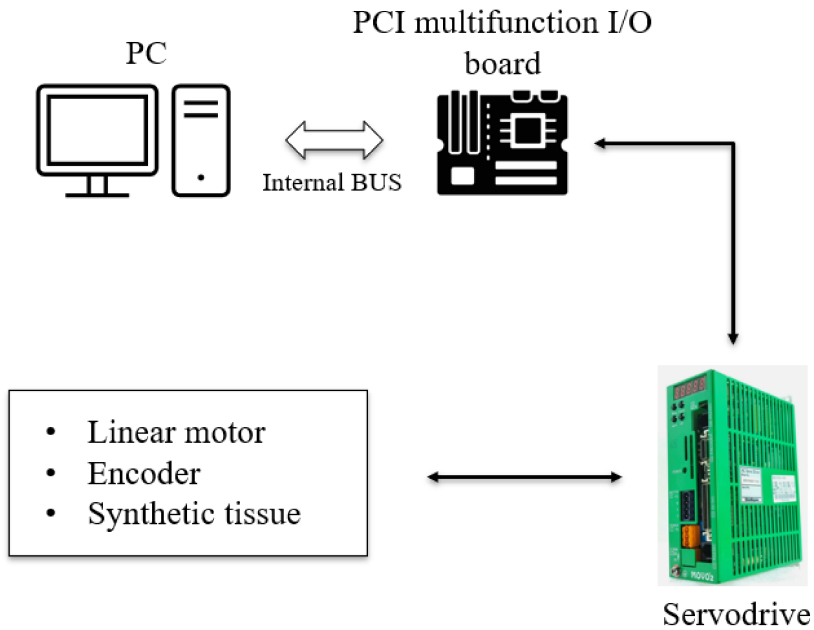


Figure 6.2: Experimental setup diagram

The experimental setup is shown in Fig. 6.1 and it can be summarized as shown in Fig. 6.2, and consists of the following components:

- **PC station:** equipped with Linux RTAI (Real-Time Application Interface), a modified version of the Linux kernel that guarantees hard real-time performance. The control algorithms were implemented in low-level C and

executed directly in kernel space. This strictly constrained environment required the development of bespoke mathematical functions, as standard user-space libraries (such as `math.h`) are unavailable, demonstrating the high computational efficiency and rigorous design required to maintain the strict real-time deadlines.

- **PCI multifunction I/O board:** used for real-time data acquisition and control signal output, by using an ADC and a DAC. The parameters of the acquisition board are the following:

Component	Parameters	Description
PCI - D/A	Type	PCI-3340
	Resolution	16 bit
	Output	± 5 V or ± 10 V
	Channels	8
	Conversion time	$5 \mu s$
PCI - A/D	Type	PCI-3178
	Resolution	16 bit
	Output	± 5 V or ± 10 V
	Channels	4
	Conversion time	$20 \mu s$
Counter Boards	Type	PCI-6201 (Linear Motor)
	Resolution	16 bit
	Input	5 V
	Channels	4
	Max input freq.	$1 MHz$

Table 6.1: Data acquisition and interface board specifications.

- **Servodrive:** responsible for actuating the linear motors based on the control signals generated by the algorithms. The servodrive was configured to operate in velocity control mode, allowing for precise regulation of the actuator's speed during both free motion and contact phases.

Component	Parameters	Description
Servo-Driver	Type	MOVO2 SVFM1 DSP-model
	Transconductance K_i	0.2 A/V
	Voltage (Vmax/rms)	$\pm 5 \text{ V}$ or $\pm 10 \text{ V}$
	Voltage	$V_{max} = 84 \text{ V}$
		$V_{rms} = 59.4 \text{ V}$
	Current	$A_{max} = 2.75 \text{ A}$
		$A_{rms} = 2.0 \text{ A}$

Table 6.2: Servo-driver technical specifications.

- **Linear Encoder:** used for precise position feedback of the linear actuator.

Component	Parameters	Description
Linear Encoder	Type	RENISHAW RGH24Y15A30A
	Resolution	$0.1 \mu m$

Table 6.3: Linear encoder specifications.

- **Linear Actuator:** the primary interface for interaction with the environment.

Component	Parameters	Description
Linear Motor	Type	GHC S160Q
	Thrust constant	$K_t = 33 \text{ N/A}$
	Mover weight (without guides and tip)	0.3 kg
	Mover weight (with guides and tip)	0.648 kg
	Windings resistance	$R = 43 \Omega$
	Max voltage	$V_{max} = 240 \text{ V}$
	Continuous current (RMS)	$I_{cont} = 0.62 \text{ A}$
	Peak current (RMS)	$I_{peak} = 2.5 \text{ A}$

Table 6.4: Linear actuator specifications.

- **Synthetic Tissue Models:** to validate the robustness and adaptability of the control algorithms across different interaction scenarios, three distinct synthetic materials were employed, each exhibiting specific mechanical properties:
 - *Material 1 (Soft):* A highly compliant, open-cell synthetic polymer foam (commercially analogous to standard cleaning sponges). This material was selected to simulate very soft, easily deformable biological tissues with low mechanical impedance.
 - *Material 2 (Medium):* A closed-cell polyurethane (PU) foam, typically found in stress-relief objects. It is characterized by moderate stiffness and a more pronounced nonlinear elastic response, making it suitable for replicating denser organ structures.
 - *Material 3 (Hard):* A high-density elastomeric foam pad. This material was utilized to replicate stiffer anatomical boundaries.

6.2 Evolution and Comparative Analysis of Control Schemes

To evaluate the force regulation capabilities of the proposed architectures, a comparative analysis was conducted between the Linear MPC, the Nonlinear MPC (NL-MPC), and the Adaptive MPC (AMPC). To provide a baseline for performance evaluation, a standard Proportional controller enhanced with a Disturbance Observer (P+DOB), with $K_p = 1$, was also implemented and tested under the same conditions.

To isolate the tracking performance and exclude the transient phenomena associated with the free-motion to contact transition, the tests were initialized with the linear actuator already in contact with the synthetic tissue. The system was then subjected to a step response test, commanding a reference force of 3 N at $t = 1$ s.

6.2.1 Linear MPC and Parameter Tuning

The experimental phase began with the implementation of the Linear MPC (L-MPC). Before evaluating its absolute tracking performance, a rigorous tuning phase was conducted to adapt the theoretical formulation to the physical constraints of the hardware.

The first critical parameter investigated was the prediction horizon H_p . Early experiments necessitated a long prediction horizon ($H_p = 200$) to smooth out a transient ‘bending’ in the experimental responses, while the control horizon was set to $H_c = 20$. The results of these early experiments are shown in Fig. 6.3.

