UNIVERSITÉ DE STRASBOURG

ÉCOLE DOCTORALE MSII

Laboratoire ICube (UMR 7357)

THÈSE présentée par

Fadi ALYOUSEF ALMASALMAH

Soutenue le : 12 juin 2025

Pour obtenir le grade de : Docteur de l'université de Strasbourg

Discipline/Spécialité : Robotique

Contributions à la Sécurité et à la Performance des Robots Téléopérés avec Retour d'Effort

"Contributions to Safety and Performance in Force-Feedback Telerobotics"

THÈSE dirigée par :

M. BAYLE Bernard	Prof., Université de Strasbourg

THÈSE encadrée par :

M. OMRAN Hassan	MdC, Université de Strasbourg
M. LIU Chao	CR CNRS (HDR), LIRMM, Université de Montpellier

RAPPORTEURS :

M. POISSON Gérard	Prof., Université d'Orléans
M. HABLY Ahmad	MdC (HDR), GIPSA-lab, Grenoble INP

EXAMINATEURS :

Mme DE MOMI Elena	Prof., NEARLAB, Politecnico di Milano
M. LAROCHE Edouard	Prof., Université de Strasbourg

Résumé

Cette thèse aborde les défis clés de la téléopération bilatérale pour des applications critiques telles que la chirurgie. Nous nous concentrons sur l'amélioration de la sécurité, de la transparence et de la robustesse en combinant la Commande Prédictive Basée sur Modèle (MPC), l'apprentissage et la préservation de la passivité. Nous introduisons d'abord un cadre MPC avec une méthode d'auto-réglage par Optimisation Bayésienne pour optimiser la transparence tout en respectant les contraintes de sécurité. Ensuite, nous développons un contrôleur adaptatif robuste de type Tube-MPC qui utilise l'apprentissage en ligne pour gérer l'incertitude du modèle et garantir le respect robuste des contraintes. Enfin, nous proposons une Commande à Impédance Variable (VIC) passive qui assure la stabilité via une optimisation en ligne. Les validations expérimentales et par simulation démontrent l'efficacité de nos approches pour une interaction homme-robot plus sûre. [1]

Mots-clefs : Téléopération Bilatérale, Chirurgie Robotique, Sécurité en Téléopération, Commande Prédictive Basée sur Modèle (MPC), Auto-réglage du MPC, MPC Adaptatif Robuste, Filtres de Passivité, Commande à Impédance Variable (VIC).

---

[1] N.B. : Un résumé détaillé en français est inclu dans l'annexe C de ce document.

Abstract

This thesis addresses key challenges in bilateral teleoperation for safety-critical applications like surgery. We focus on enhancing safety, transparency, and robustness by combining Model Predictive Control (MPC), learning-based adaptation, and passivity enforcement techniques. We first introduce an MPC framework with a Bayesian Optimization-based auto-tuning method to optimize transparency while enforcing safety constraints. Second, we develop an adaptive robust Tube-MPC controller that uses online learning to handle model uncertainty, guaranteeing robust constraint satisfaction. Third, we propose a passive Variable Impedance Control (VIC) scheme that ensures stability by integrating passivity filters with online optimization. Experimental and simulation-based validations demonstrate the robustness and effectiveness of the proposed approaches, contributing to safer and more efficient human-robot interaction.

Keywords: Bilateral Teleoperation, Robotic Surgery, Safety in Teleoperation, Model Predictive Control (MPC), MPC Auto-Tuning, Adaptive Robust MPC, Passivity Filters, Variable Impedance Control (VIC).

Acknowledgements

I would like to sincerely thank the members of the jury for evaluating my thesis. I am especially grateful to the rapporteurs, Prof. Gérard Poisson and Dr. Ahmad Hably, for their insightful reviews, and to the examiners, Prof. Elena De Momi and Prof. Edouard Laroche, for their valuable feedback and engaging discussion during the defense.

My deepest gratitude goes to my supervisors, Dr. Hassan Omran, Dr. Chao Liu, and Prof. Bernard Bayle, for giving me the freedom to follow what interests me, while providing their invaluable guidance, diverse perspectives, and lots of lessons, many of which went far beyond science.

I would also like to thank all the members of the RDH team for their support and the great working atmosphere. Special thanks to Maciej Bednarczyk, Thibault Poignonec, and Benoît Wach for their help with both hardware and software, and for teaching me many technical skills that directly contributed to this work.

To Ounay, Anne, Aude, Gui, Mahdi, Giorgia, Loïc, Thibault, Karine, and all the PhD students and interns of the RDH team: thank you for your friendship, your support, and for the countless boulangerie breakfasts and coffee breaks that made this journey so much more enjoyable. Special thanks to Dr. Jonathan Vappou for the nice discussions we shared and for the friendly rivalry on the badminton court. I am grateful for my long-time friends, Saeed, Subhy, Anas, Ghadeer, Abdul, Majd, and all the others who have been there for me throughout this journey. Your support has been invaluable.

I am deeply grateful to everyone who contributed to this journey in one way or another, especially my family, whose unconditional support has always been my foundation. Finally, to Dasha, who stood by me and helped me through the many challenges along the way, thank you!

Thank you all! I couldn't have done it alone.

Contents

Nomenclature

Acronyms

SMC: sliding mode control

TDPA: Time Domain Passivity Approach

PSPM: Passive-Set-Position-Modulation

MIS: Minimally Invasive Surgery

RMIS: Robotic Minimally Invasive Surgery

RAS: Robotic Assisted Surgery

VIC: Variable Impedance Control

LFN: Leader-Follower Network

P-P architecture: Position-Position architecture

F-P architecture: Force-Position architecture

MPC: Model Predictive Control

LTI: Linear Time-Invariant

DoF: Degree of Freedom

LMIs: Linear-Matrix-Inequalities

pHRI: physical Human-Robot Interaction

SP: Smith Predictor

MBP: Model-Based Predictors

GPC: Generalized Predictive Controller

SVD: Singular Value Decomposition

BO: Bayesian Optimization

GP: Gaussian Process

UCB: Upper Confidence Bound

HTMPC: Homothetic Tube-MPC

SML: Set Membership Learning

LMS: Least Mean Squares

sEMG: surface Electromyography

VMS: Virtual Mass-Spring

PFs: Passivity Filters

C-BO: Contextual Bayesian Optimization

Notations and Symbols

The following lists include the used mathematical notations and the main symbols used throughout the thesis. The symbols are grouped by chapters where they were first defined. In case of ambiguity, the symbol is defined clearly where it appears in the thesis. Some symbols might not be mentioned here if they were only used in one subsubsection.

Mathematical Notation

$[A]_i$: the i-th row of matrix A. If A is a vector, then $[A]_i$ is the i-th element of the vector.

$[A]_{i,j}$: the (i,j)-th element of matrix A.

1_n: a vector of ones of size n.

$\text{diag}(a_1, a_2, ..., a_n)$: a diagonal matrix with $a_1, a_2, ..., a_n$ on the diagonal.

I: identity matrix of appropriate size.

Symbols defined in Chapter 1

$x_l, \dot{x}_l, \ddot{x}_l$: the leader robot's position, velocity, and acceleration, respectively.

$x_f, \dot{x}_f, \ddot{x}_f$: the follower robot's position, velocity, and acceleration, respectively.

f_h, f_e: the human operator and the environment forces, respectively.

f_l, f_f: the leader and follower forces, respectively.

V_l, V_f: the leader and follower robot velocities in the Laplace domain, respectively.

F_h, F_e: the human operator and the environment forces in the Laplace domain, respectively.

Z_t, Z_e: the transmitted impedance to the human operator and the impedance of the environment, respectively.

$Z_{t,min}$: the minimal impedance that the teleoperation system can display to the human operator.

$Z_{t,width}$: the dynamic range of the transmitted impedance to the human operator that can be displayed by the teleoperation system.

Symbols defined in Chapter 2

t: continuous time.

τ: discrete time step.

k: discrete time step. Note that when τ and k are used together, τ means the time step of the actual system, while k is the time step of the predicted system. In other words, $x_k := x(\tau + k)$. Unless stated otherwise, both k and τ are used as subscripts to denote the discretization of the continuous-time variables.

T_s: sampling time

x, u, y: the continuous state, control input, and output of the system, respectively.

U, U^*: the sequence of inputs and the optimal sequence of inputs over the horizon of the MPC controller, respectively.

$f(\cdot)$: system discrete dynamics function, i.e., $x_{k+1} = f(x_k, u_k)$

$\mathcal{X}, \mathcal{U}$: polytopic sets that define the MPC state and control input constraints, respectively.

$\mathcal{X}_f$: the terminal set, i.e., a set that defines the terminal state constraints in the MPC controller.

N: MPC horizon.

$\mathcal{J}_s, \mathcal{J}_f, \mathcal{J}$: the stage cost function, the terminal cost function, and the cost function of the MPC controller, respectively.

$q, \dot{q}, \ddot{q}$: robot joint angle, velocity, and acceleration, respectively.

$H(q), C(q, \dot{q}), g(q)$: robot joint-space inertia matrix, Coriolis and centrifugal matrix, and gravity vector, respectively.

J: analytical Jacobian matrix.

τ_c: control torque of the robot.

f_{ext}: external force acting on the robot.

m_l, b_l: mass and damping coefficient of the leader robot, respectively.

m_f, b_f: mass and damping coefficient of the follower robot, respectively.

k_e, b_e: stiffness and damping coefficient of the environment, respectively.

A_c, B_c, C_c, D_c: continuous-time system matrices.

A, B, C, D, E, F: discrete-time system matrices.

z^{ideal}: performance vector used to evaluate the teleoperation system performance. It contains the matching error in position, velocity, and forces between the human operator and environment sides.

z: an alternative performance vector that approximates the human force using the leader robot torque.

n_x, n_u, n_y, n_z: the sizes of the state, control input, output, and performance vector, respectively.

Q, R, P: the matrices of the cost function of the MPC controller.

$H_{\mathcal{X}}, H_{\mathcal{U}}, H_{\mathcal{X}_f}$: matrices that define the polytopic constraints of the state, control input, and terminal set, respectively.

n_c, n_{cf}: number of constraints and terminal constraints, respectively.

1_n: a vector of ones of size n.

$\mathrm{diag}(a_1, a_2, \dots)$: diagonal matrix with $a_1, a_2, \dots$ on the diagonal.

$\mathcal{M}$: performance metric that is used to evaluate the controller performance for certain tuning parameters.

$\mathcal{M}^i$: the performance metric value for the i-th measurement.

$\hat{\mathcal{M}}$: a surrogate function of the performance metric, which is a model that approximates its value.

$\alpha(\cdot), \alpha_{\text{UCB}}(\cdot)$: a general acquisition function of the Bayesian optimization algorithm, and the upper confidence bound acquisition function, respectively.

$\mathcal{D}_n$: a data set that contains n measurements of the performance metric.

ϕ: a vector of tuning parameters.

Φ, a compact set where the BO searches for tuning parameters.

ϕ^i: the tuning parameters for the i-th experiment.

ϕ^*: the optimal tuning parameters that minimize the performance metric.

ϵ: noise in the performance metric evaluation.

$\mathcal{N}(\mu, \sigma^2)$: Gaussian distribution with mean μ and variance σ^2

$\mathcal{GP}$: Gaussian process

$\mu(\cdot)$: the mean function of the Gaussian process.

$\mu_n(\cdot)$: the mean value of the Gaussian process at $(\cdot)$ after n measurements.

$\sigma_n^2(\cdot)$: the variance of the Gaussian process at $(\cdot)$ after n measurements.

$\kappa(\cdot,\cdot)$: the kernel function of the Gaussian Process.

$k_{\text{GP},n}$: the kernel vector of the Gaussian process after n measurements.

$K_{\text{GP},n}$: the kernel matrix of the Gaussian process after n measurements.

Y_n: a vector of evaluations of the performance metric after n measurements.

β: exploration parameter in α_{UCB}. It balances the exploration and exploitation of the acquisition function.

$W_{\mathcal{M}}$: weight matrix for the performance metric ($\mathcal{M}$).

T_{exp}: number of time steps in the window of the performance metric evaluation.

n_{exp}: number of experiments to auto-tune the controller.

Symbols defined in Chapter 3

k_e^0, k_e^*: the initial guess and the true value of the environment stiffness, respectively.

b_e^0, b_e^*: the initial guess and the true value of the environment damping coefficient, respectively.

$\theta = [\Delta k_e, \Delta b_e]$: the uncertain parameters of the environment, which are the offset of the current estimated values from the nominal values of the environment stiffness and damping coefficient.

θ^*: the true value of the uncertain parameters θ.

d: the additive disturbance of the system.

$\mathcal{D}$: the additive uncertainty polytope, which is a set of all possible values of the additive disturbance d.

$H_{\mathcal{D}}, h_{\mathcal{D}}$: a matrix and a vector that define the additive uncertainty polytope $\mathcal{D}$.

A_θ, C_θ: the uncertain system matrices of the system, which are functions of the uncertain parameters θ.

A_0, C_0: nominal system matrices of the system, which are the system matrices for $\theta = 0$.

A_1, A_2, C_1, C_2: Constant matrices defining the affine dependence of A_θ and C_θ on θ.

$D_x(x_k)$: a function that quantifies the error in the model which comes from θ.

Θ: a polytope that bounds the environment uncertain parameter θ.

$\Theta_0, \Theta_\tau, \Theta_\tau^{HC}$: in the context of SML, Θ_0 is the initial uncertainty bounding set, Θ_τ is the learned uncertainty bounding set at time step τ, and Θ_τ^{HC} is the hypercubic approximation of Θ_τ, respectively.

η_τ: the length of the side of Θ_τ^{HC}.

$\bar{\theta}$: the nominal value of θ, i.e., the center of Θ.

$\hat{\theta}$: the point estimate of θ based on the LMS.

n_θ: size of parameters θ.

$\bar{x}_k, \hat{x}_k$: the predicted state based on the nominal model (using $\bar{\theta}$) and the estimated model (using $\hat{\theta}$), respectively.

$\mathcal{P}_0$: the initial cross-section of the tube, which is a polytope in the state space.

$\mathcal{P}_k$: a polytope in the state space that represents the tube section at time step k.

$n_{\mathcal{P}_0}$: the number of inequalities that define the initial cross-section of the tube $\mathcal{P}_0$.

s_k: the tube scaling factor at time step k.

K: a pre-stabilizing controller.

$A_{cl,\theta}$: the dynamic matrix of the closed-loop system with the controller K and the uncertainty θ.

$\rho(\Theta)$: the maximum tube contraction rate for a system with uncertainty bounded by Θ.

ρ_θ: the contraction rate of the tube for a system with uncertainty θ.

$w_k(\Theta, \mathcal{D})$: a parameter that captures growth of tube scaling factor s_k in the context of Tube-MPC.

$f_w(\cdot)$: a function that describes the dynamics of w_k.

W, W^*: the trajectory of w_k and its optimal solution over the prediction horizon.

$\mathbb{B}_n$: the unit hypercube in n-dimensional space.

v_k: an additional control input at time step k calculated by the MPC on top of the pre-stabilizing controller K.

V, V^*: the sequence of the additional control inputs and the optimal sequence of additional control inputs over the horizon of the MPC controller, respectively.

c_j: offline constants used for tightening the constraints, with $j \in \{1, \dots, n_c\}$.

Δ_τ: the set of all possible values of θ that can explain the system state transition from $x_{\tau-1}$ to x_τ.

$H_{\mathcal{P}_0}, h_{\mathcal{P}_0}$: matrix and vector that define the H-representation of polytope $\mathcal{P}_0$.

$H_{\mathcal{D}}, h_{\mathcal{D}}$: matrix and vector that define the H-representation of polytope $\mathcal{D}$.

$H_{\Delta_\tau}, h_{\Delta_\tau}$: matrix and vector that define the H-representation of polytope Δ_τ.

H_Θ: matrix that defines the H-representation of polytope Θ.

π_f: terminal controller in the MPC controller.

Symbols defined in Chapter 4

H_l, C_l, g_l: the leader robot's task-space inertia matrix, Coriolis and centrifugal matrix, and gravity vector, respectively.

H_f, C_f, g_f: the follower robot's task-space inertia matrix, Coriolis and centrifugal matrix, and gravity vector, respectively.

n, m: the sizes of the robot joint space and task space, respectively.

M_l, D_l, K_l: the imposed matrices of impedance by the VIC controller, which are the imposed inertia, damping, and stiffness matrices of the leader robot, respectively.

M_f, D_f, K_f: the imposed inertia, damping, and stiffness matrices of the follower robot by the VIC controller.

M_l^d, D_l^d, K_l^d: the desired leader impedance matrices by the user.

M_f^d, D_f^d, K_f^d: the desired follower impedance matrices by the user.

x_l^r, x_f^r: reference positions for the leader VIC and follower VIC, respectively.

$\mathcal{V}$: the storage function of the system.

$\mathcal{V}_l, \mathcal{V}_f$: the storage functions of the leader and follower VIC controllers, respectively.

$\omega(t)$: the dissipated power in the system at time t.

$\omega_l(t)$: the dissipated power in the leader VIC controller at time t.

$\omega_f(t)$: the dissipated power in the follower VIC controller at time t.

e_l, e_f: the position errors of the leader and follower VIC controllers from their corresponding references, respectively.

β_i: the parameters of the passivity optimization problem, which control the technique of the energy dissipation ($i = 1, \dots, 5$).

$\beta_{max,j}$: the maximum value of β_j for $j = 1, 2, 3, 4, 5$.

p_j: cost function weights for the passivity optimization problem.

ε: slack variable in the passivity optimization problem.

$\tilde{x}_l^r$: the error between the follower position and the leader reference position, i.e., $\tilde{x}_l^r = x_f - x_l^r$.

$\tilde{x}_f^r$: the error between the leader position and the follower reference position, i.e., $\tilde{x}_f^r = x_l - x_f^r$.

$\tilde{K}_l^d$: the error between the desired leader stiffness and the imposed leader stiffness, i.e., $\tilde{K}_l^d = K_l^d - K_l$.

$\tilde{K}_f^d$: the error between the desired follower stiffness and the imposed follower stiffness, i.e., $\tilde{K}_f^d = K_f^d - K_f$.

λ_i: the coefficients of β_i in $\omega(t)$.

Chap. 1

Introduction

1.1 Bilateral Teleoperation

1.1.1 The Concept of Teleoperation and Haptic Feedback

Since the early days of robotics, researchers have aspired to create fully autonomous machines capable of independent decision-making and complex tasks. However, the numerous challenges associated with achieving full autonomy have restricted the practical use of robots in many real-world applications. In response to these challenges, a hybrid approach emerged called *teleoperation*, which combines the computational and mechanical advantages of robots with the perceptual and cognitive skills of humans. In a teleoperation system, a human remotely controls a robot via a local interface, typically supplemented by real-time video feedback from the remote environment. As shown in Figure 1.1, a typical teleoperation setup consists of five components: a human operator, a local control device manipulated directly by the operator, a communication channel that transmits commands and sensory data, a remote robot that executes the desired task, and the environment [1]. This approach extends the human's ability to perform a wide range of tasks, such as interacting with far, inaccessible, or dangerous environments, or simply augmenting the human capacity with the precision, dexterity, force, and adaptability of robots. Teleoperation has been used for many applications such as space and ocean exploration, handling dangerous and radioactive materials, and minimally invasive surgery.

While video feedback is necessary for the human operator to be able to interact with the environment, it does not always transfer the full information needed to perform the

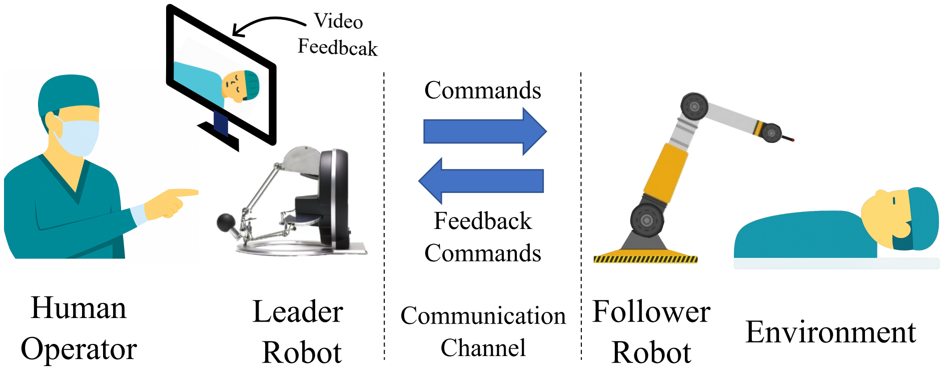

Figure 1.1: An illustration of a bilateral teleoperation system in surgery.

task safely and accurately. The absence of the touch and force feeling means that the human operator lacks a critical sensory channel used in natural interactions. This kind of information is usually referred to as *haptic feedback*. When touching an object, the human receives several types of haptic feedback. Specifically, the tips of the fingers are equipped with mechanoreceptors located on the skin, which are able to distinguish the texture, temperature, and pressure distribution on the touched object. This is called *cutaneous* or *tactile* haptic feedback [2]. Even when the tips of the fingers are not in direct contact with the environment, another important component of haptic feedback is still transferred to the human through the receptors in the muscles and tendons which sense the forces applied to the hand. This is called *kinesthetic* haptic feedback [3].

Teleoperation systems can render kinesthetic feedback to the human operator using an active local device, which can be a robot that reflects the interaction forces and position of the remote robot, in addition to capturing the operator commands. Such teleoperation systems are called *bilateral* teleoperation systems, since the commands are sent in both directions through the communication channel, compared to unilateral teleoperation systems that only transmit motion and/or force commands from the human to the teleoperated robot. In this thesis, we focus on kinesthetic (force) feedback, and we refer to it as the force feedback.

It is important to note that the nomenclature for teleoperation systems has evolved over time. In the literature, the robots used to be called master/slave robots [1], referring to the robots on the human/environment sides, respectively. This naming convention is being abandoned [4]. Other names include local and remote robots, operator and remote robots [5], and leader and follower robots [6]. We use the latter naming convention throughout the thesis. In addition, many other terms are used nearly synonymously with teleoperation, in particular telerobotics and telemanipulation. It should also be noted that the distance between the robots could vary considerably, ranging from several meters in surgical teleoperation, to thousands of kilometers in deep ocean teleoperation, and potentially millions of kilometers in space teleoperation.

Most of the work presented in this thesis can be applied to improving the safety and performance of a general bilateral teleoperation system. Although the contributions apply generally, the thesis places special focus on surgical teleoperation, motivated by its context within the Labex-CAMI (Computer Assisted Medical Interventions) project. In the following subsection, we present a historical overview of teleoperation systems.

1.1.2 Historical Overview

The development of teleoperated robots can be broadly divided into several eras, each marked by significant technical and conceptual breakthroughs. Telerobotics is one of the first applied fields in robotics. By the late 1940s and early 1950s, the first examples of telerobotics were developed in order to handle hazardous and radioactive material while protecting the human operator [7]. The first systems were electrical, controlled by an array of on-off switches to activate various motors and move various axes [8], which was not intuitive for the human to control. Later, a mechanically coupled leader-follower system was developed [9], which transmitted force and motion directly through physical linkages, ensuring that the operator could feel the interaction forces even though the system was relatively rigid and limited in operational distance (see Figure 1.2). In 1954, the first electrically-coupled teleoperation system was developed by Goertz and Thompson [10, 11], which laid the foundations for modern telerobotics and showed the significance of force feedback.

In the 1960s, the time-delay effect started being addressed by researchers [13], and supervisory control was introduced as a way to enhance system stability, where the operator gives relatively high-level commands, such as intermediate trajectory points, to be refined and executed by the local control loop of the follower robot [13, 14]. A major transformation occurred in the late 1980s and early 1990s when advances in control theory led to systematic approaches for ensuring stability in teleoperation systems. Pioneering works by Raju et al. (1989) [15], Anderson and Spong (1989) [16], and Niemeyer and Slotine (1991) [17] introduced methods based on Lyapunov stability, passivity, scattering theory, and wave variables. These techniques were specifically developed to counteract the destabilizing effects of uncertainties and time delays and

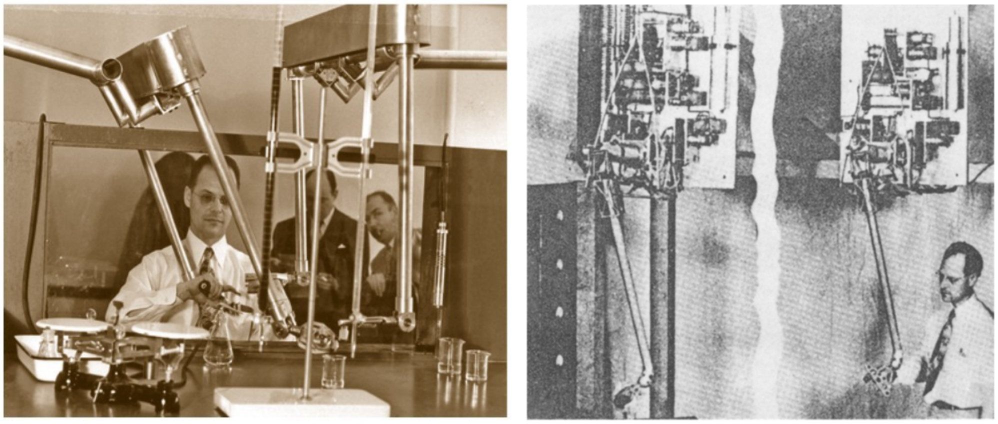

Figure 1.2: Raymond C. Goertz used mechanical (left) and electrical (right) telerobots in the early 1950s to handle radioactive material. The left photo is from [7], and the right photo is from [12].

have since become central to the design of robust bilateral teleoperation controllers.

Parallel to theoretical advances, significant hardware innovations pushed the field forward. In 1982, the Central Research Laboratory model M2 [7] became one of the first systems to separate leader and follower electronics while providing force feedback, thereby expanding the practical application of teleoperation in domains such as nuclear and industrial environments. Concurrently, research groups in France (e.g., Vertut and Coiffet, 1985) [18] and at the Jet Propulsion Laboratory [19] developed bilateral servo-manipulators tailored for specific applications, including space exploration. The 1990s witnessed teleoperation branching into new application areas. In 1993, the first telerobotic system was deployed in space during the Spacelab Mission D2 (ROTEX experiment), demonstrating remote manipulation under significant communication delays. Later, a breakthrough in telesurgery was demonstrated by what was called the Lindberg Operation in 2001 [20], where the surgeon was in New York (US), and the patient was in Strasbourg (France). Despite not having haptic force feedback, this operation underscored teleoperation's potential in performing delicate medical procedures even at a distance.

Over the last two decades, numerous variants of the previous control methods were developed to address the stability issues, improve the operator's perceived fidelity of interaction, and balance the tradeoff between them. For instance, new variations of the passivity-based methods were proposed such as Time Domain Passivity Approach (TDPA) [21], Passive-Set-Position-Modulation (PSPM) [22], and Energy Tanks [23,24], in addition to other methods that deal with non-passive environments [25–27]. Novel control paradigms, such as model-mediated control [28] and shared control strategies [29], have emerged to optimize the operator's experience by dynamically adapting system behavior to the task at hand. These approaches, combined with robust control techniques and enhanced computational capabilities, have broadened the range of teleoperation applications to include undersea exploration, agriculture, and assistive robotics. Nowadays, one of the main active research areas pushing the teleoperation field to its limit is surgical teleoperation, where the system should be able to perform very precise tasks with accurate haptic feedback while ensuring the safety of the patient. In the following subsection, we will provide an overview of the developments that happened in the surgical teleoperation field specifically.

1.1.3 Teleoperation in Surgical Robots

The evolution of teleoperated surgical systems began with pioneering platforms designed to enhance surgeon capability and patient safety. In the early 1990s, AESOP emerged as the first teleoperated robot for surgery, which was an endoscope holder operated by voice commands. Approved by the FDA in 1994, AESOP achieved significant commercial success by enabling hands-free control of the endoscope in minimally invasive procedures [30]. Building on AESOP's concept, Computer Motion developed the Zeus system. Zeus integrated three AESOP-like arms, with two teleoperated via a leader console and a third controlled by voice. The Zeus system was used in the Lindberg operation in 2001 to demonstrate the feasibility of long-distance teleoperation. Despite its innovative architecture, Zeus did not attain the same market penetration as AESOP until the company was acquired by Intuitive Surgical [30].

In 2000, Intuitive Surgical received FDA clearance for the Da Vinci system, which quickly became the benchmark for Robotic Minimally Invasive Surgery (RMIS) (see Figure 1.3). The Da Vinci system offered key advantages, including high-definition three-dimensional visualization, increased dexterity, operator tremor filtering, and variable motion scaling between the leader and the follower robots. However, a widely noted limitation of the initial versions of the Da Vinci robot was the absence of haptic force feedback, which was mentioned as a shortfall in a survey conducted by the FDA [31]. The Da Vinci system dominated the RMIS market for over two decades, with more than 6500 units installed by 2020, and more than 10 million operations performed worldwide [32]. A review paper published in November 2015 stated that the Da Vinci system was the only general RMIS system available in the market back then [33].

Figure 1.3: The Da Vinci surgical system (© 2011 Intuitive Surgical)

However, since then a large number of teleoperated surgical systems have been emerging, with some of them being specialized for certain operations, such as the NeuroArm system for neurosurgery [34], and others trying to compete, for example, by reducing the cost, reducing the size, or by offering haptic feedback [35].

While haptic force feedback has been tested in many research-oriented telesurgical robots, its integration into commercial surgical robots has been a long and slow process. Notably, the Senhance Surgical System, developed by Asensus Surgical (formerly TransEnterix), integrates haptic feedback along with a custom master device and eye-tracking for independent control of slave instruments. Launched in 2016 and FDA-cleared in 2017, Senhance has been applied in laparoscopic gynecological and colorectal procedures [36]. Meanwhile, systems like Versius from CMR Surgical have been designed to provide some form of haptic feedback. However, the initial clinical trials used the non-haptic version of the system [6]. A Hong Kong-based startup, NISI, introduced the Novel Surgical Robotic System, a miniature robot designed for natural orifice surgery that incorporates haptic feedback for precision control [37]. The mentioned systems are just a few examples of the many systems that have been emerging since the last decade. Other examples can be found in [6, 35, 37–39].

More recently in 2023, the Saroa robot was reported to be the first surgical robot used clinically to integrate a dedicated haptic feedback mechanism [40]. Saroa, being pneumatically driven, measures the pressure applied to the entire forceps at the robotic arm side, specifically providing force feedback for gripping actions. In parallel, Intuitive Surgical has continued to evolve its flagship platform. The Da Vinci 5, released in 2024, integrates instruments with built-in force sensors, enabling the surgeon to directly sense the forces applied at the instrument tips [41]. This advancement marks a significant

Figure 1.4: (left) A manual minimally invasive surgery (stock image); (right) a robotic minimally invasive surgery (© 2011 Intuitive Surgical). The figure shows the difference in the ergonomics of the surgeon's hands.

step forward in the field considering the scale and popularity of the Da Vinci robots. After presenting a brief overview of the existing telesurgical robots, we will discuss the advantages of these robots in the next subsection, with a special focus on the importance of haptic feedback.

1.1.4 The Advantage of Telesurgery and Force Feedback

One of the main fields that are being revolutionized by telesurgery is Minimally Invasive Surgery (MIS). MIS is performed through small incisions using long, slender instruments and an endoscopic camera. Compared to traditional open surgery, MIS has many advantages, such as less tissue trauma, less blood loss, less postoperative infection, and faster recovery, in addition to better aesthetics [42]. However, MIS comes with its challenges as well, where the surgeon has to operate using a long tool that is constrained by the fixed incision point, which results in a non-intuitive mirroring effect, and the motion required is non-ergonomic for the surgeon, in addition to very limited or distorted feeling of the forces applied on the tissues. Teleoperated robots are very promising for solving these challenges, such that they can remove the mirroring effect, and render the required motion back to an ergonomic one (see Figure 1.4). However, unilateral teleoperated robots result in a loss of haptic feedback.

The lack of haptic feedback in a teleoperated system forces the surgeon to rely primarily on visual cues, such as the deformation of tissue, to estimate the forces.

The likely outcome of misreading these cues is torn tissue or suture breakage [43, 44]. Moreover, the value of haptic feedback in robotic surgery becomes even more important when visual cues are compromised. For instance, when the camera's view is obstructed by fluids or smoke generated by electrosurgical tools. Although experienced surgeons get better at controlling the amount of force on the tissues by depending on the visual feedback alone, they may still unintentionally damage sutures or delicate tissues [45].

The works in [46] and [47] reported that unintentional injuries were reduced in RMIS when appropriate force feedback was available. They reported however that the operating time increased significantly in RMIS compared to manual intervention, and hypothesized that this is due to slow tracking in the position control loop. A telerobotic needle insertion study [48] showed that the error in the detection of transition between tissue layers was reduced by 55% with the force feedback. Another possible use case for the haptic feedback is to give the surgeon clues about collisions that happen between the tool and the tissues outside the camera view, allowing the surgeon to adjust the positions of the tool [30]. Review of studies that discuss the effect of haptic feedback on surgical robots can be found in [6, 33], and a meta-analysis of these studies can be found in [49].

Regarding the recent commercial surgical robots that integrated haptics force feedback, initial studies have been recently published. Ueda et al. [40] did several experiments to evaluate the effect of force feedback with the Saroa robot. In one of the experiments, 6 physicians were asked to move 20 grains of puffed rice from one tray to another as quickly as possible (Figure 1.5), while the haptic feedback was turned on and off blindly. Overall, the authors reported that the grasping force was significantly lower with the haptics on (mean value: 0.63 N), while without haptics (mean value: 2.14 N). In addition, the completion time was also reduced, where the mean time with the haptics was (mean value: 127.8 s), and without haptics was (mean value: 153 s). The rice grains were then scanned by electron microscopy for comparison, and the results showed that the surface of the rice had almost no damage when force feedback was on, while noticeable damage happened when it was off. Figure 1.5 shows the results of these scans. Clinical case studies on the Saroa robot were reported by Ueda et al. [50] and by Iwatani et al. [51], both of which showed similar results concerning the decreased grasping force when the haptic feedback was on.

A pre-clinical study with the Da Vinci 5 robot was presented in [52], where twenty-nine novice surgeons were randomized into two groups. Both groups used the Da Vinci 5 robot, but one with force feedback while the other without. Participants performed

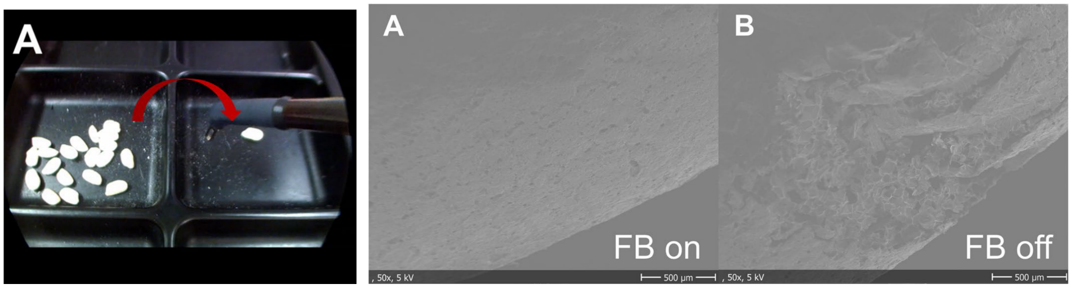

Figure 1.5: The puffed rice experiment with the Saroa robot. (left) The rice grains were moved from one tray to another. (right) The electron microscope scans of the rice grains after the experiment, where (A) is the rice that was moved with haptic feedback on, and (B) is the rice moved with haptic feedback off. (Figure is taken from [40]).

several types of stitches on ex-vivo porcine bladder and porcine aorta. Force feedback significantly lowered the mean force applied, the average number of errors, and the completion time. The force feedback group caused less tissue trauma with a higher skill score. Other initial studies about the Da Vinci 5 robot can be found in [41, 53]. It is worth mentioning that the above-cited studies on the new commercial robots are still preliminary, and further research has to be carried out to evaluate the effect of force feedback on the surgical outcome in the long term.

1.2 Core Concepts in Bilateral Teleoperation

In this section, we introduce several core concepts of bilateral teleoperation, which include system modeling, performance, stability, and communication delay. These topics are essential for understanding the control design of the system in this thesis. In Chapters 2 and 3, we assume working with simple robots that have one Degree of Freedom (DoF) each. This assumption is justified by the robotics background provided in Appendix A, which demonstrates how a complex multi-DoF robot can be approximated by a set of decoupled 1-DoF models, for which a separate controller can be designed. This assumption simplifies the analysis while preserving the essential characteristics needed for teleoperation control.

1.2.1 System Modeling

A bilateral teleoperation system consists of five components: a human operator, a leader robot, a communication channel, a follower robot, and an environment (Figure 1.6). It is common to represent such a system as three interacting subsystems which are illustrated in Figure 1.7. Each one of the operator and environment subsystems is represented by a one-port network, while the leader, the communication channel, and the follower are lumped into a two-port network called the Leader-Follower Network (LFN). In this context, a port refers to an interface where energy is exchanged between two networks. Specifically, each one-port network is characterized by a single pair of conjugate variables (for example, force and velocity). In Figures 1.6 and 1.7, the signals transmitted through the system are shown, where the symbols V and F denote the velocities and forces in Laplace domain, respectively. The subscripts h, l, f, and e correspond to the human operator, the leader robot, the follower robot, and the environment, respectively.

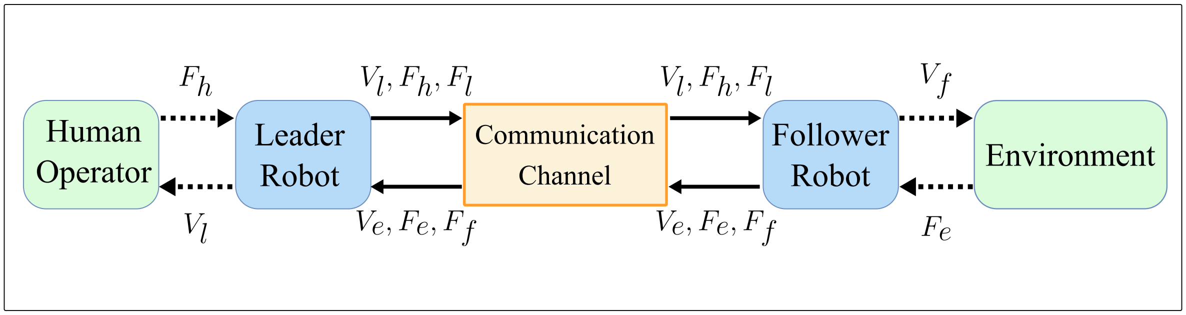

Figure 1.6: General schematic of a bilateral teleoperation system. The dashed arrows mean a physical interaction, while the solid arrows mean a signal transmission.

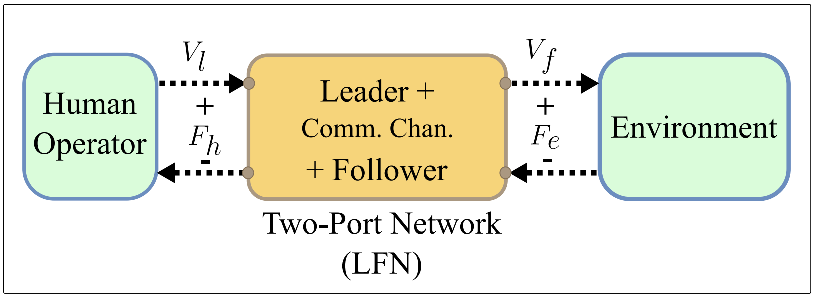

Figure 1.7: Network model of a teleoperation system.