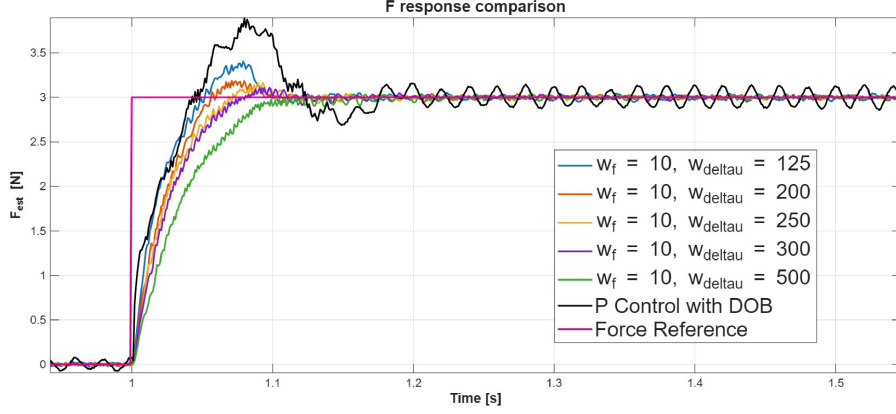


Figure 6.3: Early tuning results with a long prediction horizon ($H_p = 200$).

However, after solving some minor issues related to the hardware setup and the friction compensation, the transient bending phenomenon was completely eliminated, allowing for a drastic reduction of the prediction horizon. This is a critical improvement, as a long prediction horizon significantly increases the computational load, which can lead to real-time constraint violations. The horizon was subsequently drastically reduced to $H_p = 50$. As shown in Fig. 6.4, experimental evaluations demonstrated that this reduction did not degrade the tracking performance. A 50-step window (50 ms) adequately covers the fundamental transient dynamics of the actuator, providing optimal predictions without violating the real-time constraints.

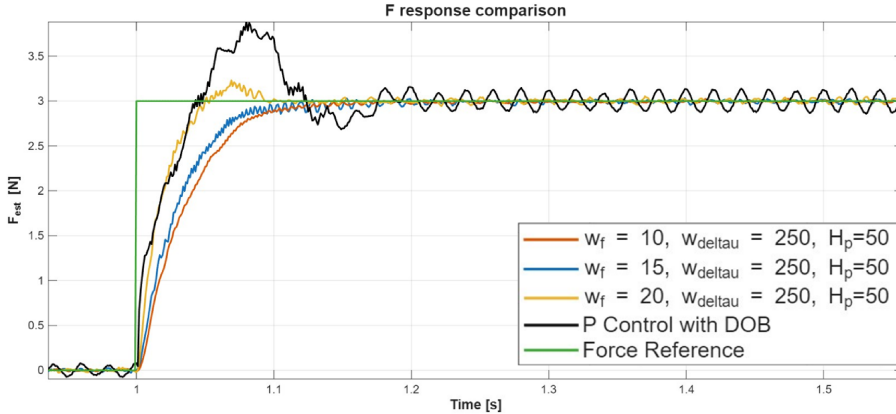


Figure 6.4: Experimental results with the optimized prediction horizon ($H_p = 50$).

The second fundamental tuning step involved the cost function weights, $w_{\Delta u}$ and w_f . Fig. 6.3 clearly shows how $w_{\Delta u}$ can work as an active damping factor. In fact, it can be appreciated how progressively increasing $w_{\Delta u}$ naturally smoothed the force transient. A high penalty on the control variations effectively acts as an artificial damper, absorbing the energy of the step and preventing the bouncing phenomenon. A similar effect can be observed about w_f in Fig. 6.4, where a low tracking weight results in a slower response, while a high tracking weight leads to a more aggressive control action, reducing the rise time but potentially increasing the overshoot. The optimal trade-off was found at $w_{\Delta u} = 250$, combined with a tracking weight $w_f = 15$.

Then from both Fig. 6.3 and Fig. 6.4, is also possible to appreciate the ability of the L-MPC to achieve less overshoot and a faster settling time compared to the classical P+DOB controller, which is shown in the same figures for comparison.

However, while the properly tuned Linear MPC provided stable and completely overshoot-free tracking, its performance was intrinsically limited by the assumption of a constant environment stiffness. Specifically, it struggled to capture the initial low-resistance region of the synthetic tissue, resulting in a highly over-damped response and a sluggish rise time.

6.2.2 Nonlinear MPC Implementation

To overcome the inherent limitations of the linear baseline, the Nonlinear MPC (NL-MPC) was subsequently deployed. By incorporating a polynomial representation of the tissue directly into the equivalent secant stiffness update, the controller was made aware of the changing environmental impedance.

The chosen polynomial model was a cubic function, which provided a good balance between modeling accuracy and computational complexity. The parameters of the cubic model were identified through offline system identification experiments, where the force response of the synthetic tissue was recorded for a range of displacements in steady state, to remove the viscous effects. The resulting force-displacement data was then fitted with a cubic polynomial, yielding the coefficients that were directly embedded into the MPC model. The synthetic tissue used in the experiments with the NL-MPC was the hardest one and the results of the identification are shown in the following equations and figures 6.5 and 6.6.

$$F_{env}(x) = ax^3 + bx^2 + cx + d \quad (6.1)$$

$$K_{eq}(x) = \frac{F_{env}(x)}{x} = ax^2 + bx + c \quad (6.2)$$

$$(6.3)$$

where the empirically identified coefficients are:

$$a = -1.58 \times 10^6$$

$$b = 1.36 \times 10^5$$

$$c = 5.01 \times 10^2$$

$$d = 0.0$$

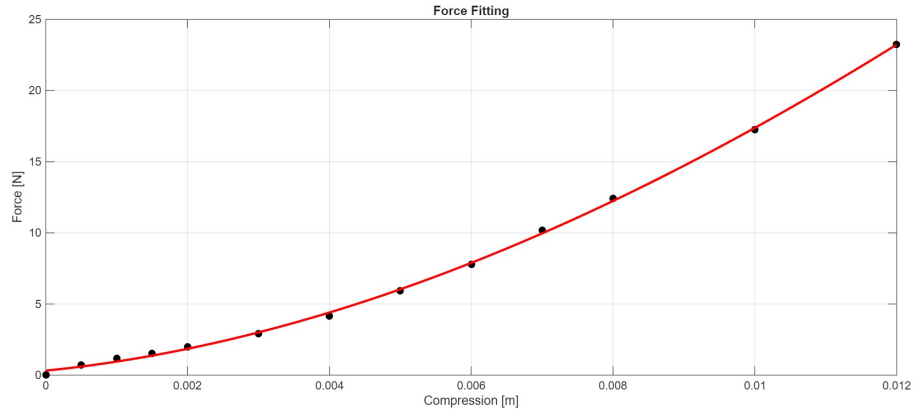


Figure 6.5: Force-displacement data of the synthetic tissue and the corresponding cubic fit.

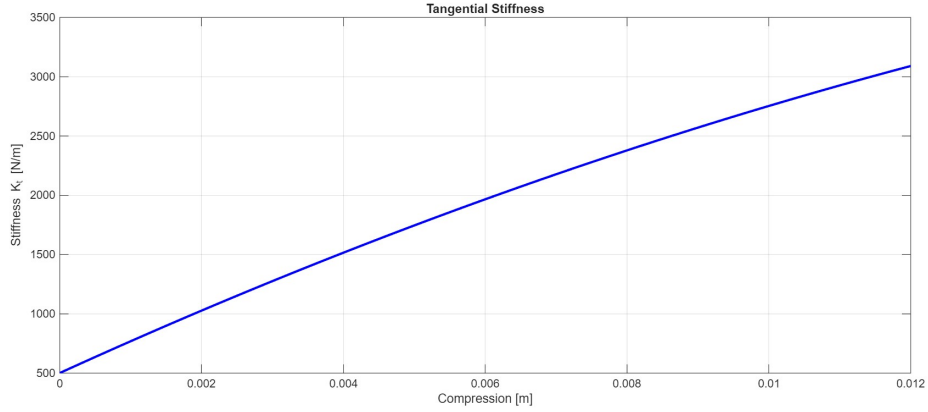


Figure 6.6: Equivalent secant stiffness as a function of the compression, derived from the cubic force model.

At this stage, it is crucial to address a potential physical inconsistency inherent to the polynomial formulation. Specifically, deriving the equivalent secant stiffness from a cubic force model yields a quadratic function ($K_{eq}(x) = ax^2 + bx + c$). From a purely mathematical perspective, the squared term would result in a positive stiffness even for negative displacements (i.e., when the actuator retracts and breaks contact with the tissue), which is physically nonsensical. However, this mathematical artifact is completely neutralized by the overall control architecture. The cubic model is strictly defined as a local approximation, valid only within the compression domain ($x > 0$). As extensively detailed in Chapter 4, whenever the actuator breaks contact and moves away from the tissue, the Contact Detection supervisor instantly forces the system into the **APPROACH** state, completely decoupling the environment model from the MPC predictions. Consequently, the controller never evaluates the quadratic stiffness for negative penetrations, guaranteeing absolute physical consistency at all times.

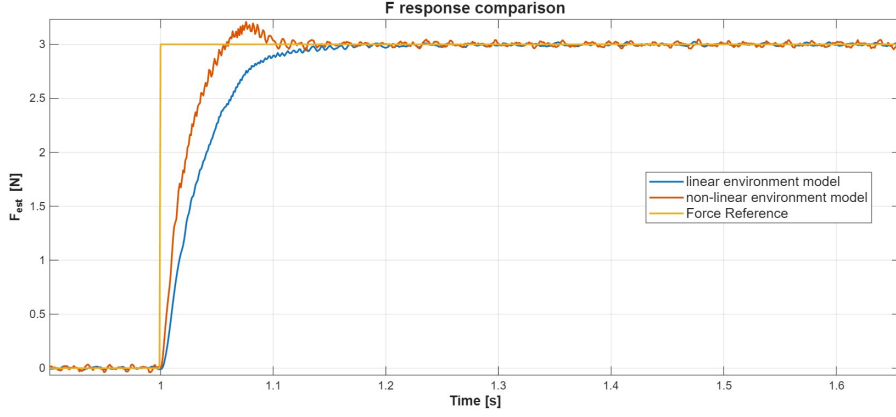


Figure 6.7: Experimental force response comparison between the Linear MPC (blue) and the Nonlinear MPC (orange) subjected to a 3 N step reference.

Fig. 6.7 depicts the experimental force response comparison between the Linear MPC (blue) and the Nonlinear MPC (orange) subjected to a 3 N step reference and with the same tuning parameters ($w_f = 15$, $w_{\Delta u} = 250$, $H_p = 50$, $H_c = 20$). The L-MPC response is characterized by a slower rise time and a long settling time, while the NL-MPC achieves a much faster rise time and a significantly reduced settling time, at the cost of a small overshoot (approximately 5%). This performance improvement is directly attributable to the NL-MPC's ability to capture the nonlinear stiffness characteristics of the tissue, allowing for a more aggressive control action during the initial contact phase without compromising stability.

6.2.3 Adaptive MPC and Overall Comparison

While the NL-MPC demonstrated excellent performance, it remained constrained by the necessity of offline preliminary tissue identification. To address this, the Adaptive MPC (AMPC) driven by the Recursive Least Squares (RLS) estimator was finally tested.

The parameters of the RLS algorithm were tuned to achieve a fast convergence while maintaining estimation stability. The chosen values are:

- Forgetting factor $\lambda = 0.98$: this value resulted in a good balance between responsiveness to changes and noise sensitivity. A lower λ would have made the estimator more reactive but also more susceptible to noise, while a higher λ would have resulted in a more stable but slower convergence.
- Initial covariance $P_0 = 10000 \cdot I_{2 \times 2}$: this high initial covariance reflects a

high level of uncertainty about the initial parameter estimates, allowing the RLS algorithm to quickly adapt to the true system parameters based on the incoming data.

In addition to that, the raw estimated data were bounded within a physically plausible range to prevent the MPC from receiving unrealistic parameter values during the initial convergence phase. Specifically, the estimated stiffness K_{env} and damping B_{env} were constrained to be non-negative and below a maximum value derived from the known properties of the synthetic tissues. This bounding strategy ensured that even during the early stages of estimation, when the RLS algorithm is still converging, the control actions remain safe and effective.

$$0 \leq \hat{K}_{env} \leq 5000 \text{ N/m}, \quad 0 \leq \hat{B}_{env} \leq 1000 \text{ Ns/m} \quad (6.4)$$

Then, K_{env} and B_{env} were initialised with physically plausible values (e.g., $K_{env} = 300 \text{ N/m}$, $B_{env} = 50 \text{ Ns/m}$) to provide the MPC with a reasonable starting point for control actions, even before the RLS estimator converges to the true parameters.

To analyze the internal dynamics of the AMPC, it is crucial to observe the behavior of the online estimator during the impact transient. Fig. 6.8 illustrates the RLS parameter estimation during a simulated¹ step response where the simulated environment was modeled using the identification done for the nonlinear case.

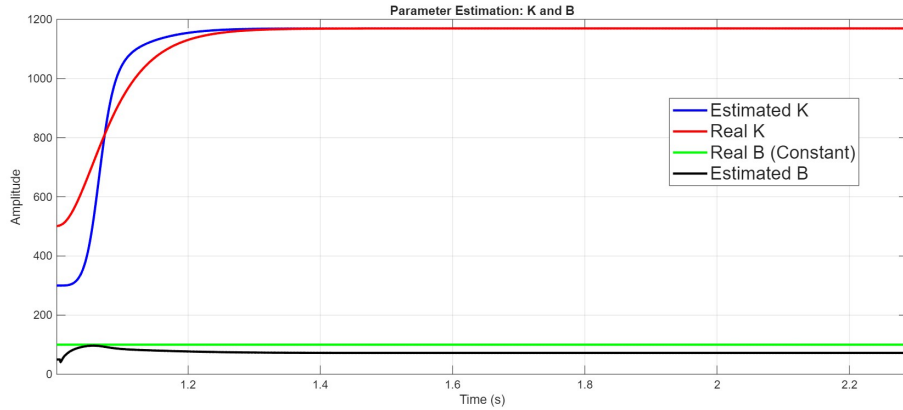


Figure 6.8: Time evolution of the estimated stiffness $\hat{K}_{env}$ and damping $\hat{B}_{env}$ parameters during the step response simulation.