Several bilateral control architectures exist in the literature. These architectures can be categorized based on the number and the type of signals transmitted from one side of the teleoperation system to the other. In the general case, both force and velocity signals are exchanged in both directions. This is known as a four-channel control architecture [54]. However, other architectures are possible, such as three-channel or two-channel configurations [1]. The configuration type also depends on the type and direction of exchanged signals. For example, if only position signals are exchanged, the configuration is called a two-channel Position-Position (P-P) configuration. Another very common example is the Force-Position (F-P) configuration, where the position of the leader robot is sent to the follower robot, while the force of the follower robot is sent back to the leader robot and displayed to the operator. The choice of the configuration depends on the application and the requirements of the task, in addition to the availability of the sensors. Regardless of the control architecture used, the operator closes the perception-action loop primarily through visual feedback (although it is not depicted in Figure 1.6).

1.2.2 Stability

The stability of a teleoperation system is the primary objective of the controller design. It is defined as the boundedness of the system response to a bounded excitation or to non-zero initial conditions [15]. One of the reasons that make the stability analysis complicated is that the teleoperation systems have to deal with the nonlinear, unpredictable, and time-varying human operator model. Moreover, the follower robot often interacts with various (possibly unknown and time-varying) environments. Communication channel issues such as time delay and loss of packets are other factors that worsen the stability of the system. These challenges have motivated extensive research over the past decades to come up with various methods to guarantee stability while minimizing the effect on performance.

If the model of the system including the human and the environment is Linear Time-Invariant (LTI), one can use linear methods such as Nyquist [1,54,55], root locus method [56], or absolute stability for linear controllers [1]. Robust controllers have also been applied to guarantee the stability while accounting for the uncertainty in the system. For example, μ-synthesis approach was applied in [57,58] to take into account the environment uncertainty and time delay, and a method based on Linear-Matrix-Inequalities (LMIs) was proposed in [59] to design a robust controller to bounded but arbitrarily fast time-varying parametric uncertainties.

For a general linear or nonlinear teleoperation system, Lyapunov methods and passivity-based methods are commonly used in the literature. Specifically, passivity has been extensively considered to ensure stability for teleoperation systems [60]. Passivity is a sufficient condition for stability [1]. Passive systems can be intuitively defined as systems that do not generate energy, but rather store or dissipate it. A desirable feature in passivity-related methods is that they generally focus only on the energy exchanged by the system through its ports with other external systems, which often simplifies the implementation of these controllers.

Two very important rules make passivity particularly suitable for bilateral teleoperation: "First, two passive systems can be combined to form a new passive system. Second, the feedback connection of two passive systems is stable" [7]. Therefore, if the passivity of the LFN is guaranteed, then the interaction with any passive operator and passive environment is stable. This condition covers a very wide range of environments that do not generate energy or include active elements such as actuators or motors. In addition, in most scenarios, the human operator is considered a stabilizing element rather than a destabilizing one, especially in the frequency range of interest for haptics, where the behavior of the human operator can be assumed to be passive [61].

While passivity is very convenient for stabilizing bilateral teleoperation systems, it is a conservative method [1]. Also, in some cases, the environment cannot be considered a passive element, such as in beating-heart surgery, or if the follower robot is interacting with a human as in telerehabilitation robots. To reduce this conservatism, several methods were proposed. For example, a measure called the excess of passivity was estimated by measuring the human grasping force. It is then used to absorb some of the energy generated by the non-passive environment which relaxes the passivity condition [25, 26]. Also, small-gain theorem approaches were proposed for dealing with non-passive environments [27, 62, 63].

In this thesis, we will only consider a passive operator and environment. In Chapter 4, we will design a passivity-based method for a special class of controllers for bilateral teleoperation, specifically, Variable Impedance Control (VIC).

1.2.3 Performance

Quantitatively assessing the performance of a teleoperation system is challenging but essential, particularly for surgical applications, where submillimeter accuracy, real-time force feedback, and strict safety standards must all be met. In the case of surgical teleoperated robots, performance metrics can be divided into objective and subjective measures. Objective metrics include statistical outcomes such as error rates (e.g., incidences of tissue damage or suture breakage), operating time, and postoperative complications [51, 52, 64]. Subjective metrics often involve evaluations of the operator's mental workload and ergonomic comfort using established instruments (e.g., the NASA-TLX) [65]. However, these high-level measures can be difficult to interpret and may not directly translate into actionable design changes, which is why more detailed performance metrics are needed.

An alternative perspective that is directly related to teleoperation focuses on achieving *telepresence*, which refers to the operator's sense of being fully immersed and directly interacting with the remote environment. This concept is closely related to *transparency*, which, in bilateral teleoperation, means that the technical medium between the operator and environment is not felt. In other words, the dynamics of the leader and follower robots and the communication channel are effectively canceled out. A transparent system is one in which the following conditions are maintained in the time domain

$$f_h = f_e \quad \text{and} \quad \dot{x}_l = \dot{x}_f \tag{1.1}$$

where f_h and $\dot{x}_l$ represent the force and velocity at the human (operator) side, and f_e and $\dot{x}_f$ represent those at the environment (follower) side [54].

In the case of LTI systems, transparency can be analyzed in the frequency domain using impedance matching. The transmitted impedance to the human operator is defined as

$$Z_t(s) = \frac{F_h(s)}{V_l(s)}$$

while the environment impedance is given by

$$Z_e(s) = \frac{F_e(s)}{V_f(s)}$$

with s representing the Laplace variable, and F_h, V_l, F_e, and V_f being the Laplace transforms of the operator force and velocity, and environment force and velocity, respectively. The transparency condition is met when [54]

$$Z_t(s) = Z_e(s) \quad \text{and} \quad V_l(s) = V_f(s) \tag{1.2}$$

A common performance metric includes calculating the two following values, first $Z_{t,\min}$, defined as

$$Z_{t,\min}(s) = Z_t(s)\big|_{Z_e=0}$$

which corresponds to the minimal impedance perceived by the operator when the follower is not in contact with any environment, and second, $Z_{t,\text{width}}$, defined as

$$Z_{t,\text{width}}(s) = Z_t(s)\big|_{Z_e\to\infty} - Z_{t,\min}(s)$$

which represents the dynamic range of the transmitted impedance that can be achieved by the teleoperation system. An ideal bilateral teleoperation system would satisfy

$$|Z_{t,\min}(s)| \to 0 \quad \text{and} \quad \frac{1}{|Z_{t,\text{width}}(s)|} \to 0 \tag{1.3}$$

Alternative frequency-domain measures include the sensitivity of transmitted impedance to changes in environment impedance [66] and the maneuverability index defined in [67], which evaluates how closely the actual system responses match the defined ideal responses by integrating the error in the frequency domain.

For nonlinear systems or for online optimization of transparency based on a time-varying model, time-domain measures can be used. For instance, the weighted integral of matching errors in positions, velocities, and forces can be used [5, 68], where it is usually evaluated on a finite-time window. This measure is derived directly from the definition of transparency in (1.1). In this thesis, we primarily adopt this type of time-domain measure of transparency, which is discussed in detail in Chapters 2 and 3.

It is worth noting that there exists a tradeoff between stability and transparency as has been shown by many works [1, 54, 69]. This means that achieving perfect transparency may lead to instability in practice. This tradeoff motivates the design of controllers that can guarantee stability while optimizing for transparency, which is one of the reasons why we will use optimization-based methods in the controller design in this thesis. Finally, in some applications, scaled transparency is desired. For instance, in construction tasks, scaling down the transmitted environmental forces and velocities can reduce operator fatigue [54], whereas in microsurgery, scaling up these signals can enhance the operator's perception.

1.2.4 Communication Time Delay

Communication time delay is one of the most studied issues in the literature on bilateral teleoperation due to its severe impact on the stability and transparency of the system [14, 70, 71]. Depending on the specific teleoperation system, the time delay can range from a few microseconds or milliseconds to several minutes for outer-space teleoperation (e.g., on Mars).

As we mentioned before, the main focus of this thesis is teleoperated robots in the context of surgery. The majority of currently available teleoperated surgical robots are designed for operations where the surgeon is present in the same room as the patient, leveraging the benefits of teleoperation aside from long-distance telesurgery. Despite the absence of significant communication delay in current surgical systems, ensuring safety and high performance remains a critical research challenge due to interactions with delicate tissues and the need for precise haptic feedback. This highlights that improving the safety and performance of non-delayed teleoperated robots remains an active research area. Therefore, throughout this thesis, we focus on non-delayed bilateral teleoperation, assuming that the communication channel is sufficiently fast such that the delay is negligible.

1.3 Safety in Bilateral Teleoperation

Safety is a very wide concept that encompasses many factors, such as mechanical, electrical, high-level awareness, and control aspects. In this thesis, we focus on developing robust and safe controllers for bilateral teleoperation systems. Safety in teleoperation is closely related to safety in physical Human-Robot Interaction (pHRI), which is a very active area of research with more established safety standards. This is why, in the following, we draw inspiration from pHRI to define some of the safety directions for our work.

1.3.1 Safety Concept for Physical Human-Robot Interaction

The safety in pHRI can be regarded from many points of view. For example, it could be either an added component to the controller design such as emergency backup plans, or it could be an intrinsic aspect of the controller design itself, for instance, stability. Another classification is presented in a recent survey on safe and ergonomic human-robot interaction control [72]. In this survey, the papers related to safety were categorized as: collision avoidance, collision detection, motion planning, and safety-oriented control system design.

By comparison to this classification, in teleoperation, the first three aspects are mainly handled by the human operator, assuming the operator is well aware of the environment to detect the collision, and is able to control the follower robot accurately to avoid the collision and plan the motion. This makes high transparency — which enables the operator to better perceive the remote environment and enhances the tracking of their commands — a key aspect for safety. In this regard, we consider transparency as one of the main goals of safe controller design, and in Chapter 2, we propose a systematic and efficient method for auto-tuning a special class of controllers for bilateral teleoperation, specifically for achieving high transparency.

The fourth mentioned aspect of safe pHRI is safety-oriented control system design. The aim here is to build controllers that are inherently safe by design. Again, this concept covers a broad range of strategies. In the following subsections, we introduce several features that are directly related to this context, which will form the main focus of the thesis. First, we discuss imposing explicit safety constraints on the system. These constraints are closely aligned with safety standards, as will be shown later. Second, we discuss the robustness of the system in the face of uncertainties, which is critical for maintaining safety in many applications. Finally, variable compliance control is discussed as an advanced safety feature. Such a controller allows the system to adapt dynamically to varying interaction forces, which further contributes to safety by ensuring that unexpected impacts or excessive forces do not lead to hazardous conditions, while still being able to reach high forces when needed.

1.3.2 Safety Constraints

pHRI safety standards impose limits on robot behavior to ensure safe interaction with humans. For example, ISO-10218 (2006), which is related to industrial robots, included limits on flange speed (≤ 0.25 m/s), dynamic power (≤ 80 W), and static force (≤ 150 N) [73]. In his thesis in 2014, Haddadin established new grounds for more thorough safety standards and suggested that the characteristics of the tool should be taken into account (e.g., sharp or blunt), or body parts (soft tissues, head, hand) and other factors [73]. Later, and to enhance safety, the ISO/TS-15066 (2016) specified new power and force thresholds that have been established on the basis of pain sensitivity thresholds. The guideline also reports the maximum contact pressure and force for each body area [74].

While teleoperated surgical robots are not identical to collaborative robots, they do share many similarities due to their close interaction with the patient. In some cases, nurses and assistants can be working close to the teleoperated robot, which makes it even more similar to collaborative robots. Other standard guidelines were published in IEC 80601-2-77 (2019) [75], which concern Robotic Assisted Surgery (RAS). These guidelines require the manufacturer to consider the device's speed of movement in the context of all other RAS-related movements, both inside and outside the patient. A common feature between these standards is that they establish safety limits on the signals of the robotic system such as velocity and force.

Another important safety measure in teleoperated surgical robots is a concept called forbidden virtual fixtures. These fixtures define virtual boundaries within the robot's workspace that the system must not cross, effectively preventing the robot from entering dangerous zones, such as areas near critical tissues or anatomical structures, regardless of the operator's commands [76]. Forbidden virtual fixtures could be regarded as a constraint on the robot position. All the mentioned safety aspects can be summarized by one idea: a safe controller must be able to respect specified constraints on the robot. To implement these constraints effectively, we will later use a control method called Model Predictive Control (MPC). MPC is an optimization-based control method that allows incorporating constraints on the system. The method uses a model of the system to predict the system state trajectory over a short time-horizon in the future, allowing it to plan and adjust the robot's trajectory in real time. This ensures that the robot's movements and forces remain within safe limits as defined by relevant safety standards. MPC will be discussed more in detail in Chapter 2.

1.3.3 Robustness to Uncertainty

The explicit enforcement of safety constraints, as discussed above, is a cornerstone of safe system design. However, a critical distinction must be made between safety under nominal conditions and safety in practice. A controller designed using a perfect, nominal model of the robot and its environment might satisfy constraints in simulation, but it offers no guarantees in the real world where uncertainties are inevitable. Modeling errors, unmodeled dynamics (like friction), sensor noise, and unexpected environmental interactions can cause the system's actual behavior to deviate from its predicted trajectory, leading to constraint violations and compromising safety. Therefore, safety and robustness are fundamentally intertwined. A truly safe system must be robustly safe, meaning it is designed to explicitly handle uncertainty and guarantee that constraints will be satisfied across all expected variations and disturbances. Robustness is the property that translates theoretical safety into practical, reliable safety. This thesis addresses this challenge by developing control strategies that are not only aware of constraints but are also robust to the uncertainties that threaten to violate them.

Moreover, in the particular case of MPC which solves an optimization problem online, it is important to ensure that the problem stays feasible throughout the operation and that it does not fail to be solved suddenly, which might compromise the safety. For all the mentioned reasons, in Chapter 3, we design a control strategy to account for these uncertainties using robust MPC techniques, and we augment it with learning methods to obtain a better estimation of the system model and uncertainties.

1.3.4 Variable Compliance Control

Another aspect that contributes to safety in teleoperation is variable compliance control. This approach shifts the focus from enforcing rigid, predetermined limits to enabling the robot to adapt its behavior dynamically in response to the environment. In this work, we adopt a widely used method within this category, called Variable Impedance Control (VIC) [77]. VIC allows the robot to track a desired reference trajectory while modulating its stiffness and damping characteristics in real time. By doing so, the system can react differently to external disturbances. For example, it may become more compliant when encountering unexpected obstacles or forces, or around fragile objects, thereby reducing the risk of injury or damage [78]. Conversely, when accurate tracking is needed to implement a certain task such as cutting, the system increases its rigidity and applies the required forces to achieve the desired trajectory [79]. This adaptive behavior not only enhances safety by providing a more human-like interaction but also improves task performance in certain teleoperation contexts.

Nevertheless, varying the impedance can introduce stability challenges, specifically by injecting energy into the system and compromising the passivity condition [80]. To address this issue, we develop a method that guarantees the passivity and stability of the system. Specifically, we build an optimization-based method that combines different energy dissipation mechanisms. This method is presented in Chapter 4.

1.4 Content of the Thesis

1.4.1 Contributions

This thesis contributes to the research field of bilateral teleoperation by addressing key challenges in safety, transparency, and robustness. Motivated by the need to handle safety constraints, we first design an MPC controller for a bilateral teleoperation system with the goal of optimizing transparency while respecting safety constraints. Our main contributions related to this topic are presented in the following.

Due to modeling errors and other factors, the applied MPC does not result in a transparent system automatically, and accurate tuning is necessary, which is a process that is often difficult and time-consuming. Our first contribution is the design of a framework to auto-tune the MPC controller for enhancing transparency. The method was directly tested on hardware, effectively improving the system's transparency within a relatively short time. The framework and the results are discussed in Chapter 2.

In critical applications such as surgical robots, it is necessary to satisfy the safety constraints, which are dealt with by the MPC controller. However, in the presence of environment model uncertainties, the constraints are prone to violation by the MPC. Our second contribution deals with this issue by employing a learning-based robust MPC. The method guarantees robust constraint satisfaction under bounded environment uncertainty while enhancing the model at the same time. A learning method is used to obtain a better estimation of the uncertainty bounds, reducing the conservatism of the robust controller. The details are presented in Chapter 3.

The third issue addressed in this thesis is the safety in VIC-based teleoperation. While VIC renders the robots more flexible and provides them with natural and compliant interaction characteristics, time-varying impedance (particularly increasing stiffness) can inject energy into the system, potentially violating the passivity condition and compromising stability. Our third contribution is the design of a passive VIC controller for bilateral teleoperation. We extend a method called passivity filters to the bilateral teleoperation case, and we combine it with other passivity-based methods using online optimization. The method is presented in Chapter 4.

In addition to the theoretical developments, this thesis also contributes practically by simulation-based and experimental validation of the methods. Particularly, the proposed auto-tuning framework and the passive VIC were implemented on two different bilateral teleoperation setups. These experimental results demonstrate the feasibility and effectiveness of the proposed approaches. The experimental validations are presented in the corresponding chapters for each contribution.

1.4.2 Scientific Publications

The contributions of this thesis have been published in the following proceedings:

[81] F. A. Almasalmah, H. Omran, C. Liu, T. Poignonec, and B. Bayle, "Auto-Tuning of Model Predictive Control for Bilateral Teleoperation with Bayesian Optimization*," *IFAC-PapersOnLine*, vol. 58, pp. 85–90, Jan. 2024.

[82] F. A. Almasalmah, H. Omran, C. Liu, and B. Bayle, "Adaptive Robust Model Predictive Control for Bilateral Teleoperation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7069–7074, Oct. 2023.

[83] F. A. Almasalmah, T. Poignonec, H. Omran, C. Liu, and B. Bayle, "Passivity Filters for Bilateral Teleoperation with Variable Impedance Control," in *2025 IEEE International Conference on Robotics and Automation (ICRA)* — (Accepted).

1.4.3 Thesis Outline

After introducing the history, context, and main concepts of bilateral teleoperation in Chapter 1, the remainder of this thesis is organized as follows. Each chapter includes a focused literature review that provides the necessary background and positions our contributions with respect to the current state of research.

In Chapter 2, we design an MPC controller for the bilateral teleoperation system, and we present the proposed MPC auto-tuning framework. Chapter 3 details our approach to design an adaptive and robust MPC controller, which deals with the uncertainties of the environment to guarantee the constraints satisfaction, and learns a better model and tighter uncertainty bounds. In Chapter 4, we introduce a VIC strategy for bilateral teleoperation, and we augment it with an optimization-based passivity filter. This enables the leader and the follower robots to adapt their compliance dynamically while preserving the system's passivity and stability. Finally, Chapter 5 concludes the thesis by summarizing the main contributions and discussing future research directions.

Chap. 2

MPC Auto-Tuning for Bilateral Teleoperation

Building upon the discussion in Chapter 1 regarding the importance of safety and transparency in bilateral teleoperation, this chapter focuses on developing a control strategy that explicitly addresses these requirements. We identified transparency as essential for effective interaction and safety, while noting that constraints are fundamental to safe operation, especially in delicate tasks. MPC provides a suitable framework for integrating these aspects. Therefore, this chapter presents the design and formulation of an MPC controller tailored for bilateral systems, aimed at maximizing transparency subject to safety constraints. Furthermore, recognizing the practical difficulties in achieving optimal performance through manual parameter tuning, we introduce an auto-tuning method for the MPC controller.

This chapter is structured as follows: Section 2.1 provides an overview of MPC and designs an MPC controller for bilateral teleoperation. Section 2.2 details the auto-tuning method approach. Section 2.3 validates the method on a 1-DoF teleoperation system experimentally. Section 2.4 concludes the chapter.

2.1 MPC for Bilateral Teleoperation

2.1.1 Introduction and Literature Review

We begin by summarizing the MPC concept and its broad industrial success. Next, we review its application in interaction control, in which a robot physically interacts with an environment, since this topic is directly related to our problem. We then discuss the idea and history of predictive approaches in teleoperation, including MPC and other related predictive controllers, and we conclude by focusing on state-of-the-art MPC in teleoperation.

2.1.1.1 Model Predictive Control

MPC is an advanced control method that uses a dynamic model of the system to predict future states over a horizon and to optimize the control input in order to achieve a certain goal. Its ability to handle constraints on the states, inputs, and outputs makes it especially suitable for applications that require safety and high performance. MPC has seen major developments throughout the decades, driven mainly by the hardware upgrades that allow for real-time computations, the advances of optimization algorithms such as interior-point and active-set methods, and the development of specialized numerical solvers and libraries (e.g., CVX, OSQP, qpOASES, Acados, and GPU-accelerated MPC).

MPC utilizes a mathematical model of the system dynamics to predict future behavior over a finite time horizon N. This prediction is used to solve an optimization problem at every time step, aiming to minimize a defined cost function while respecting system constraints. Although the optimization problem results in a sequence of N optimal control inputs, only the first control input of the computed sequence is applied to the system before the process repeats at the next time step. This is known as the receding horizon principle, which makes it a closed-loop controller with more robustness against disturbances and uncertainties. In classical MPC, a general form of the optimization problem at a discrete time step τ is given by

$$\begin{aligned} U^* = \arg\min_{U} & \left(\mathcal{J}_f(x_N) + \sum_{k=0}^{N-1} \mathcal{J}_s(x_k, u_k) \right) \\ \text{s.t.} \quad & x_0 = x(\tau) \text{ (measured)} \\ & x_{k+1} = f(x_k, u_k) \\ & (x_k, u_k) \in \mathcal{X} \times \mathcal{U} \\ & x_N \in \mathcal{X}_f \end{aligned} \tag{2.1}$$

where $x(\tau) \in \mathbb{R}^{n_x}$ is the measured system state at the discrete time step τ, $x_k \in \mathbb{R}^{n_x}$ and $u_k \in \mathbb{R}^{n_u}$ are respectively the discrete system state and input at the future instant k (relative to time step τ). N is the prediction horizon, $x_0 \in \mathbb{R}^{n_x}$ and $x_N \in \mathbb{R}^{n_x}$ are the initial and final states, respectively. n_x and n_u are the sizes of the state and input vectors, respectively. $f(.)$ represents the system dynamics, $\mathcal{J}_s$ and $\mathcal{J}_f$ are stage and terminal cost functions, respectively. $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ are sets that define constraints on the states and inputs, and $\mathcal{X}_f \subseteq \mathbb{R}^{n_x}$ is the terminal constraint set. U is the control input trajectory over the prediction horizon defined as

$$U = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

Finally, U^* is the optimal solution that contains the optimal control input trajectory over the prediction horizon. It is well known that an unconstrained MPC with a linear system and quadratic cost function yields an optimal solution that is equivalent to a state-feedback controller which can be computed offline. However, when the problem is constrained, it has to be solved online using suitable solvers to guarantee constraint satisfaction [84]. Figure 2.1 shows the operating principle of MPC.

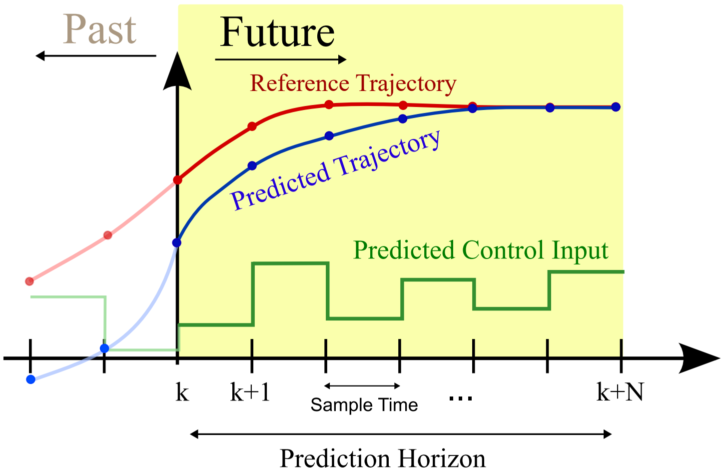

Figure 2.1: The basic concept of MPC. The controller predicts the future behavior of the system (blue line) over a finite horizon and optimizes the control input (green line) to achieve a desired reference behavior (red line).

Thanks to the progress of MPC, it is competing with many classical controllers in many applications. For example, in automotive industry, a multivariable constrained MPC system was designed for torque tracking in turbocharged gasoline engines for mass production by General Motors [85]. In the field of aeronautics, a robust MPC was developed and evaluated for an uncertain nonlinear F-16/MATV aircraft model [86]. The work in [87] designed a stochastic MPC controller for the management of water distribution networks, aiming to optimize energy consumption. In a recent survey of 2020 [88], Samad et al. considered MPC more impactful to the industry than most classical control-theory techniques such as robust control, adaptive control, and non-linear control. The authors also predicted that the impact of MPC will keep growing in the future. Many other applications can be found in an early survey of MPC in industry [89], or a more recent one in [90]. Following its success, MPC has been widely used in robotics, including in interaction control.

2.1.1.2 MPC in Interaction Control

Over the past decade, MPC has been increasingly applied in robot interaction control, where robots are supposed to interact physically with an environment or humans. By leveraging its inherent versatility and flexibility, MPC was used in various ways and was combined with different components to enhance the interaction while ensuring safety and performance.

Several studies employed MPC as a high or mid-level controller to set commands for a lower-level controller. In [91], MPC was used as a high-level controller for joint path set-point generation. This enabled a robot to rapidly reach a target position in dense clutter while regulating whole-body contact forces to be below a given threshold. In [92], Haninger et al. used MPC with a Gaussian process to model human force in various task modes. Based on inferred human intent, MPC optimizes the robot trajectories and generates the desired force trajectory for a lower-level admittance controller, allowing for flexible and efficient collaboration. Gold et al. [93] generalized the idea and proposed a general framework of model predictive interaction control for robotic manipulation tasks. The proposed controller acts as a mid-level trajectory planner and calculates the desired joint positions and velocities for the subordinate controller to realize a desired interaction force, compliance behavior, or motion.

Other studies used MPC as a lower-level controller to directly manage the interaction forces. Bednarczyk et al. [94] developed a method based on MPC to replicate an impedance controller behavior when interacting with an environment, while respecting practical robotic constraints. In [95], Matschek et al. proposed a learning-based MPC scheme that optimizes the robot torques and provides stochastic safety guarantees by limiting the forces when interacting with an uncertain environment. They demonstrated the method with a robot writing on a board with a certain reference force.

Building on these diverse implementations, it is evident that MPC's adaptability makes it an excellent candidate to safely manage physical interactions, from high-level trajectory generation to low-level force regulation. Next, we present early works that employed predictive approaches in teleoperation, which laid the groundwork for using MPC in this field.

2.1.1.3 Prediction-Based Control in Teleoperation

Many types of predictive controllers were used in teleoperation, including Smith Predictor (SP) [96, 97], Model-Based Predictors (MBP) [98, 99], Generalized Predictive

Control (GPC) [100, 101], and MPC [5, 68]. The concept of prediction-based control was applied for many reasons, such as enhancing stability and performance in general, or more specific goals such as reducing the effect of time delay [102], or dealing with specific types of predictable disturbances [103]. An extensive survey on predictive control approaches in teleoperation is provided in [104]. In the following, we will focus mostly on GPC, which is a close relative to MPC. GPC solves an optimization problem with no constraints. In the linear case, it could be solved offline and formulated as a state-feedback controller. The method has been widely applied in teleoperation, both unilateral and bilateral.

An example of disturbance compensation was done by Ginhoux et al. [103] designed a unilateral teleoperation scheme that compensates for the repetitive respiratory motions or cardiac motions in a surgical robot. The method uses the predictive feature of GPC to predict and compensate for such repetitive motions. Later, Joinie-Maurin et al. [105] developed the method to include force feedback while removing the part that results from the respiratory motion, such that the surgeon feels as if the environment is motionless. This allows for a more intuitive operation on moving organs.

The work of Slama et al. [102] included a GPC on the operator side that generates a reference velocity for the follower robot's PI controller. The GPC model explicitly takes into account the effect of the haptic force feedback which modifies the intended trajectory of the operator. This enhances the robustness against environment uncertainty and time delay. The method was tested on a scaled teleoperation through the internet in [106]. Chen et al. [101] used an event-based GPC strategy for teleoperation via internet, which used a path governor on the leader side that generates online parameterization of the desired motion reference, and a GPC on the follower side to generate the redundant control information to diminish the influence of the packet loss and the large time delay. Other papers that employed GPC in teleoperation include [107–111].

2.1.1.4 MPC in Teleoperation

MPC has gained attention in teleoperation applications due to its success in solving practical problems such as safety constraints and improving performance using learning techniques. In addition, MPC is an online optimization-based method, which has the potential to manage online the compromise between safety and transparency in a better way than the controllers that separate those two goals [112]. An early attempt was made by Bemporad in 1998 [113], where the author used an MPC at the leader side to handle time delays in a unilateral teleoperation scenario. The first value of the optimal control sequence, computed at each step, is applied as usual, but in case of delayed signals, the remaining sequence from the last computed trajectory is used as a fallback input on the follower robot, while still satisfying the constraints. In [68], Sheng and Spong proposed a modified MPC method to deal with the time delay and input or output constraints in bilateral teleoperation systems. The controller adds a correction signal based on the previous prediction error compared to the new measured signals. The authors formulated separate controllers for free motion and contact motion. Ghazi et al. [114] designed an MPC for bilateral teleoperation and focused on the optimization algorithm that minimizes the cost function by comparing several optimization methods in simulation. The authors found that the interior-point method, Singular Value Decomposition (SVD), and recurrent neural networks were the most efficient methods for this application. In [115], Hatori et al. proposed a compensation method for systems with time-varying delays using MPC on the follower side. The method improves responsiveness and compensates for disturbances and modeling errors by incorporating a model error compensator.

Due to the flexibility of MPC, there exist many other interesting examples illustrating the use of MPC constraints and cost function for a variety of goals in the literature. In a more recent work by Piccinelli and Muradore [112], the authors designed a nonlinear MPC with energy tanks for both the leader and follower robots. In order to guarantee passivity, constraints on the energy level in the tanks were added to the MPC problem. The authors showed how MPC reduced the chattering issue that happens when the tank has low energy, and the method was tested on a 1-DoF teleoperation system in a hardware experiment. Later in [5], the authors used the same framework and added a force constraint on the follower robot to prevent it from damaging the unknown environment. The method used recursive least squares to learn the environment model, starting from an overestimation of the model parameters. The method was tested on a 6-DoF teleoperation system in simulation. In [116], the authors proposed using a hybrid linear MPC, where they defined an integer variable that counts the number of switches between free motion and contact state. By including the variable in the cost function, they reduced the tool bouncing at the moment of contact. To keep the passivity, the authors used a linearized version of the energy tanks.

In several works, MPC was used on a relatively complex follower robot to provide it with additional abilities and a degree of autonomy. For example, in [117], Hu et al. implemented a nonlinear MPC to provide real-time motion planning and collision avoidance capabilities to the follower robot, since it has redundant parts that cannot be directly controlled by the operator or by the haptic device. Risiglione et al. [118] used a nonlinear MPC on the follower side for controlling a teleoperated robotic arm on top of a legged robot. The MPC was used as a high-level controller that gives reference commands to a lower-level whole-body controller. Both sides have energy tanks to guarantee passivity and stability. A similar application was studied by Cheng et al. in [119], where a feedback MPC was implemented to give reference commands to the lower-level whole-body controller which runs at a much faster rate while guaranteeing collision-free teleoperation by defining suitable MPC constraints. On the operator side, the haptic force feedback is reflected from a force/torque sensor on the follower robot with additional damping for better stability. While the majority of works on predictive control in teleoperation consider time delay, other problems such as tuning for performance and robust constraints in the presence of uncertainties are still not deeply explored. In this chapter, we will design a linear MPC for bilateral teleoperation, and then we will focus on the tuning of the MPC controller for bilateral teleoperation applications.

2.1.2 MPC Formulation for Bilateral Teleoperation

2.1.2.1 Prediction Model for Bilateral Teleoperation

An ideal MPC controller requires accurate models of all system components, including the human operator, the robots, and the environment. However, this is almost impossible in practice, especially for the human operator. A simplified model could still be useful considering the closed-loop nature and the fact that the optimization problem is solved at each time step, which reduces the effect of model mismatch. Therefore, the following assumptions are made to simplify the model throughout Chapters 2 and 3:

1. accurate models of the leader and follower robots are available, which allows for linearizing and decoupling the axes as will be shown later.

2. the human force is measurable and slowly varying, i.e., $\dot{f}_h \approx 0$. As a result, f_h is measured at each time step and assumed constant over the prediction horizon in the optimization problem. While this is a simplification, it is commonly adopted in the literature and is justified by the short duration of the prediction window relative to typical variations in human input.

3. the robot is always in contact with the environment, which is represented suffi-

ciently accurately by an LTI model with decoupled axes, i.e., pushing the environment in one axis does not generate a force in the other axes.

4. the leader robot has low impedance, meaning that the human operator feels mainly the actuator force, rather than the dynamics of the robot.

To model a general robotic manipulator with n-DoF, moving in m-dimensional task space and interacting with an external system with force f_{ext}, the dynamics are expressed in the joint space as follows

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau_c + J^\top f_{\text{ext}} \tag{2.2}$$

where $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ are the joint position, velocity, and acceleration, respectively. $H(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ is the Coriolis matrix, $g(q) \in \mathbb{R}^n$ is the gravity vector, $\tau_c \in \mathbb{R}^n$ is the control torque, $J \in \mathbb{R}^{m\times n}$ is the Jacobian matrix, and $f_{\text{ext}} \in \mathbb{R}^m$ is the external force.

A common practice is to build a low-level controller to linearize and decouple the robot axes in the task space, which can be done by implementing the steps in Appendix A. Building on this, in Chapters 2 and 3, we will assume that both the leader and follower robots are linearized and decoupled using a low-level controller while being in contact with an environment with LTI model and decoupled axes. Therefore, we design our MPC for a single axis, which deals with a 1-DoF LTI teleoperation system that involves one axis on the human side and the corresponding axis on the follower side. The resulting dynamics of this system can be described as

$$m_l\ddot{x}_l + b_l\dot{x}_l = f_l + f_h \tag{2.3}$$

$$m_f\ddot{x}_f + b_f\dot{x}_f = f_f + f_e \tag{2.4}$$

where $m_l, b_l, m_f, b_f \in \mathbb{R}$ are the mass and damping coefficients of 1-DoF axis of the leader and follower robots, respectively. $f_l \in \mathbb{R}$ and $f_f \in \mathbb{R}$ are the control forces applied by the motors of the leader and follower robots, respectively. $f_h \in \mathbb{R}$ and $f_e \in \mathbb{R}$ are the human force and environment force, respectively.

In the literature, the environment is often assumed to be modeled sufficiently accurately by an LTI spring-damper system [5, 120]. Assuming such a model for the environment and that it is always in contact with the follower robot, its force can be written as

$$f_e = -k_e x_f - b_e \dot{x}_f \tag{2.5}$$

where k_e is the stiffness of the environment, and b_e is its damping coefficient. The linearized dynamics of the follower robot (2.4) when in contact with the environment can be rewritten as

$$m_f \ddot{x}_f + (b_f + b_e)\dot{x}_f + k_e x_f = f_f \tag{2.6}$$

From (2.3) and (2.6), the continuous-time state space representation can be written as

$$\underbrace{\begin{bmatrix} \dot{x}_l \\ \ddot{x}_l \\ \dot{x}_f \\ \ddot{x}_f \\ \dot{f}_h \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -b_l/m_l & 0 & 0 & 1/m_l \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -k_e/m_f & -(b_f+b_e)/m_f & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A_c} \underbrace{\begin{bmatrix} x_l \\ \dot{x}_l \\ x_f \\ \dot{x}_f \\ f_h \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 & 0 \\ 1/m_l & 0 \\ 0 & 0 \\ 0 & 1/m_f \\ 0 & 0 \end{bmatrix}}_{B_c} \underbrace{\begin{bmatrix} f_l \\ f_f \end{bmatrix}}_{u} \tag{2.7}$$

We also define the output of the system as

$$y = \begin{bmatrix} x_l \\ \dot{x}_l \\ x_f \\ \dot{x}_f \\ f_h \\ f_e \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -k_e & -b_e & 0 \end{bmatrix}}_{C_c} \underbrace{\begin{bmatrix} x_l \\ \dot{x}_l \\ x_f \\ \dot{x}_f \\ f_h \end{bmatrix}}_{x} \tag{2.8}$$

To be used in the traditional MPC, the continuous-time system can be discretized with sampling time T_s and rewritten as

$$\begin{aligned} x_{k+1} &= A x_k + B u_k \\ y_k &= C x_k \end{aligned} \tag{2.9}$$

where the subscript $k \in \mathbb{N}$ refers to the discrete moment k. A, B, and C are the discretized versions of A_c, B_c, and C_c from (2.7) and (2.8).

In the following, we formulate the problem as a linear MPC problem, where the cost function can be quadratic (hence convex), and the constraints are written as combined linear constraints on the states and inputs. Note that this also allows adding constraints to outputs or a certain performance vector that is linear in the states and inputs. This formulation allows for efficient solving of the optimization problem using specialized solvers.

2.1.2.2 MPC cost function for bilateral teleoperation

To quantify transparency, we define the ideal performance vector z^{ideal} as follows

$$z^{\text{ideal}} = \begin{bmatrix} x_l - x_f \\ \dot{x}_l - \dot{x}_f \\ f_h + f_e \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} y \tag{2.10}$$

The transparency is achieved if $z^{\text{ideal}} = 0$. Hence, the goal of the MPC could be to minimize the magnitude of the z^{ideal}.

Note, however, that naively minimizing $(f_h + f_e)$ using the prediction model (2.9) limits the capacity to reflect the environment force to the operator, since f_h is not controllable by the input u. At the same time, the MPC can control f_e indirectly by modifying $x_f, \dot{x}_f$. This means that from the point of view of the controller, only f_e can be controlled to match f_h, and not the opposite. Therefore, this formulation cannot implement any teleoperation architecture that displays environment force on the operator side such as F-P architecture. In order to provide the MPC with the ability to manipulate the future values of f_h, we can either incorporate the human model in the MPC which is not trivial, or we can use the value of the leader robot force f_l as an approximation of $-f_h$ as done in [68]. This takes advantage of the fact that the human feels mainly the force applied by the leader robot motor (see assumption 4), i.e., $f_h \approx -f_l$. Such assumption is valid for most commercial haptic devices thanks to their light weight [121]. For that reason, we modify the definition of the performance vector to include $-f_l + f_e$ instead of $f_h + f_e$

$$z = \begin{bmatrix} x_l - x_f \\ \dot{x}_l - \dot{x}_f \\ -f_l + f_e \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{E} y + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -1 & 0 \end{bmatrix}}_{F} u \tag{2.11}$$

and the discrete version of the performance vector is defined as

$$z_k = Ey_k + Fu_k \tag{2.12}$$

The MPC cost function is defined in order to minimize the matching errors between both robots and maximize transparency along the prediction horizon as follows

$$\mathcal{J}(x_{0\ldots N}, u_{0\ldots N-1}) = \sum_{k=0}^{N-1} \left(z_k^\top Q z_k + u_k^\top R u_k \right) + z_N^\top P z_N \tag{2.13}$$

where $Q \in \mathbb{R}^{n_z \times n_z}$ is a semi-positive definite matrix that penalizes the matching errors between the leader and follower robots, and n_z is the size of the vecotr z. $R \in \mathbb{R}^{n_u \times n_u}$ is a positive definite matrix that penalizes the control effort. $P \in \mathbb{R}^{n_z \times n_z}$ is a semi-positive definite matrix that penalizes the matching errors at the end of the prediction horizon.