As depicted in the graph, upon initial contact, the estimated stiffness (blue

¹The simulation was done using a model of the regulated system on MATLAB Simulink.

line) requires a brief transient period to fully converge to the actual nonlinear stiffness of the tissue (red line). This plot is fundamental to understanding the AMPC's transient behavior. During this initial estimation lag, the controller operates under the assumption that the environment is significantly softer than it actually is. Consequently, the optimization solver dictates a more aggressive control action to reach the desired 3 N target.

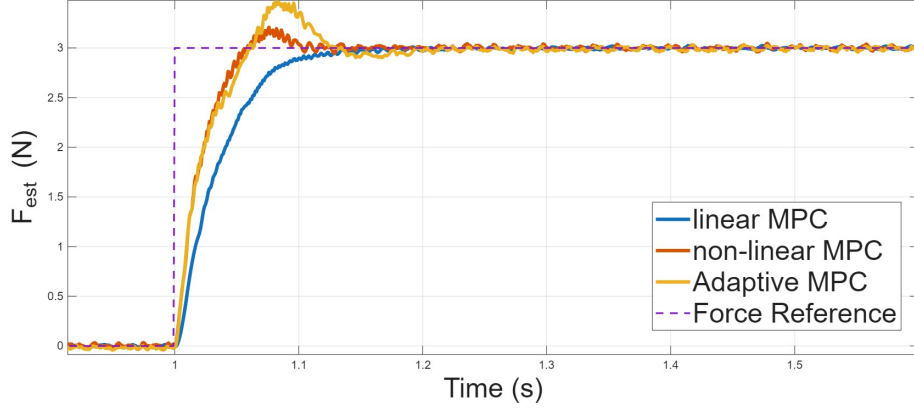


Figure 6.9: Experimental force response comparison between the Linear MPC (blue), the Nonlinear MPC (red) and the Adaptive MPC (yellow) subjected to a 3 N step reference.

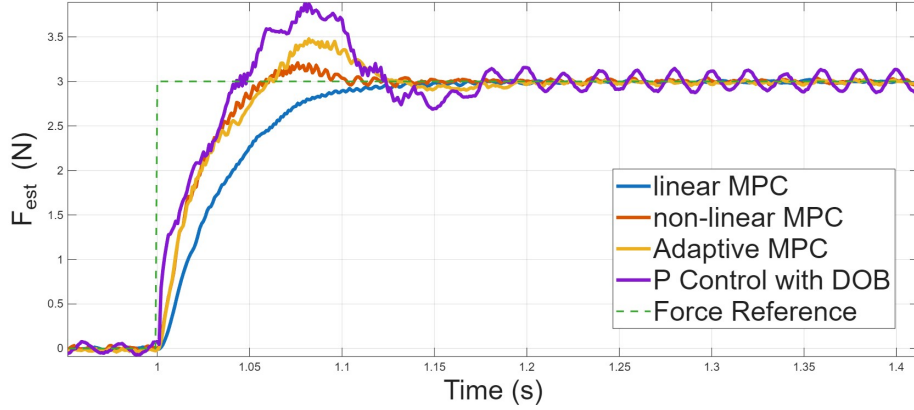


Figure 6.10: Experimental force response comparison between the Linear MPC (blue), the Nonlinear MPC (red), the Adaptive MPC (yellow) and the P+DOB controller (purple) subjected to a 3 N step reference.

Fig. 6.9 depicts the experimental force response comparison between the Linear MPC (blue), the Nonlinear MPC (red) and the Adaptive MPC (yellow) subjected to a 3 N step reference and with the same tuning parameters ($w_f = 15$,

$w_{\Delta u} = 250$, $H_p = 50$, $H_c = 20$). Fig. 6.10 adds the P+DOB response (purple) to the comparison.

The quantitative performance indicators for each controller, including Rise Time, Settling Time (5%), and Overshoot, are summarized in Table 6.5.

Controller	Rise Time [s]	Settling Time (5%) [s]	Overshoot [%]
Linear MPC	0.067	0.092	0.78
NL-MPC	0.041	0.080	5.01
AMPC	0.046	0.115	14.15
P+DOB	0.039	0.164	25.86

Table 6.5: Experimental performance comparison of the implemented control architectures.

The comparative analysis yields several key observations:

- **MPC Superiority:** All three proposed MPC formulations vastly outperformed the standard P+DOB architecture. While the classical reactive controller exhibits a mathematically fast rise time (0.039 s), it suffers from a massive overshoot (25.86%) and prolonged oscillations, rendering it unsuitable for delicate surgical interactions.
- **Model Accuracy Impact:** The transition from L-MPC to NL-MPC reduces the rise time (0.067 s to 0.041 s) and improves the settling time (0.080 s), demonstrating the critical importance of embedding the tissue nonlinearity into the control law.
- **AMPC Trade-offs:** The Adaptive MPC successfully regulates the force without any prior environment knowledge. The slight degradation in settling time (0.115 s) and the noticeable overshoot (14.15%) are direct physical consequences of the RLS estimator's convergence time. During the very first milliseconds of the step, the estimator is still computing the correct $\hat{K}_{env}$, leading to a brief phase of model mismatch. However, this transient is quickly resolved, and the system reaches a stable equilibrium, confirming the AMPC as the most flexible and clinically relevant solution among those tested.

6.3 Safety and Robustness Stress Test

To evaluate the robustness and effectiveness of the proposed AMPC architecture, the system was tested against the three considered synthetic materials. For these trials, the damping weight was increased to $w_{\Delta u} = 450$ to reduce the controller's aggressiveness during the contact transient and prioritize stability. Fig. 6.11 illustrates the experimental outcomes, while Table 6.6 shows the performance indicators:

- **Medium and Hard:** In both scenarios, the controller successfully reaches the force setpoint with an almost critically damped response, exhibiting a low overshoot.
- **Soft:** A more visible overshoot is still present despite the higher damping. This phenomenon indicates that softer tissues are more sensitive to the inherent estimation lag of the RLS algorithm.

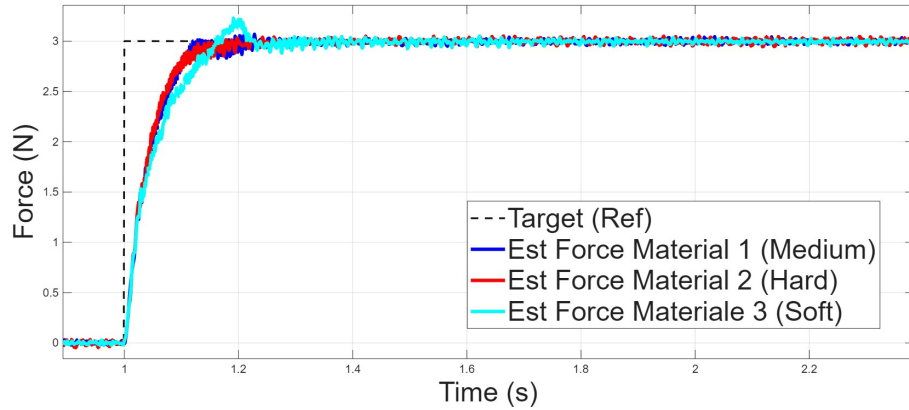


Figure 6.11: AMPC with increased damping weight ($w_{\Delta u} = 25$) across three different environments.