2.1.2.3 MPC Constraints for Bilateral Teleoperation

The main advantage of MPC is the ability to handle constraints on inputs and states. In general, it makes sense to add hard constraints on control inputs to account for motor saturation. In bilateral teleoperation, additional constraints could be defined to enhance safety. For example, the velocity and the kinetic energy can be limited based on safety standards [74, 75]. Constraints on the follower robot position can be defined to prevent it from exiting the safe operating area, which can be seen as a forbidden region virtual fixture [122]. In addition, the force applied to the environment can be limited to prevent its damage [5]. Other works constrained the energy level in the tanks to be higher than a threshold to guarantee passivity [112]. In the following, we consider a general form of convex mixed-input-state linear constraints that can be expressed by the row-wise inequality

$$H_{\mathcal{X}} x_k + H_{\mathcal{U}} u_k \leq 1_{n_c} \tag{2.14}$$

where $H_{\mathcal{X}} \in \mathbb{R}^{n_c \times n_x}$ and $H_{\mathcal{U}} \in \mathbb{R}^{n_c \times n_u}$ are the constraint matrices and vector, respectively, and n_c is the number of constraints. 1_{n_c} is a vector of ones of size n_c. Note that in the case of a good model, constraints on y or z can be reformulated to the form of (2.14) using the model (2.9) and (2.12). The mentioned form of constraints represents a polytopic region in the space of states and inputs. Constraints on the terminal state can also be added as

$$H_{\mathcal{X}_f} x_N \leq 1_{n_{c_N}} \tag{2.15}$$

where $H_{\mathcal{X}_f} \in \mathbb{R}^{n_{c_N} \times n_x}$ is the constraint matrix for the final state, and n_{c_N} is the number of constraints on the final state, and $1_{n_{c_N}}$ is a vector of ones of size n_{c_N}.

2.1.2.4 MPC Optimization Problem for Bilateral Teleoperation

Since we consider a teleoperation at close distance with negligible delays, we design the MPC as a centralized controller that takes the states of both robots and the human force as inputs at the same time. Figure 2.2 shows a block diagram of the centralized MPC for bilateral teleoperation. Decentralized MPC is usually used when the robots are far from each other such as in [116], where the two MPC controllers are implemented separately, and each controller takes the latest solution of the other controller as an input. However, in our case, a centralized MPC gives more flexibility to the optimizer to adjust both control inputs at the same time.

We can now formulate the full MPC optimization problem using the prediction model (2.9) and (2.12), the cost function (2.13), and the constraints (2.14) and (2.15). At every discrete time step τ, the MPC takes $x_0 = x(\tau)$ as input and solves the following problem

$$\begin{aligned}
U^* = \arg\min_{U} & \sum_{k=0}^{N-1} \left(z_k^\top Q z_k + u_k^\top R u_k \right) + z_N^\top P z_N && (2.16)\\
\text{s.t.} \qquad x_0 &= x(\tau) && (2.17)\\
x_{k+1} &= A x_k + B u_k && (2.18)\\
y_k &= C x_k && (2.19)\\
z_k &= E y_k + F u_k && (2.20)\\
H_{\mathcal{X}} x_k + H_{\mathcal{U}} u_k &\leq 1_{n_c} && (2.21)\\
H_{\mathcal{X}_f} x_N &\leq 1_{n_{c_N}} && (2.22)
\end{aligned}$$

At each time step, the state $x(\tau)$ is measured and updated, including the positions and velocities of the robots, and the human force. Then, the optimization problem (2.16) is solved to get the optimal control sequence U^*. After that, only the first control input u_0 is applied to both robots, and the process is repeated at the next time step.

Note that in traditional bilateral teleoperation architectures, signals are directly sent and used by the controllers on each side, and thus creating a clear architecture such as F-P teleoperation. On the other hand, MPC-based bilateral teleoperation uses the optimization problem (2.16) to connect the robot models and to minimize the matching errors in z from (2.12). Therefore, the architecture is not as clear as in traditional teleoperation, and it depends on the weights of the cost function and the model, as well as the signals sent from each robot. In any case, choosing the cost function weights (Q, R, P) reflects directly on the behavior of the teleoperation system. However, it is not straightforward to select these weights to achieve the desired behavior, especially when the system parameters are not known accurately. In the next section, we will present a method for auto-tuning the MPC controller for bilateral teleoperation.

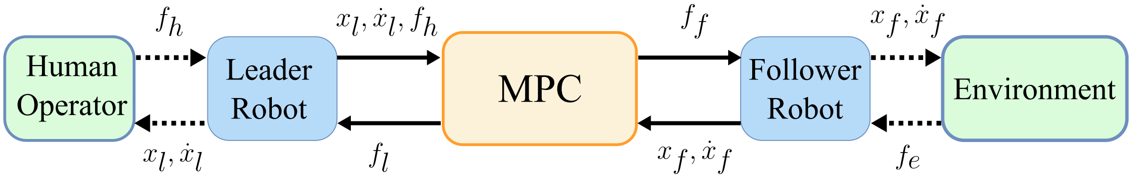

Figure 2.2: Block diagram of the centralized MPC for bilateral teleoperation. Note that the MPC plays the role of the controller and the communication channel in this diagram since it takes the signals from both sides and directly computes the control inputs for each robot.

2.2 Auto-Tuning MPC for Bilateral Teleoperation

2.2.1 Introduction and Literature Review

Despite its importance, the choice of the MPC parameters is rarely an intuitive task, and it generally requires experience and iterative tuning through trial and error. The MPC cost function in teleoperation is often designed to reduce the predicted matching errors between the leader and follower robots in terms of position, velocity, and force. In addition, the cost function usually contains a term related to the control inputs, as defined in (2.13). However, it should be noted that accurately matching forces and matching positions are two competing objectives, which requires the controller to compromise between the two [1]. Furthermore, the control input penalty weight R governs the level of control effort exerted by the controller, but determining its appropriate value relative to Q is not straightforward. Moreover, assigning the weights in the MPC cost function based on intuition may not always yield the intended results in the closed-loop behavior due to several reasons, such as model errors, short prediction horizon, and the choice of control input weights. Thus, an efficient method for auto-tuning is still needed.

To tune the cost function weights, two perspectives can be adopted. The first approach, known as the inverse optimality problem, involves deriving a cost function that would yield a controller with desirable behavior, often by mimicking a pre-existing, well-performing controller. Examples include learning from human demonstrations [79], solving the inverse LQR problem using LMIs [123], or controller matching techniques [124]. However, these methods require that a good controller already exists as a benchmark. In this work, we focus on a different approach, which involves directly optimizing a known performance metric, which will be discussed next.

2.2.1.1 Tuning as an Optimization Problem

The controller tuning process can be seen as an optimization problem whose goal is to minimize a certain metric that assesses the controller performance. This metric depends implicitly on the controller parameters, such as the MPC cost function weights, since they do in fact affect the performance. Unfortunately, such a metric may not have a simple analytical form, and it can only be evaluated point-wise by performing experiments (or simulations) using the candidate weights that are being evaluated. Hence, in the literature of teleoperation controller tuning, black-box optimization methods were often used. For example, early work in [125] used the Hooke and Jeeves search method to search for optimal PID gains for a simulated 2-DoF robot. Other authors used particle swarm optimization [57, 126], genetic algorithms [127], or artificial bee colony algorithm [128]. Evolutionary learning neural networks were used in [129] to learn the optimal gains for a PID controller based on the current environment. In [130], the authors used a method called relay auto-tuning to tune the PID controller of the leader robot in a medical needle insertion scenario. Most of the aforementioned methods require a large number of experiments to find the optimal tuning, which limits their applicability to simulations. Optimal tuning obtained in simulations can be applied to the real system, but this might make it suboptimal due to model-plant mismatch. Therefore, a method that is sample-efficient and can be applied directly to the real system is needed. A promising approach is to use Bayesian optimization (BO), which is a black-box optimization method that has gained popularity in recent years due to its sample efficiency and ability to deal with noise-corrupted objective functions.

2.2.1.2 Bayesian Optimization

Among the black-box methods, BO stands out for its ability to find optimal parameters with minimal evaluations. BO achieves this by explicitly modeling the objective function using a model, which guides the search process. This model is referred to as the *surrogate model* in the literature. To look for the minimizer, the algorithm iteratively balances the exploration of promising regions in the search space and the exploitation of known high-performing areas. More details about the algorithm will be provided in Section 2.2.2. A review on the method can be found in [131]. Recently, BO has been applied to tune controllers in many contexts. For instance, the work in [132] auto-tuned an LQR controller for balancing an inverted pendulum held by a robotic arm in hardware experiments. The authors of [133] used BO to find energy-efficient trajectories for a robotic manipulator by tuning the MPC weights. In [134], the authors auto-tuned a robot controller to interact with unknown objects robustly without causing any damage. In [135], a user-adaptive variable damping controller was designed for human-robot interaction tasks. The method used BO to find the optimal controller parameters for each user efficiently.

In the following, we use the BO algorithm to auto-tune the MPC controller for bilateral teleoperation tasks. Therefore, safety guarantees can be enforced by the MPC controller, while ensuring an accurate teleoperation by finding the optimal MPC weights. The method balances the trade-off between position tracking and force tracking for the leader and follower robots. Moreover, the sample efficiency of the method makes it applicable for tuning in experimental hardware directly with a relatively low number of experiments as will be demonstrated through our simulations and experiments. The auto-tuning method focuses on finding the best weights for the MPC cost function to achieve good teleoperation performance by minimizing the closed-loop matching errors of position, velocity, and force between both robots.

2.2.2 Bayesian Optimization for MPC Tuning

In the following, we provide a brief background about the BO algorithm that we use to find the optimal weights Q, R of the MPC cost function. Interested readers are referred to [136] for more details about the algorithm.

2.2.2.1 Problem Formulation

We consider Q, R as diagonal matrices, and $P = 0$ as was done in [137] to reduce the search space dimension and we write

$$Q = \text{diag}(\phi_1, \phi_2, \phi_3) \quad , \quad R = \text{diag}(\phi_4, \phi_5) \tag{2.23}$$

where diag(.) is a diagonal matrix with the arguments on its main diagonal. Finally, we search for the optimal $\phi = [\phi_1, \phi_2, \phi_3, \phi_4, \phi_5] \in \mathbb{R}^5$ in a bounded set we call the search space $\Phi \subset \mathbb{R}^5$.

We define a scalar performance metric $\mathcal{M}(\phi)$ that measures the performance of a certain MPC controller parameterized by ϕ. The tuning process is equivalent to finding the solution to

$$\phi^* = \arg\min_{\phi \in \Phi} \mathcal{M}(\phi) \tag{2.24}$$

where $\mathcal{M} : \mathbb{R}^5 \to \mathbb{R}$ is the performance metric. This metric does not have a known analytical form, and the evaluation of its value at a certain point ϕ could be done experimentally by running the MPC performance with the candidate set of weights ϕ plugged into the cost function matrices as in (2.23). This is a time-consuming process even for evaluating a single ϕ. Moreover, evaluating the performance metric $\mathcal{M}(\phi)$ is often corrupted by noise due to the stochastic nature of the environment and the human operator. That is, repeating the same experiment with the same ϕ will yield different values of $\mathcal{M}(\phi)$, which makes it difficult to find the optimal ϕ^* using traditional optimization methods. BO helps find ϕ^* efficiently with a low number of experiments.

2.2.2.2 Bayesian Optimization Algorithm

The BO algorithm constructs a surrogate model $\hat{\mathcal{M}}(\phi)$ that approximates $\mathcal{M}(\phi)$ in important regions, and then iteratively seeks to find the minimizer of the surrogate model instead of the actual function $\mathcal{M}(\phi)$. When the surrogate model becomes accurate enough, the minimizer of the surrogate model will be close to the minimizer of the function. Using this idea, the optimization problem (2.24) (the tuning) turns into the following iterative process

1. choose a point ϕ^i at which to evaluate $\mathcal{M}(\phi)$.

2. evaluate the performance metric $\mathcal{M}(\phi^i)$.

3. update and fit the surrogate model $\hat{\mathcal{M}}(\phi)$.
4. minimize the surrogate model $\hat{\mathcal{M}}(\phi)$ using more efficient methods such as gradient-based methods.

To choose which points to sample $\mathcal{M}(\phi)$ at, the algorithm uses an *acquisition function* $\alpha(\phi)$ that balances the exploration-exploitation trade-off. In other words, $\alpha(\phi)$ encourages exploring new regions in the space where the surrogate function has high uncertainty and is likely not to be close to the actual function in order to enhance the fitting (exploration). Conversely, $\alpha(\phi)$ promotes exploitation by selecting points that minimize the surrogate function, which are likely to approximate the minimizer of the true function $\mathcal{M}(\phi)$.

Gaussian Processes (GPs) are a popular choice for constructing a surrogate model due to their sample efficiency and robustness in handling noisy objective functions. The GP is a probabilistic model that approximates the performance metric and provides the estimated mean and variance of the value at each point. Given a set of n noisy evaluations

$$\mathcal{D}_n = \{(\phi^i, \mathcal{M}^i = \mathcal{M}(\phi^i) + \epsilon^i)\}_{i=1}^n$$

where $(\cdot)^i$ denotes the i-th evaluation, ϵ^i is Gaussian noise with $\epsilon^i \sim \mathcal{N}(0, \sigma_\epsilon^2)$, and $\mathcal{M}^i$ represents the i-th evaluation of the performance metric. The GP is completely specified by its mean function $\mu(\cdot)$ and covariance function (or kernel function) $\kappa(\cdot, \cdot)$. The surrogate model of the performance metric is a GP which can be expressed as

$$\hat{\mathcal{M}}(\phi) \sim \mathcal{GP}\big(\mu(\phi), \kappa(\phi^i, \phi^j)\big)$$

where ϕ^i, ϕ^j represent two points in the search space. Assuming a zero prior mean, the predicted mean and variance at a new point ϕ are given by

$$\begin{aligned} \mu_n(\phi) &= k_{\mathrm{GP},n}^\top K_{\mathrm{GP},n}^{-1} Y_n \\ \sigma_n^2(\phi) &= \kappa(\phi, \phi) - k_{\mathrm{GP},n}^\top K_{\mathrm{GP},n}^{-1} k_{\mathrm{GP},n} \end{aligned} \tag{2.25}$$

with $\kappa(\phi, \phi) \in \mathbb{R}$ is the kernel function evaluated at (ϕ, ϕ), $k_{\mathrm{GP},n} \in \mathbb{R}^n$ is the vector whose i-th entry is $\kappa(\phi, \phi^i)$, and $K_{\mathrm{GP},n} \in \mathbb{R}^{n\times n}$ the kernel matrix with entries

$$[K_{\mathrm{GP},n}]_{ij} = \kappa(\phi^i, \phi^j) + \delta_{ij}\sigma_\epsilon^2$$

where δ_{ij} is the Kronecker delta, and $Y_n = [\mathcal{M}^1, \ldots, \mathcal{M}^n]^\top$ is the vector of observed performance metrics. Equation (2.25) defines an analytical form of the probabilistic GP model $\hat{\mathcal{M}}(\phi)$, which tries to capture the actual performance metric $\mathcal{M}(\phi)$. Note that calculating the predicted mean and variance at a new point depends directly on all previously measured points in $\mathcal{D}_n$. Given the predictive distribution (2.25), the algorithm proposes the next point to be evaluated ϕ^{n+1} by minimizing the acquisition function. We use the standard Upper Confidence Bound (UCB) function given by

$$\alpha_{\text{UCB}}(\phi|\mathcal{D}_n) = \mu_n(\phi) - \beta\sigma_n(\phi) \tag{2.26}$$

with the exploration hyperparameter β, where higher values of β encourage the exploration of unknown regions with high uncertainty in the GP, and lower values encourage finding the minimum using the current learned surrogate model. For the GP kernel, we employ the standard *Matérn kernel 3/2*, and infer its hyperparameters and noise level by evidence maximization [138]. In order to search for the optimal weights of the MPC, we need to define a bounded search space and a performance metric. The algorithm builds a surrogate GP model that captures the relationship between the MPC weights and the value of the performance metric and encodes it in the form of a probabilistic distribution (2.25). This is then used iteratively to find the optimal weights.

2.2.3 Proposed Approach of Auto-Tuning

In the following, we present the proposed approach to automatically tune the MPC weights using the BO algorithm in bilateral teleoperation.

2.2.3.1 Performance Metric Design

To compare different weights of the MPC, we need to design a function that measures the performance based on an experiment (or simulation). One of the advantages of BO is its ability to handle complex non-convex performance metrics [131]. Many examples of such functions were used in the literature in different applications. For example, in autonomous racing cars, the authors of [137] used lap time and the deviation from the center line as a performance metric. The work in [139] used human preference as a measure of performance to tune a haptic rendering system. In [135], the authors used BO to adapt the damping in a pHRI scenario. The authors used a combination of agility measures, user effort, and stability-related measures such as overshoot and settling time. The authors of [134] used compliance and force smoothness at the moment of contact to tune a robot that interacts with an uncertain environment.

In teleoperation, as we mentioned in Subsection 1.3.1, our main goal is to enhance system transparency in order to contribute to the awareness and control accuracy of the human operator, thus contributing to safety. Therefore, we use the matching errors of positions, velocities, and forces to measure the performance after each experiment (i) as

$$\mathcal{M}^i = \frac{1}{T_{exp}} \sum_{j=1}^{T_{exp}} z_j^\top W_{\mathcal{M}} z_j \tag{2.27}$$

where $W_{\mathcal{M}}$ is a scaling matrix that takes into account measurement units and the relative importance between the signals as defined by the user, T_{exp} is the duration of the experiments measured in time steps, and z is the performance vector which contains the matching errors as defined in (2.12). In practice, we set an upper bound on the performance metric value to prevent large values since a point with an excessively high value can deteriorate the accuracy of the GP fitting for neighboring low points. It is important to point out the difference between the MPC cost function (2.13) and the performance metric (2.27). Although both depend on the matching errors, the former minimizes the predicted errors based on the system model on a short horizon, while the latter evaluates the closed-loop measured errors on a much longer window in a model-independent manner. Importantly, the closed-loop errors differ from the predicted ones due to the limited prediction horizon and the potential model-plant mismatch. The MPC cost function also includes a cost for control inputs while the performance metric does not.

2.2.3.2 Auto-Tuning the MPC for Bilateral Teleoperation

To tune the MPC, we perform n_{exp} experiments of interaction with an environment with parameters k_e, b_e. Each experiment is run for T_{exp} time steps. The human operator applies a force profile on the leader robot, while the follower robot interacts with the environment. Before each experiment (i), the BO algorithm proposes a certain ϕ^i that contains the diagonal values of Q and R. After T_{exp} time steps of interaction, we calculate the performance metric as in (2.27), and we update the surrogate model with the new noisy measurement of the performance metric $(\phi^i, \mathcal{M}(\phi^i) + \epsilon^i)$ where ϵ^i is the unknown noise value included in the measurement. Using the updated mean and kernel functions, the next promising point is found by minimizing the acquisition function from (2.26). After testing n_{exp} different points, we can either take the best one as the optimal tuning, or we can find the minimizer of the learned surrogate (GP) model using gradient-based techniques with multiple restarts. The latter choice is preferred due to its robustness against noisy measurements and human input variability.

In this work, we assume that we know a range of parameters that can stabilize the system, which is used as the search space for the algorithm. In practice, theoretical stability guarantees can be achieved through other means, for instance, using energy tanks [112]. Figure 2.3 shows the architecture of the teleoperation system with the proposed auto-tuning algorithm.

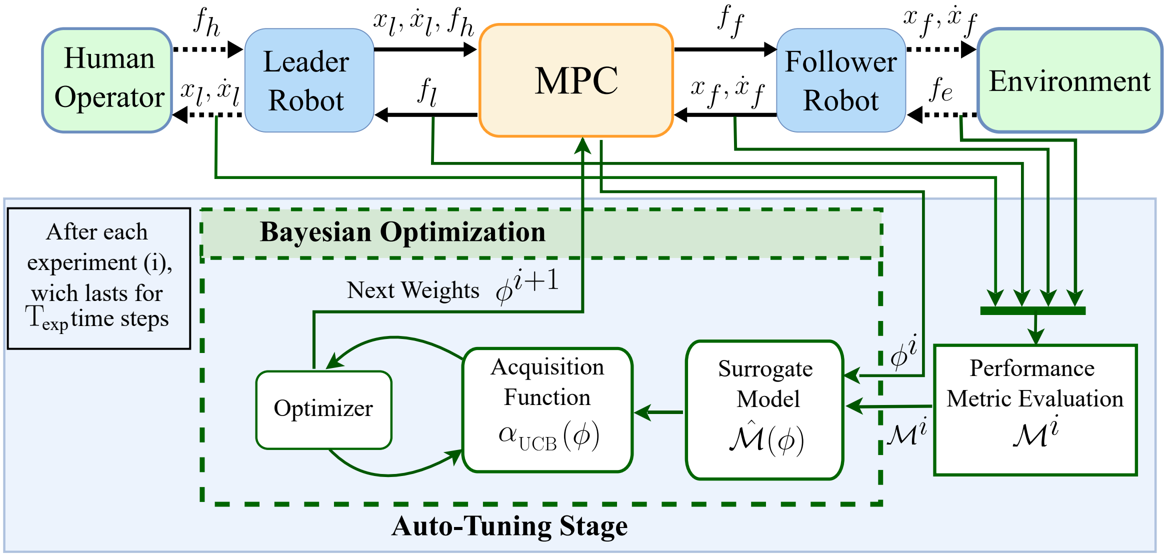

Figure 2.3: Block diagram of the proposed auto-tuning approach with the teleoperation system.

2.3 Validation

To evaluate the effectiveness of the proposed auto-tuning algorithm, we validate it through both simulation and hardware experiments. The simulation provides a controlled environment to test the algorithm under ideal conditions with accurate models and no noise, while the hardware experiments demonstrate its performance under realistic conditions, including model inaccuracies and sensor noise. We apply the proposed algorithm in both simulation and hardware experiments using the same setup to compare the results.

2.3.1 Experiment and Simulation Description

To evaluate the performance metric at a certain point (ϕ^i), we perform what we refer to as a *trial*, which can be either a short simulation or a hardware experiment. Each trial involves 3 seconds of interaction with a low-frequency sinusoidal-like human force as input to the system ($1 \sim 2$ Hz). At the end of each interaction, we calculate the noisy performance metric value ($\mathcal{M}(\phi^i) + \epsilon^i$), and we assign it to the corresponding MPC weights. The (unmeasurable) noise ϵ^i can be caused, for instance, by human behavior variability. The BO algorithm updates the surrogate model with the measured value, and the minimization of the acquisition function (2.26) proposes new weights to test, which are then used in the next trial. The process is repeated for n_{exp} trials until acceptable tuning is found. The goal of the tuning is to minimize the performance metric in (2.27). We set $W_{\mathcal{M}}$ from (2.27) as $W_{\mathcal{M}} = \text{diag}(1500, 5, 500)$, and we impose an upper bound of 10 on the performance metric value. In the BO algorithm, we set the exploration parameter in (2.26) to $\beta = 3$ initially, and decrease it linearly until 0.5 at the last point to encourage the exploitation. The BO algorithm also requires defining the bounds of the parameters, which are set as the element-wise inequalities

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \leq \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{bmatrix} \leq \begin{bmatrix} 10^4 \\ 10 \\ 10^4 \\ 10^3 \\ 10^3 \end{bmatrix} \tag{2.28}$$

where the search space of stabilizing parameters results from empirical tests and scaling of measuring units.

The code is implemented in the ROS2 framework under Linux. Acados library [140] is used to formulate and implement the MPC solver in C++. The BO code is based on the software in [141]. The implemented code automates the tuning process such that the human operator is only asked to apply a repeatable force profile on the leader robot continuously. Meanwhile, the code runs the MPC controller on both robots, measures the performance metric, performs the BO computations, proposes new weights, and switches the controller weights automatically for the next trial. The code automatically finishes the tuning after a predefined number of trials n_{exp} and finds the optimal weights.

The experimental setup includes two identical 1-DoF robots which are shown in Figure 2.4. Each robot is composed of a *Maxon* motor, which is controlled by *EPOS3* driver board, and connected to a high-precision encoder with 4000 readings per rotation. The motor shaft is connected to the robot joint by a cable-driven capstan mechanism with a reduction ratio of 10. Finally, a load cell force sensor is embedded in each robot handle to measure the external forces. The follower robot is in contact with a linear spring, which could be approximated by a rotational spring in the range of motion with stiffness $k_e = 0.7$ Nm.rad^{-1}. The communication is implemented under *EtherCat* protocol.

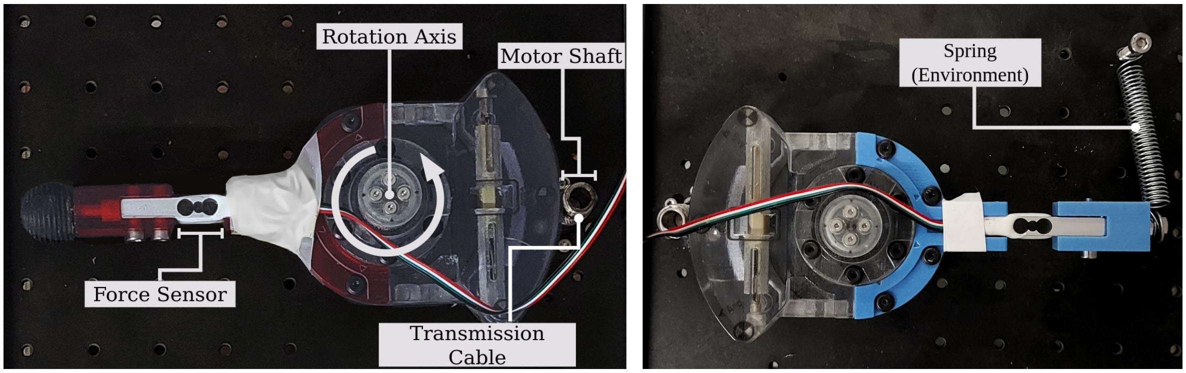

Figure 2.4: The hardware setup. The leader robot is on the left, and the follower robot is on the right. The follower robot is attached to a spring (the environment).

The robot models were identified using *MATLAB* Identification Toolbox. The model parameters for the leader and follower robots in (2.3) and (2.6) are reported in Table 2.1. The identified values of the model parameters are used in the simulation for comparison.

Table 2.1: Identified parameters of the robots

robot moment of inertia	$m_l = m_f = 0.00092$	kg.m^2
robot damping coefficient	$b_l = b_f = 0.0032$	Nm.s.rad^{-1}
environment stiffness	$k_e = 0.7$	Nm.rad^{-1}
environment damping coefficient	$b_e = 0.0$	Nm.s.rad^{-1}

All forces and torques are projected on the joint axis and measured as torques (Nm) for comparison and plotting. The sampling time is set to $T_s = 0.002$ s, and the MPC prediction horizon is set to $N = 15$ time steps. We set a constraint on both control inputs as follows: -0.33 Nm $\leq f_l, f_f \leq 0.33$ Nm.

2.3.2 Simulation Results

The simulation was done in ROS2 with the exact same code that will be tested on the hardware later. The BO performs the auto-tuning using 9 simulated interactions, and the results are reported in Figure 2.5. Figure 2.5(a) illustrates the performance metric computed at the end of each trial, plotted with time, which corresponds to the actual time the simulation and optimization took. The first 3 points are chosen around the middle of the search space to initialize the surrogate model with no optimization yet. Afterward, a general downward trend can be noticed, indicating that the algorithm is finding better weights, with occasional rises due to exploration. In the bottom part of Figure 2.5, we compare the results that correspond to different tested weights. In each subfigure, we show the positions and forces of the leader robot (blue), and the follower robot (red). In Figure 2.5(b), we show the performance of an example of the explored

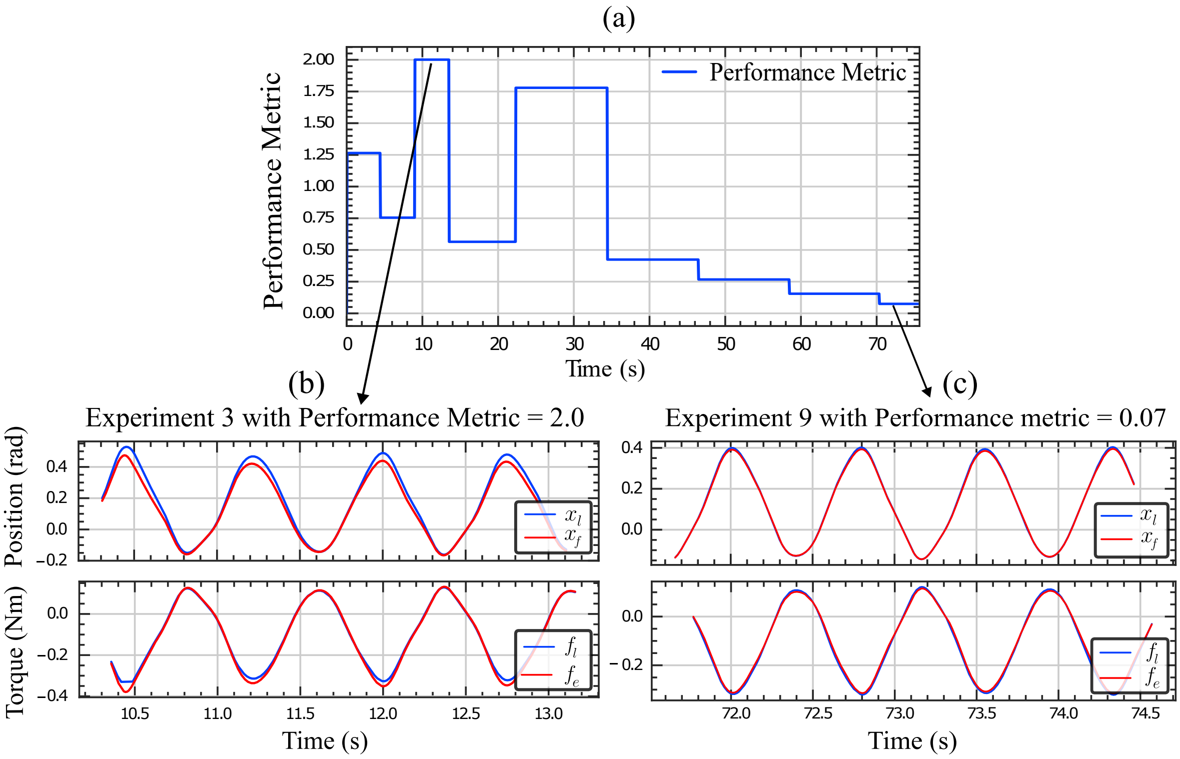

Figure 2.5: Auto-tuning experiment in simulation. (a) Performance metric value computed at the end of each trial and plotted with time. (b) The performance of the worst tested weights which shows the position and force signals on both sides. (c) The performance of the best tested weights which shows the position and force signals on both sides.

weights, which correspond to a high value of the performance metric (≈ 2, worst tested performance). Some errors in position and force tracking can be seen between both robots. In Figure 2.5(c), we present the performance of the last explored weights in the trials, which correspond to the lowest found value of the performance metric. Both position and force tracking are almost perfect with performance metric value close to zero (≈ 0.07, best tested performance).

We test the performance of the found optimal tuning which minimizes the learned surrogate model. The testing is done using the same type of input from the human operator. Figure 2.6 displays the performance of the optimal weights after the auto-tuning, where we plot the positions of both robots in Figure 2.6(a), and the velocities in Figure 2.6(b). In Figure 2.6(c), the environment torque and the leader robot's torque are shown, both projected on the robots' joint axis. In Figure 2.6(d), the moving average of the performance metric is depicted with a window of 3 seconds, which holds a low, near-constant value. Overall, almost perfect tracking in position, velocity, and force can be observed for the optimal auto-tuned weights.

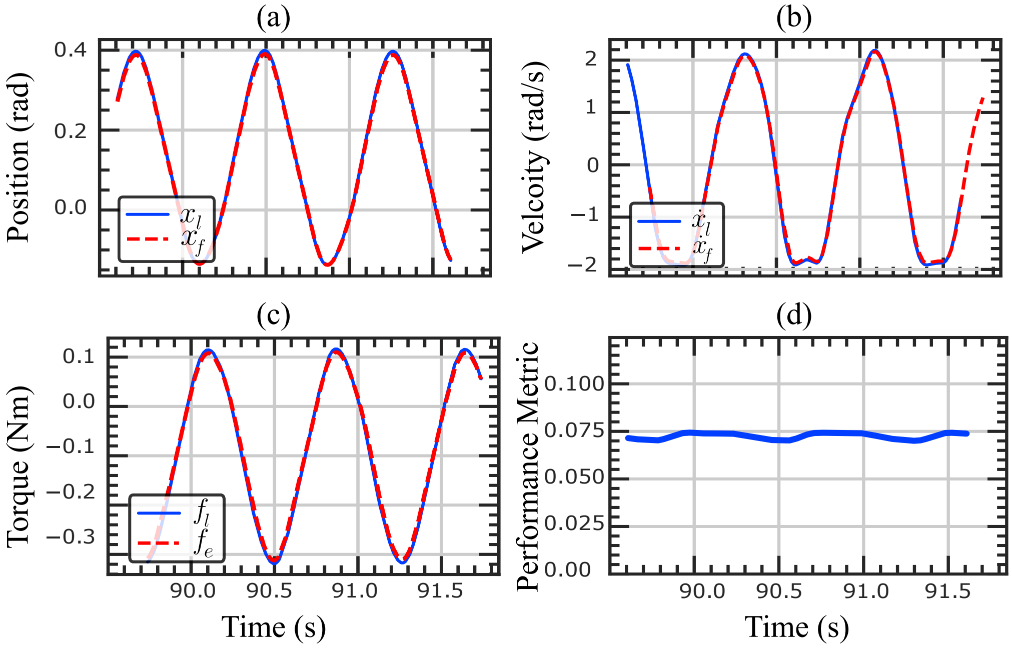

Figure 2.6: Performance of the found optimal weights in simulation with near-perfect tracking. (a) Positions of the leader and follower robots. (b) Velocities of the leader and follower robots. (c) Forces of the leader robot and the environment. (d) Moving-window average of the performance metric.

2.3.3 Hardware Experiments Results

The same protocol as the simulation part is repeated on hardware for comparison. The MPC auto-tuning is performed through 14 trials of 3 seconds each. The following figures present results analogous to those in the simulation subsection for direct comparison. Figure 2.7 illustrates the results of the hardware auto-tuning experiment. Figure 2.7(a) shows the performance metric, computed at the end of each trial, plotted with time. The model is initialized with 3 trials using 3 different weights chosen randomly to explore the search space. After that, the optimization starts and the performance metric starts decreasing. We show a comparison between different tested weights during the search for the optimal value. The comparison displays the improvement in the matching between the robots positions (x_l, x_f) and forces (f_l, f_e). In the bottom part of Figure 2.7, we compare the results that correspond to different tested weights. In each subfigure, we show the positions and forces of the leader robot (blue), and the follower robot (red).

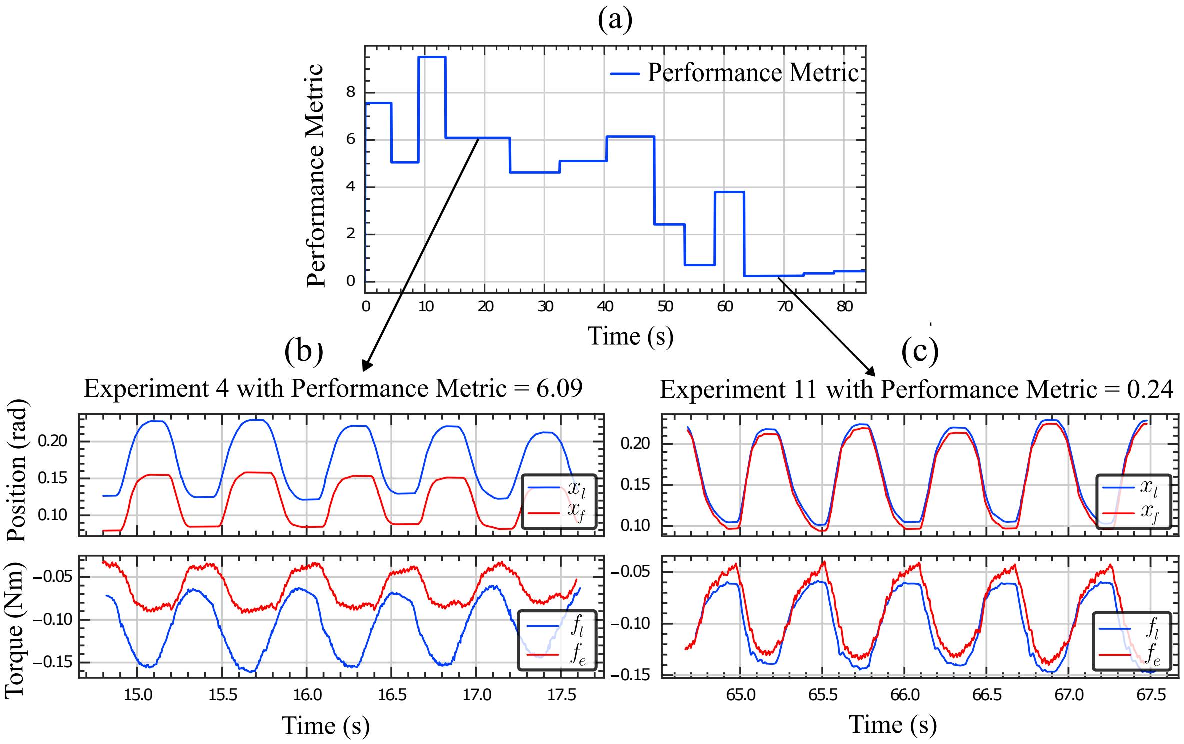

Figure 2.7: Auto-tuning experiment with hardware. (a) Performance metric value computed at the end of each trial and plotted with time. (b) The performance of bad tested weights which shows the position and force signals on both sides. (c) The performance of the best tested weights which shows the position and force signals on both sides.

In Figure 2.7(b), we show the performance of one example of the explored weights, which corresponds to a high value of the performance metric (≈ 6, bad performance). Large errors in position and force tracking can be seen between both robots. In Figure 2.7(c), we present the performance of the best found weights, which corresponds to the lowest value of the performance metric (≈ 0.24).

We present the performance of the found optimal controller in Figure 2.8, where the positions, velocities, and forces of both robots are plotted. A 3-second moving average of the performance metric is shown in Figure 2.8(d), which holds a low, near-constant value (≈ 0.25) due to the good controller performance.

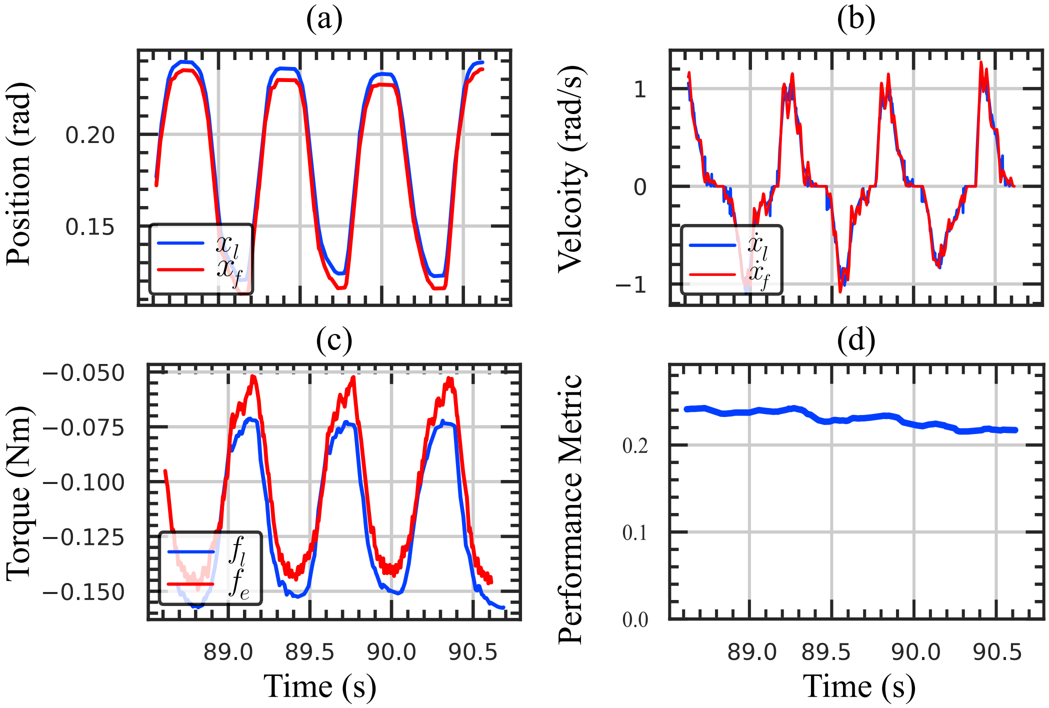

Figure 2.8: Performance of the found optimal weights in hardware experiment. (a) Positions of the leader and follower robots. (b) Velocities of the leader and follower robots. (c) Forces of the leader robot and the environment. (d) Moving-window average of the performance metric.

2.3.4 Discussion

Overall, the hardware experiment results are consistent with the simulation results. In both cases, the auto-tuning shows an improvement in performance compared to the initial tuning, without prior knowledge about the role of each parameter on the closed-loop performance. The auto-tuning achieved nearly perfect tracking in simulation. In the hardware experiment, however, we observe that while force and position tracking improves after tuning, it remains imperfect. This performance gap can be explained partially by unmodeled nonlinearities such as static friction, which are present in the physical hardware but not in our linear MPC model. Addressing such nonlinearities requires more than tuning the MPC cost weights. Since the goal of this work is to show the feasibility of auto-tuning, handling unmodeled nonlinearities is beyond the work scope. We observe in Figure 2.5 that even the initial performance was not very poor in simulation, which means that under a perfect model, most weights in the chosen area give acceptable results. This is unlike the initial weights in the hardware experiment in Figure 2.7, which gave poor tracking results. This shows the importance of tuning directly on hardware compared to using the optimal weights found in simple simulations. Even under a non-perfect model, the auto-tuning greatly enhanced the performance in the hardware experiment. Finally, the auto-tuning took around 80 seconds in hardware, which corresponds to 3 seconds for each trial, plus 1.5 seconds of transient time between the trials that we added in order to remove the effect of weight switching on the performance measure. The rest of the time is for the BO computations after each trial.

By the end of the exploration phase (i.e., after n_{exp} trials), we assume that a good surrogate model of the performance metric is built. Then we look for the weights that minimize the surrogate model assuming they are close enough to the minimizers of the real performance metric. As we mentioned before, we could have taken the best weights that were tested during the exploration phase, however, since we assume that those are noisy evaluations —particularly due to the challenge of ensuring perfect human force repeatability between trials— a better strategy is to minimize the learned GP surrogate model which has good noise rejection features due to its probabilistic nature. To understand the idea of noise rejection, we could think of the following example: if the behavior of the human who is performing the auto-tuning was very inconsistent, such that they did not excite the system well enough when they were testing certain weights, while they excited it too much for other weights. This could generate less tracking errors for the first case, which results in a (potentially misleading) best measured point. However, if enough points are tested, the probabilistic nature of the GP model can recognize this as an outlier and likely not identify it as an actual minimizer in the learned GP model. This is why we choose to minimize the learned GP model instead of taking the best point of the measurements. Note that the repeatability of the human force is important to obtain a fair comparison between different weights. On the other hand, some amount of human error can also be beneficial to the tuning, since it adds a range of useful frequencies to the input force instead of tuning for only one frequency, which can better approximate real-life scenarios.

Regarding the choice of the prediction horizon, we set $N = 15$ time steps, which we tuned empirically. Note that the value of N could also be added to the auto-tuning parameters in future work. Finally, in the performed hardware experiment, the stopping criterion was a maximum number of trials. Other stopping criteria could be used, such as a maximum time limit or to reach a threshold value of the performance metric.

2.4 Chapter Conclusion

In this chapter, we presented an MPC formulation for bilateral teleoperation which optimizes for transparency. After that, and motivated by the effort required for tuning the MPC weights, we presented an approach for auto-tuning the MPC which can be applied directly to hardware. We formulated the tuning problem as a black-box optimization and used the Bayesian Optimization algorithm to efficiently search for the optimal MPC weights, which gave good results in a few minutes in simulation and hardware experiments. One limitation of the method is that it requires a range of stable parameters to search from. Another limitation is that the method was tested on a single environment. More discussion about the limitations and future work will be presented in Chapter 5.

Chap. 3

Bilateral Teleoperation with Robust Constraints

While the MPC framework in Chapter 2 addressed constraints and transparency regardless of modeling errors, real-world environmental uncertainties can compromise safety by leading to constraint violations. This chapter tackles this challenge by developing an adaptive robust MPC controller for bilateral teleoperation. This approach guarantees robust constraint satisfaction under uncertainty while using online learning to enhance performance.

This chapter begins in Section 3.1 by motivating the need for robust constraints in teleoperation and reviewing relevant literature. Section 3.2 then details the proposed adaptive robust MPC methodology. Simulation results validating the approach are presented in Section 3.3, followed by the chapter conclusion in Section 3.4.

3.1 Introduction and Literature Review

3.1.1 Motivation for Robust Constraints in Teleoperation

Bilateral teleoperation systems that use MPC are sensitive to uncertainties in the environment, such as unknown stiffness, damping, or dynamic changes in contact conditions. These uncertainties pose a challenge in ensuring constraint satisfaction, which is critical for safe and effective teleoperation. For instance, in surgical robotics, exceeding force constraints can lead to tissue damage or violate position or velocity limits.

Existing MPC approaches in teleoperation struggle with two primary limitations: (1) the lack of robustness to uncertainties in the environment model, and (2) the absence of recursive feasibility guarantees. Without robustness, the controller may produce solutions that fail to account for worst-case scenarios, leading to constraint violations. Similarly, without recursive feasibility, the optimization problem may become infeasible during operation, causing failure in system control.

Addressing these issues is crucial for advancing the deployment of MPC in teleoperation for risky scenarios. A long-term vision is to enable MPC to handle uncertain and dynamic environments, where the model parameters and uncertainty bounds are estimated using traditional or advanced techniques such as computer vision or artificial intelligence. With these estimations, the controller would be able to deal robustly with the uncertainty. By incorporating such guarantees, teleoperation systems can achieve a new level of safety and reliability, paving the way for broader adoption in critical applications such as risky industrial manipulation, remote exploration, or robotic surgery.

This chapter aims to design an MPC control approach for robust constraint satisfaction in a bilateral teleoperation scenario. The focus will be on studying feasibility and conservatism, and re-evaluating the difficulties that can arise from such a control design. This represents a step towards incorporating such methods for more reliable real-life applications.

3.1.2 Literature Review

In the following, we will review the main related topics in the literature to illustrate our position in the field and to justify our choices. We begin by discussing the different types of robustness in bilateral teleoperation. We then present works addressing robust constraints, and we finally introduce the robust MPC techniques and choose the one that fits our goals.

3.1.2.1 Robustness in Bilateral Teleoperation

Robustness in bilateral teleoperation is a critical aspect that has been extensively studied in the literature. Generally speaking, related works could be classified based on several criteria, such as the type of disturbance being handled, or the goal of the robustness. The robustness could be against different disturbances, such as constant time delay [142], time-varying delay [143], packet loss [106], model parameter uncertainty or a mix of these factors [144]. As for the goal of robustness, numerous works aimed to achieve robust stability and performance using a variety of control methods. For example, classical robust control techniques such as H_∞ design a feedback controller that minimizes the worst-case gain from disturbances to the system output. The controller is synthesized by solving an optimization problem with a set of LMIs [145]. The work in [58] used μ-synthesis, considering a mix of uncertainties and constant time delay, which achieved robust stability and performance. Other works that also used μ-synthesis include [57, 146, 147]. Passivity-based methods achieve robust stability by ensuring that the system never generates energy but only stores or dissipates it [21, 24, 60]. Passivity-based control could be considered a robust control method that focuses on robust stability, as such controllers usually guarantee stability for a wide range of situations, specifically for any passive environment and passive human operator [1].

While the literature on robustness often focuses on robust stability, some works gave special attention to achieving robust performance in order to enhance transparency under different teleoperation conditions. For instance, [148] developed several controllers for different environment parameters and proposed a robust switching method to switch between them smoothly based on the environment to achieve better performance. In [59], the authors exploited the bounds on the uncertainty in the environment model and used LMIs to construct a robust controller with predefined performance. Other papers that focus on the robust performance include [58, 146, 149, 150].

Sliding Mode Control (SMC) has also been used in bilateral teleoperation scenarios to enhance performance in the presence of uncertainties. SMC defines a sliding surface, which is a manifold in the state-error space along which the closed-loop dynamics exhibit the desired convergence properties. A discontinuous, high-gain control law is then synthesized to drive the trajectory onto this surface in finite time and keep it there, thereby maintaining "sliding" motion that is robust to matched disturbances and model uncertainties. The authors in [144] presented a robust adaptive-sliding-mode control scheme for bilateral teleoperation systems operating in the presence of bounded but unknown uncertainties and time-varying delays. The results showed significant improvement compared to a traditional (proportional + damping) controller. Other works that used SMC in bilateral teleoperation include [151–153].

3.1.2.2 Robust Constraints in Bilateral Teleoperation

Beyond maintaining stability, another important aspect of robust control is ensuring the satisfaction of state and input constraints despite uncertainties. We refer to this as the problem of robust constraint satisfaction, which means that for all possible realizations of uncertainty, the system's trajectories will not violate predefined operational limits. Many papers have addressed such constraints, for example, those on inputs, states, or tracking errors. Robust constraint satisfaction has been explored in the literature through various methods. For instance, in [154] a predictive strategy for teleoperation systems operating over communication channels with unknown and possibly unbounded time delays is presented. The method ensures stability and prevents constraint violations on input- and state-related variables regardless of the time delays. The approach is demonstrated on a leader-follower teleoperation system over the internet, where a cart/rod inverted pendulum is stabilized under constraints on the cart motor voltage and the rod angle. In [155], the authors used a terminal SMC and employed barrier Lyapunov functions to prove stability and ensure that synchronization error constraints are not violated. In [156], the authors used an MPC with mixed H_2/H_∞ approach for space bilateral teleoperation. The method addresses large time-varying delays and takes input constraints into account. However, it does not consider state constraints or model parameter uncertainty.

The work in [157] designed a data-driven H_∞ control method for discrete-LTI systems subject to norm-bounded disturbances. This method handles state and input constraints, ensuring robust performance and constraint satisfaction in bilateral teleoperation systems. The above-mentioned works assume fixed uncertainty bounds, which result in an overly conservative controller in case of large estimated bounds. In addition, robust methods generally optimize for the performance of the worst-case scenario. However, when the system is operating far from the constraints, better performance can be achieved by optimizing for typical or expected scenarios instead of the worst case.

In the context of MPC in bilateral teleoperation, the design usually includes constraints on position, velocities, and environmental force [5, 68, 112]. However, to the best of our knowledge, no work has specifically addressed handling input and state constraints in MPC-based bilateral teleoperation under model uncertainty. The work in this chapter aims to fill that gap by designing a robust MPC controller for bilateral teleoperation that ensures constraint satisfaction under both parametric uncertainty and additive disturbance in the system model. We focus on the computational efficiency of the controller, reducing conservatism, and enhancing performance. These goals will guide the choice of the robust MPC and learning methods in the following subsection.

3.1.2.3 Robust MPC Methods

The literature on robust MPC presents a wide range of methods to handle constraints robustly, as reviewed in [158–160]. Our goal is to implement a robust MPC for bilateral teleoperation while addressing the real-time constraints without being overly conservative with respect to the performance of the teleoperation system.

One of the most common methods for robust MPC is Tube-MPC [161]. Unlike traditional MPC, which predicts only the state trajectory based on a nominal model, Tube-MPC also predicts a whole tube around it that encompasses all the trajectories under all possible uncertainties. In discrete-time systems, the method finds a sequence of sets $\mathcal{P}_k$ that represent the tube cross-section at each time step. Figure 3.1 visually illustrates the polytopic tube concept, where the red sets are the tube cross-sections at different future time steps, which are constructed around the nominal predicted trajectory (black line). By controlling these polytopes to stay within the constraints (shown as a cyan vertical plane), we ensure that the true system trajectory remains within the safe region as well.

The variations of tube controllers mainly differ by the type and shape of the cross-section. Common implementations include: 1) ellipsoid cross-section which offers less control over the shape but has fewer parameters to handle (ellipsoid semi-axes, one length per dimension); 2) polytopic cross-section which provides more control over the shape but requires handling more parameters (vertices or inequalities) and does not scale well with high-dimensional state spaces. Some basics of polytopes are presented in a short section in Appendix B.1 to help the reader understand the following sections.

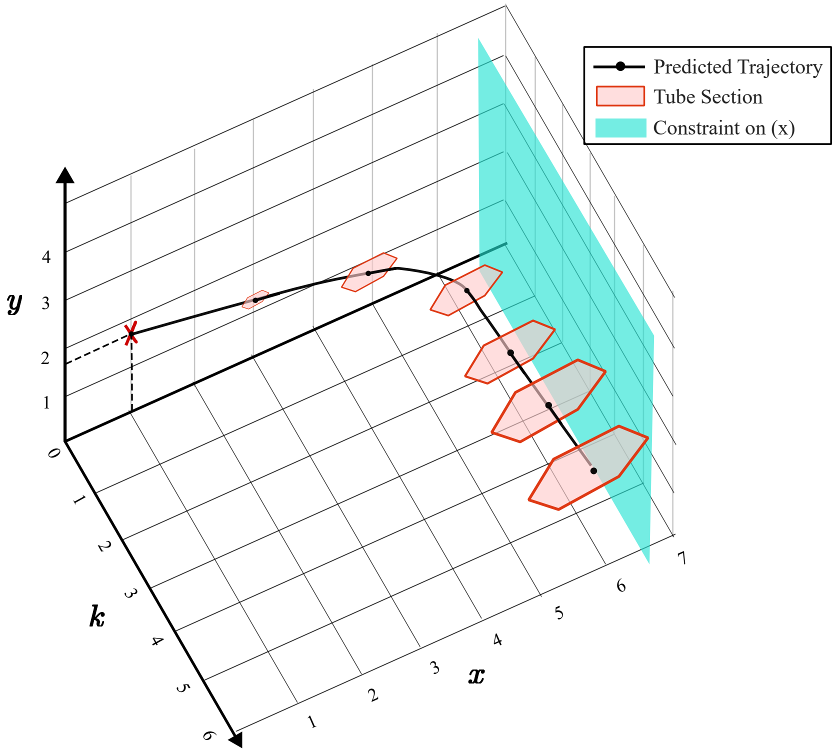

Figure 3.1: An illustration of the Tube-MPC concept. Axes x and y represent the state space. Axis k represents the future discrete time steps. The blue vertical plane represents a constraint in the state space ($x \leq x_0$). The red cross is the current state. The red polygons (polytopes) are the tube sections at each predicted time step (i.e., $\mathcal{P}_k$), and the black line is the nominal predicted trajectory. Note that the tube sections are translated and scaled versions of the same polytope, and they do not intersect with the constraint, meaning that all the trajectories in the tube respect the constraint.

Since its inception, many variations of Tube-MPC have been developed. These variations mainly differ in the type and flexibility of the cross-section. For instance, the following types of Tube-MPC have been proposed in the literature:

- Rigid Tube-MPC: allows only translations of the initial set $\mathcal{P}_0$ which makes the cross-section fixed [162].
- Homothetic Tube-MPC (HTMPC): allows both translations and scaling of the initial set $\mathcal{P}_0$ [163], as shown in Figure 3.1.
- Elastic Tube-MPC: permits scaling the vertices of $\mathcal{P}_0$ separately, potentially changing the shape while maintaining convexity [164].
- Configuration-Constrained Tube-MPC: optimizes the shape of the polytopic tube subject to conic vertex configuration constraints and associated vertex control laws by solving convex optimization problems online [165].
- Fully-Parametrized Tube-MPC: enables the optimization of all parameters defining the tube, offering maximum flexibility in adapting its shape [166].

The choice of the type of the Tube-MPC depends mainly on the computational complexity and the level of controller conservatism. HTMPC offers a balance between conservatism and computational complexity [164, 167]. For that reason, we choose to design an HTMPC for our application as a starting point to test the feasibility of the method. In the following sections, we will detail our implementation of an HTMPC controller and describe how data-driven learning methods are integrated to refine the uncertainty bounds, thereby reducing conservatism while maintaining robustness.

3.1.3 Teleoperation System Modeling with Uncertainties

As in the previous chapter, we consider that the follower robot is always in contact with an environment that can be modelled by an LTI system with stiffness k_e^* and damping coefficient b_e^*.

Starting from an initial estimate of these parameters k_e^0 and b_e^0, we bound the possible parameter offsets by a known compact polytope

$$\Theta = \left\{\theta \in \mathbb{R}^2 \mid H_\Theta\, \theta \leq h_\Theta\right\} \tag{3.1}$$

where H_Θ and h_Θ are the matrices that define the bounding polytope in the H-representation (see Appendix B.1) and they are assumed to be known. θ is the uncertainty parameter defined as

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \Delta k_e \\ \Delta b_e \end{bmatrix} = \begin{bmatrix} k_e - k_e^0 \\ b_e - b_e^0 \end{bmatrix} \in \Theta \tag{3.2}$$

so that the true offset

$$\theta^* = \begin{bmatrix} k_e^* - k_e^0 \\ b_e^* - b_e^0 \end{bmatrix} \tag{3.3}$$

also satisfies $\theta^* \in \Theta$. Figure 3.2 illustrates the initial guess and the true value of the environment parameters and the uncertainty. We assume that the system state is

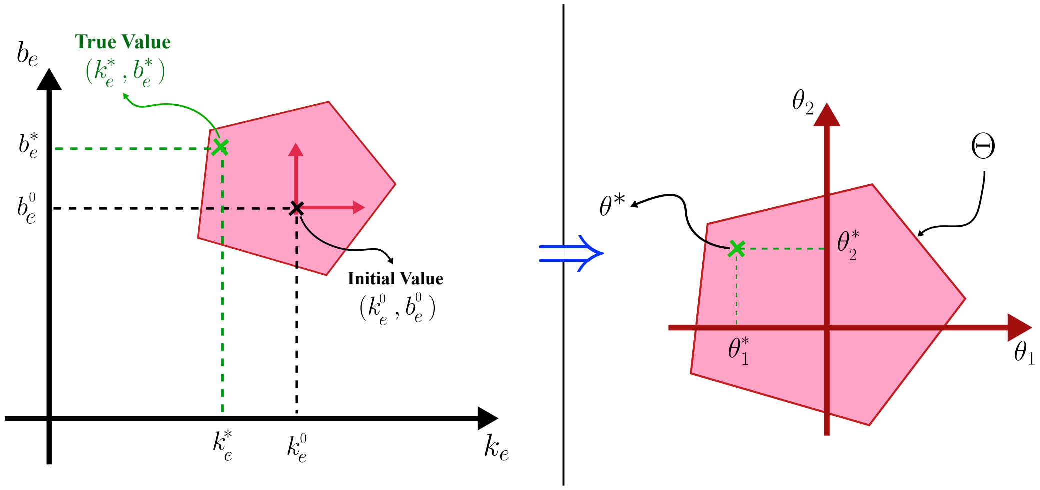

Figure 3.2: An illustration of the initial guess and the true value of the environment model parameters, with the polytopic bounding set Θ, represented in the (k_e, b_e) coordinates (left), and the uncertainty (θ_1, θ_2) coordinates (right)

also affected by an additive uncertainty $d(t) \in \mathcal{D}$ where $\mathcal{D} \in \mathbb{R}^{n_x}$ is a known compact polytopic set. The system dynamics from (2.7) can be rewritten to include uncertainties θ and d as follows

$$\begin{aligned} \dot{x} &= A_{c,\theta} x + B_c u + d \\ y &= C_{c,\theta} x \end{aligned} \tag{3.4}$$

where $A_{c,\theta}$ and $C_{c,\theta}$ are the system's continuous-time matrices with the parametric uncertainty. Note that the matrix B_c defined in (2.7) is not impacted by θ since it does not depend on the environment model. Using Euler discretization with a sampling time

T_s, the discrete dynamics can be written as

$$\begin{aligned} x_{\tau+1} &= A_\theta x_\tau + Bu_\tau + d_\tau \\ y_\tau &= C_\theta x_\tau \end{aligned} \tag{3.5}$$

where τ is the discrete time step, $d_\tau = d(\tau)$ is the discrete additive uncertainty at time step τ, and the matrices are defined as

$$\begin{aligned} A_\theta &= I + A_{c,\theta} T_s \\ B &= B_c T_s \\ C_\theta &= C_{c,\theta} \end{aligned} \tag{3.6}$$

where I is the identity matrix with the appropriate size. The resulting matrices A_θ and C_θ are affine in the parameter θ such that

$$\begin{aligned} A_\theta &= A_0 + A_1\theta_1 + A_2\theta_2 \\ C_\theta &= C_0 + C_1\theta_1 + C_2\theta_2 \end{aligned} \tag{3.7}$$

where A_0, A_1, A_2, C_0, C_1, and C_2 are constant known matrices. A_0 and C_0 are the initial system matrices with the initial parameters k_e^0 and b_e^0. The explicit formulas of the uncertain matrices mentioned here are given in Appendix B.2. We also define the following matrix to quantify the effect of the parametric uncertainty on the system dynamics

$$D_x(x_\tau) = [A_1 x_\tau, A_2 x_\tau] \tag{3.8}$$

where $D_x(x_\tau) \in \mathbb{R}^{n_x \times n_\theta}$, and the comma separates between the two columns $A_1 x_\tau$ and $A_2 x_\tau$. With this new matrix, the dynamics from (3.5) can be rewritten as

$$x_{\tau+1} = A_0 x_\tau + D_x(x_\tau)\theta + Bu_\tau + d_\tau \tag{3.9}$$

As can be seen from this equation, the matrix $D_x(x_\tau)$ defines the effect of the parametric uncertainty on the system dynamics.

After reviewing the related works and presenting the necessary mathematic formulations, in this chapter, we design a robust Tube-MPC controller for bilateral teleoperation, and we use learning methods to enhance the performance of the controller and reduce the conservatism. The main tasks required to achieve a robust and adaptive controller are threefold: first, to robustly guarantee the satisfaction of the MPC constraints on the states and inputs despite the parametric and additive uncertainties; second, to reduce the conservatism of the controller by refining the uncertainty bounds through learning from data, that is by finding a tighter set $\Theta_{\text{new}} \subset \Theta$ which robustly contains the true parameter θ^*; and third, to find a point estimate of the parameter θ^*, meaning to find a point $\hat{\theta}$ that is close to θ^*.

3.2 Adaptive Robust MPC for Bilateral Teleoperation

In this section, we present the designed adaptive robust MPC controller for bilateral teleoperation. The method is based on the work by Kohler et al. [168] and adapted to the context of bilateral teleoperation. Due to the complexity of the method, we will present it in a modular and simplified way, and we will refer to the Appendix and the literature for some details. We first present the ideas related to robust Tube-based MPC. We then explain the concepts related to learning the uncertainty bounds and the model parameters. Finally, we present the full controller formulation that combines both aspects of learning and robustness.

3.2.1 Robust Tube-MPC Controller

3.2.1.1 Tube-MPC Controller Concept

The core objective of Tube-based MPC is to ensure robust stability and constraint satisfaction despite model uncertainties. Rather than predicting a single nominal trajectory, this approach computes a tube, which is a sequence of sets $\{\mathcal{P}_k\}_{k=1}^{N}$ around the nominal trajectory $\bar{x}_k$ (as was illustrated before in Figure 3.1). This tube encloses all possible state trajectories under the uncertainties.

The general robust MPC optimization problem can be written as a modified version of the nominal MPC problem (2.1) as follows

$$U^* = \arg\min_{U} \left(\mathcal{J}_f(\bar{x}_N) + \sum_{k=0}^{N-1} \mathcal{J}_s(\bar{x}_k, u_k) \right) \tag{3.10}$$

$$s.t. \quad x_0 = \bar{x}_0 = x(\tau) \qquad \text{(measured state)} \tag{3.11}$$

$$\bar{x}_{k+1} = A_0 \bar{x}_k + B u_k \qquad \text{(nominal model)} \tag{3.12}$$

$$x_{k+1} = A_\theta x_k + B u_k + d_k \qquad \text{(actual system)} \tag{3.13}$$

$$x_k \in \mathcal{X}, x_N \in \mathcal{X}_f \tag{3.14}$$

$$u_k \in \mathcal{U} \tag{3.15}$$

$$\forall d_k \in \mathcal{D}, \forall \theta \in \Theta, k \in \{0 \dots N-1\} \tag{3.16}$$

where x_k is the true system state at the future time step k. $\bar{x}_k$ is the nominal state at time step k predicted using the nominal model. Note that the constraints (3.14) are defined on the true state trajectory x_k which is unknown and depends on the uncertain parameters θ and d_k.

Solving problem (3.10) for all values of d_k and θ is not tractable in the general case since it has an infinite number of constraints. To make the problem computationally feasible, HTMPC turns the constraints into a finite number of constraints. The technique consists of two main steps that will be explained more in the next subsections. The first step is to design a sequence of polytopic sets $\{\mathcal{P}_k\}_{k=1}^{N}$ such that they contain all possible uncertain trajectories by construction, which will be a useful feature. The second step is to reexpress the constraints on the infinite number of trajectories x_k as a finite number of *tightened constraints* on the nominal trajectory $\bar{x}_k$ by using the mentioned feature of $\mathcal{P}_k$. This means that $\bar{x}_k$ should satisfy the original constraints with an additional appropriate margin to account for the uncertainty.

3.2.1.2 Constructing the Sets $\mathcal{P}_k$ to Contain the Uncertain Trajectories

The tube sections $\mathcal{P}_k$ are constructed such that they contain all uncertain trajectories, i.e., the following condition should be guaranteed

$$\begin{gathered} x_k \in \bar{x}_k \oplus \mathcal{P}_k \\ \forall d_k \in \mathcal{D}, \forall \theta \in \Theta, k \in \{1 \dots N\} \end{gathered} \tag{3.17}$$

where $\oplus$ is the Minkowski sum (Appendix B.1.2), which can be seen as translating the set $\mathcal{P}_k$ by the vector $\bar{x}_k$ in the state space. In HTMPC, to have a fixed computational complexity, the sets $\mathcal{P}_k$ are designed as scaled and translated versions of a fixed polytopic set $\mathcal{P}_0$ as follows

$$\begin{aligned} \mathcal{P}_k &= s_k \mathcal{P}_0 \\ \mathcal{P}_0 &= \{x \in \mathbb{R}^{n_x} : H_{\mathcal{P}_0} x \leq 1_{n_{\mathcal{P}_0}}\} \end{aligned} \tag{3.18}$$

where $\mathcal{P}_0$ is a polytopic set fixed offline and represented in the H-representation by $n_{\mathcal{P}_0}$ inequalities defined by the matrix $H_{\mathcal{P}_0} \in \mathbb{R}^{(n_{\mathcal{P}_0} \times n_x)}$ and the vector of ones $1_{n_{\mathcal{P}_0}} \in \mathbb{R}^{n_{\mathcal{P}_0}}$. Online, $\mathcal{P}_0$ is translated by $\bar{x}_k$ and scaled by the scalar s_k to form the cross-sections of the tube. This makes the task of constructing the sets $\{\mathcal{P}_k\}_{k=1}^N$ equivalent to finding the scaling parameters $\{s_k\}_{k=1}^N$ such that (3.17) holds. The set $\mathcal{P}_{k+1}$ is computed based on $\mathcal{P}_k$ such that it accounts for the propagation of all the states $x_k \in \mathcal{P}_k$ while considering all uncertainties. Therefore, the scaling parameters $\{s_k\}_{k=0}^N$ from (3.18) are calculated recursively to capture both the parametric and additive uncertainties [168]. Specifically, s_k is propagated via the following dynamics

$$s_{k+1} = \rho(\Theta) s_k + w_k(\Theta, \mathcal{D}) \tag{3.19}$$

where $\rho(\Theta)$ is the maximum tube contraction rate over the uncertainty bounding set Θ, which describes the contraction of the tube assuming a stable system, and it depends on the system dynamics without the MPC. $w_k(\Theta, \mathcal{D})$ captures the growth of the tube due to uncertainties. The work in [168] finds $\rho(\Theta)$ by solving a separate optimization problem, and it is resolved only if Θ is changed. For w_k, the authors add it as an additional decision variable in the MPC optimization problem such that it is found by the MPC solver. These dynamics will be used and discussed later in the adaptive robust MPC formulation in subsection 3.2.3 and then detailed in Appendix B.3.

3.2.1.3 Reformulating the Constraints into Tightened Constraints

After building $\mathcal{P}_k$ such that it satisfies (3.17) by construction, the constraints on the true state x_k defined in (3.14) and (3.16) can be reformulated as a finite number of tightened constraints on the nominal state $\bar{x}_k$. That is, for the following conditions to hold

$$\begin{aligned} & x_k \in \mathcal{X}, x_N \in \mathcal{X}_f \\ & \forall d_k \in \mathcal{D}, \forall \theta \in \Theta, k \in \{0 \ldots N-1\} \end{aligned}$$

it is sufficient to satisfy the following condition

$$\bar{x}_k \oplus \mathcal{P}_k \subseteq \mathcal{X} \tag{3.20}$$

and the same for the final state x_N

$$\bar{x}_N \oplus \mathcal{P}_N \subseteq \mathcal{X}_f \tag{3.21}$$

This means that the whole tube section should be within the constrained set $\mathcal{X}$, which is a sufficient condition for the true state x_k to be within the constraints as well. Using this, the constraints on x_k can be replaced using (3.18), (3.19), (3.20), and (3.21) to rewrite the optimization problem (3.10) as follows

$$(U^*, W^*) = \underset{U,W}{\arg\min} \left(\mathcal{J}_f(\bar{x}_N) + \sum_{k=0}^{N-1} \mathcal{J}_s(\bar{x}_k, u_k) \right) \tag{3.22}$$

$$\tag{3.23}$$

$$\begin{aligned}
s.t. \quad & x_0 = \bar{x}_0 = x(\tau) \quad , \quad s_0 = 0 && (3.24)\\
& \bar{x}_{k+1} = A_0 \bar{x}_k + B u_k && (3.25)\\
& \bar{x}_k \oplus s_k \mathcal{P}_0 \subseteq \mathcal{X} \quad , \quad \bar{x}_N \oplus s_N \mathcal{P}_0 \subseteq \mathcal{X}_f \qquad \text{(tightened constraints)} && (3.26)\\
& s_{k+1} = \rho(\Theta) s_k + w_k(\Theta, \mathcal{D}) && (3.27)\\
& w_k \geq f_w(\bar{x}_k, s_k, \Theta, \mathcal{D}) && (3.28)\\
& u_k \in \mathcal{U} && (3.29)\\
& \forall k \in \{0 \ldots N-1\} && (3.30)
\end{aligned}$$

where f_w is a function that describes the dynamics of the tube growth due to uncertainties at the point $(\bar{x}_k, s_k)$. W is the vector of decision variables that contains the tube growth w_k for all time steps, and W^* is the optimal value of W that is to be found by the MPC solver. Note that this formulation no longer has the infinite number of constraints as (3.16). However, the condition (3.28) on f_w and the set inclusion condition (3.26) have to be reformulated into a standard form that can be solved by the MPC solver. To simplify the presentation, the steps to reformulate these conditions into a set of linear inequalities are presented in Appendix B.3 and B.4. However, a short summary of simplifying (3.26) is provided in subsubsection 3.2.1.5 below, after presenting a necessary step to reduce the tube growth.

3.2.1.4 Reducing the Tube Growth

One problem with the uncertainty tube is that it can grow significantly over time if the uncertainty is not negligible. This can lead to overly conservative control actions and reduced performance. To mitigate this issue, two strategies are commonly employed in the design of Tube-MPC controllers. First, a pre-stabilizing lower-level controller K is designed offline and applied to the system to reject the uncertainties and keep all the trajectories close to the nominal trajectory. The MPC is applied on top of K such that the control input to the system is decomposed as follows

$$u_\tau = Kx_\tau + v_\tau \tag{3.31}$$

where v_τ is an additional control input calculated by the MPC to achieve the desired goals, that is to minimize the cost function and respect the constraints. In this case, the MPC controls the new pre-stabilized system that has the modified dynamics from (3.5) as follows

$$x_{\tau+1} = A_{cl,\theta}x_\tau + Bv_\tau + d_\tau \tag{3.32}$$

where $A_{cl,\theta} = A_\theta + BK$.

Second, $\mathcal{P}_0$ is often designed to be the so-called maximal contractive polytopic set [169] or the minimal robust positively invariant set [170] for the pre-stabilized system. This means that the trajectories stay in a bounded or even shrinking set under the effect of K when the MPC is turned off.

Note that when the controller K is applied as in (3.31), the MPC constraints on the input (3.29) should be also tightened due to the fact that K applies a control input that depends on the uncertain system. The tightened input constraints can be expressed as follows

$$K(\bar{x}_k \oplus s_k\mathcal{P}_0) + v_k \subseteq \mathcal{U} \tag{3.33}$$

3.2.1.5 Writing the Tightened Constraints in a Standard Form

To implement the HTMPC (3.22) with a real-time solver, the tightened constraints are reformulated from the set-inclusion form (3.26) and (3.33) to linear inequalities. In particular, Kohler et al. proposed an efficient formulation that precomputes some constant parameters offline based on the shape of $\mathcal{P}_0$ and rewrites the online constraints in a linear form [168]. As explained in Appendix B.4, this can be done using the offline-precomputed constant scalars for each constraint row j in the original constraints (2.14) as follows

$$c_j = \max_{x \in \mathcal{P}_0} ([H_{\mathcal{X}}]_j + [H_{\mathcal{U}}]_j K)x, \quad j \in \{1, \dots, n_c\} \tag{3.34}$$

where $H_{\mathcal{X}}$ and $H_{\mathcal{U}}$ are the matrices that describe the constraints defined in (2.14). These constants are then used to project the tightened constraints from (3.26) into linear inequalities

$$s_k c_j + ([H_{\mathcal{X}}]_j + [H_{\mathcal{U}}]_j K)\bar{x}_k + [H_{\mathcal{U}}]_j v \leq 1 \tag{3.35}$$

Note that this is equivalent to adding a margin to the original constraints, where the margin is equal to c_j scaled by s_k for each constraint row j. The detailed derivation of these constants is provided in Appendix B.4. The robust Tube-MPC formulation will be used later after the introduction of the data-driven learning methods in the following sections.

3.2.2 Learning the Model

A notable drawback of robustness in control design is the inherent conservatism of the controller. To ensure robust constraint satisfaction for all $\theta \in \Theta$, the MPC might select a suboptimal solution, sacrificing performance for robustness. This conservatism becomes more pronounced if the model uncertainty is large, leading to less optimal solutions for the true system. To mitigate this, data-driven techniques such as Set Membership Learning (SML) can be employed to refine the estimation of the uncertainty bounds Θ [171,172].

3.2.2.1 Set-Membership Learning

Unlike traditional learning methods that estimate the value of the true parameter θ^*, SML focuses on identifying a set of parameters that robustly encapsulates θ^*. This approach aligns well with the requirements of robust Tube-based controllers. Starting from a parametric uncertainty bounding set Θ_0, each observation of the state transition from time step $\tau - 1$ to τ could be used to reduce the uncertainty bound. The new bounding set Θ_τ is found by eliminating from $\Theta_{\tau-1}$ the regions that are inconsistent with the observed state transition. This update leverages the fact that there exists a specific set of parameters $\Delta_\tau \subseteq \mathbb{R}^{n_\theta}$ capable of explaining the state transition under a bounded additive disturbance $d_\tau \in \mathcal{D}$. If θ^* remains constant, it will lie within each

Δ_τ, and thus within the following intersection

$$\theta^* \in \bigcap_{\tau=0,1,\ldots} \Delta_\tau \tag{3.36}$$

Figure 3.3 illustrates the concept of SML, where it shows the initial uncertainty set Θ_0 and the feasible sets Δ_τ at different time steps. In dark purple is the resulting feasible set Θ_τ after the intersection of all Δ_τ with Θ_0.

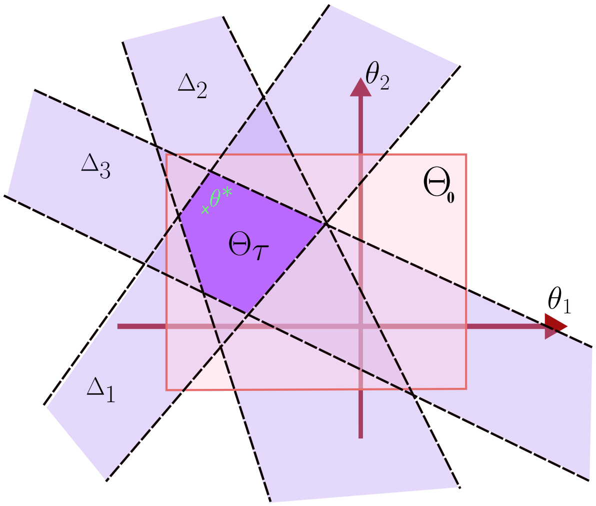

Figure 3.3: An illustration of the Set-Membership Learning concept. The initial bounding set Θ_0 is shown in light red, and the feasible sets Δ_τ are shown as strips in the space. The intersection of all Δ_τ with Θ_0 gives a tighter bound Θ_τ (purple), which contains the true parameter θ^* (green cross).