Material	Rise Time [s]	Settling Time (5%) [s]	Overshoot [%]
Material 1 (Medium)	0.086	0.109	0.5
Material 2 (Hard)	0.084	0.117	1.4
Material 3 (Soft)	0.114	0.207	5.96

Table 6.6: AMPC performances with different tissues

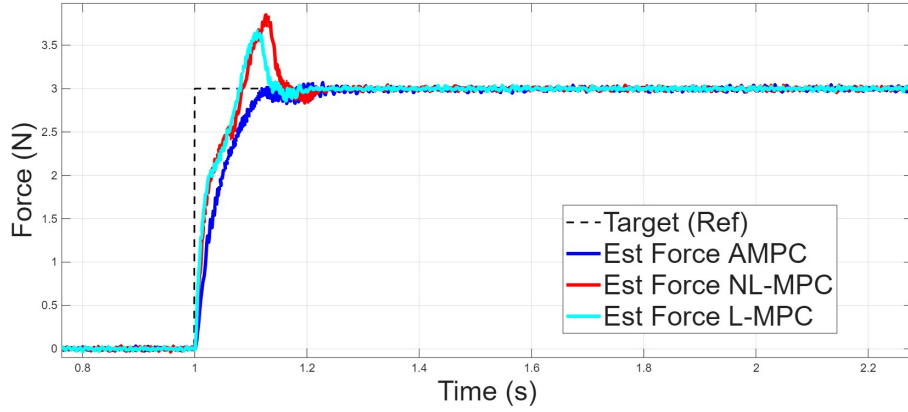


Figure 6.12: Comparison of the force responses of the different MPCs on an unknown environment

Fig. 6.12 shows a comparison of the force responses of the three proposed architectures on an unknown environment that corresponds to Material 1 in Table 6.6, while the L-MPC and the NL-MPC were calibrated on Material 2. As expected, both non-adaptive controllers exhibit a significant overshoot, approximately 20% and 25% respectively, when interacting with the softer Material 1. This behavior is primarily due to the model-plant mismatch. In contrast, the AMPC architecture, by leveraging the RLS algorithm, is able to ensure a smoother response. These results confirm that the AMPC, although it requires a higher damping penalization to handle very compliant tissues, can offer higher flexibility and robustness thanks to its ability to identify unknown environments without prior calibration.

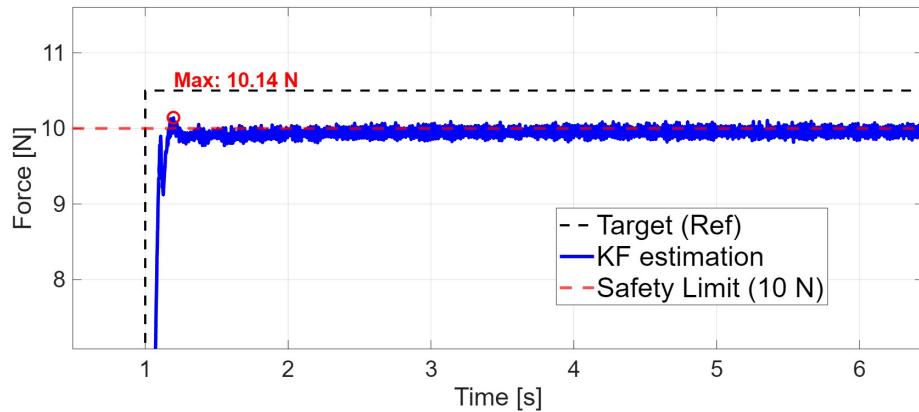


Figure 6.13: AMPC response to a force reference that exceeds the safety threshold.

To evaluate the safety of the proposed control architecture, a "stress test"

was performed by providing a force reference $F_{ref} = 10.5$ N, which intentionally exceeds the predefined safety threshold of 10 N and could potentially damage the handled tissue. As shown in Fig. 6.13, the MPC controller successfully identifies the violation of the constraint and saturates the control action to keep the interaction force within safe boundaries. The system exhibits a negligible peak overshoot of 1.4%, reaching a maximum value of 10.14 N before settling at the 10 N limit. This transient behavior could be attributed to the discrete-time nature of the controller. The fact that the force initially approaches the threshold, slows down, and then recovers toward the limit is a direct consequence of the anticipatory nature of the AMPC controller. The optimizer predicts future violations and reduces the control effort to prevent a collision with the constraint. However, due to possible model-plant mismatch and system non-idealities (such as friction or unmodeled actuator dynamics), this preventive damping leads to a temporary underestimation of the required force, which is then corrected as the system reaches a steady-state condition on the constraint boundary.

This result shows that the controller prioritizes the patient’s safety over the tracking performance, ensuring that the contact force never dangerously exceeds the imposed limit, even under improper commands.

6.4 Free-Motion to Contact Transition

The ultimate validation of the proposed control architecture involves the execution of a complete interaction task, transitioning from unconstrained free-space motion to active tissue indentation. As extensively modeled in Chapter 4, this scenario poses significant control challenges due to the abrupt change in system dynamics and the risk of severe impact oscillations. To manage this phase, the supervisory logic, comprising the Contact Detection, Dynamic Reference Generator, and Weight Scheduling algorithms, was executed synchronously with the Adaptive MPC.

The experimental trial shown in Fig. 6.14 and Fig. 6.15 evaluated a single approach-to-contact maneuver. The actuator was initialized at a distance from the synthetic tissue and commanded to approach the tissue at 2 cm/s and establish a 3 N contact force. For this test, the utilized synthetic tissue was the hard one, $w_{\Delta u}$ was set to 250 during the free-motion phase and increased to 450 at the moment of contact to prioritize stability during the transient.

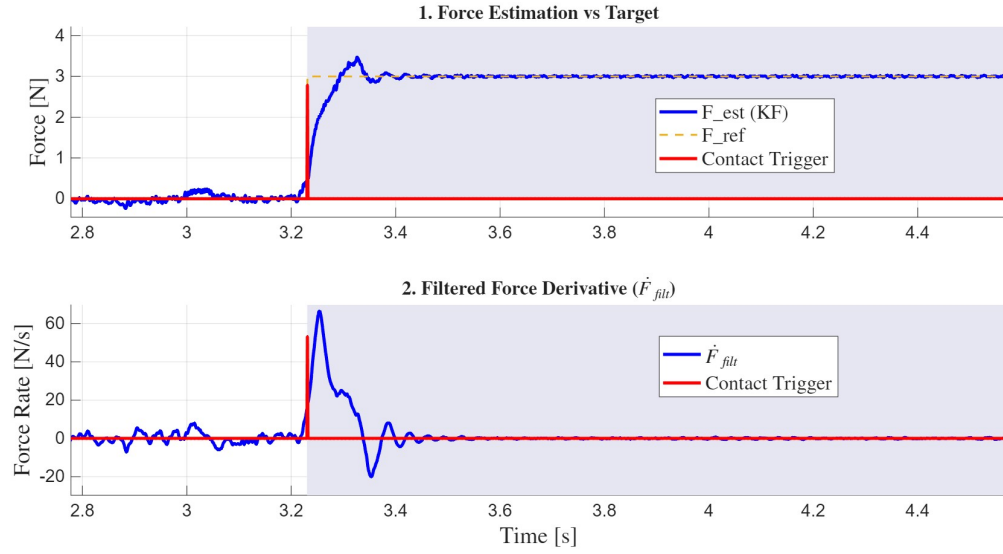


Figure 6.14: Experimental response of the AMPC during a free-motion to contact transition. (a) Interaction force before and after contact. (b) Filtered force derivative and contact trigger signal.