To explain the exact steps of SML, we start by defining the initial uncertainty bounding set Θ_0 and the additive uncertainty set $\mathcal{D}$ as the polytopes

$$\Theta_0 = \{\theta : H_{\Theta_0}\theta \leq h_{\Theta_0}\} \tag{3.37}$$

$$\mathcal{D} = \{d : H_\mathcal{D} d \leq h_\mathcal{D}\} \tag{3.38}$$

where $H_{\Theta_0}, H_\mathcal{D}$ and $h_{\Theta_0}, h_\mathcal{D}$ are the matrices and vectors with appropriate sizes that define the polytopes. We assume that these matrices are known since we have an estimated upper bound on the uncertainties. After the transition from τ to $(\tau + 1)$, the additive uncertainty bound is written from the uncertain system dynamics (3.9) as follows

$$d_\tau = x_{\tau+1} - (A_0 x_\tau + D_x(x_\tau)\theta + Bu_\tau) \in \mathcal{D} \tag{3.39}$$

Rewriting this using the definition of $\mathcal{D}$ from (3.38) gives

$$H_{\mathcal{D}}(x_{\tau+1} - (A_0 x_\tau + D_x(x_\tau)\theta + Bu_\tau)) \leq h_{\mathcal{D}} \tag{3.40}$$

From this inequality, new information about the true value of θ can be derived

$$\begin{aligned} &H_{\mathcal{D}}(x_{\tau+1} - A_0 x_\tau - D_x(x_\tau)\theta - Bu_\tau) \leq h_{\mathcal{D}} \\ &\qquad\qquad \Rightarrow \underbrace{-H_{\mathcal{D}} D_x(x_\tau)}_{H_{\Delta_\tau}} \theta \leq \underbrace{h_{\mathcal{D}} - H_{\mathcal{D}}(x_{\tau+1} - A_0 x_\tau - Bu_\tau)}_{h_{\Delta_\tau}} \\ &\qquad\qquad \Rightarrow H_{\Delta_\tau}\theta \leq h_{\Delta_\tau} \end{aligned} \tag{3.41}$$

Here, $H_{\Delta_\tau}\theta \leq h_{\Delta_\tau}$ defines the region Δ_τ, containing parameter values that can explain the transition $(\tau \rightarrow \tau + 1)$ under the bounded additive uncertainty assumption. Since θ^* is constant, the feasible parameter set evolves as

$$\forall \tau \geq 0, \quad \Theta_\tau = \{\theta : \theta \in \Theta_0 \cap \Delta_0 \cap \cdots \cap \Delta_\tau\} \tag{3.42}$$

At each time step, the uncertainty bounding set is refined by intersecting it with the latest Δ_τ.

One potential problem is the increasing computational complexity of the intersection with time, since the polytope Θ_τ can have a large number of sides. This issue can be mitigated by several methods, for example by limiting the number of sides. The work in [168] used a hypercube that overapproximates the feasible set alongside with keeping the last $M \in \mathbb{N}$ sets Δ_τ in a moving window fashion to compute a tighter feasible set Θ_τ^{HC}. The hypercubic set is finally defined by its center $\bar{\theta}_\tau$ and size η_τ as

$$\Theta_\tau^{HC} = \bar{\theta}_\tau \oplus \eta_\tau \mathbb{B}_{n_\theta} \tag{3.43}$$

where $\mathbb{B}_{n_\theta} = \{\theta \in \mathbb{R}^{n_\theta} | \|\theta\|_\infty \leq 0.5\}$ is the unit hypercube in the parametric space.

This restriction to a hypercube reduces the computational complexity of the intersection and the tube controller, but at the cost of less efficient learning or slower convergence. However, in applications such as teleoperation, real-time responsiveness take priority over controller conservatism. An illustration of the overapproximation idea is shown in Figure 3.4, where the bounding set Θ_τ is shown in dark purple, and its hypercubic overapproximation Θ_τ^{HC} is shown in blue. The center of the hypercube $\bar{\theta}_\tau$ is shown as a blue cross, and the size of the hypercube is defined by the side length η_τ.

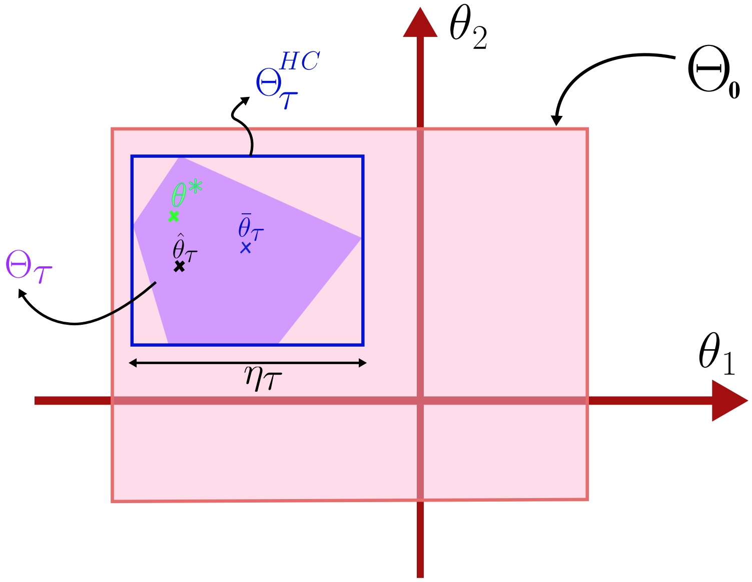

Figure 3.4: An illustration of the parameters learning with SML and LMS. The true value of the environment parameter θ^* is shown as a green cross. The initial bounding set Θ_0 is shown in light red. The learned bounding set Θ_τ is shown in purple, and its hypercubic overapproximation Θ_τ^{HC} is shown in blue, with its center $\bar{\theta}_\tau$ and side length η_τ. The point estimate $\hat{\theta}$ learned by LMS is shown as a black cross.

3.2.2.2 Learning for Performance

The Tube-MPC guarantees robust constraint satisfaction by ensuring that there always exists at least one feasible trajectory respecting constraints for all possible uncertainty realizations. Nevertheless, the MPC still has some freedom to choose among multiple feasible solutions, particularly when the states and control inputs are far from constraint boundaries. This choice is guided by the MPC cost function. To optimize performance, the cost function should ideally be designed using the most accurate model available. This corresponds to a model based on a point estimate $\hat{\theta}$ of the parameters, ideally very close to the true (unknown) value θ^*. A straightforward approach for such point estimation is choosing the center of the uncertainty set, i.e., $\hat{\theta}_\tau = \bar{\theta}_\tau$, which represents the center of the hypercube Θ_τ^{HC}. More sophisticated approaches for parameter estimation include the Least Mean Squares (LMS) or Kalman filtering, which iteratively refine the parameter estimate as new data becomes available.

An improvement on these estimation methods could be done by using the robust information of $\theta^* \in \Theta_\tau^{HC}$. In particular, $\hat{\theta}_\tau$ can be projected on Θ_τ to avoid bad scenarios of overshooting or diverging estimation.

Figure 3.4 illustrates the two types of parameter learning, particularly SML and LMS. The figure shows the true value θ^* as a green cross. For the SML, it shows the initial bounding set Θ_0, and the results of the learning after τ time steps are a tighter learned bounding set Θ_τ, and its hypercubic overapproximation Θ_τ^{HC}. The figure also shows the point estimate $\hat{\theta}$ learned by LMS as a black cross. In the illustrated case, $\hat{\theta}$ is closer to the true value θ^* than the center of the bounding set $\bar{\theta}$, which means that by optimizing the cost function for this value, the performance of the controller is improved when applied to the true system.

3.2.3 Adaptive Robust MPC Formulation For Bilateral Teleoperation

After presenting the general concepts of the Tube-MPC and the model learning in the previous sections, this section presents the full adaptive robust MPC formulation for the bilateral teleoperation case.

3.2.3.1 Adaptive Robust Tube-MPC Controller Formulation

The implemented adaptive robust Tube-MPC takes advantage of two model learning methods at the same time, where SML learning reduces the controller conservatism by enlarging the feasible region while keeping the robust guarantees against uncertainties. On the other hand, LMS learning directly enhances the performance. To show how the MPC can simultaneously incorporate both learning methods, we present the following simple optimization example. For certain values $\hat{a}, \bar{a} \in \mathbb{R}$

$$\begin{aligned} u^* = \arg\min_{u\in\mathbb{R}} (||u - \hat{a}||^2) \\ \text{s.t. } u + \bar{a} \geq 0 \end{aligned} \tag{3.44}$$

In this optimization problem, the value of $\bar{a}$ defines the feasible region of u, while the value of $\hat{a}$ affects the optimal solution. If $\hat{a}$ lies within the feasible region, then the constraints are not active, and the problem becomes a non-constrained optimization, meaning that only the cost function is used. However, if the optimal solution is outside the feasible region, a suboptimal solution is chosen such that u^* stays within the feasible region.

Similar to this example, the MPC optimization problem could depend on two predicted trajectories, $\{\bar{x}_k\}_{k=0}^N$ which is used to define the robust feasible region, and $\{\hat{x}_k\}_{k=0}^N$ which is used in the cost function for performance. $\bar{x}_k$ is propagated using the model with $\bar{\theta}_\tau$, the uncertainty set center learned by the SML. While $\hat{x}_k$ is propagated using the point estimate $\hat{\theta}_\tau$ learned by the LMS, which represents the most likely value of θ^*. Therefore, optimizing for $\hat{\theta}_\tau$ is likely to improve the performance as long as the optimal solution lies within the feasible region. Using this idea, the optimization problem of the adaptive robust MPC at time step τ can be formulated by combining the previously explained robust Tube-MPC equations (3.22), (3.31), (3.35), with the SML learning method that computes a set Θ_τ^{HC} from (3.43), and the LMS learning method that computes a point estimate $\hat{\theta}_\tau$. The optimization problem is written as follows

$$(V^*, W^*) = \underset{V,W}{\arg\min} \sum_{k=0}^{N-1} \left(\hat{z}_k^\top Q \hat{z}_k + \hat{u}_k^\top R \hat{u}_k\right) + \hat{z}_N^\top P \hat{z}_N \tag{3.45}$$

$$\text{s.t.}\quad \bar{x}_0 = \hat{x}_0 = x_\tau,\ s_0 = 0, \tag{3.46}$$

$$\hat{x}_{k+1} = A_{cl,\hat{\theta}_\tau}\hat{x}_k + Bv_k, \tag{3.47}$$

$$\bar{x}_{k+1} = A_{cl,\bar{\theta}_\tau}\bar{x}_k + Bv_k, \tag{3.48}$$

$$s_{k+1} = \rho_{\bar{\theta}_\tau} s_k + w_k, \tag{3.49}$$

$$w_k \geq f_w(\bar{x}_k, s_k, \Theta_\tau^{HC}, \mathcal{D}) \tag{3.50}$$

$$\hat{y}_k = C_{\hat{\theta}}\hat{x}_k + D\hat{u}_k, \tag{3.51}$$

$$\hat{z}_k = E\hat{y}_k + F\hat{u}_k, \tag{3.52}$$

$$[H_{\mathcal{X}}]_j \bar{x}_k + [H_{\mathcal{U}}]_j \bar{u}_k + c_j s_k \leq 1, \tag{3.53}$$

$$\bar{u}_k = v_k + K\bar{x}_k,\ \hat{u}_k = v_k + K\hat{x}_k, \tag{3.54}$$

$$(\bar{x}_N, s_N) \in \mathcal{X}_f, \tag{3.55}$$

$$j = 1, \ldots, n_c,\ k = 1, \ldots, N-1, \tag{3.56}$$

where V is a vector that contains the MPC control inputs v_k, and V^* is the optimal value of V. $\rho_{\bar{\theta}_\tau}$ is the contraction rate of the tube assuming the uncertainty $\bar{\theta}_\tau$. The solution to (3.45) is decided by the optimal decision variables $v_\cdot^*$ and $w_\cdot^*$. Based on these solutions, the optimal trajectories of the following variables are found: $\hat{x}_\cdot^*, \bar{x}_\cdot^*, \hat{u}_\cdot^*, \bar{u}_\cdot^*, s_\cdot^*, \hat{y}_\cdot^*, \hat{z}_\cdot^*$.

The optimization problem (3.45) has 3 types of propagated state trajectories:

- (3.47) describes $\hat{x}_k$, the state trajectory propagated based on the point estimate $\hat{\theta}_\tau$, and it is used in the cost function;

- (3.48) describes $\bar{x}_k$, the nominal state trajectory based on SML-learned $\bar{\theta}_\tau$, which is used for robustness guarantees;
- (3.49) describes s_k, the tube scaling parameter, propagated based on the contraction rate of the tube $\rho_{\bar{\theta}_\tau}$ and w_k, which over-approximates the effect of parametric and additive uncertainty.

Equations (3.51) and (3.52) describe the expected output and the performance vectors trajectories, respectively, which are defined based on the point estimate $\hat{\theta}_\tau$. The inequality (3.50) describes the dynamics of variable w_k, which overapproximates the additional effect of the uncertainty on the tube scaling parameter. This is done using the function f_w which controls the dynamics of w_k. The details of the function f_w are explained in Appendix B.3, where it is turned into a set of inequalities linear in $\bar{x}_k$ and s_k. The inequality (3.53) describes the tightened version of the constraints to ensure that the whole predicted tube $\mathcal{P}_k$ satisfies the constraints. The equality (3.54) describes the control inputs, which consist of the precalculated state-feedback part (Kx_k) and the additional optimal input v_k. (3.55) describes the terminal condition using the set $\mathcal{X}_f$. This condition is used to guarantee the recursive feasibility of the MPC as will be explained in subsubsection 3.2.3.2

This optimization formulation corresponds directly to the bilateral teleoperation system with uncertain environment discussed in Section 3.1.3. In this practical context, the state trajectories $\hat{x}_k$ and $\bar{x}_k$ represent the predicted future states of the teleoperation system, including the positions and velocities of the leader and follower robots. The vector $\hat{z}_k$ contains the matching errors between the leader and follower robots in terms of position, velocity, and force. Note that $\hat{z}_k$ is used in the cost function to optimize the transparency, which takes into account the environment model with the parameters estimated by the LMS $(\hat{\theta}_\tau)$. Minimizing this performance vector ensures accurate and transparent bilateral teleoperation. The equations that are related to $\bar{x}_k, \bar{u}_k, s_k$ and $\bar{\theta}$ represent the nominal model of the teleoperation system, which is used to ensure robust constraint satisfaction. However, $\hat{u}_k$ is the control command applied to the leader and follower robots, which is directly related to the estimated model with $(\hat{\theta}_\tau)$.

The optimization problem (3.45) has a quadratic cost with linear constraints. Some solvers require writing the constraints in a standard form. This could be done by writing a system with an augmented state $x_{g,k} = [\hat{x}_k^\top, \bar{x}_k^\top, s_k]^\top$, and an augmented control input defined as $u_{g,k} = [v_k^\top, w_k]^\top$ such that

$$\begin{aligned} & x_{g,k+1} = A_{cl,g}(\hat{\theta}_\tau, \bar{\theta}_\tau, \rho_{\bar{\theta}_\tau}) x_{g,k} + B_g v_{g,k} \\ & H_{\mathcal{X},g} x_g + H_{\mathcal{U},g} v_g \leq 1_{n_{c,g}} \end{aligned} \tag{3.57}$$

where the subscript g denotes the augmented variables. The matrices $A_{cl,g}$, B_g, $H_{\mathcal{X},g}$, and $H_{\mathcal{U},g}$ are constructed by stacking the corresponding matrices for each state and control input from (3.50), (3.53), and (B.18).

Figure (3.5) shows the general architecture of the teleoperation system with the controller and the learning mechanisms. At each time step τ, the signals on each side are measured and sent to the centralized controller and learning components. The SML uses these measurements to learn the uncertainty bounds Θ_τ^{HC}. If Θ_τ^{HC} is updated, a new contraction rate $\rho_{\bar{\theta}_\tau}$ should be recomputed as in (B.10), and the corresponding parameters should be updated in the optimization problem (3.45). The LMS-learning finds a point estimate $\hat{\theta}_\tau$ and projects it on Θ_τ^{HC}, which is used in the cost function for performance, especially when the constraints are not active. The solver solves (3.45) and computes the optimal control input $v_\cdot^*$. Finally, $\hat{u}_\cdot^*$ is calculated accordingly from (3.54), and only the first value $\hat{u}_0$ is applied by the robots. At the next time step $(\tau+1)$, the whole process is repeated again.

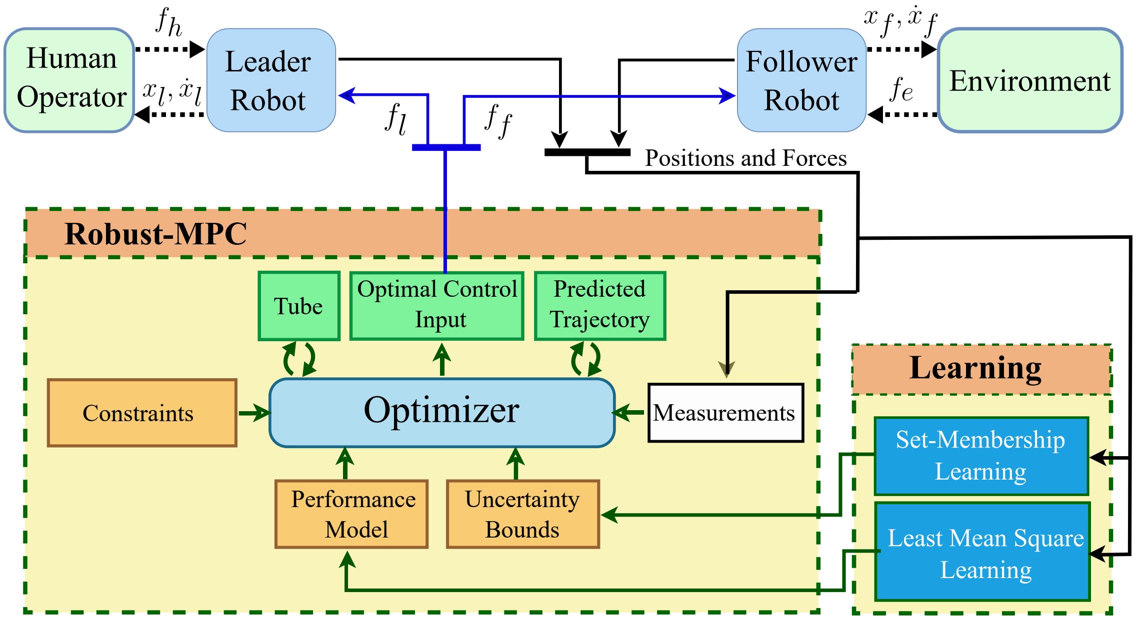

Figure 3.5: The general architecture of the teleoperation system with the robust Tube-MPC controller and the learning mechanisms.

3.2.3.2 Recursive Feasibility Outline

Recursive feasibility means that if the system starts from a state where the MPC problem is feasible, then it will keep being feasible in the future. This feature of robust MPC is very important to prevent the online optimization failure during the teleoperation and to keep the safety of the system. Below, we present the general outline of the proof that is often used in Tube-MPC. The idea of the proof is to define a terminal set $\mathcal{X}_f$ that has the following features:

- there exists a stabilizing controller $\pi_f(x)$ that makes the set robustly forward invariant set under all the possible uncertainty. i.e., if $x_k \in \mathcal{X}_f$ and $\pi_f(x_k)$ is applied, then the next state $x_{k+1} \in \mathcal{X}_f$, for all possible values of θ and d_k.
- all the points of the set satisfy the safety constraints. i.e.,

$$H_{\mathcal{X}} x_k + H_{\mathcal{U}} \pi_f(x_k) \leq 1_{n_{c_N}}, \quad \forall x_k \in \mathcal{X}_f$$

For recursive feasibility in the uncertainty-free system, it is sufficient to guarantee that the optimal trajectory will end up in the defined terminal set at the end of the prediction horizon, and it is assumed that the controller $\pi_f(x)$ will handle the guarantees after that (although this does not happen in practice, but only for theoretical guarantees). The following example illustrates this idea.

Assuming the feasibility of the MPC optimization problem at time step τ and that the MPC gave the following optimal control input and state trajectory

$$U_0^* = \{u_0^*, u_1^*, \ldots, u_{N-1}^*\}$$
$$X_0^* = \{x_\tau, x_1^*, \ldots, x_N^*\}$$

At the next time step $(\tau + 1)$, and assuming an ideal model, the system state will be equivalent to the predicted one, that is $x_{\tau+1} = x_1^*$. In that case, we can find the following new feasible solution to the MPC optimization problem

$$U_1^* = \{u_1^*, u_2^*, \ldots, u_{N-1}^*, \pi_f(x_N^*)\}$$
$$X_1^* = \{x_1^*, x_2^*, \ldots, x_N^*, x_{N+1}^*\}$$

where the new terminal state is given by

$$x^*_{N+1} = Ax^*_N + B\pi_f(x^*_N) \in \mathcal{X}_f$$

which is guaranteed by the terminal set and the controller $\pi_f(x)$ definitions. This solution shows how the problem is feasible at time step $(\tau+1)$ only from the feasibility at time step τ, and thus recursively feasible under the assumption of an ideal model. Figure (3.6) illustrates the concept of recursive feasibility without uncertainty.

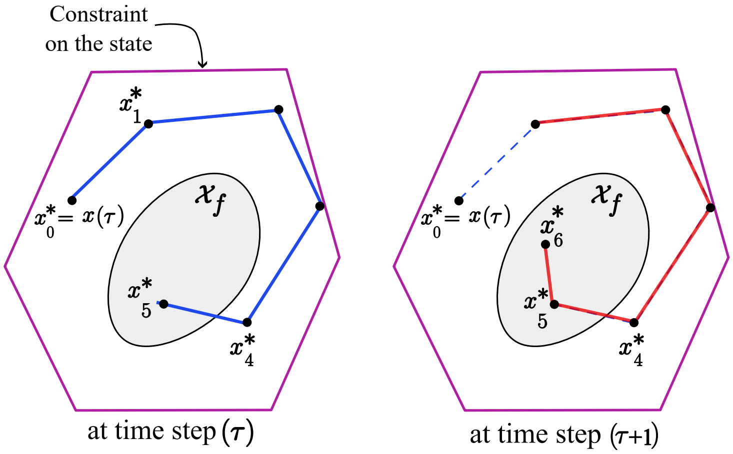

Figure 3.6: An illustration of the recursive feasibility concept without uncertainty for a horizon $N = 5$. (Left) at time step τ, the predicted feasible trajectory is shown as a blue line which ends up in the terminal set $\mathcal{X}_f$. (Right) at time step $(\tau+1)$, a feasible trajectory is found by reusing the previous one (red line), and the last state x^*_{N+1} is computed using the terminal controller $\pi_f(x)$, which guarantees that it stays in the terminal set.

For the system with uncertainty, the same idea can be used to prove recursive feasibility, except that all the trajectories inside the tube should end up in the terminal set. It is also required to show that the tube predicted at time step $(\tau+1)$ is included in the tube predicted at time step τ. Figure (3.7) illustrates the recursive feasibility concept with uncertainty.

The controller K and terminal cost matrix P are designed offline using standard robust control methods based on LMIs to guarantee the stability of the closed-loop system. The terminal set $\mathcal{X}_f$ is designed to be a robustly invariant set for the closed-loop system with the terminal control $\pi_f(x_k) = Kx_k$, in addition to other conditions that guarantee the recursive feasibility of the MPC. Details of the design of K, P, and

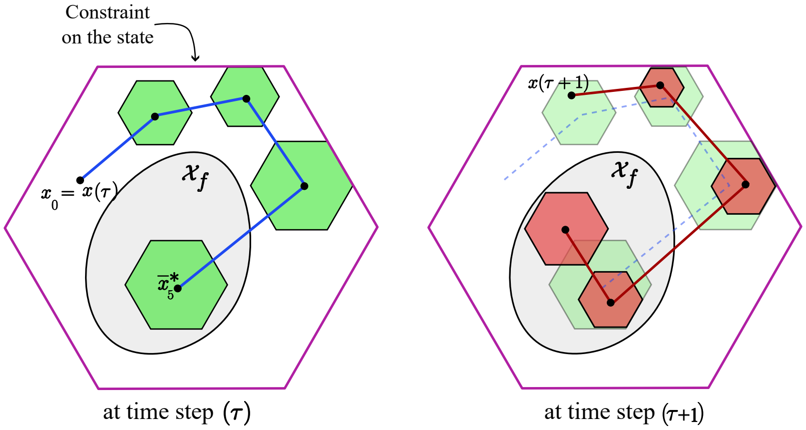

Figure 3.7: An illustration of the recursive feasibility concept with uncertainty, where the predicted trajectories are bounded within a tube. (left) At time step τ, the predicted nominal trajectory is shown as a blue line; the tube sections are shown as green polygons that contain the states under all possible uncertainties. The last set of the tube lies within the terminal set $\mathcal{X}_f$. (right) At time step $(\tau + 1)$, the previous predicted trajectory and tube are kept (transparent dashed blue line and green polygons); the first state is not equal to the nominal predicted one, but it is still inside the tube. The new predicted trajectory is shown as a red line, and the new tube sections are shown as red polygons. The new tube is contained by the previous one, and the last set of the new tube lies within the terminal set $\mathcal{X}_f$, which is guaranteed by the terminal controller $\pi_f(x)$.

$\mathcal{X}_f$ can be found in [168].

3.3 Validation

The proposed controller is validated through simulation studies. Two different simulations were performed. The first one focuses on showing the controller's effectiveness in robust constraint satisfaction, and on the SML learning of the uncertainty bounds. It shows the effect of SML on the conservatism of the robot. The second simulation highlights the controller's performance improvement with LMS learning.

3.3.1 Simulation Setup

The proposed method was validated through simulation with MATLAB 2022a, using *mpt3 Toolbox* for polytopes [173], *yalmip* for formulating the optimization problem [174], and *mosek* as a solver for the MPC [175]. We assume to have prior knowledge about the possible range of the environment model parameters, so the controller is initialized with the following values

$$\begin{aligned} k_e^0 &= 1000 \text{ N.m}^{-1}, & b_e^0 &= 10 \text{ N.s.m}^{-1} \\ \Delta k_e &\in [-200, +200] \text{ N.m}^{-1}, & \Delta b_e &\in [-3, +3] \text{ N.s.m}^{-1} \end{aligned} \tag{3.58}$$

The values of leader and follower robots models are the same as in [59] (converted from rotational to translational values), as reported in Table 3.1. The sampling period is $T_s = 0.005$ s. The prediction horizon is $N = 20$, and the MPC weight matrices Q, R were chosen experimentally as

$$Q = \begin{pmatrix} 5 \times 10^4 & 1 & 1 \\ 1 & 10^{11} & 1 \\ 1 & 1 & 10^5 \end{pmatrix} \quad R = \begin{pmatrix} 2 & 1 \\ 1 & 10 \end{pmatrix} \times 10^{-5} \tag{3.59}$$

Table 3.1: Numerical parameters of simulated robots

robot mass	$m_l = 0.24$	$m_f = 0.23$	kg
robot damping coefficient	$b_l = 1.34$	$b_f = 0.8$	N.s.m^{-1}

We designed the pre-stabilizing feedback controller K, the terminal cost matrix P, the tube-section $\mathcal{P}_0$, and the terminal set $\mathcal{X}_f$ using the method defined in Appendix A of [168]. $\mathcal{P}_0$ is designed as a $\rho-$contractive set using the method proposed in [169] with $\rho = 0.6$. The result is a 4-D polytope represented by $n_{\mathcal{P}_0} = 14$ inequalities. The additive noise bounds are set to

$$\begin{aligned} |[d]_1| < 0.0001 \times T_s, \quad |[d]_2| < 0.4 \times T_s \\ |[d]_3| < 0.0001 \times T_s, \quad |[d]_4| < 0.01 \times T_s \end{aligned} \tag{3.60}$$

where $[d]_i$ is the i-th component of d. Larger noise is considered on the velocity compared to position since position measurements have very high quality in most haptic devices. It is worth pointing out that having tight bounds on the additive noise accelerates the SML. However, the learning might fail if the noise exceeds the pre-specified bounds, which is why we initialized the SML with 4 times the expected bounds as a safety margin. Particularly, we defined the polytope $\mathcal{D}$ from (3.38) using 4 times the expected bounds (3.60).

3.3.2 Simulation 1: The Effectiveness of SML-Learning

The goal of this simulation is to demonstrate the controller's ability to robustly satisfy the constraints and to show the role of SML in reducing controller conservatism. We simulate a 3-second interaction with the previously defined environment. The true values of the environment parameters that is simulated here are as follows

$$k_e^* = 810 \text{ N.m}^{-1}, \quad b_e^* = 7.3 \text{ N.s.m}^{-1}$$

which makes the true value of uncertainty θ^* as follows

$$\theta_1^* = -190 \text{ N.m}^{-1}, \quad \theta_2^* = -2.7 \text{ N.s.m}^{-1}$$

We set a position constraint on the follower robot as: $x_f \leq 5 \times 10^{-3}$ m. This type of constraint is usually used in medical robots as a virtual fixture for the safety of the patient. The simulation is done in two stages, where the first stage is done without SML, and the second stage is done with SML in order to see the effect of the learning. The SML starts only at $t = 1.5$ s, while the LMS-learning starts from $t = 0$ s. Figures 3.8 and 3.9 illustrate the results of the first simulation.

3.3.2.1 First Stage (before SML)

At $t = 0$ s, the follower robot is in contact with the environment (i.e., $x_f > 0$). The human operator applies a filtered step force, as shown in Figure (3.8b) (blue line). This force moves the leader robot, and the follower robot follows closely (red line) until the follower robot gets close to the constraint limit, that is when the controller prevents it from continuing the motion, and the leader robot starts resisting the human hand to keep its position matched with the follower robot. The green line represents the virtual target position that the human is trying to reach (based on f_h and $\hat{k}_e$). In Figure (3.8a), we show the boundaries of the tube projected on x_f (red-dashed lines) at $t \in \{0.4s, 1s, 2s\}$. The tube is predicted over a horizon of 100 ms. All possible state trajectories under the uncertainty are bounded between the two red dashed lines of the tube. During this stage, the position matching error is small, but the reflected force to the human operator does not match the environment force as shown in Figure (3.8b) because the follower robot is not allowed to go deeper into the environment.

3.3.2.2 Second Stage (with SML)

At $t = 1.5$ s, the SML starts, and quickly, the bounds on the parametric uncertainty begin to shrink as seen in Figure (3.9) (black dashed lines). At $t = 1.6$ s, the human operator increases the force to the same previous level, but this time, both robots move very closely to the constraints, since the uncertainty became much smaller. The tube shown at $t = 2$ s is much smaller than the one of $t = 0.4$ s, which means that the controller is now able to find a smaller tube that lies inside the constrained set. At $t = 2.6$ s, the human tries to violate the constraint with a bigger force, but the follower

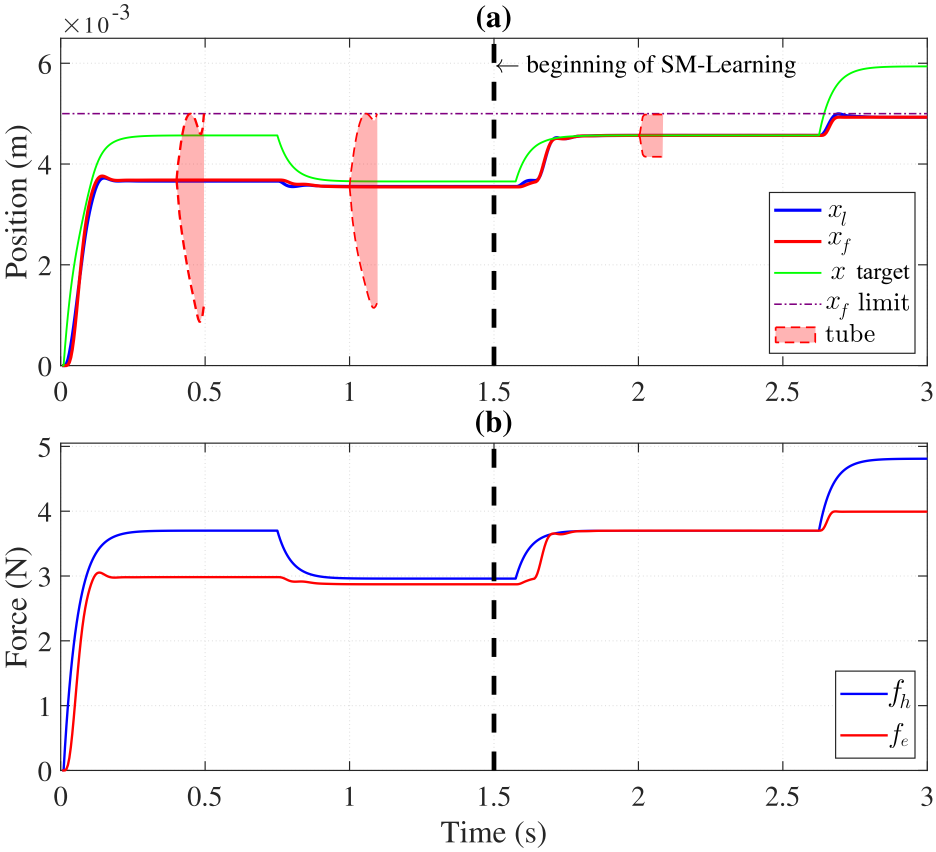

Figure 3.8: Simulation 1, soft contact with a position constraint on the follower robot. The SML starts at $t = 1.5$. (a) The position of the leader (blue) and follower (red); the virtual target position based on f_h and k_e (green); the constraint on x_f (violet); the predicted tube around x_f at $t \in \{0.4, 1, 2\}$ s (dashed-red). (b) The human force (blue); the environment force (red).

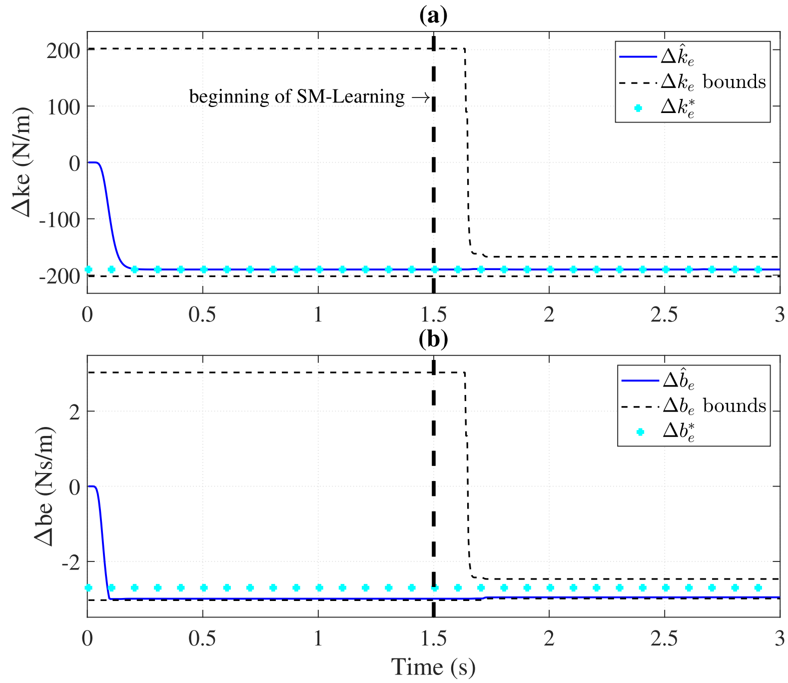

Figure 3.9: Simulation 1, LMS and SML learning of the environment parametric uncertainty. The figure shows the projection of Θ_τ^{HC} on each of the parameters (a) environment stiffness k_e; (b) environment damping coefficient b_e. On each figure, the estimated parameter by LMS is in blue; the boundaries of the uncertainty are in black-dashed lines, learned by SML; the true value of the parameter is in cyan. The SML starts at $t = 1.5$ s.

robot stays inside the safe region, and the leader robot resists the motion and keeps the positions matched. Figure (3.9) shows how the LMS-learning converges quickly to the true values of the parameters.

3.3.3 Simulation 2: The Effectiveness of LMS-Learning

In case the bounds on the additive noise are not tight enough, SML might be slow. For this case, the parameters learned using the LMS method are used in the cost function to enhance the performance. In this simulation, we remove the position constraints and the SML. The controller is initialized with the same values from (3.58). The true values of the simulated environment parameters are

$$k_e^* = 1190 \text{ N.m}^{-1}, \quad b_e^* = 12.7 \text{ N.s.m}^{-1} \tag{3.61}$$

which makes the true value of uncertainty θ^* as follows

$$\theta_1^* = 190 \text{ N.m}^{-1}, \quad \theta_2^* = 2.7 \text{ N.s.m}^{-1}$$

Figure (3.10) shows that before learning, the position matching errors between both robots were relatively large. At $t = 0.5$ s, the LMS-learning starts, and the parameters start converging to their true values as shown in Figure (3.11). By $t = 0.8$ s, the matching errors become much smaller since the controller is using a good estimate of the parameters in the cost function. This simulation shows how LMS-learning can enhance performance even if the bounds on parametric uncertainty are still wide, as long as the system is far from the constraints. In practice, we found that the damping coefficient b_e is more difficult to learn than the stiffness k_e, and it is learned only during the dynamic motions. This can be observed in Figure (3.11). This could be due to the fact that the damping coefficient is not directly observable when the system is stationary, and the effect of the damping is only visible when the system is in motion.

3.4 Chapter Conclusion

In this chapter, we have presented an adaptive robust MPC framework for bilateral teleoperation systems, which combines tube controllers with two online learning mechanisms to learn the environment model. The SML-learning method learns the bounds of parametric uncertainty to reduce the conservatism of the robust tube controller, and the LMS method improves the prediction model. The framework ensures constraint satisfaction robustly and recursively, which means that if the optimization problem starts from a feasible point, it will be feasible in the future even under the worst-case uncertainty. Our simulation studies have shown that the proposed approach is effective in maintaining constraints, enhancing performance, and reducing controller conservatism for bilateral teleoperation tasks.

Several limitations, however, should be acknowledged. The current framework assumes a constant, albeit unknown, environment parameter θ^*. Furthermore, the SML algorithm is sensitive to measurement outliers that fall outside the defined noise bounds,

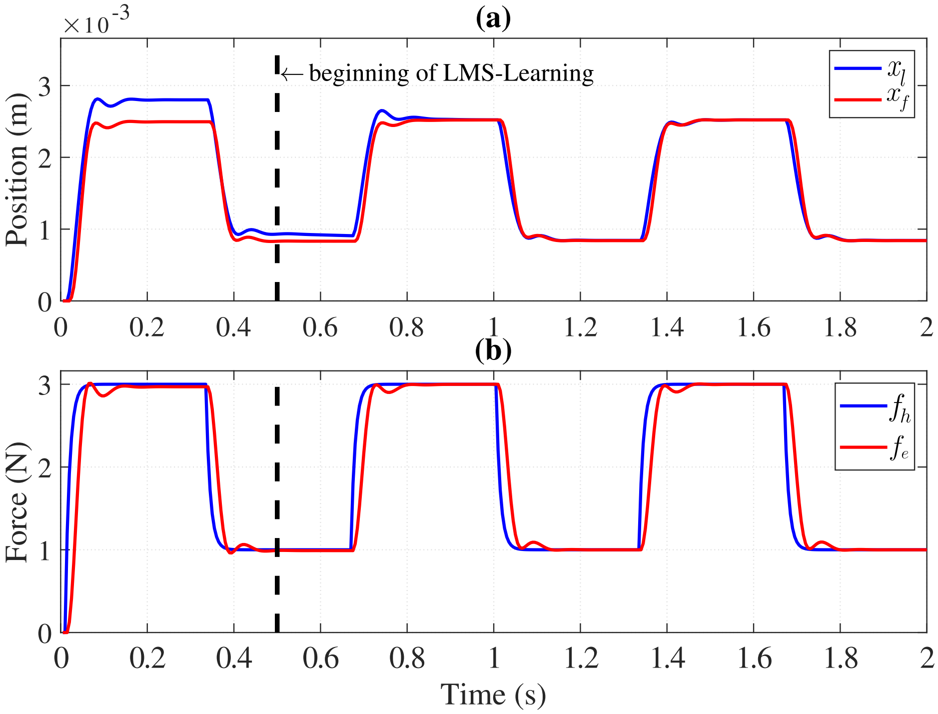

Figure 3.10: Simulation 2, soft contact with LMS-learning and no constraints. (a) Position of the leader robot (blue) and the follower robot (red). (b) Human force (blue); environment force (red).

which could corrupt the learned uncertainty set. Finally, the design of the underlying Tube-MPC components, such as the pre-stabilizing controller K and the terminal set, still involves complex offline computations and a degree of conservatism. These limitations offer clear avenues for future work, as discussed in Chapter 5, and highlight the inherent trade-offs between performance, safety guarantees, and controller complexity.

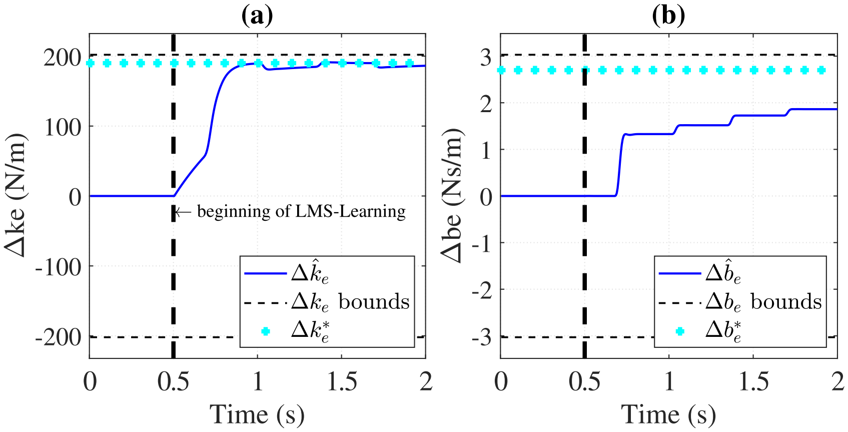

Figure 3.11: Simulation 2, LMS parameter learning without SML. (a) Learning of environment stiffness uncertainty. (b) Learning of environment damping coefficient uncertainty.

Chap. 4

Variable Impedance Control in Bilateral Teleoperation

Chapters 2 and 3 focused on ensuring safe and high-performance bilateral teleoperation under constraints and uncertainty, primarily through MPC techniques. This chapter shifts focus to adaptive interaction by exploring Variable Impedance Control (VIC) in bilateral teleoperation. We address the critical stability challenge posed by potential passivity violations in VIC and develop an optimization-based method to ensure safe, stable, and adaptive impedance modulation.

The chapter is organized as follows: Section 4.1 provides an introduction to VIC in teleoperation and discusses the associated passivity challenges, reviewing relevant literature. Section 4.2 presents the proposed passive VIC controller, detailing the impedance formulation and the optimization-based passivity filter mechanism. Section 4.3 describes the experimental setup and validation results on a 3-DoF bilateral system. Finally, Section 4.4 concludes the chapter.

4.1 Introduction and Literature Review

4.1.1 Variable Impedance Control in Teleoperation

Humans deal with various tasks instinctively by adjusting the impedance of their hands and arms [176]. Similarly, in robotics, VIC strategies have been developed to dynamically adjust a robot's interaction with its environment, improving both performance and safety in complex environments [77]. A challenge in teleoperation systems is enabling the robot to handle a wide range of tasks in diverse contexts such as space exploration or minimally invasive surgery, where human intervention might be impossible, unsafe, or impractical. The advantages of VIC motivate its use in such a context to control both the leader and follower robots. In the literature, VIC implementations vary widely in how the impedance profile is generated and in the specific performance or safety benefits they deliver.

One example of VIC in teleoperation is the concept of *teleimpedance* control, which has emerged in the last decade to describe controllers that vary the impedance of the follower robot based on criteria defined by the leader side, and it was applied in both unilateral and bilateral teleoperation [4]. In this method, the impedance of the human arm is often imitated on the follower robot. The human arm impedance could be estimated, for example, using surface Electromyography (sEMG) on the human arm [177]. In [178], the authors designed a teleimpedance scheme where the follower impedance is adapted based on the operator grip force, which is generally correlated with the operator impedance. On the operator side, the leader impedance is adapted to enhance the situational awareness of the operator, such that the haptic device feels light if the follower has low impedance, making it feel safer to move freely. When the follower has high impedance, and it is more dangerous to the environment, the haptic device feels heavier to the user. In [79], Michel et al. designed an adaptive controller for a cutting task, for which the follower's impedance profile is learned from human demonstrations, while the leader robot reflects the measured environment force. A similar framework was developed in [179], where the leader and follower sensor information was used to modify the follower impedance based on a force-stiffness map learned from human demonstrations. Other methods for transferring the operator's arm impedance to the follower robot include using biomechanical models of the arm to estimate its impedance [180], or inducing a perturbation on the master side and measuring the displacement [181]. External devices, such as buttons can be used as well [182]. We refer the reader to [4] for a comprehensive survey of such teleimpedance approaches.

Beyond teleimpedance schemes that rely on directly transferring the operator's impedance, other works have used VIC in teleoperation to adapt impedance profiles based on environmental interaction, task dynamics, or learned strategies. In [78], VIC was used for compliant assembly, where the follower impedance profile is changed online based on velocity and torque measurements to reduce impact forces resulting from collisions with the environment. In [121], Motaharifar et al. developed a collaborative haptic training system with two VIC-controlled haptic interfaces, one for the trainer and the other for the trainee. The impedance of each interface is adapted to operate either in force-reflection or position-reflection mode so as to allow different levels of authority or guidance to the trainee, based on the estimated impedance of the trainer.

Varying the impedance on the leader side was also used for different purposes. In [183], Corredor et al. introduced an adaptive impedance control strategy for teleoperation that dynamically adjusts the admittance on the leader side based on a decision-making model to provide a haptic guidance force. This artificially generated force prevents human movements in certain areas of the task space and/or guides them toward a desired area for better task execution. The decision-making model continuously evaluates tracking errors against interaction conflicts, choosing between increasing the assistance or favoring task performance. This dynamic adjustment ensures that the leader's impedance is adapted in real time to balance effective guidance with natural human control. In [184], the environment's impedance was estimated online and used as a local reference for the leader VIC, which improved transparency in a delayed teleoperation system.

As we can see from the mentioned literature, VIC has many applications in teleoperation, since it provides advantages on both the leader and follower sides. However, one issue that faces VIC is that it can inject energy into the system, thereby risking the violation of passivity, which is a widely used condition to guarantee stability. In the following, we will discuss the passivity and its associated issues in VIC.

4.1.2 Passivity Issues in Variable Impedance Control

Although VIC is an interesting solution to perform flexible and complex interaction tasks, it also poses potential stability issues [80], which are even more significant in the context of teleoperation where stability is a major concern [79]. Most of the methods that guarantee stability in bilateral teleoperation are based on passivity, which is a sufficient condition for stability, provided that the operator and environment are passive [70]. A famous example of such versatile methods is TDPA proposed by Ryu et al. [21]. TDPA monitors the energy exchanged at the system's interaction ports in real time through passivity observers. When these observers detect non-passive energy flow (i.e., the system generates energy), passivity controllers are activated to inject adaptive damping, thereby dissipating the excess energy. This process prevents the system from generating net energy that could destabilize the interaction. Many variations of the method were developed afterward. For example, a Virtual Mass-Spring (VMS) filter was added to TDPA in [185] to filter out the high-frequency force modification or chattering. Another chattering-free TDPA was also designed in [186] which defines a low velocity threshold, under which the dissipation is reduced to prevent the chattering issue at low velocities. A less-conservative version of TDPA, called TDPA-Energy Reflection was also proposed in [187]. This method was used by Wang et al. in [179] to passivate a bilateral teleoperation system with VIC.

Another versatile approach to ensure passivity is Energy Tanks [23, 24]. In this method, a virtual tank is used to store (a virtual equivalent to) the energy dissipated by the physical system. This virtual energy can be used later to perform non-passive interactions without generating net energy. When the energy in the tank is not enough to compensate for the non-passive behavior, the control input is modified, for example by damping injection, to preserve the passivity. Energy tanks were used in [79] and [188] to ensure passivity of bilateral teleoperation systems that include VIC. Another passivity framework was proposed for bilateral teleoperation called Passive-Set-

Position-Modulation or PSPM [22]. The PSPM modulates the set-position signal sent to each robot to guarantee the passivity. An optimization problem is solved to find the closest point to the original reference point that still satisfies the passivity condition.

Panzirsch et al. [189] presented a novel gradient-based passivity control concept for variable stiffness. The approach is based on a potential energy storage reference and prevents phases of zero stiffness. An observer and gradient-based method adjusts the desired stiffness in real time while ensuring that the observed energy remains non-negative, thus guaranteeing passivity without resorting to complete force attenuation, which is a common drawback in traditional methods such as TDPA or energy tanks.

A method specific to passivity of VIC in pHRI was designed in [80] by Kronandar and Billard. The authors proposed a state-independent stability condition which can verify the passivity of preplanned impedance profiles offline. Thus, this concept is not applicable to scenarios in which the profile needs to be adaptive in real time. Bednarczyk et al. developed this condition and introduced Passivity Filters (PFs) [190]. In VIC implementation, these filters adjust the desired impedance profile online to ensure that the energy generated by varying the impedance is lower than the energy dissipated by the system. The work in [188] implemented a similar idea, where an energy tank was used to monitor the passivity condition, and the non-passive stiffness changes are set to zero if the tank is below a minimum threshold. Compared to previous methods, PFs ensure a continuous and smooth impedance variation due to their low-pass filtering nature. PFs were initially proposed for collaborative manipulation, and the method does not apply directly to bilateral teleoperation. Hence, further developments are required to dissipate the additional energy generated by the architecture of the teleoperation system itself. In this chapter, we propose an optimization-based framework for passive VIC in bilateral teleoperation, inspired by the PFs, PSPM, and TDPA methods.

4.2 Variable Impedance Control for Bilateral Teleoperation

4.2.1 Variable Impedance Controller Formulation

Consider a general bilateral teleoperation system with leader and follower robots, both with n-DoF, in m-dimensional task space. In the following, we consider the superscript with $i \in \{l, f\}$ to denote the leader and the follower. The dynamic model of each robot can be written as

$$H_l(x_l)\ddot{x}_l + C_l(x_l, \dot{x}_l)\dot{x}_l + g_l(x_l) = f_l + f_h \tag{4.1}$$

$$H_f(x_f)\ddot{x}_f + C_f(x_f, \dot{x}_f)\dot{x}_f + g_f(x_f) = f_f + f_e \tag{4.2}$$

where $x_i, \dot{x}_i, \ddot{x}_i \in \mathbb{R}^m$ denote the task space positions, velocities, and accelerations for the leader and the follower, respectively, and $i \in \{l, f\}$. $f_i \in \mathbb{R}^m$ denotes the actuators force. $H_i(x) \in \mathbb{R}^{m \times m}$, $C_i(x_i, \dot{x}_i) \in \mathbb{R}^{m \times m}$ and $g_i(x_i) \in \mathbb{R}^m$ are the inertia matrix, the Coriolis and centrifugal matrix, and the gravity effects vectors, respectively. The time dependency has been omitted for readability, and the matrices' dependency on $x_i, \dot{x}_i$ will be omitted in the following for simplicity. $f_h, f_e \in \mathbb{R}^m$, are the interaction forces with the human and environment, respectively.

Several VIC techniques have been proposed in the literature, depending on the desired architecture for the teleoperation control system [79,121,181,191]. In our work, the control inputs (actuators forces) are calculated such that the imposed impedance behavior of the leader robot is

$$M_l\ddot{x}_l + (D_l + C_l)\dot{x}_l + K_l(x_l - x_l^r) = f_h + f_e \tag{4.3}$$

and for the follower robot it is

$$M_f(\ddot{x}_f - \ddot{x}_f^r) + D_f(\dot{x}_f - \dot{x}_f^r) + K_f(x_f - x_f^r) = f_e \tag{4.4}$$

where $M_i, D_i, K_i \in \mathbb{R}^{m \times m}$, for $i \in \{l, f\}$, are time-dependent matrices that describe the imposed inertia, damping, and stiffness, respectively. The inertia matrix is designed to be positive definite, while the damping and stiffness matrices are chosen as positive semi-definite matrices. x_l^r is the reference position that the leader robot should track. $x_f^r, \dot{x}_f^r, \ddot{x}_f^r$ are the follower's reference position, velocity, and acceleration, respectively. Ideally, the reference signals for one robot should reflect the measurements made on the other one, leading to $x_l^r = x_f$, $x_f^r = x_l$, $\dot{x}_f^r = \dot{x}_l$ and $\ddot{x}_f^r = \ddot{x}_l$. However, it will be shown later that doing so could lead to a lack of passivity. As a result, the following developments will propose a method to track the ideal trajectory without compromising passivity.

A few remarks should be noted here:

- for the leader, smaller values of K_l allow prioritizing the reflection of the environment force. In contrast, higher values of K_l create a higher coordinating force that pushes the leader's position to match the follower's position as can be seen in (4.3);

- we consider a teleoperation control architecture with no force sensor on the leader side, and therefore we do not reflect the operator's force on the environment. Also, imposing the inertia matrix would require such a force sensor [7]. However, since most haptic interfaces have low inertia, it is reasonable to set $M_l = H_l$ [121], though it makes M_l time-varying as H_l is a function of $x_l(t)$;
- on the follower side, where a force sensor is attached, M_f is designed to be constant;
- the Coriolis matrix C_l is left in the imposed impedance to use the passivity feature that lies in the skew symmetry of $\dot{M}_l - 2C_l$.

To implement the behavior described by (4.3) and (4.4), the control inputs f_l, f_f from (4.1) and (4.2) are chosen as follows

$$\begin{aligned} f_l &= g_l + H_l u_l^{\mathrm{imp}} && (4.5) \\ u_l^{\mathrm{imp}} &= H_l^{-1}(f_e - D_l \dot{x}_l - K(x_l - x_l^r)) && (4.6) \end{aligned}$$

and on the follower side

$$\begin{aligned} f_f &= C_f \dot{x}_f + g_f - f_e + H_f u_f^{\mathrm{imp}} && (4.7) \\ u_f^{\mathrm{imp}} &= \ddot{x}_f^r - M_f^{-1}\left(D_f(\dot{x}_f - \dot{x}_f^r) + K_f(x_f - x_f^r) - f_e\right) && (4.8) \end{aligned}$$

4.2.2 Passivity Filters for VIC in Bilateral Teleoperation

4.2.2.1 Passivity Condition

A dynamic system of state $x(t)$, input $u(t)$, and output $y(t)$ is said to be passive with respect to an interaction port (u, y) if there exists a non-negative storage function $\mathcal{V}(t) = \mathcal{V}(x(t))$ such that

$$\mathcal{V}(x(t)) - \mathcal{V}(x(0)) \leq \int_0^t u^\top(\xi) y(\xi)\, d\xi \tag{4.9}$$

where the storage function represents the internal stored energy of the system.

Passivity has been extensively considered to ensure stability for teleoperation systems [60]. The idea is to make the system passive with respect to the ports $(\dot{x}_l, f_h)$ and $(\dot{x}_f, f_e)$. Thus, a sufficient condition for the passivity is ensured if we can find a storage function such that

$$\dot{\mathcal{V}} \leq \dot{x}_l^\top f_h + \dot{x}_f^\top f_e \tag{4.10}$$

Let us consider the following storage function

$$\mathcal{V} = \underbrace{\frac{1}{2}\left(\dot{x}_l^\top M_l \dot{x}_l + e_l^\top K_l e_l\right)}_{\mathcal{V}_l} + \underbrace{\frac{1}{2}\left(\dot{e}_f^\top M_f \dot{e}_f + e_f^\top K_f e_f\right)}_{\mathcal{V}_f} \tag{4.11}$$

where $e_l = x_l - x_l^r$, $e_f = x_f - x_f^r$. To get to an equation similar to (4.10), we can treat $\mathcal{V}_l$ and $\mathcal{V}_f$ separately. The first part of the storage function is

$$\mathcal{V}_l = \frac{1}{2}\left(\dot{x}_l^\top M_l \dot{x}_l + e_l^\top K_l e_l\right)$$

Thus, its time derivative is

$$\dot{\mathcal{V}}_l = \dot{x}_l^\top M_l \ddot{x}_l + \frac{1}{2}\dot{x}_l^\top \dot{M}_l \dot{x}_l + e_l^\top K_l \dot{e}_l + \frac{1}{2} e_l^\top \dot{K}_l e_l$$

We substitute $M_l \ddot{x}_l$ from (4.3) as

$$\begin{aligned}
M_l \ddot{x}_l &= -(D_l + C_l)\dot{x}_l - K_l(x_l - x_l^r) + f_h + f_e \\
\Rightarrow \dot{\mathcal{V}}_l &= \dot{x}_l^\top \left(-(D_l + C_l)\dot{x}_l - K_l(x_l - x_l^r) + f_h + f_e\right) \\
&+ \frac{1}{2}\dot{x}_l^\top \dot{M}_l \dot{x}_l + (x_l - x_l^r)^\top K_l(\dot{x}_l - \dot{x}_l^r) + \frac{1}{2}(x_l - x_l^r)^\top \dot{K}_l (x_l - x_l^r) \\
&= -\dot{x}_l^\top (D_l + C_l)\dot{x}_l - \cancel{\dot{x}_l^\top K_l(x_l - x_l^r)} + \dot{x}_l^\top f_h + \dot{x}_l^\top f_e \\
&+ \frac{1}{2}\dot{x}_l^\top \dot{M}_l \dot{x}_l + (x_l - x_l^r)^\top K_l(\cancel{\dot{x}_l} - \dot{x}_l^r) + \frac{1}{2}(x_l - x_l^r)^\top \dot{K}_l (x_l - x_l^r) \\
&= \dot{x}_l^\top f_h + \dot{x}_l^\top (\frac{1}{2}\dot{M}_l - (D_l + C_l))\dot{x}_l + \dot{x}_l^\top f_e \\
&- (x_l - x_l^r)^\top K_l \dot{x}_l^r + \frac{1}{2}(x_l - x_l^r)^\top \dot{K}_l (x_l - x_l^r) \\
&= \dot{x}_l^\top f_h - \dot{x}_l^\top D_l \dot{x}_l + \dot{x}_l^\top (\frac{1}{2}\dot{M}_l - C_l)\dot{x}_l + \dot{x}_l^\top f_e - e_l^\top K_l \dot{x}_l^r
\end{aligned}$$

Note that the term $\dot{x}_l^\top(\frac{1}{2}\dot{M}_l - C_l)\dot{x}_l = 0$ due to the skew symmetry of $(C_l - \frac{1}{2}\dot{M}_l)$.

Therefore, we can write

$$\dot{x}_l^\top f_h = \dot{\mathcal{V}}_l + \omega_l \tag{4.12}$$

where we define the term ω_l as

$$\omega_l = \dot{x}_l^\top D_l \dot{x}_l - \dot{x}_l^\top f_e + e_l^\top K_l \dot{x}_l^r - \frac{1}{2} e_l^\top \dot{K}_l e_l \tag{4.13}$$

The second part of the storage function from (4.11) is

$$\mathcal{V}_f = \frac{1}{2}\left(\dot{e}_f^\top M_f \dot{e}_f + e_f^\top K_f e_f\right)$$

and its time derivative is

$$\dot{\mathcal{V}}_f = \dot{e}_f^\top M_f \ddot{e}_f + \frac{1}{2}\dot{e}_f^\top \dot{M}_f \dot{e}_f + e_f^\top K_f \dot{e}_f + \frac{1}{2} e_f^\top \dot{K}_f e_f$$

By substituting $M_f \ddot{e}_f$ from (4.4) as

$$\begin{aligned}
M_f \ddot{e}_f &= (-D_f \dot{e}_f - K_f e_f + f_e) \\
\Rightarrow \dot{\mathcal{V}}_f &= \dot{e}_f^\top (-D_f \dot{e}_f - \cancel{K_f e_f} + f_e) + \frac{1}{2}\dot{e}_f^\top \dot{M}_f \dot{e}_f \\
&\quad + \cancel{e_f^\top K_f \dot{e}_f} + \frac{1}{2} e_f^\top \dot{K}_f e_f \\
&= \dot{e}_f^\top f_e + \dot{e}_f^\top (-D_f + \frac{1}{2}\dot{M}_f)\dot{e}_f + \frac{1}{2} e_f^\top \dot{K}_f e_f \\
&= \dot{x}_f^\top f_e - (\dot{x}_f^r)^\top f_e - \dot{e}_f^\top (D_f - \frac{1}{2}\dot{M}_f)\dot{e}_f + \frac{1}{2} e_f^\top \dot{K}_f e_f
\end{aligned}$$

We set the desired inertia M_f to be a constant matrix, and we get the following

$$\dot{x}_f^\top f_e = \dot{\mathcal{V}}_f + \omega_f \tag{4.14}$$

where we define the term ω_f as

$$\omega_f = \dot{e}_f^\top D_f \dot{e}_f - \frac{1}{2} e_f^\top \dot{K}_f e_f + (\dot{x}_f^r)^\top f_e \tag{4.15}$$

From (4.12) and (4.14), we obtain the following power balance

$$\dot{x}_l^\top f_h + \dot{x}_f^\top f_e = (\dot{\mathcal{V}}_l + \dot{\mathcal{V}}_f) + (\omega_l + \omega_f) \tag{4.16}$$

$$\Rightarrow \dot{x}_l^\top f_h + \dot{x}_f^\top f_e = \dot{\mathcal{V}} + \omega \tag{4.17}$$

where $\omega = \omega_l + \omega_f$. Equation (4.17) describes the energy exchange with the human and the environment, where the left side is the energy that passes through the system ports $(\dot{x}_l, f_h)$ and $(\dot{x}_f, f_e)$, and the right side shows the change in the stored energy $\dot{\mathcal{V}}$ and an additional term ω. In order to guarantee the passivity of the system such as in (4.10), it is sufficient to ensure that $\omega \geq 0$. In this case, ω is the power dissipated by the system. Note that some terms in ω are not guaranteed to be positive by default, such as the ones related to $\dot{K}_l$ and $\dot{K}_f$, which can inject energy into the system and potentially violate the passivity condition. A similar problem has been addressed in [190] for pHRI control using PFs. The solution was based on the idea of filtering the variations of the impedance parameters to make the term ω always positive. The filtered values track the desired ones as fast as possible while limiting their change rate in a way to guarantee passivity.

4.2.2.2 The Proposed Method

Inspired by the PFs, we modify the terms in ω in real time to satisfy the passivity condition. ω is written from (4.13) and (4.15) as

$$\omega = \underbrace{\dot{x}_l^\top D_l \dot{x}_l - \dot{x}_l^\top f_e + e_l^\top K_l \dot{x}_l^r - \frac{1}{2} e_l^\top \dot{K}_l e_l}_{\omega_l} + \underbrace{\dot{e}_f^\top D_f \dot{e}_f - \frac{1}{2} e_f^\top \dot{K}_f e_f + (\dot{x}_f^r)^\top f_e}_{\omega_f} \tag{4.18}$$

We consider the impedance behavior defined (4.3) and (4.4), and we assume that the impedance matrices desired by the users are the following

$$\text{for the leader: } D_l^d, K_l^d \qquad (\text{ assumed } M_l^d = H_l) \tag{4.19}$$

$$\text{for the follower: } D_f^d, K_f^d \qquad (\text{ assumed } M_f^d = \text{constant}) \tag{4.20}$$

while the reference signals in (4.3) and (4.4) are set in a way to achieve the teleoperation task, i.e.,

$$\text{for the leader: } x_l^r = x_f \tag{4.21}$$

$$\text{for the follower: } x_f^r = x_l,\ \dot{x}_f^r = \dot{x}_l,\ \ddot{x}_f^r = \ddot{x}_l \tag{4.22}$$

Ideally, the desired impedance and references should be applied directly to get the desired teleoperation behavior. However, to guarantee passivity, we follow the PFs method, and we use a filtered version of the user-desired values. First, for the stiffness matrices, we define the following first-order filter

$$\dot{K}_l = \beta_1(t)(K_l^d - K_l) = \beta_1(t)\tilde{K}_l \tag{4.23}$$

$$\dot{K}_f = \beta_2(t)(K_f^d - K_f) = \beta_2(t)\tilde{K}_f \tag{4.24}$$

where $\beta_1(t) \geq 0, \beta_2(t) \geq 0$ are time-varying stiffness filter gains. By choosing the values of $\beta_1(t)$ and $\beta_2(t)$ appropriately, we can modify the terms $(\frac{1}{2}e_l^\top \dot{K}_l e_l)$ and $(\frac{1}{2}e_f^\top \dot{K}_f e_f)$ in (4.18) if needed. Particularly, if the desired stiffness is larger than the current one, i.e., $K_f^d > K_f$ (or $K_l^d > K_l$), then $\dot{K}_f > 0$ ($\dot{K}_l > 0$), and the corresponding term in ω is negative and injects energy into the system. In the worst-case scenario, by selecting $\beta_2(t) = 0$ ($\beta_1 = 0$) means that the imposed stiffness is fixed, and no energy is introduced by the corresponding term.

Varying the other impedance matrices do not need filtering. Notably, the damping terms $(\dot{x}_l^\top D_l \dot{x}_l)$ and $(\dot{e}_f^\top D_f \dot{e}_f)$ are always positive, and only dissipate energy. Therefore, they do not need to be filtered. Inertia matrices M_l and M_f do not appear in ω, and they are not filtered either.

Filtering the impedance variations was sufficient to guarantee passivity in pHRI in [190]. However, in teleoperation, the energy generation does not come only from varying the impedance, but other terms have to be taken into account. In particular, the terms $(e_l^\top K_l \dot{x}_l^r)$, $(\dot{x}_f^{r\top} f_e)$, and $(\dot{x}_l^\top f_e)$. Since the signs of these terms are not guaranteed, any one of them could inject energy into the system. Concerning the first two terms, and inspired by the PSPM method [22], the position reference for each side can be filtered such as

$$\dot{x}_l^r = \beta_3(t)(x_f - x_l^r) = \beta_3(t)\tilde{x}_l^r \tag{4.25}$$

$$\dot{x}_f^r = \beta_4(t)(x_l - x_f^r) = \beta_4(t)\tilde{x}_f^r \tag{4.26}$$

where $\beta_3(t) \geq 0, \beta_4(t) \geq 0$ are time-varying position reference filter gains.

In order to guarantee the ability to dissipate the energy related to the term $\dot{x}_l^\top f_e$, and inspired by the traditional TDPA method [21], we use additional damping injection. Therefore, we set the imposed leader damping matrix to be

$$D_l = D_l^d + I\beta_5(t) \tag{4.27}$$

where $\beta_5(t) \geq 0$ is a time-varying additional damping parameter on the leader side.

The rest of the imposed impedance matrices is set as the desired value, i.e.,

$$D_f = D_f^d \tag{4.28}$$

$$M_f = M_f^d \quad \text{constant} \tag{4.29}$$

$$M_l = M_l^d = H_l \tag{4.30}$$

Note that the filtering and the damping injection would lead to a deterioration of the performance as the desired impedance behavior and position trajectory will not be tracked perfectly. This is done of course in favor of guaranteeing the passivity. In order to minimize the conservative effect of passivation on the performance, the time-varying values of $\beta_j(t)$ are chosen by solving the following optimization problem

$$\beta^* = \arg \min_{\beta_1 \dots \beta_5} (\sum_{j=1}^{5} p_j \beta_j) \tag{4.31}$$

$$\begin{aligned} \text{s.t.} \quad & 0 \leq \omega(t) \\ & 0 \leq \beta_j \leq \beta_{\max,j} : j \in \{1, \dots, 5\} \end{aligned}$$

where

$$\begin{aligned} \omega &= \omega_l + \omega_f \\ &= \dot{x}_l^\top (D_l^d + \beta_5 I)\dot{x}_l - \dot{x}_l^\top f_e + e_l^\top K_l \tilde{x}_l^r \beta_3 - \frac{1}{2} e_l^\top \tilde{K}_l e_l \beta_1 \\ &+ \dot{e}_f^\top D_f^d \dot{e}_f - \frac{1}{2} e_f^\top \tilde{K}_f e_f \beta_2 + \tilde{x}_f^{r\top} f_e \beta_4 && (4.32) \\ &= \sum_{j=1}^{5} \lambda_j \beta_j + \zeta && (4.33) \end{aligned}$$

$\beta^* = [\beta_1^*, ..., \beta_5^*]$ is the optimal solution of the optimization problem, p_j are cost function weights that tune the importance of each variable, $\beta_{max,j}, j \in \{1, \cdots, 4\}$ are chosen based on practical aspects of the filters, such as the bandwidth of the maximum permissible variations of the impedance profile. $\beta_{max,5}$ is selected based on the maximum expected energy coming from the uncontrolled term $(\dot{x}_l^\top f_e)$. λ_j are the coefficients of β_j in (4.32) and $\zeta = \dot{x}_l^\top D_l^d \dot{x}_l + \dot{e}_f^\top D_f^d \dot{e}_f - \dot{x}_l^\top f_e$ contains the terms that are not controllable by the decision variables β_i.

It is worth mentioning that the optimization problem (4.31) has a linear cost with linear constraints, and that we aim to maximize $\beta_1 \dots \beta_4$ while minimizing β_5, that is why it is necessary to choose $(p_1, \dots, p_4 \leq 0)$ and $(p_5 \geq 0)$.

One important feature of the optimization problem (4.31) is that it is always feasible under a reasonable upper bound $\beta_{max,5}$. For example, one naive solution that is always feasible is the following

$$\beta^* = [0, 0, 0, 0, \bar{\beta}_5]$$

The zeros in the solution cancel the energy generated by all the terms in ω from (4.32), except for the energy generated by the term $(-\dot{x}_l^\top f_e)$. If the energy generated by this term is larger than the energy dissipated by the damping terms, then β_5 is chosen to dissipate the excess of energy. This feature of the proposed method is important to guarantee passivity under all conditions of teleoperation. Although feasible, in practice, a slack variable $\varepsilon \in \mathbb{R}_+$ can be added to the optimization problem to reduce numerical issues. The optimization problem becomes

$$\begin{aligned} \beta^* = \arg \min_{\beta_1 \dots \beta_5, \varepsilon} & (\sum_{j=1}^{5} p_j \beta_j + p_6 \varepsilon) \qquad (4.34) \\ \text{s.t.} \quad & 0 \leq \omega(t) + \varepsilon \\ & 0 \leq \beta_j \leq \beta_{\max,j} : j \in \{1, \dots, 5\} \end{aligned}$$

where p_6 is chosen to be positive and much larger in magnitude than other weights. ε relaxes the passivity condition when needed, while keeping the violation negligible since the penalty decided by p_6 is very large.

The choice of each one of the weights $p_1, \dots, p_5$ decides the balance between the dissipation mechanisms. The effect of each mechanism on the performance can be very different in nature. For example, the parameter related to stiffness filtering on the leader side (β_1) governs the change of stiffness K_l. A similar effect happens on the follower side by using (β_2) to dissipate the energy. Particularly, this happens when the desired controller stiffness K_f^d increases, meaning that the follower robot might be slower to get into the precise-tracking mode, without losing control over it by the operator. β_3 and β_4 which are related to reference position filtering on the leader and follower robots, modify the reference position on each side. In the worst case when $\beta_4 = 0$ and lasts for a period of time, the reference position of the follower robot stops updating. This results in disconnecting the follower from the leader robot, and the follower keeps tracking a fixed reference position until β_4 becomes non-zero, and the reference position goes back to updating and following the leader's position. On the leader side, if $\beta_3 = 0$ and lasts for a period of time, the same effect happens to the position reference. However, the leader robot still renders the force feedback f_e. β_5 is a damping injection parameter on the leader side, which means that it introduces additional resistance to the human operator when it is not zero, and the leader robot feels more difficult to move. In some cases, this might lead to spikes in leader's force. Arguably, filtering the stiffness offers advantages over other mechanisms since it does not introduce spikes in force and it does not disconnect the follower robot from the operator. The above-mentioned factors influence the selection of a role in deciding the weights $p_1, \ldots, p_5$ and $\beta_{max,1}, \ldots, \beta_{max,5}$. However, this task requires some tuning by the user. Choosing these parameters automatically could be possible using, for example, learning-based methods, but it is beyond the scope of this thesis.

Figure 4.1 shows the schematic diagram of the proposed VIC with the passivity filter in bilateral teleoperation, and algorithm 1 summarizes the steps to implement the proposed method in practice.

Algorithm 1 Passivity Filter Algorithm

```
Input:
  $p_j, \beta_{max,j} : j \in \{1 \ldots 5\}$
Online:
  for each time step do
      get states and sensors values $(x_l, \dot{x}_l, \ddot{x}_l)$, $(f_e, x_f, \dot{x}_f, \ddot{x}_f)$
      get the user-desired impedance values $M_l^d, K_l^d, D_l^d, M_f^d, K_f^d, D_f^d$   ▷ (4.19), (4.20)
      calculate $\omega$ coefficients $(\lambda_j, \zeta)$   ▷ (4.33)
      get optimal $\beta_j : j \in \{1 \ldots 5\}$   ▷ by solving (4.34)
      update $M_l, M_f, K_l, K_f, D_l, D_f, x_l^r, x_f^r, \dot{x}_f^r, \ddot{x}_f^r$   ▷ (4.23),. . . ,(4.30)
      calculate $f_l, f_f$ in order to obtain the VIC behavior   ▷ (4.5), (4.7)
  end for
```

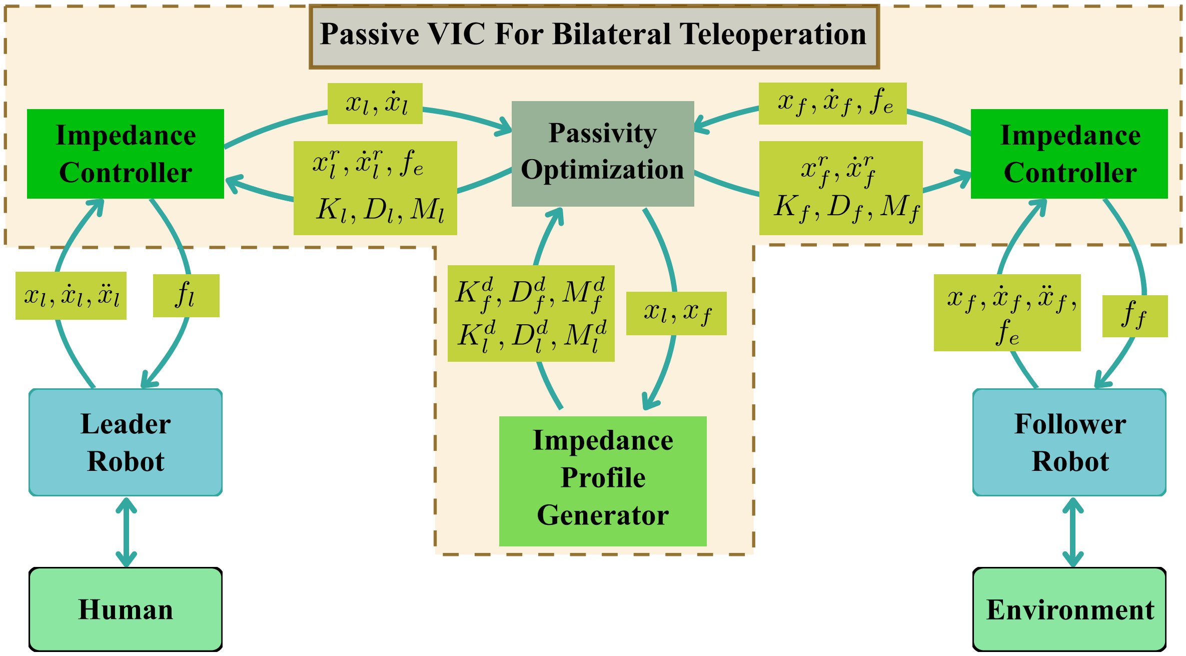

Figure 4.1: Schematic of the proposed VIC with passivity filter in bilateral teleoperation. Note that M_f^d is assumed constant, and M_l^d is assumed equal to the natural inertia matrix H_l. Other desired impedance matrices are set by the impedance profile generator, which is a user-defined map that is a function of the robots' positions (or could be any other function).

4.3 Experimental Validation

In this section, we present the experimental validation of the proposed VIC with passivity guarantees in bilateral teleoperation. First, we present the experiment setup, and an experimental task is designed to evaluate the performance of the proposed method. Finally, the results are presented and analyzed.

4.3.1 Experiment Setup And Task Description

Figure 4.2 illustrates the experimental setup. On the left, the leader is implemented using the *Omega7* haptic interface (Force Dimension, Switzerland), while on the right the follower is implemented with the *Omega3* haptic interface. A force sensor (ATI Nano17, ATI Industrial Automation, USA) was mounted on the follower using a custom 3D-printed adapter. The remote environment consists of a foam block featuring a roughly carved maze-like pattern with a depth of 10 mm (see Figure 4.2, right). The control software is implemented under the ROS2 framework with a 500 Hz operating frequency. The optimization problem is solved using the ProxSuite optimization framework [192], an open-source, efficient, and robust solver for quadratic programs, designed for real-time robotics applications. Both robots are connected to the same computer, ensuring negligible communication delays.

To validate the proposed method, we design a pedagogical example that demonstrates its behavior under varying task requirements. In this example, the advantages of VIC are highlighted by varying the task requirements, including high precision and high forces in certain regions, low forces and safe motion in others, and fast efficient motion along certain directions. We consider a repetitive task where the follower robot is supposed to interact with the Target regions (T1) and (T2) (see Figure 4.3), and to navigate through the tunnel between these two targets as fast as possible without damaging the tunnel walls. This experiment can emulate a scenario where, for example, several parts have to be unmounted from the region T1, and then to be moved through the tunnel to be mounted again in the region T2. To reduce the cognitive effort of the user during the motion through the tunnel, the controller is designed to automatically reduce the forces applied on the tunnel walls, while still following the leader's fast motion along the tunnel axis. This is done by appropriately varying the impedance profile as will be explained next.

We assume that a map is available to determine the desired impedance profile as a

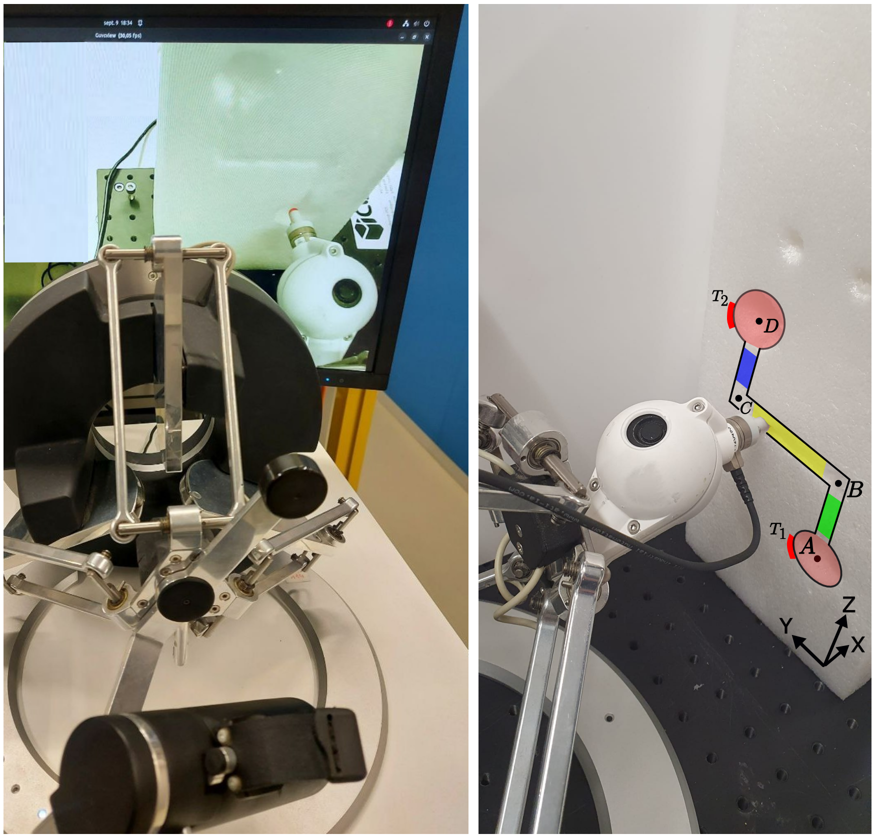

Figure 4.2: (Left) The leader robot with real-time video feedback of the environment. (Right) The follower robot interacting with the environment and the different colored zones

function of the follower robots' position. In practice, such a map could be preplanned by the user or learned from demonstration as in [79,179]. The environment is segmented into different zones, each one is indicated by a distinct color in Figure 4.3. A reference frame is attached to the environment with axes parallel to the different sections of the tunnel to facilitate the analysis of the results.