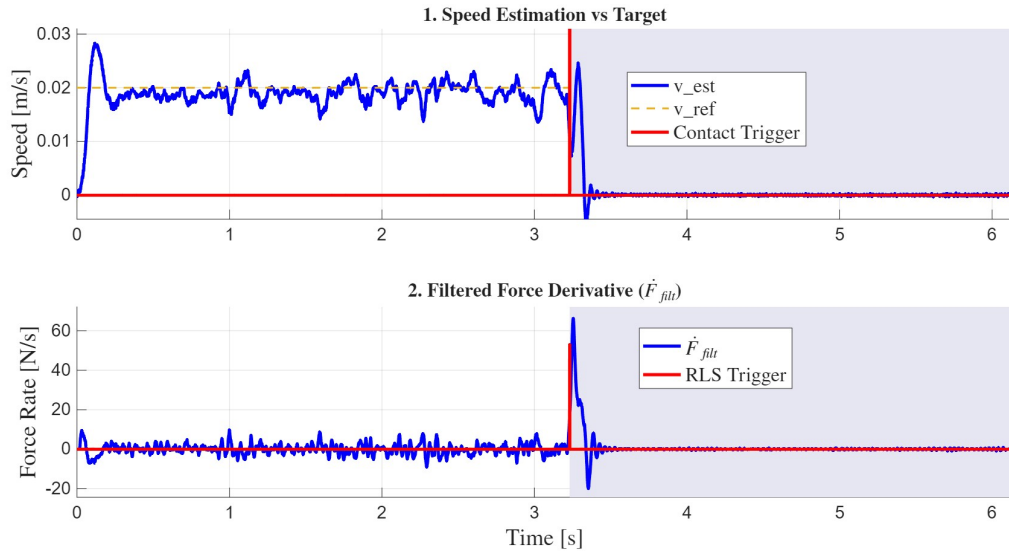


Figure 6.15: Experimental response of the AMPC during a free-motion to contact transition. (a) Actuator speed before and after contact. (b) Filtered force derivative and contact trigger signal.

The graphs clearly highlight the distinct operational phases managed by the supervisor. During the initial **APPROACH** phase, the Dynamic Reference algorithm commands a constant safe velocity while nullifying the force reference. Simulta-

neously, the Weight Scheduling completely deactivates the force tracking weight. Consequently, the actuator approaches the tissue smoothly, bounding its kinetic energy.

At the moment of impact, the Contact Detection algorithm triggers the state transition and the system switches to the **CONTACT** state (highlighted in light blue in the graphs). Several critical actions occur simultaneously: the spatial coordinate is locked to compute the relative penetration Δx , the RLS covariance is reset to identify the new tissue parameters, and the reference vector shifts from velocity to the 3 N force target. Crucially, the immediate intervention of the Weight Scheduling algorithm, which increases the control increment penalty ($w_{\Delta u}$) to provide active damping, absorbs the kinetic energy of the collision. As shown in the force plot, this orchestrates a truly bumpless transition: the interaction force rises smoothly towards the target without exhibiting the dangerous high-frequency bouncing effects typically associated with standard PID controllers upon impact.

6.5 Dynamic Trajectory Tracking in Continuous Contact

In real surgical applications, robotic tools rarely perform a single, static indentation. Procedures such as tissue palpation, dynamic organ retraction, or suturing require the end-effector to continuously modulate the applied force while maintaining uninterrupted contact with the biological structure. To evaluate the clinical applicability and the tracking robustness of the proposed architecture under these conditions, the system was subjected to a highly dynamic, time-varying force reference trajectory. The experimental result is depicted in Fig. 6.16. This test was still performed on the hard synthetic tissue.

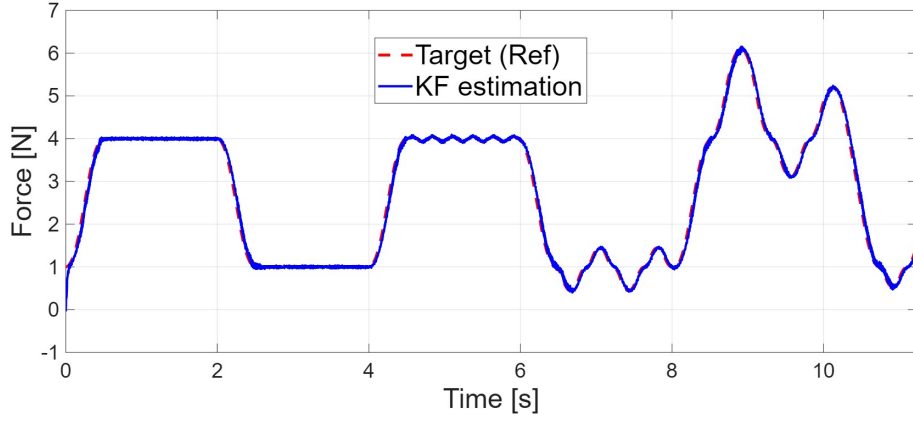


Figure 6.16: Experimental test of the AMPC subjected to a realistic, continuous time-varying force reference trajectory without breaking contact with the tissue.

This stress test demonstrates the absolute reliability of the Adaptive Model Predictive Control (AMPC) and its underlying Recursive Least Squares (RLS) estimator during prolonged dynamic operations.

Because the measured force never falls below the $F_{release}$ hysteresis threshold, the Finite State Machine (FSM) persistently maintains the system in the **CONTACT** state. Throughout this phase, the RLS estimator maintains a coherent and stable identification of the environmental parameters ($\hat{K}_{env}$ and $\hat{B}_{env}$), continuously feeding physically accurate data to the controller’s internal model without any computational degradation.

Furthermore, the predictive nature of the AMPC ensures that the tracking performance remains precise and virtually delay-free across ascending and descending force ramps. By continuously solving the Quadratic Programming (QP) optimization over the prediction horizon H_p , the controller smoothly anticipates the dynamic variations. The overall tracking error remains bounded and stable, completely avoiding actuator saturation, high-frequency oscillations, or unsafe force spikes. This continuous tracking accuracy confirms that the integrated AMPC architecture is fully capable of managing safe, sustained human-robot interactions in unstructured and dynamically changing surgical scenarios.

6.6 Experimental Validation on Complex Biological Tissues

To rigorously assess the translational potential and structural robustness of the proposed Adaptive MPC architecture, a final series of experiments was conducted

using real biological tissues. While synthetic phantoms provide a repeatable benchmark, genuine organic structures introduce complex, unmodeled physical phenomena such as non-linear viscoelasticity, stress relaxation, and multi-layer anisotropy. To evaluate the controller’s adaptability under these realistic clinical conditions, two distinct biological tissue configurations characterized by contrasting mechanical impedances and structural boundary conditions were investigated:

- **Layered Musculo-Adipose Tissue Specimen:** This configuration was selected to analyze the interaction with a highly compliant and dissipative environment. From a physical standpoint, this tissue features a pliable dermal surface backed by a significant subcutaneous adipose layer underlaid by a dense muscular mass. This multi-layered composition yields a highly non-linear elastic response and pronounced viscous damping properties.
- **Bone-Backed Dermal Tissue Specimen:** This setup was chosen to evaluate the control performance when interacting with a stiffer environment characterized by a hard underlying boundary. Mechanically, this specimen consists of a thin soft-tissue layer directly resting upon a rigid bony structure. This drastically limits the maximum allowable displacement and causes a rapid increase in reaction force upon compression.

The results of these experiments are shown in Fig. 6.17 and Fig. 6.18, while the performance indicators are summarized in Table 6.7.

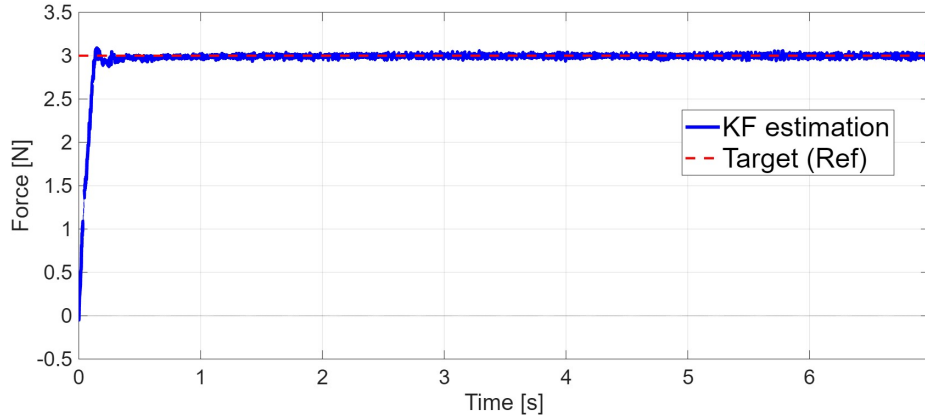


Figure 6.17: Experimental result on the layered musculo-adipose specimen.

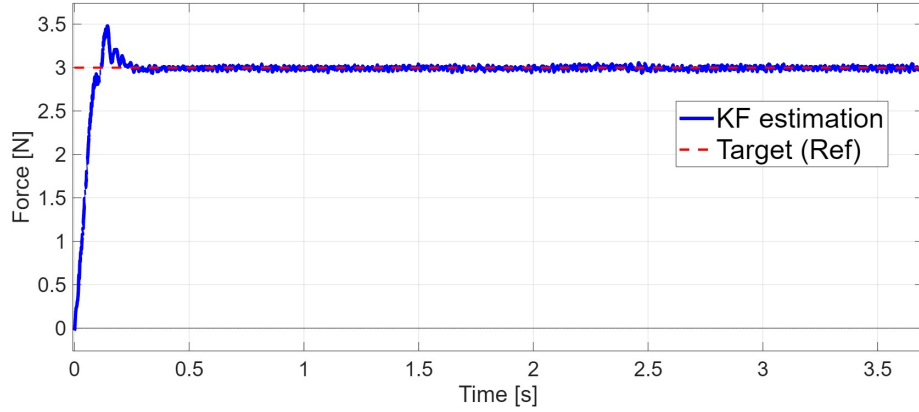


Figure 6.18: Experimental result on the bone-backed dermal specimen.

Material	Rise Time [s]	Settling Time (5%) [s]	Overshoot [%]
Layered Musculo-Adipose Specimen	0.100	0.124	4.07
Bone-Backed Dermal Specimen	0.069	0.189	16.06

Table 6.7: AMPC performances with biological tissues

Prior to these trials, the control increment penalty was strategically increased to $w_{\Delta u} = 600$. Physically, targeting an unmodeled, highly non-linear, and structurally unpredictable organic tissue demands a more conservative and safety-oriented control action. Raising $w_{\Delta u}$ injects a high level of artificial active damping into the closed-loop system, effectively bounding high-frequency control efforts. This preventive measure eliminates actuator chattering and structural force spikes that could otherwise induce tissue micro-trauma during the transient phase.

The system was commanded to execute a force step up to a safe target of 3 N. Experimental observations, summarized in Table 6.7, confirmed that the AMPC successfully adapted to the varying organic environments, although the distinct structural boundaries dictated significantly different transient responses. Despite the presence of complex viscoelasticity and unmodeled structural dynamics, the online RLS estimator continuously updated the estimated parameters ($\hat{K}_{env}$, $\hat{B}_{env}$) without experiencing numerical divergence in either scenario.

For the layered musculo-adipose specimen, the interaction force asymptotically approached the target with a highly damped, bumpless profile, yielding a

minimal overshoot of 4.07% and a fast settling time of 0.124 s. In this case, the high compliance and viscous dissipation of this particular tissue synergized with the active damping provided by the tuned $w_{\Delta u}$, resulting in an exceptionally smooth transient.

Conversely, the bone-backed dermal specimen exhibited a much faster rise time (0.069 s) but a noticeable overshoot of 16.06%, settling at 0.189 s. From a physical perspective, the rapid compression of the thin soft-tissue layer leads to an abrupt encounter with the rigid underlying bone impedance. This sudden structural stiffening causes the reaction force to spike faster than the physiological convergence time of the RLS estimator. However, crucially, the preventive tuning of $w_{\Delta u}$ effectively contained this transient: while the lag in parameter estimation caused a temporary force overshoot, the active damping absorbed the collision energy, preventing sustained high-frequency oscillations and ensuring a stable settling phase.

These results demonstrate that the integrated AMPC framework possesses the necessary flexibility and inherent safety features to operate robustly in unstructured, real-world clinical applications, successfully managing even highly stiff boundaries.

Chapter 7

Conclusions and Future Work

This concluding chapter summarizes the core contributions of this thesis, synthesizes the insights gained from the experimental validation, discusses the inherent limitations of the current implementation, and outlines promising directions for future research in the field of robotic surgery force control.

7.1 Summary of Contributions

The primary objective of this research was the development, implementation, and comprehensive evaluation of advanced Model Predictive Control (MPC) strategies for high-precision force regulation in surgical linear actuators. Traditional reactive control methods, such as the Proportional controller enhanced with a Disturbance Observer (P+DOB), struggle to balance fast transients with strict safety boundaries and often exhibit oscillatory behaviors during physical interactions. To overcome these limitations, this thesis established a rigorous predictive control framework tailored to the non-linearities and safety demands of robotic surgery.

The core contributions of this work can be summarized as follows:

- **Development of a Sensorless Force Observer:** A distinctive and foundational pillar of this architecture is the design of a Kalman Filter configured as an Unaccessible Input Observer (UIO). Operating as a high-performance state observer, this framework enables the accurate, real-time estimation of the interaction force exerted by the motor on the environment without requiring bulky or delicate force sensors at the distal tip of the actuator.
- **Formulation of Advanced MPC Frameworks:** Three distinct predictive control schemes were developed and deployed: a baseline Linear MPC

optimized for real-time efficiency, an offline-identified Nonlinear MPC (NL-MPC) embedding a cubic tissue stiffness model to capture environmental variations, and an Adaptive MPC (AMPC) driven by an online Recursive Least Squares (RLS) estimator to manage unknown environments without prior calibration.