At this stage, it is worth recalling that setting high desired stiffness for the follower's controller along a specific direction leads to better tracking of the motion reference x_f^r coming from the leader robot along that direction. Hence, in the tunnel, a high desired stiffness is set along the main direction of the tunnel. Conversely, a low desired stiffness is set perpendicularly to reduce the contact forces applied to the tunnel walls. As a result, to control the teleoperation system and perform the two distinct behaviors mentioned previously, the impedance profile of the follower controller is defined based on the follower robot tip position, where it is set based on the colored zone as shown in Figure 4.3. The values are chosen as follows

- in all the experiments, the desired stiffness is 100 $\mathrm{N.m}^{-1}$ along the X-axis, orthogonal to the surface of the environment, which allows an average tracking and forces along this direction;
- working zones: high desired stiffness (500 $\mathrm{N.m}^{-1}$) along the Y- and Z-axes in regions (A) and (D), allowing to apply high force with limited motions on the dark red targets visible in Figure 4.3;
- vertical motion zones (green and blue): high desired stiffness (400 $\mathrm{N.m}^{-1}$) along the Z-axis, and low desired stiffness (50 $\mathrm{N.m}^{-1}$) along the Y-axis;
- horizontal motion zone (yellow): high desired stiffness (400 $\mathrm{N.m}^{-1}$) along the Y-axis and low desired stiffness (50 $\mathrm{N.m}^{-1}$) along the Z-axis;
- corner zones: relatively high desired stiffness (300 $\mathrm{N.m}^{-1}$) along Y- and Z-axes close to the corner points (C) and (B), to facilitate guiding during changes in directions.

The desired damping matrix is set such that the damping ratio is 0.9 for the follower, and the desired inertia is $M_f^d = 0.3$ $\mathrm{kg.m}^2$ along all three axes. To simplify the discussion, the leader robot controller is chosen to work in force reflection mode, which means that $K_l^d = D_l^d = 0$ and that the desired inertia is the same as the natural inertia, meaning $M_l^d(t) = H_l(x_l)$. However, additional damping can be introduced by the optimization if the passivity condition requires so. For the optimization problem (4.31) the maximum values of β_j are chosen as $\beta_{max,1} = \beta_{max,2} = 316$, $\beta_{max,3} = \beta_{max,4} = 75$, $\beta_{max,5} = 100$. After scaling each optimization variable to have a value in $[0, 1]$, we set the cost function weights as

$$[p_1, \ldots, p_5] = [-10, -10, -10, -10, 100]$$

where negative values maximize the corresponding optimization parameter, and positive values minimize it. We mention that we also added a slack variable for the optimization constraint with a weight equal to 10^6.

4.3.2 Experimental Results

In the following figures, the background colors correspond to the ones defining the various environment zones in Figure 4.3. The study is limited to Y- and Z-axes since the motions and forces of interest are along these directions. We illustrate only a part of the repetitive motion for clarity, where it starts by an interaction with the target (T_1), goes through the tunnel, and ends by an interaction with the target (T_2). We demonstrate the performance of the designed teleoperation system in Figure 4.4. In this figure, the blue lines correspond to the leader signals, and the red lines correspond to the follower signals. The position tracking results along the Y-axis and along the Z-axis are shown in Figure 4.4(a) and Figure 4.4(b), respectively. The velocity tracking results along the Y-axis and Z-axis are shown in Figure 4.4(c) and Figure 4.4(d), respectively. Similarly, the force tracking performance along the Y-axis and Z-axis are shown in Figure 4.4(e) and Figure 4.4(f), respectively.

Figure 4.5 describes the curves related to the variable stiffness of the follower robot through time and in different environment zones. Specifically, Figure 4.5(a) depicts the

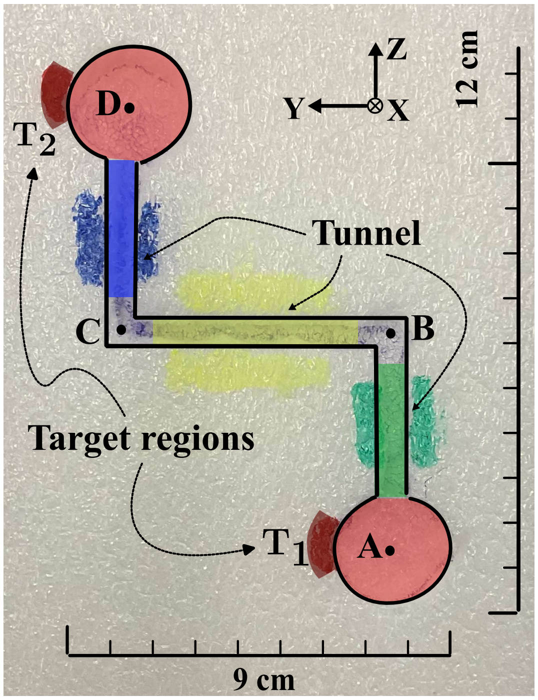

Figure 4.3: The different zones in the environment. The dark red parts, (T_1) and (T_2), are the target regions where the robot has to apply high forces. The other colored zones are carved in the foam material to around 10 mm depth.

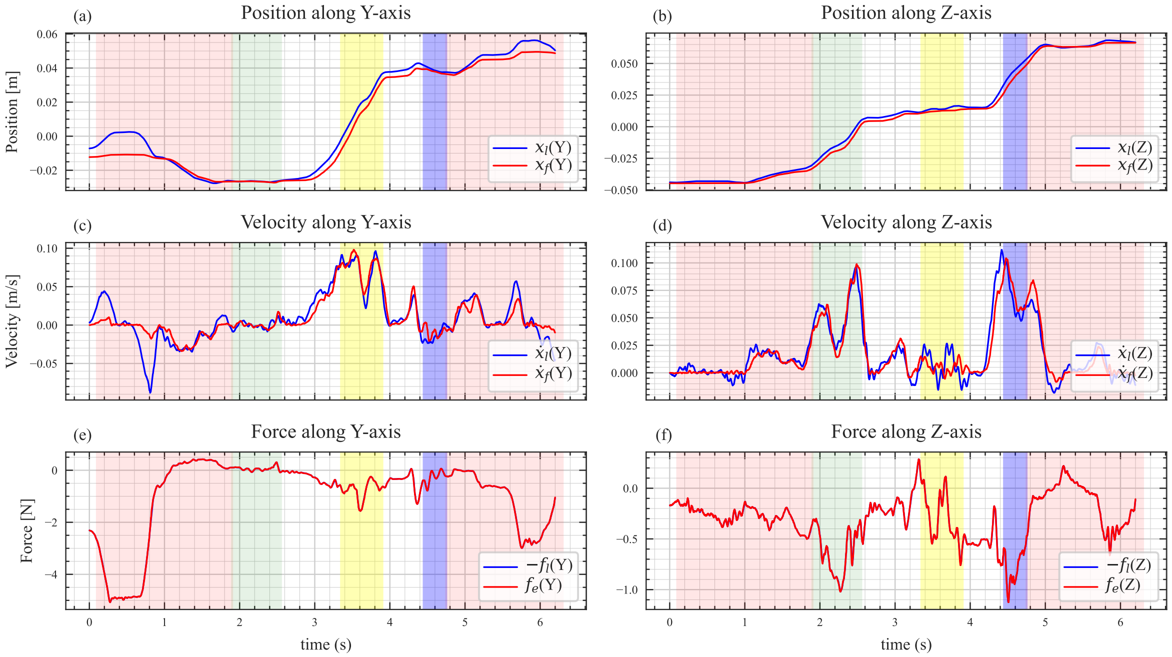

Figure 4.4: The signals of both robots: along Y-axis, (a) positions, (c) velocities, (e) forces. Along Z-axis, (b) positions, (d) velocities, (f) forces. The blue lines correspond to the leader, and the red lines correspond to the follower. The background colors correspond to the different zones in the environment.

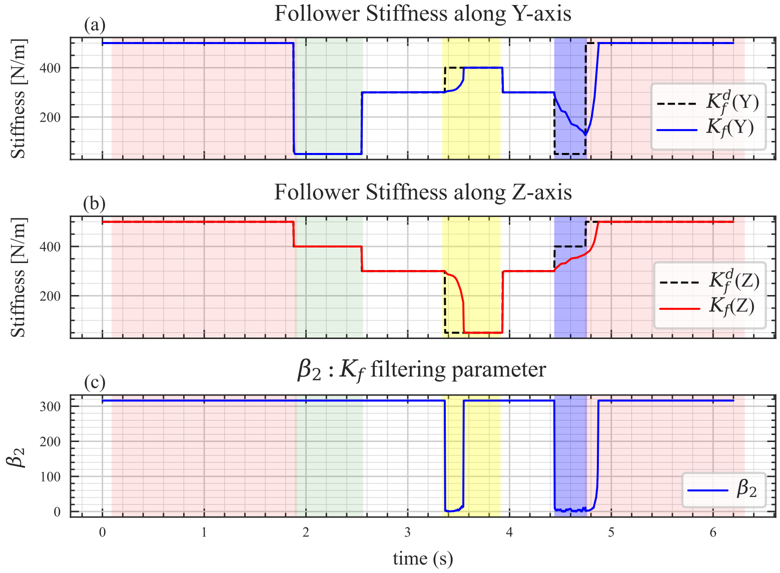

Figure 4.5: Follower's stiffness filtering results. It shows the desired and imposed stiffness along Y-axis (a), and along Z-axis (b). (c) shows the follower's stiffness filtering gain.

desired stiffness profile along the Y-axis $K_f^d(\mathrm{Y})$ (black dashed lines) and the filtered profile $K_f(\mathrm{Y})$ (blue line) which results from filtering the desired one with gain β_2 and will be actually applied on the robot. Figure 4.5(b) depicts similar information along the Z-axis. Figure 4.5(c) shows the gain β_2 that is used to filter the follower's stiffness, which results from solving the optimization problem (4.31). Figure 4.6 illustrates the dissipated power $\omega(t)$ by the system throughout the experiment.

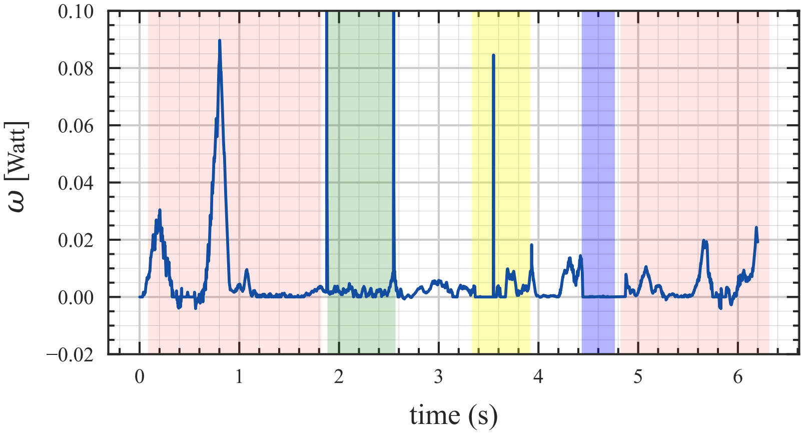

Figure 4.6: $\omega(t)$, the dissipated power by the system defined in (4.32).

The experiment begins in the red working region (A), where the user applies a relatively high force along the Y-axis to interact with the red target (T_1) as can be noticed from Figure 4.4(e) at time $t = 0.2$ s. This phase mimics the task that should be performed on the target.

Once the robot enters the tunnel (green region), the desired stiffness drops along the Y-axis to 50 $\mathrm{N.m^{-1}}$ to prevent high contact forces against the wall (see Figure 4.5(a)). Since decreasing the stiffness is generally a dissipative act (because the term involving $\dot{K}_f$ in ω becomes positive (4.18)), the passivity condition was not violated, and the filtering gain β_2 was set to its maximum value (316) by the solver (see Figure 4.5(c)). Therefore, the filtered impedance tracks the desired one very fast, and they are almost identical (Figure 4.5(a)). Notice the spike in the dissipated power ω at the beginning of the green region (Figure 4.6), which is due to the energy dissipated by decreasing the stiffness.

When the robot comes closer to the corner (first uncolored region, $t \approx 2.5$ s), the impedance is set to 300 $\mathrm{N.m^{-1}}$ along Y- and Z-axes to regain better control of the follower robot and to change the direction as wanted. In this case, varying the impedance did not violate the passivity which is why it was not filtered.

When the robot enters the horizontal part of the tunnel (yellow region), the desired stiffness is increased along the Y-axis and lowered along the Z-axis. To guarantee passivity, the optimization problem (4.31) generates a solution $(\beta_1, \ldots, \beta_5)$ that keeps the passivity. Specifically, the filter gain β_2 is reduced to a low value (Figure 4.5(c)), which filters the stiffness change and slows it down until the dissipated energy allows reaching the desired stiffness around $t \approx 3.6$ s (Figure 4.5(a, b)). During this phase of filtering the stiffness change, ω could be seen to be equal to zero for a period of time (Figure 4.6). A similar case of stiffness lowpass filtering can be noticed in the blue region for the same reason.

At the end of the task, the robot reaches the red region (D), where the stiffness is increased along both Y- and Z-axes, and the β_2 gain slows down this change again to satisfy the passivity condition. In that region, the follower robot applies a relatively high force to the second dark red target (T_2), as required in the task specifications (Figure 4.4(e), at $t \approx 5.6$ s).

Figure 4.7 describes the signals related to the follower's reference position signals along the Y-axis. Figure 4.7(a) depicts the leader's position along the Y-axis x_l(Y) (blue line) and the follower's reference position signal x_f^r(Y) (red dashed line). Figure 4.7(b) illustrates the leader's velocity $\dot{x}_l$(Y) (blue line), and the follower's reference velocity signal $\dot{x}_f^r$(Y) (red dashed line). Figure 4.7(c) shows the gain β_4 which modifies the follower's reference position signal. Ideally, the reference position and velocity signals of the follower robot should come directly from the leader's position and velocity signals. However, in our method, these references are computed from (4.26) by filtering the position x_l with gain β_4. In the presented results, although β_4 sometimes drops from its maximal value because of the necessary filtering (Figure 4.7(c)), no visible effect can be observed in the signals $\dot{x}_f^r$ and x_f^r in Figure 4.7(a, b), except in the yellow region where the velocity reference drops to zero for a very short period around $t \approx 3.5$ s. Similar results were found along the Z-axis as illustrated in Figure 4.8, where again, no visible effect on the tracking can be seen.

The damping injection parameter β_5 is displayed in Figure 4.9. Increases in its value can be noticed at $t \approx 0.4$ s and $t \approx 5.8$ s. However, it remained equal to zero most of the time. Parameters β_1 and β_3, related to the leader's stiffness and position filtering, did not participate in the energy dissipation since leader's stiffness was set to zero. Both parameters did not drop from their max values throughout the experiment, which is logical since dropping them does not dissipate energy in this particular case, and thus they were not displayed here.

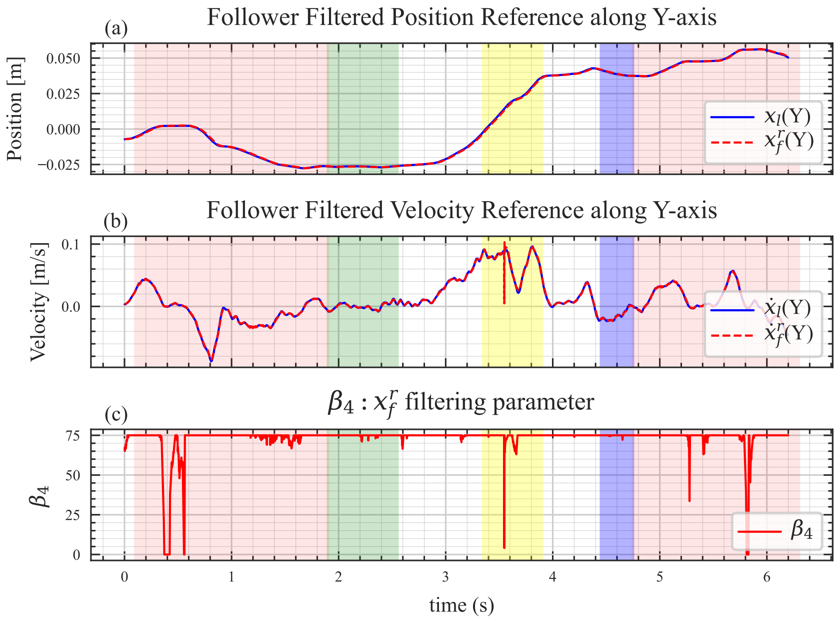

Figure 4.7: Follower's position reference filtering along the Y-axis. (a) leader's position and follower's position reference, (b) leader's velocity and follower's velocity reference, (c) follower's position filtering gain.

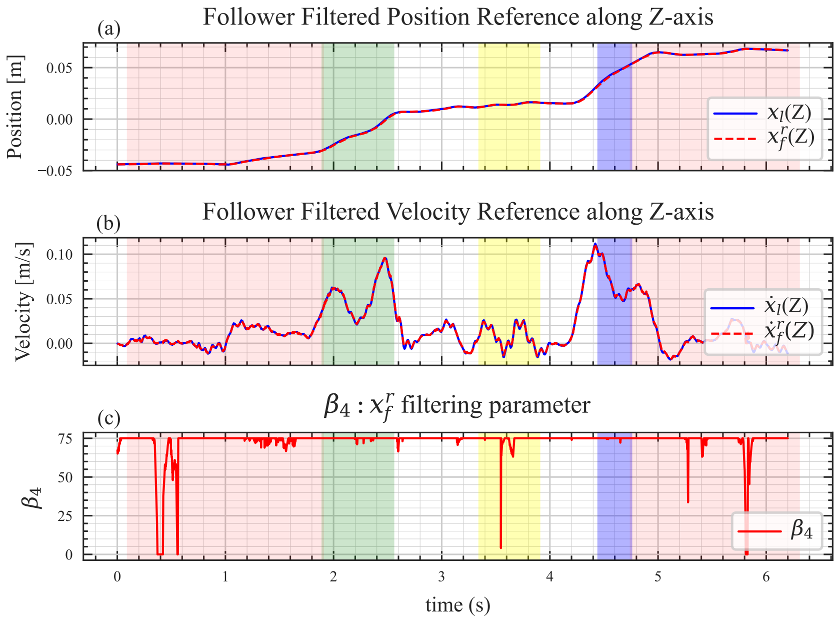

Figure 4.8: Follower's position reference filtering along the Z-axis. (a) leader's position and follower's position reference, (b) leader's velocity and follower's velocity reference, (c) follower's position filtering gain.

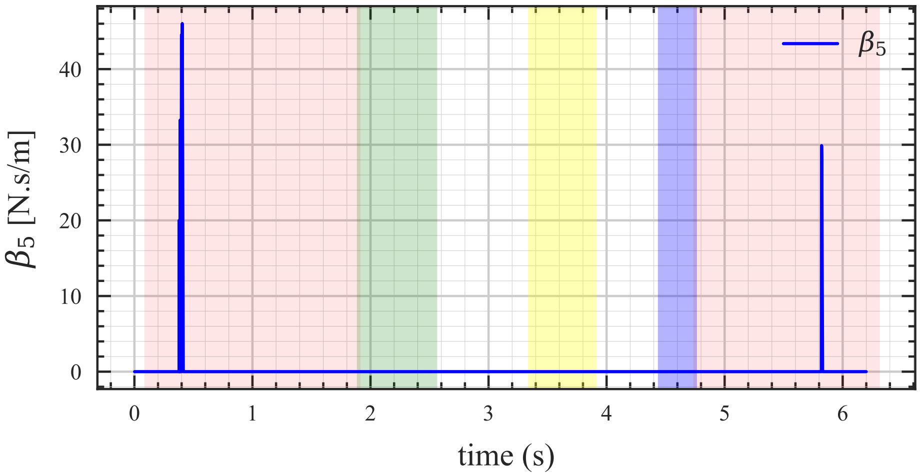

Figure 4.9: β_5: damping injection parameter

4.3.3 Discussion

In this experiment, the main contribution of the optimization comes from parameter β_2, which adapts the impedance parameters of the follower robot. However, the other parameters are also important for guaranteeing the feasibility of the optimization problem and thus the system passivity. It is worth mentioning that the parameter β_1 which filters the leader's impedance has a similar importance. However, it did not contribute to energy dissipation during the study due to the fact that the leader's stiffness $K_l^d = 0$. It should be noted that other conditions of teleoperation and optimization settings (such as the cost weights in (4.31)) could change the effects of these parameters as well.

While it is well known that passivity greatly influences system performance [1], an advantage of our method is that it integrates different ways of dissipating energy, which include filtering the impedance profile, filtering position references, and damping injection. Therefore, by choosing the weights in the optimization problem, the user can prioritize one method over the others depending on the specific control task.

Figure 4.5 shows that in the yellow region, decreasing the stiffness along Z-axis was delayed, although it dissipates energy. The reason behind that is that β_2 filtered the stiffness variation to stop the stiffness increase along the Y-axis at the same time. Therefore, introducing an optimization parameter per each direction separately could reduce the conservatism of the filters and allow for only modifying the stiffness in the needed direction.

One problem with the damping injection is that it can introduce high pulses of force (Figure 4.9), which may cause high-frequency vibrations or chattering. Several methods have been proposed in the literature to deal with this issue, e.g., Virtual-Mass-Spring method [185]. Since this problem usually happens at low velocities, a simple-to-implement method was proposed in [186] where a certain velocity threshold was set, below which there is no need to inject all the calculated damping, especially since the system is safe at low velocities. We also note that although damping injection only on the leader side is theoretically enough for passivity, adding an optimization parameter for damping injection on the follower side can also help in reducing the damping spikes of the leader.

It can be noticed that the dissipated power ω displayed in Figure 4.6 is generally positive, except for very small negative values that come from the slack variable (e.g., at $t = 0.4$ s). Although the optimization problem is theoretically always feasible, these negative values are caused by numerical issues, and a slack variable is often necessary in practice to guarantee the efficient convergence of the solver. Spikes in the dissipated power can be noticed at the beginning and end of the green zone, and in the yellow zone (Figure 4.6). These spikes are likely due to the sudden dropping of the stiffness along one of the axes (Figure 4.5(a, b)), which is a dissipative act. Note that these spikes do not violate the passivity condition since they increase ω.

4.4 Chapter Conclusion

In this chapter, we proposed a variable impedance control formulation for bilateral teleoperation with passivity guarantees. This method addresses the system passivity challenge when the impedance of the follower or leader robots depends on the task and hence is time-varying. To tackle this problem, the method extends the concept of passivity filter to the context of bilateral teleoperation and combines it with different energy dissipation methods (namely, time-domain passivity and passive-set-position-modulation methods). The proposed method uses linear optimization to find the optimal parameters for the energy dissipation. An experimental study based on two 3-DoF robots was carried out to evaluate the effectiveness of this method and the overall performance of the teleoperation scheme.

The main advantage of the method is that it allows for task-specific customization of the energy dissipation strategy. By appropriately tuning the weights in the optimization formulation, the user can choose the preferred dissipation mechanisms based on the control objectives. The method mainly uses reference filtering of the impedance profile and Cartesian position reference. Therefore, these references are altered in a transparent and smooth manner to guarantee passivity.

Future works involve reducing the conservatism of the current method by exploring the possibility of separating the optimization parameters along each axis to allow for different filtering for different axes. Another improvement can be made by storing the dissipated energy for later use, similar to energy tanks, which could reduce the conservatism of the method. However, this requires a careful study of the maximum amount of energy that can be stored safely. Other future works include studying the effect of the cost weights on the performance of the method. Finally, the incorporation of time domain passivity approach can result in some chattering problems due to the damping injection. So the method can be improved by implementing a chattering-free mechanism such as the VMS method [185] or using a velocity threshold under which the damping will not be injected [186].

Chap. 5

Conclusion and Perspectives

5.1 Conclusion

This thesis addresses critical challenges in enhancing the safety, transparency, and robustness of bilateral teleoperation systems. The core objective is to develop and validate advanced control strategies that enable more intuitive, safer, and effective bilateral telemanipulation, especially when dealing with constraints, uncertainties, and the need for adaptable interaction behaviors.

The presented research resulted in three primary contributions. First, acknowledging the importance of constraint handling in safety-critical applications, we developed a Model Predictive Control (MPC) framework tailored for bilateral teleoperation (Chapter 2). Subsequently, and motivated by the practical difficulty of manually tuning MPC parameters for optimal performance, particularly concerning transparency, we introduced an auto-tuning framework based on Bayesian Optimization (BO). This data-driven approach efficiently optimizes the MPC cost function weights by directly minimizing a performance metric evaluated on the system. The effectiveness of this auto-tuning method was demonstrated through both simulation and hardware experiments on a 1-DoF system, significantly improving the transparency with minimal tuning time.

Second, we tackled the crucial issue of robustness against uncertainties, which is paramount for ensuring safety and reliability in real-world scenarios like robotic surgery (Chapter 3). Standard MPC struggles with constraint satisfaction under model uncertainty. To overcome this, we designed an adaptive robust Tube-MPC controller. This controller guarantees robust constraint satisfaction and recursive feasibility despite parametric uncertainties in the environment model (stiffness and damping) and additive disturbances. The framework integrates two learning mechanisms: Set-Membership Learning (SML) refines the uncertainty bounds online, reducing controller conservatism, while Least Mean Squares (LMS) provides a point estimate of the environment parameters to enhance performance when constraints are not active. The efficacy of this approach in satisfying constraints robustly and adapting to the uncertain environment properties was validated through simulation.

Third, addressing the need for robots to dynamically adapt their interaction behavior, we investigated Variable Impedance Control (VIC) within the context of bilateral teleoperation (Chapter 4). While VIC offers significant advantages in flexibility and human-like interaction, it poses challenges to system stability due to potential passivity violations. We propose a novel optimization-based control framework that ensures passivity, and hence stability, for VIC-based bilateral teleoperation. This framework extends the concept of passivity filters and integrates multiple energy dissipation strategies, including impedance profile filtering, reference position modulation, and adaptive damping injection. The controller uses online linear optimization to dynamically adjust these mechanisms, preserving passivity while minimally impacting performance. The feasibility and effectiveness of this passive VIC scheme were validated experimentally using two 3-DoF haptic devices in a task requiring varying interaction dynamics.

In summary, this thesis proposes practical and theoretically grounded methods for advancing bilateral teleoperation control. By developing techniques for efficient MPC tuning, robust constraint handling under uncertainty, and passivity-preserving variable impedance control, this work paves the way for more capable, reliable, and safer teleoperation systems, particularly in demanding fields such as surgical teleoperation.

5.2 Perspectives

The research presented in this thesis lays a foundation for safer and high-performance bilateral teleoperation systems. At the same time, it also opens several avenues for future investigation aimed at enhancing robustness, adaptivity, performance, and practical applicability. We outline these directions hereafter, grouping them into key research themes.

5.2.1 Adaptive-Tuning for Multi Environments

The auto-tuning framework presented in Chapter 2 successfully demonstrated the use of Bayesian Optimization to optimize MPC parameters for transparency in a specific environment. As highlighted by the system model (2.7), variations in environment stiffness and damping coefficient significantly alter the dynamics, implying that a tuning optimized for one environment may be suboptimal for another. This necessitates re-tuning for each new environment condition. One direction to address this limitation is to extend the auto-tuning methodology to explicitly account for the environment variations, enabling the controller tuning to adapt automatically. A reasonable assumption is that finding an optimal tuning for one environment could provide valuable information for similar environments. This idea motivates the use of Contextual Bayesian Optimization (C-BO) [193]. C-BO extends standard BO by modeling the objective function (in our case, the teleoperation performance metric $\mathcal{M}(\phi)$) as dependent on both the decision variables (the MPC weights ϕ that we optimize) and context variables (environment parameters c that are observed but not directly controlled) i.e., $\mathcal{M}(\phi, c)$. The core idea is that the C-BO surrogate model learns the relationship between the context and the optimal decision variables, allowing it to generalize and predict good tuning parameters even for environment contexts not explicitly encountered during the initial learning phase.

C-BO has shown significant potential in robotics for enabling efficient generalization of learned behaviors across different conditions. For instance, the work in [137] employed C-BO for auto-tuning an MPC for autonomous racing, using environment factors like tire friction as context variables to adapt the controller online. Similarly, C-BO was used for adapting robot-learned skills based on object properties or target locations as context parameters [194, 195].

Implementing such a method would likely involve a two-stage process:

1. Offline Learning Stage: Conduct a systematic series of experiments (simulation or hardware) where the teleoperation system interacts with a range of environments characterized by different known stiffness and damping values (c). For each context c, C-BO would be used to explore the MPC weight space ϕ and evaluate the performance $\mathcal{M}(\phi, c)$. This builds the surrogate GP model that captures the mapping $c \to \phi^*$, where ϕ^* represents the optimal weights for context c.

2. Online Implementation Stage: During actual teleoperation, an online mechanism

would be needed to estimate the current environment context $\hat{c}$ (e.g., using LMS). Given $\hat{c}$, the learned C-BO surrogate model would then be queried to predict the optimal MPC weights $\phi^*(\hat{c})$. These weights would be applied to the MPC controller, allowing it to dynamically adapt its behavior to the perceived environment.

While C-BO offers significant potential for adaptive tuning, some challenges must be addressed. For example, the adaptation should ensure seamless transitions, preventing any discontinuities that could disrupt the user experience, or even induce stability issues.

5.2.2 Enhancing the Uncertainty Learning

The adaptive robust Tube-MPC framework developed in Chapter 3 represents an important step towards safe and efficient teleoperation under uncertainty by guaranteeing robust constraint satisfaction and recursive feasibility while reducing conservatism through online learning. However, certain assumptions and characteristics of the current framework require further investigation to broaden its applicability and enhance its robustness in complex real-world scenarios. Two key topics for future work are extending the framework to handle time-varying environment parameters and improving the robustness of the SML component against measurement outliers.

5.2.2.1 Handling Time-Varying Environment Parameters

The current formulation assumes that the true environment parameters $\theta^* = [k_e^*, b_e^*]^\top$ are unknown but constant. However, certain applications might involve environments whose properties drift over time. If θ^* drifts significantly, the SML process, which relies on intersecting the sets (Δ_τ), might eventually lead to an empty feasible parameter set (Θ_τ), thereby invalidating the robustness guarantees of the Tube-MPC.

To extend the work to such cases and prevent the feasible set Θ_τ from becoming empty due to parameter drift, a mechanism of set inflation could be used. At each time step, the updated feasible set Θ_τ can be inflated by an appropriate amount. This allows the set to expand and re-capture the true parameter if it has drifted slightly outside the previous intersection. The rate of inflation should be based on the expected rate of parameter variation. This makes the uncertainty set alternate between growing with time and shrinking due to learning.

5.2.2.2 Improving SML Robustness to Outliers

An issue of standard SML is its sensitivity to outliers. The method fundamentally assumes that every measurement $(x_\tau, u_\tau, x_{\tau+1})$ is consistent with the uncertain system model (3.9) for the true parameter $\theta^* \in \Theta_0$ and disturbance $d_\tau \in \mathcal{D}$. If a single measurement violates these assumptions (e.g., due to unmodeled dynamics, sensor noise spikes, external disturbances exceeding $\mathcal{D}$, or even numerical errors), the resulting set Δ_τ might not contain θ^*. Intersecting the current feasible set $\Theta_{\tau-1}$ with such Δ_τ can irreversibly corrupt the learning process, leading to Θ_τ becoming empty or excluding the true parameter value, thereby compromising the safety guarantees of the robust controller. Future work can focus on enhancing the robustness of SML against outliers by detecting and rejecting them as done in [196, 197], which is crucial for the practical deployment of the proposed controller in noisy scenarios.

5.2.3 Reducing Conservatism and Design Complexity in Robust MPC

While providing safety guarantees, the robust Tube-MPC approach from Chapter 3 introduces conservatism and significant design complexity, particularly in the offline phase. A key source of conservatism lies in the design of the pre-stabilizing controller K and the resulting tube cross-section $\mathcal{P}_0$, defined in subsubsection 3.2.1.4. Since the MPC is applied on top of the low-level controller K, a high-gain K minimizes the tube size (reducing state deviation from the nominal state) but consumes significant control authority. This leaves less room for the MPC to choose the additional control input v_k from (3.45) and potentially forces conservative actions to satisfy the tightened constraints (3.35). Conversely, a low-gain K leads to larger tubes, requiring more constraint tightening and potentially limiting achievable performance even if more control authority is available for v_k. This trade-off is difficult to balance manually.

Determining the controller K, the terminal cost P, the terminal set $\mathcal{X}_f$, and the initial tube shape $\mathcal{P}_0$ typically involves solving complex offline optimization problems (e.g., LMIs). This process can be time-consuming, sensitive to tuning parameters within the design algorithms, and may sometimes fail to find a feasible solution (e.g., a suitable positively invariant set for $\mathcal{P}_0$). Furthermore, the complexity of the resulting $\mathcal{P}_0$ (number of facets) can vary, impacting the online computational load of the MPC, particularly the constraint (3.50) which translates into a set of conditions (B.18), the number of which depends on the complexity of $\mathcal{P}_0$.

The challenges above suggest an opportunity to apply the auto-tuning philosophy from Chapter 2 to the design of the robust MPC itself. Instead of only tuning the MPC cost weights (Q and R) for transparency, future research could explore using BO or similar methods to tune these weights in a way that takes into account their effect on the resulting controller parameters $K, P, \mathcal{X}_f$, and $\mathcal{P}_0$. The auto-tuning in this case should consider the balance between tube size, aggressiveness of K, the conservatism of $\mathcal{X}_f$, and the computational complexity, in addition to the transparency.

5.2.4 Refining Passive Variable Impedance Control

As was mentioned throughout Chapter 4, there are several improvements that could be made in the future to enhance the proposed passivity framework. Specifically, the following points could be addressed.

5.2.4.1 Direction-specific passivity control

The current optimization uses scalar filter gains (β_1 and β_2 from (4.23), and (4.24)) to filter the stiffness matrices of the leader and follower robots, respectively. The effect of this was clear in the experiment in Figure 4.5 at the beginning of the yellow region, where the stiffness was filtered along all directions. The filtering included the stiffness along the Z-axis even though it was decreasing and dissipating energy. Extending the framework to allow independent filtering parameters for different Cartesian axes could reduce conservatism by applying passivity interventions only along directions where they are strictly necessary.

5.2.4.2 Integral passivity and energy management

It should be noted that in this work, we restrict the instantaneous dissipated energy to always be positive, that is $\omega(t) \geq 0$. However, many works use a less conservative condition, only requiring the integral of the dissipated power to be positive which is the passivity condition defined in (4.9). This allows for storing the dissipated energy by the system and using it later for potentially non-passive behaviors, as in energy tanks [198]. Although extending this idea for our method is relatively straightforward, determining the maximum amount of energy that can be safely stored is highly task-dependent and a challenging issue, which is why we left it for future work.

5.2.4.3 Chattering mitigation

A specific challenge associated with the proposed passive VIC framework relates to the damping injection mechanism, controlled by the parameter $\beta_5(t)$ from (4.27). While theoretically sufficient for ensuring passivity by dissipating excess energy associated with the term $-\dot{x}_l^\top f_e$, the practical implementation can lead to undesirable interaction artifacts. The optimization problem (4.34) determines the minimal required β_5 at each time step to satisfy the passivity constraint $\omega(t)+\varepsilon \geq 0$. If the system generates energy rapidly, the optimizer might command a large, sudden increase in β_5 from zero. This abrupt injection of significant damping, especially when the leader robot's velocity $\dot{x}_l$ is low, can be perceived by the human operator as a sharp, unnatural braking force or a high-frequency vibration, commonly referred to as chattering. Future work should focus on integrating established chattering reduction techniques into the optimization framework. Two promising directions, inspired by the literature reviewed in Section 4.1.2, are the following. First, inspired by chattering-free TDPA [186], damping injection via β_5 could be significantly reduced or eliminated when the leader velocity $|\dot{x}_l|$ falls below a predefined threshold. This would avoid the issue at low speeds, where chattering is most probable, potentially at the cost of minor, temporary passivity violations that are often acceptable if the velocity is low. The second possible method would be to use virtual dynamics filtering, drawing from the VMS concept used with TDPA in [185]. The damping force commanded by β_5 could be applied through a virtual dynamic filter. This acts as a low-pass filter, smoothing the force experienced by the user and mitigating high-frequency components caused by rapid changes in β_5.

5.2.4.4 Auto-tuning of passivity weights

The cost weights (p_j) manage the balance between different energy dissipation techniques in the passivity optimization problem (4.34). These weights currently require manual tuning. Developing methods (e.g., learning-based, Bayesian Optimization) to automatically tune these weights based on high-level task requirements or user preferences would enhance usability.

Appendix A

Robotics Background

A.1 Robot Dynamics Linearization

The dynamics of n-DoF serial robot in the joint space can be described by the following equation

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau_c + \tau_{\text{ext}} \tag{A.1}$$

where $q, \dot{q}, \ddot{q}$ are the position, velocity, and acceleration of the robot joints, respectively. $H(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the Coriolis and centrifugal matrix, $g(q) \in \mathbb{R}^n$ is the gravity vector, $\tau_c \in \mathbb{R}^n$ is the control torque, and $\tau_{\text{ext}} \in \mathbb{R}^n$ is the external torque projected into the joint space. A classical approach to control the robot is called computed torque method, where the equation (A.1) is first linearized by canceling the nonlinear terms. After that, a more advanced controller is designed for the linearized decoupled system. Such a method requires the knowledge of the robot model. In the following, we present two linearization cases in the task space, with and without the force sensor.