- **Integrated Transition Management Feature:** To complement the core force tracking capabilities and enable practical deployment in unstructured clinical environments, a supervisory control system was integrated as a key feature. This system seamlessly orchestrates the switch from unconstrained motion to contact via an anticipatory Contact Detection logic driven by the filtered force derivative ($\dot{F}_{filt}$), a Dynamic Reference Generator, and a Weight Scheduling algorithm that dynamically adjusts the controller objective.

7.2 Synthesis of Experimental Findings

The empirical validation conducted on the real-time Linux RTAI platform confirmed that the advanced predictive strategies and the underlying observer infrastructure significantly outperform classical reactive control architectures.

Early experiments first validated the sensorless force observer infrastructure. The UIO Kalman Filter estimation demonstrated good steady-state accuracy and dynamic coherence when benchmarked against both a standard Disturbance Observer (DOB) and direct physical readings from a dedicated load cell, proving that high-fidelity force feedback can be reliably achieved purely through software-based sensorless estimation.

In static step-response trials, all MPC formulations demonstrated superior tracking compared to the P+DOB baseline, which suffered from a substantial 25.86% overshoot and persistent oscillations. The NL-MPC successfully exploited the cubic force-displacement relationship to halve the rise time of the linear baseline (0.041 s) while maintaining a minimal 5% overshoot.

The Adaptive MPC proved to be the most flexible and clinically promising architecture, successfully regulating interaction forces on both synthetic foams and multi-layer biological specimens without any prior environmental knowledge. The experimental data revealed a key biomechanical trade-off: while the highly compliant *Layered Musculo-Adipose Specimen* allowed an exceptionally smooth and well-damped transient (overshoot of 4.07%, settling time of 0.124 s), the rigid *Bone-Backed Dermal Specimen* induced a faster rise time (0.069 s) but a higher

overshoot of 16.06%. This transient peak was physically justified as an estimation lag of the RLS algorithm when encountering sudden structural stiffening, which was safely contained by the active damping of the tuned control penalty ($w_{\Delta u} = 650$).

Crucially, the integrated transition management system successfully demonstrated its efficacy as a robust operational feature. During approach-to-contact maneuvers, the supervisory logic seamlessly coordinated the shift from unconstrained motion to force regulation, eliminating high-frequency bouncing effects upon impact. Furthermore, a constraint-handling stress test using an unsafe 10.5 N reference validated that the MPC framework consistently prioritizes structural safety boundaries, successfully saturating the interaction force precisely at the 10 N limit.

7.3 System Limitations

Despite the excellent performance achieved, a rigorous scientific evaluation requires acknowledging the limitations of the current experimental setup, which open avenues for further improvements:

- **1-DoF Dimensionality:** The experimental validation was restricted to a single degree-of-freedom (1-DoF) linear actuator. While this setup successfully isolated the axial force dynamics and observer performance, real surgical interventions involve multi-DoF manipulators characterized by highly coupled multi-axis kinematics.
- **Sensitivity to Friction Model Mismatch:** The accuracy of the sensorless UIO Kalman Filter heavily relies on the precision of the feedforward friction compensation framework. Unmodeled localized variations in Coulomb or viscous friction parameters could introduce structural bias or estimation offsets in the force feedback loop.
- **RLS Estimation Lag:** The convergence delay of the standard RLS estimator is responsible for the temporary force overshoot observed during impacts with the bone-backed dermal specimen. In highly critical surgical environments, even a temporary model mismatch could lead to localized tissue trauma if the target structures are exceptionally delicate.

7.4 Future Work

Based on the insights and limitations identified in this work, future research directions could focus on the following developments:

- **Multi-Axis Systems:** Developing and validating the proposed Sensorless MPC framework on multi-DoF surgical manipulators.
- **Adaptive and Robust Friction Observers:** Integrating online friction identification schemes to work in parallel with the UIO Kalman Filter, ensuring that the sensorless force estimation remains immune to thermal drifts, mechanical wear, or structural variations in the actuator guides.
- **Intelligent and Fast Parameter Identification:** Integrating machine learning algorithms, such as shallow neural networks or physics-informed neural networks (PINNs), to work in tandem with or replace the RLS estimator. This approach aims to instantaneously predict tissue stiffness parameters upon the very first millisecond of contact, effectively reducing the estimation lag.
- **Advanced Viscoelastic Modeling within the MPC Horizon:** Incorporating higher-order tissue models, allowing the MPC to explicitly anticipate phenomena like stress relaxation and creep during continuous contact tracking.

Bibliography

- [1] Fadi Alyousef Almasalmah, Hassan Omran, Chao Liu, and Bernard Bayle. Adaptive robust model predictive control for bilateral teleoperation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7069–7074. IEEE, Oct 2023.
- [2] Smita De, Jacob Rosen, Aylon Dagan, Blake Hannaford, Paul Swanson, and Mika Sinanan. Assessment of tissue damage due to mechanical stresses. *The International Journal of Robotics Research*, 26(11-12):1159–1171, 2007.
- [3] C. Freschi, V. Ferrari, F. Melfi, M. Ferrari, F. Mosca, and A. Cuschieri. Technical review of the da vinci surgical telemanipulator. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 9(4):396–406, 2013.
- [4] Alejandro GJ González, Eduardo J Adam, and Jacinto L Marchetti. Offset-free model predictive control: A tutorial. *Industrial & Engineering Chemistry Research*, 47(24):10230–10244, 2008.
- [5] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.
- [6] Eric C. Kerrigan and Jan M. Maciejowski. Soft constraints and exact penalty functions in model predictive control. *Proc. UKACC International Conference (Control 2000)*, 2000, (2319-2327), 2000-09.
- [7] Yeisson Rivero-Moreno, Sophia Echevarria, Carlos Vidal-Valderrama, Luigi Pianetti, Jesus Cordova-Guilarte, Jhon Navarro-Gonzalez, Jessica Acevedo-Rodríguez, Gabriela Dorado-Avila, Luisa Osorio-Romero, Carmen Chavez-Campos, and Katheryn Acero-Alvarracín. Robotic surgery: A comprehensive review of the literature and current trends. *Cureus*, 15(7):e42370, 2023.

- [8] Kouhei Ohnishi Seiichiro Katsura, Yuichi Matsumoto. Modeling of force sensing and validation of disturbance observer for force control. *IEEE Transactions on industrial electronics*, 54(1), 2007.
- [9] Y. Wang, H. Liu, and J. Zhang. Force control of surgical robot with time delay using model predictive control. In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pages 421–428. SciTePress, 2018.
- [10] Maciej Ławryńczuk. *Nonlinear Predictive Control Using Wiener Models: Computationally Efficient Approaches for Polynomial and Neural Structures. Studies in Systems, Decision and Control*. Cham, 2022.