A.1.1 Model Linearization with Force/Torque Sensor

If a measurement of τ_{ext} is available (e.g., using a force/torque sensor), the following control law can be applied [7]

$$\tau_c = \tau' + \tau_{aux} \tag{A.2}$$

where τ' is the desired torque, and τ_{aux} is the auxiliary torque which is calculated as follows

$$\tau_{aux} = H(q)\nu + C(q, \dot{q}) + g(q) - \tau_{\text{ext}}$$

where ν is the new input which represents the desired acceleration, resulting in a double integrator model in the joint space

$$\ddot{q} = \nu \tag{A.3}$$

The dynamics of the robot in the m-dimension task space can be then found by derivating the forward differential kinematic model of the robot as follows

$$\ddot{x} = J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q} \tag{A.4}$$

where $\ddot{x} \in \mathbb{R}^m$ is the task space acceleration, $J(q) \in \mathbb{R}^{m\times n}$ is the Jacobian matrix, and $\dot{J}(q) \in \mathbb{R}^{m\times n}$ is the time derivative of the Jacobian matrix. By setting the desired joint acceleration ν as

$$\nu = J(q)^{+}u - J(q)^{+}\dot{J}(q, \dot{q})\dot{q}$$

where $J(q)^{+} \in \mathbb{R}^{n\times m}$ is the pseudo-inverse of the analytic Jacobian matrix, the result is a double integrator model in the task space

$$\ddot{x} = u \tag{A.5}$$

where $u \in \mathbb{R}^m$ is the desired acceleration in the task space. The input u can be easily chosen such that the robot behaves in a desired manner. For example, a common practice is to render the robot as a mass-damper system in the task space under the influence of the external force f_{ext} and an additional control force f_c. To achieve that, the following control law can be applied

$$u = M^{-1}(-D\dot{x} + f_c + f_{\text{ext}}) \tag{A.6}$$

where $f_c = J(q)^{-\top}\tau_{\text{ext}}$ and $f_c = J(q)^{-\top}\tau'$, which results in the following dynamics in the task space

$$M\ddot{x} + D\dot{x} = f_c + f_{\text{ext}} \tag{A.7}$$

where $M \in \mathbb{R}^{m\times m}$ is the desired inertia matrix, and $D \in \mathbb{R}^{m\times m}$ is the desired damping matrix. Note that if M and D are chosen as diagonal matrices, the robot behaves as a set of decoupled mass-damper systems, one for each task space dimension.

A.1.2 Model Linearization without Force/Torque Sensor

In case the force/torque sensor is not available, the same behavior of (A.7) can be achieved in the task space using the joints and task space accelerations [199]

$$\tau_{aux} = H(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) - J^{\top}(q)M\ddot{x} - J^{\top}(q)D\dot{x} \tag{A.8}$$

this results in the following dynamic equation

$$M\ddot{x} + D\dot{x} = J^{-\top}(q)\tau' + J^{-\top}(q)\tau_{\text{ext}} \tag{A.9}$$

which can be rewritten as

$$M\ddot{x} + D\dot{x} = f_c + f_{\text{ext}} \tag{A.10}$$

In practice, a low-pass filter is usually applied to the acceleration to reduce the derivation noise. However, since the low-pass filter introduces a delay, the controller can not provide large corrections of the robot inertia, which is why the inertia matrix M is usually chosen to be close to the real inertia of the robot [199].

Appendix B

Calculations for Adaptive Robust MPC Implementation

B.1 Polytopes

In this section, we provide an overview of polytopes and key related concepts to the robust MPC formulation in Chapter 3. An understanding of this concept is essential for understanding the Tube-based MPC.

B.1.1 Definition

An n-dimensional polytope is a bounded convex set in $\mathbb{R}^n$ defined by a finite number of linear inequalities. It can either be represented as the intersection of half-spaces, or as the convex hull of its vertices. In this thesis, we primarily use the half-space representation, or *H-representation*. In n-dimensional space, a polytope is defined by the H-representation with $n_{\mathcal{P}}$ linear inequalities as follows

$$\mathcal{P} = \{x \in \mathbb{R}^n \mid H_{\mathcal{P}} x \leq h_{\mathcal{P}}\},$$

where $H_{\mathcal{P}} \in \mathbb{R}^{n_{\mathcal{P}} \times n}$ and $h_{\mathcal{P}} \in \mathbb{R}^{n_{\mathcal{P}}}$. In Chapter 3, the same naming convention is used for different polytopes by replacing the name of the polytope. Intuitively, a polytope is a geometric object that generalizes polygons in two dimensions and polyhedra in three dimensions.

Figure B.1 shows a simple example of a polytope in $\mathbb{R}^2$ defined by its vertices (on the left) and by the intersection of half-spaces (on the right). Although visualization

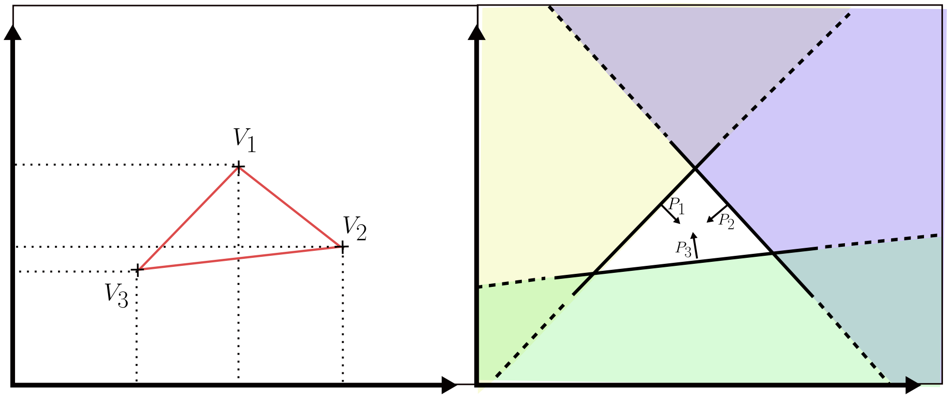

Figure B.1: Example of a polytope in $\mathbb{R}^2$. The left side shows the polytope defined by its vertices (V_1, V_2, V_3), while the right side shows the same polytope defined by its half-spaces with normal vectors (P_1, P_2, P_3).

becomes challenging in higher dimensions, the same principles apply.

To write the H-representation of the shown polytope in the figure, we write the inequalities that define these half-spaces as follows

$$P_1^\top x \geq b_1$$
$$P_2^\top x \geq b_2$$
$$P_3^\top x \geq b_3$$

where $x \in \mathbb{R}^2$. $P_i \in \mathbb{R}^2$ are the normal vectors of the half-spaces, and $b_i \in \mathbb{R}$ are corresponding offset scalars. In this case, the H-representation of the polytope is given by matrices $H_\mathcal{P}$ and $h_\mathcal{P}$ as follows

$$H_\mathcal{P} = \begin{bmatrix} -P_1^\top \\ -P_2^\top \\ -P_3^\top \end{bmatrix}, \quad h_\mathcal{P} = \begin{bmatrix} -b_1 \\ -b_2 \\ -b_3 \end{bmatrix}$$

In this case, the polytope is represented by 3 inequalities and $H_\mathcal{P} \in \mathbb{R}^{3\times 2}$ and $h_\mathcal{P} \in \mathbb{R}^3$.

B.1.2 Minkowski Sum and Tube Construction

A crucial operation in robust MPC is the *Minkowski sum.* For two sets A and B in $\mathbb{R}^n$, the Minkowski sum is defined as

$$A \oplus B = \{a + b \mid a \in A,\, b \in B\}.$$

In case A is a point in space, the Minkowski sum is equivalent to a translation of the set B by the point A.

B.2 Uncertain Model Matrices

The matrices of the uncertain system from (3.5) and (3.7) are given by

$$A_0 = \begin{bmatrix} 1 & T_s & 0 & 0 & 0 \\ 0 & 1 - \frac{b_l Ts}{m_l} & 0 & 0 & \frac{Ts}{m_l} \\ 0 & 0 & 1 & T_s & 0 \\ 0 & 0 & -\frac{k_e^0 Ts}{m_f} & 1 - \frac{(b_f + b_e^0)Ts}{m_f} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.1}$$

$$A_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{Ts}{m_f} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{Ts}{m_f} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{B.2}$$

$$C_\theta = C_0 + C_1\theta_1 + C_2\theta_2 \tag{B.3}$$

$$C_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -k_e^0 & -b_e^0 & 0 \end{bmatrix}, C_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}, C_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \end{bmatrix} \tag{B.4}$$

B.3 Dynamics of Uncertainty Propagation

In this section, we will present more details about uncertainty propagation with linear inequalities in the HTMPC. Specifically, the dynamics of w_k and s_k from equations (3.49) and (3.50). Due to the complexity of the problem, we show the main ideas and the needed equations for reimplementing the work. However, we do not go into the proof or detailed derivations. Interested readers are referred to [168] for a more detailed explanation.

As discussed in Chapter 3, the tube cross-section $\mathcal{P}_k$ should be scaled to ensure that it contains all possible values of x_k under all uncertainties. To achieve that, the scaling factor s_k should take into account the effect of both parametric and additive uncertainties. For that, the following condition should be satisfied

$$x_k \in \bar{x}_k \oplus \mathcal{P}_k \Rightarrow x_{k+1} \in \bar{x}_{k+1} \oplus \mathcal{P}_{k+1} \tag{B.5}$$

$$\forall d_k \in \mathcal{D}, \forall \theta \in \Theta, k \in \{0 \dots N\} \tag{B.6}$$

here, $\mathcal{P}_k = s_k \mathcal{P}_0$, where $\mathcal{P}_0$ is the initial cross-section defined as

$$\mathcal{P}_0 = \{x \in \mathbb{R}^{n_x} : H_{\mathcal{P}_0} x \leq 1_{n_{\mathcal{P}_0}}\} \tag{B.7}$$

The dynamics of the tube scaling factor can be expressed as

$$s_{k+1} = \rho(\Theta) s_k + w_k(\Theta, \mathcal{D}) \tag{B.8}$$

which can be seen as a projection of the uncertainty from the state space into scalar dynamics. This could be used to achieve robustness with significantly less number of constraints.

To reduce online computational complexity, Kohler et al. [168] proposed transforming some components of (B.8) into inequalities based on constants computed offline. This approach introduces conservatism but allows for better real-time implementation. The contraction rate $\rho(\Theta)$ is computed as the worst case over $\theta \in \Theta$. When Θ is a hypercube, $\rho(\Theta)$ can be upper-bounded using the center and the size of the set Θ and a constant L_B computed offline as

$$\rho(\Theta) \leq \rho_{\bar{\theta}} + \eta L_B \tag{B.9}$$

where $\rho_{\bar{\theta}}$ is the contraction rate at $\bar{\theta}$, the center of Θ, given by

$$\rho_{\bar{\theta}} = \max_{i} \max_{x \in \mathcal{P}_0} [H_{\mathcal{P}_0}]_i A_{cl,\bar{\theta}} x \tag{B.10}$$

$$i = 1, \ldots, n_{\mathcal{P}_0} \tag{B.11}$$

and L_B is a constant computed offline, defined as

$$L_B = \max_{i,r} \max_{x \in \mathcal{P}} [H_{\mathcal{P}_0}]_i D_x(x) \nu_r \tag{B.12}$$

$$r = 1, \ldots, 2^{n_\theta}, i = 1, \ldots, n_{\mathcal{P}_0} \tag{B.13}$$

where D_x was introduced in (3.8), ν_r are the vertices of the unit hypercube $\mathbb{B}_{n_\theta} = \{\theta \in \mathbb{R}^{n_\theta} : \|\theta\|_\infty \leq 0.5\}$, and r is the index of the vertices. The term $w_k(\Theta, \mathcal{D})$ in the scaling dynamics can be separated as

$$w_k(\Theta, \mathcal{D}) = w_k(\Theta) + w_k(\mathcal{D}) \tag{B.14}$$

where the effect of the additive uncertainty $\mathcal{D}$ is bounded by a constant

$$w_k(\mathcal{D}) \leq \bar{d} := \max_{i} \max_{d \in \mathcal{D}} [H_{\mathcal{P}_0}]_i d \tag{B.15}$$

$$i = 1, \ldots, n_{\mathcal{P}_0} \tag{B.16}$$

and the effect of θ on w_k at the nominal point $(\bar{x}_k, \bar{u}_k)$ is bounded by

$$\begin{aligned} & w_k(\Theta) \leq \max_{i,r} \eta [H_{\mathcal{P}_0}]_i D_x(\bar{x}_k) \nu_r \\ & r = 1, \ldots, 2^{n_\theta}, i = 1, \ldots, n_{\mathcal{P}_0} \end{aligned} \tag{B.17}$$

Finally, by substituting (B.9), (B.14), (B.15), and (B.17) into (B.8), we can rewrite the dynamics of the scaling factor as

$$\begin{aligned} & s_{k+1} = (\rho_{\bar{\theta}_\tau} + \eta_\tau L_B s_k) s_k + w_k \\ & w_k \geq \bar{d} + \eta_\tau [H_{\mathcal{P}_0}]_i D(\bar{x}_k) \nu_r \\ & r = 1, \ldots, 2^{n_\theta}, i = 1, \ldots, n_{\mathcal{P}_0} \end{aligned} \tag{B.18}$$

Note that this set of inequalities replace the dynamics of s_k and w_k in the MPC optimization constraints (3.49) and (3.50), which introduces $(2^{n_\theta} \cdot n_{\mathcal{P}_0})$ constraints for each k.

B.4 Detailed Derivations for Constraint Tightening

As was shown in Chapter 3, the tightened constraints from (3.20) considering the uncertainty is given by

$$\bar{x}_k \oplus s_k \mathcal{P}_0 \subseteq \mathcal{X} \tag{B.19}$$

To be able to solve the problem using traditional linear solvers, this condition should be reformulated from a set inclusion form into a linear form. Once again, the tightened constraints can be simplified by separating the parts that can be computed offline. The original polytopic MPC constraints with no uncertainty were written in (2.14) as

$$H_{\mathcal{X}} x_k + H_{\mathcal{U}} u_k \leq 1_{n_c} \tag{B.20}$$

which includes a mixed constraint on states and control inputs. However, the tightening idea applies only to states since it is the part that has uncertainty, while the control input is decided by the MPC itself. Therefore, the tightened constraints for all uncertainties, and after applying the pre-stabilizing controller K from (3.31), can be written as

$$\begin{aligned} & H_{\mathcal{X}} x_k + H_{\mathcal{U}}(K x_k + v) \leq 1_{n_c} \\ & \forall x_k \in \bar{x}_k \oplus \mathcal{P}_k \end{aligned} \tag{B.21}$$

Next, each row of the constraint can be analyzed separately. For a given row (j), the previous condition is equivalent to the following

$$\max_{x_k \in \bar{x}_k \oplus \mathcal{P}_k} \underbrace{([H_{\mathcal{X}}]_j + [H_{\mathcal{U}}]_j K)}_{H_{\mathrm{cl},j}} x_k + [H_{\mathcal{U}}]_j v \leq 1 \tag{B.22}$$

where $[\cdot]_j$ denotes the (j)-th row of the matrix. This can be rewritten as

$$\max_{z \in \mathcal{P}_k} H_{\mathrm{cl},j}(\bar{x}_k + z) + [H_{\mathcal{U}}]_j v \leq 1 \tag{B.23}$$

Using the fact that $\mathcal{P}_k$ contains the origin, we know that

$$\max_{z \in \mathcal{P}_k} H_{\mathrm{cl},j} z \geq 0 \tag{B.24}$$

From that, a sufficient condition for the constraint (B.23) to hold is [1]

$$H_{\mathrm{cl},j}\bar{x}_k + \max_{z\in\mathcal{P}_k}\left(H_{\mathrm{cl},j}z\right) + [H_{\mathcal{U}}]_j v \leq 1. \tag{B.25}$$

Note that the maximization does not depend on x_k anymore, and knowing that ($\mathcal{P}_k = s_k\mathcal{P}_0$), this maximization could be computed offline. Thus, we define the constant c_j for each constraint (j) as follows

$$c_j := \max_{z\in\mathcal{P}_0}\left(H_{\mathrm{cl},j}z\right). \tag{B.26}$$

Finally, online, the tightened constraint (j) from (B.25) can be simplified using the computed constants c_j as

$$s_k c_j + H_{\mathrm{cl},j}\bar{x}_k + [H_{\mathcal{U}}]_j v \leq 1 \tag{B.27}$$

which resembles the original constraint form, but with the addition of the tightening scalar term $s_k c_j$. This is the tightened constraint that is used in the HTMPC optimization problem (3.45).

---

[1] This transition from (B.23) to (B.25) uses the following inequality for a constant a $\max_{b\in\mathcal{P}}(a+b) \leq \max_{b\in\mathcal{P}} a + |\max_{b\in\mathcal{P}} b| = a + \max_{b\in\mathcal{P}} b$ where $a \in \mathbb{R}$ and $\mathcal{P} \subset \mathbb{R}^{+}$.

Appendix C

Extended Abstract (French)

Introduction Générale

La téléopération bilatérale, qui permet à un opérateur humain de commander à distance des manipulateurs robotiques tout en recevant un retour d'effort kinesthésique, revêt une importance capitale dans de nombreuses applications. Celles-ci s'étendent de la chirurgie minimalement invasive, où la précision et la sensation tactile sont primordiales, à l'intervention en environnements hostiles (nucléaire, sous-marin) où la présence humaine directe est impossible ou dangereuse, jusqu'à l'exploration spatiale qui repousse les frontières de l'interaction à distance. Le retour d'effort est un élément crucial de ces systèmes, car il vise à procurer à l'opérateur une sensation d'interaction directe avec l'environnement distant, améliorant ainsi l'efficacité de la manipulation et la sécurité des opérations.

Malgré les progrès significatifs réalisés, la conception de systèmes de téléopération bilatérale performants et sûrs demeure un défi scientifique et technologique majeur. Plusieurs problématiques fondamentales doivent être adressées. Premièrement, il est impératif de garantir la *stabilité* du système couplé, qui inclut l'opérateur humain, le robot maître (manipulé par l'opérateur), le robot esclave (interagissant avec l'environnement distant) et l'environnement lui-même. Les retards de communication, les non-linéarités et les incertitudes peuvent facilement déstabiliser un tel système. Deuxièmement, une haute *transparence* est recherchée, c'est-à-dire la capacité du système à transmettre fidèlement les informations de position et de force entre l'opérateur et l'environnement, donnant à l'opérateur l'illusion d'une manipulation directe. Troisièmement, la *sécurité* de l'opération est une préoccupation centrale, impliquant le respect strict de contraintes sur les mouvements et les forces, tant pour protéger l'opérateur que l'environnement manipulé, qui peut être fragile (par exemple, des tissus biologiques en chirurgie). Enfin, les systèmes doivent faire preuve d'*adaptabilité* face aux incertitudes inhérentes aux modèles de l'environnement (rigidité, amortissement variables et souvent inconnus) et aux perturbations externes.

Cette thèse se propose d'apporter des contributions significatives pour relever ces défis, en se concentrant sur le développement et la validation de nouvelles stratégies de commande. Les travaux présentés s'articulent autour de trois axes principaux visant à améliorer la sécurité, la transparence et la robustesse des robots téléopérés avec retour d'effort.

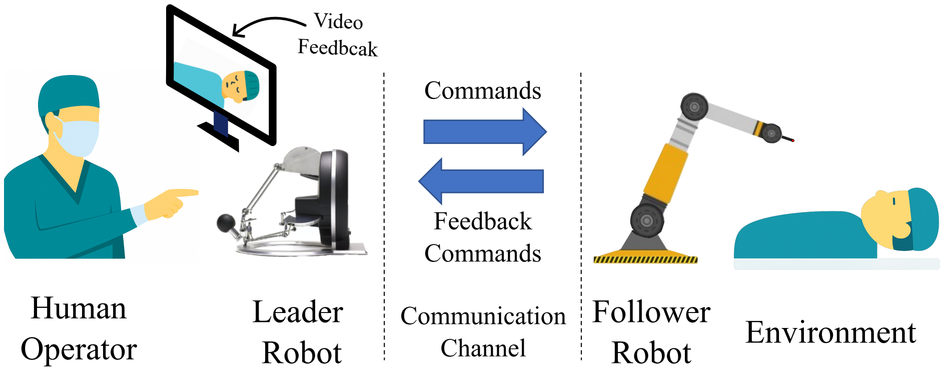

Figure C.1: Schéma de principe d'un système de téléopération bilatérale maître-esclave avec retour d'effort, illustrant les flux d'information.

Pour aborder les problématiques énoncées, cette thèse s'appuie sur et développe plusieurs concepts clés de la théorie de la commande et de la robotique. La **passivité** est un concept fondamental utilisé pour garantir la stabilité des systèmes téléopérés, en s'assurant que le système ne génère pas d'énergie nette. La **transparence**, quantifiée par diverses métriques, mesure la fidélité avec laquelle l'opérateur perçoit l'environnement distant. La **sécurité** est abordée comme une combinaison de plusieurs aspects, notamment la transparence, les contraintes de sécurité, la robustesse et l'adaptabilité. La **commande à impédance variable (VIC)** est explorée pour permettre une adaptation dynamique du comportement interactif du robot. Enfin, la **Commande Prédictive Basée sur Modèle (MPC)** est un outil central utilisé pour sa capacité à optimiser la performance tout en respectant des contraintes complexes sur les états et les commandes du système.

Les contributions principales de cette thèse se structurent comme suit :

1. **Auto-réglage de la Commande Prédictive (MPC) pour la Transparence en Téléopération Bilatérale :** Face à la difficulté du réglage manuel des nombreux paramètres du MPC, un cadre d'auto-réglage basé sur l'Optimisation Bayésienne (BO) a été développé. Cette méthode permet d'optimiser systématiquement les poids de la fonction de coût du MPC afin de maximiser la transparence de l'interaction, évaluée par une métrique de performance quantitative.

2. **Téléopération Bilatérale avec Contraintes Robustes : MPC Robuste Adaptatif :** Pour garantir la sécurité des opérations en présence d'incertitudes sur le modèle de l'environnement, un contrôleur MPC robuste adaptatif, basé sur l'approche "Tube-MPC", a été conçu. Ce contrôleur assure la satisfaction robuste des contraintes et la faisabilité récursive du problème d'optimisation, tout en intégrant des mécanismes d'apprentissage en ligne pour réduire le conservatisme et améliorer la performance.

3. **Commande à Impédance Variable (VIC) Passive pour la Téléopération Bilatérale :** La VIC offre une grande flexibilité d'interaction mais peut compromettre la passivité du système. Un nouveau cadre de commande a été proposé pour garantir la passivité des systèmes de téléopération bilatérale utilisant la VIC. Cette approche étend le concept des filtres de passivité et intègre, via une optimisation en ligne, plusieurs mécanismes de dissipation d'énergie pour maintenir la stabilité tout en minimisant l'impact sur la performance.

Le reste de ce résumé détaillera successivement chacune de ces contributions, en présentant la problématique spécifique abordée, l'approche méthodologique développée et les principaux résultats de validation obtenus.

Contribution 1 : Auto-réglage du MPC pour la Téléopération Bilatérale

La Commande Prédictive Basée sur Modèle (MPC) est une stratégie de commande avancée particulièrement adaptée à la téléopération bilatérale en raison de sa capacité à gérer explicitement les contraintes sur les états (positions, vitesses) et les commandes (forces/couples des actionneurs), tout en optimisant un critère de performance. En téléopération, ce critère est souvent lié à la transparence, c'est-à-dire la capacité à minimiser les erreurs de suivi entre le robot maître et le robot esclave en termes de position, de vitesse et de force. Cependant, la performance d'un contrôleur MPC dépend fortement du choix des poids dans sa fonction de coût. Le réglage manuel de ces poids est une tâche ardue, empirique et chronophage, surtout lorsque la dimension de l'espace des paramètres est élevée et que leur influence sur le comportement en boucle fermée n'est pas intuitive.

Pour pallier cette difficulté, cette première contribution propose un cadre d'auto-réglage des paramètres du MPC spécifiquement conçu pour optimiser la transparence en téléopération bilatérale. L'approche a été initialement formulée et validée pour un système de téléopération maître-esclave à un degré de liberté (1-DoF).

La démarche comprend d'abord la formulation du contrôleur MPC. Le modèle dynamique du système 1-DoF est établi, incluant le robot maître, le robot esclave et une représentation simplifiée de l'environnement avec lequel le robot esclave interagit. La fonction de coût du MPC est définie pour minimiser une somme pondérée des erreurs quadratiques prédites entre les positions, les vitesses et les forces du maître et de l'esclave sur un horizon de prédiction fini. Des contraintes sont imposées sur les commandes des actionneurs pour refléter leurs limites physiques.

Ensuite, le cadre d'auto-réglage basé sur l'Optimisation Bayésienne (BO) est introduit. Le BO est une technique d'optimisation globale efficace pour les fonctions "boîte noire" coûteuses à évaluer, ce qui est le cas de la performance d'un système de téléopération avec un jeu de paramètres MPC donné. Une métrique de performance quantitative, basée sur l'intégration des erreurs de suivi mesurées en boucle fermée (position, vitesse, force) sur une durée d'interaction typique, est définie. L'algorithme de BO est alors utilisé pour explorer l'espace des poids de la fonction de coût du MPC, en cherchant itérativement à minimiser cette métrique de performance. Le BO construit un modèle statistique (processus Gaussien) de la fonction objectif et utilise une fonction d'acquisition pour décider intelligemment du prochain jeu de poids à évaluer, équilibrant l'exploration de nouvelles régions et l'exploitation des régions prometteuses.

La validation du cadre d'auto-réglage du MPC a été réalisée à la fois par des simulations et par des expérimentations sur le banc d'essai 1-DoF maître-esclave décrit précédemment (voir Figure 2). Le protocole expérimental impliquait des interactions répétées où un opérateur (ou un profil de force simulé) appliquait des efforts sur le robot maître, tandis que le robot esclave interagissait avec un environnement de type ressort.

Les résultats ont démontré l'efficacité de l'approche d'auto-réglage. En simulation, l'algorithme d'Optimisation Bayésienne a convergé en un nombre réduit d'itérations

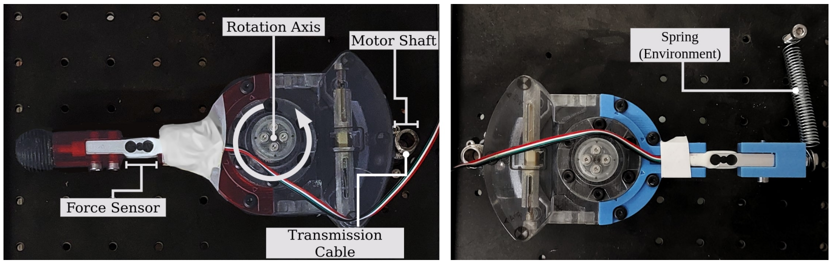

Figure C.2: Banc d'essai 1-DoF pour l'auto-réglage du MPC.

(typiquement moins de 15 essais de 3 secondes chacun) vers un jeu de poids pour le MPC qui améliorait de manière significative la transparence du système. Cette amélioration se traduisait par une réduction notable des erreurs de suivi en position, vitesse et force entre le robot maître et le robot esclave, se rapprochant d'un comportement idéalement transparent.

Les expérimentations sur le banc d'essai matériel ont confirmé ces observations. Malgré les imperfections inhérentes à un système réel (frictions non modélisées, bruit de mesure), l'auto-réglage a permis d'obtenir une amélioration substantielle de la transparence par rapport à un réglage initial empirique. Le temps total requis pour l'auto-réglage sur le matériel est resté de l'ordre de quelques minutes, ce qui est considérablement plus rapide qu'un processus de réglage manuel itératif. La Figure 3 illustre typiquement l'évolution de la métrique de performance durant le processus d'optimisation et une comparaison qualitative des signaux avant et après l'application des poids optimisés.

En discussion, les principaux avantages de cette méthode résident dans son efficacité en termes de nombre d'évaluations requises et dans la réduction significative de l'effort et du temps de réglage pour l'ingénieur. Une limite de cette étude initiale est que l'auto-réglage a été validé pour un seul type d'environnement. L'adaptation à des environnements variés constitue une perspective intéressante. Néanmoins, ce travail établit la faisabilité et l'intérêt de l'Optimisation Bayésienne pour l'auto-réglage des contrôleurs MPC en téléopération bilatérale.

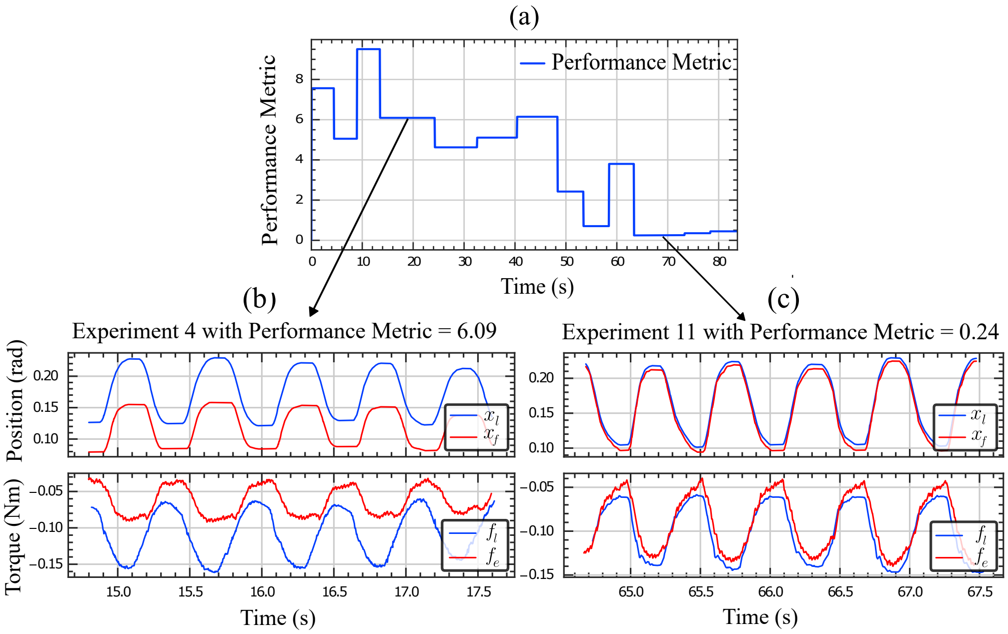

Figure C.3: Graphique illustrant l'amélioration de la transparence avec l'auto-réglage du MPC. (a) Évolution de la métrique de performance durant l'optimisation Bayésienne. (b) Comparaison des signaux de position (haut) et de force (bas) du robot maître (bleu) et du robot esclave (rouge) avant (gauche) et après (droite) auto-réglage.

Contribution 2 : Téléopération Bilatérale avec Contraintes Robustes (MPC Robuste Adaptatif)

Si le MPC standard permet de gérer les contraintes, sa performance et sa capacité à garantir la satisfaction de ces contraintes peuvent être compromises en présence d'incertitudes significatives sur le modèle du système, en particulier sur les paramètres de l'environnement avec lequel le robot esclave interagit (par exemple, la rigidité et l'amortissement de tissus biologiques en chirurgie robotique). Une violation des contraintes de sécurité (par exemple, dépassement de limites de l'espace de travail) peut avoir des conséquences graves.

Cette deuxième contribution vise à relever ce défi en proposant un cadre de Commande Prédictive Robuste et Adaptative pour la téléopération bilatérale, basé sur l'approche "Tube-MPC". L'objectif est de garantir la satisfaction robuste des contraintes (c'est-à-dire pour toutes les réalisations possibles des incertitudes bornées) et la faisabilité récursive du problème d'optimisation MPC, tout en s'adaptant en ligne aux propriétés de l'environnement pour améliorer la performance et réduire le conservatisme inhérent aux approches robustes.

Le principe du Tube-MPC consiste à calculer à chaque instant une trajectoire nominale pour le robot esclave et un "tube" d'incertitude autour de cette trajectoire. Ce tube est construit de manière à contenir toutes les trajectoires réelles possibles du système compte tenu des incertitudes. Les contraintes sont alors resserrées sur la trajectoire nominale de telle sorte que si la trajectoire nominale respecte ces contraintes resserrées, alors toutes les trajectoires réelles à l'intérieur du tube respecteront les contraintes originales.

Le modèle du système de téléopération est formulé pour inclure des incertitudes paramétriques bornées sur la rigidité (k_e) et l'amortissement (b_e) de l'environnement, ainsi que des perturbations additives bornées. Deux mécanismes d'apprentissage en ligne sont intégrés au contrôleur Tube-MPC :

1. **Apprentissage par Appartenance à un Ensemble (Set-Membership Learning - SML) :** Cette technique affine en ligne l'ensemble des valeurs possibles pour les paramètres incertains de l'environnement (le "set" d'incertitude Θ). À chaque nouvelle mesure de l'interaction, les régions de l'ensemble Θ qui sont incohérentes avec les observations sont éliminées. Cela permet de réduire la taille du tube d'incertitude et donc le conservatisme du contrôleur, car les marges de resserrement des contraintes deviennent moins importantes.

2. **Moindres Carrés Moyens (Least Mean Squares - LMS) :** Parallèlement, un algorithme LMS est utilisé pour estimer une valeur ponctuelle ($\hat{\theta}$) des paramètres de l'environnement. Cette estimation est utilisée dans la fonction de coût du MPC pour prédire le comportement le plus probable du système et ainsi optimiser la performance (par exemple, la transparence) lorsque le système opère loin des contraintes.

La formulation du contrôleur MPC robuste adaptatif combine les éléments du Tube-MPC avec les informations issues des modules d'apprentissage SML et LMS. Plus précisément, la propagation du tube d'incertitude (sa taille et son évolution) dépend de l'ensemble d'incertitude Θ_τ fourni par le SML à l'instant τ. La trajectoire nominale, sur laquelle les contraintes sont resserrées, est calculée en utilisant le centre de cet ensemble Θ_τ. La fonction de coût du MPC, quant à elle, est optimisée en utilisant la trajectoire prédite avec l'estimation ponctuelle $\hat{\theta}_\tau$ des paramètres fournie par le LMS.

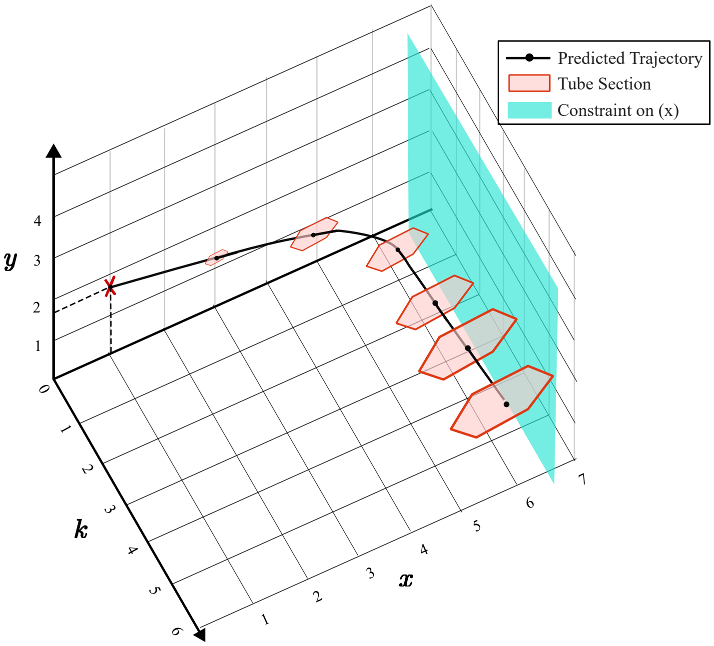

Figure C.4: Illustration schématique du concept de Tube-MPC. La figure montre une trajectoire nominale, le tube d'incertitude qui l'entoure, et une contrainte sur l'état. Le tube est contenu à l'intérieur de la région admissible.

Cela permet d'obtenir une meilleure performance lorsque l'estimation $\hat{\theta}_\tau$ est précise, tout en garantissant la robustesse grâce au tube basé sur Θ_τ. La faisabilité récursive est assurée par l'utilisation d'un ensemble terminal robuste positivement invariant.

La validation de cette approche a été réalisée par des simulations d'un système de téléopération bilatérale 1-DoF interagissant avec un environnement dont les paramètres (rigidité et amortissement) étaient incertains mais bornés. Une contrainte de position a été imposée au robot esclave pour simuler une fixture virtuelle de sécurité.

Les résultats de simulation ont démontré la capacité du contrôleur MPC robuste adaptatif à satisfaire la contrainte de position de manière robuste, c'est-à-dire que le robot esclave n'a jamais violé la limite imposée, malgré l'incertitude initiale sur les paramètres de l'environnement. Simultanément, l'apprentissage SML a montré son efficacité : au fur et à mesure des interactions, l'ensemble d'incertitude Θ_τ s'est resserré autour de la vraie valeur des paramètres, permettant au contrôleur d'être moins conservateur (par exemple, le robot esclave pouvait s'approcher plus près de la contrainte sans la violer, car le tube d'incertitude était plus petit). L'apprentissage LMS a également convergé vers la vraie valeur des paramètres, ce qui a permis d'améliorer la transparence du système lorsque celui-ci n'était pas limité par la contrainte. La Figure 5 illustre typiquement le respect de la contrainte et l'évolution des bornes d'incertitude apprises par SML.

Ces travaux montrent qu'il est possible de concevoir des contrôleurs MPC pour la téléopération qui sont à la fois robustes aux incertitudes, garantissant la sécurité, et adaptatifs, améliorant la performance et réduisant le conservatisme au fil du temps grâce à l'apprentissage en ligne.

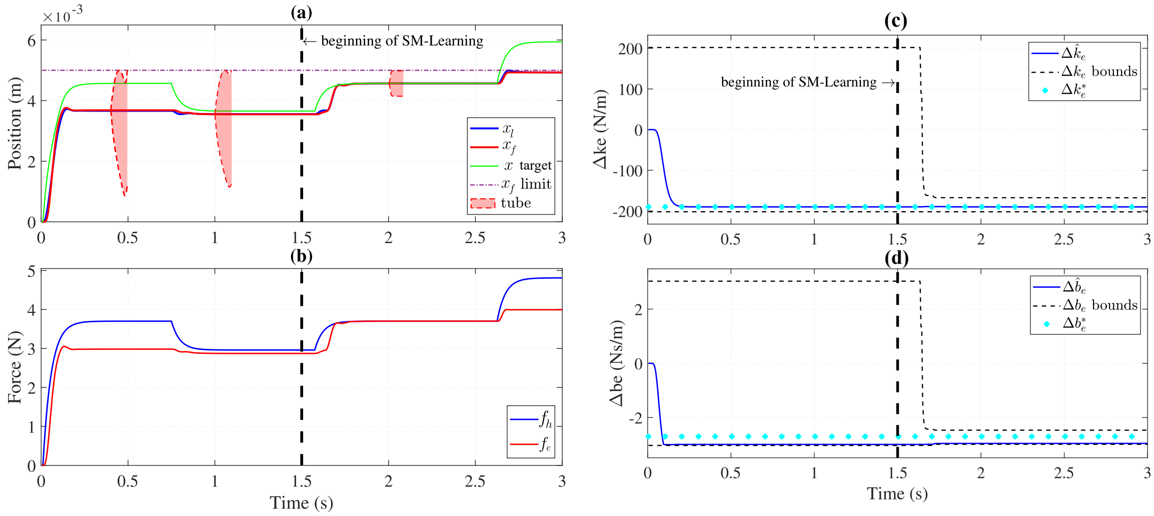

Figure C.5: Graphique de simulation illustrant : (a-b) Les signaux et Le respect d'une contrainte de position par le robot esclave. (c-d) L'évolution des bornes d'incertitude sur un paramètre de l'environnement apprises par SML.

Contribution 3 : Commande à Impédance Variable (VIC) Passive pour la Téléopération Bilatérale

La Commande à Impédance Variable (VIC) est une stratégie de contrôle attrayante car elle permet d'adapter dynamiquement le comportement interactif d'un robot, notamment sa rigidité et son amortissement apparents, en fonction de la tâche ou de l'environnement. En téléopération bilatérale, la VIC peut être utilisée pour que le robot esclave se comporte de manière souple lors de contacts inattendus ou pour protéger des environnements fragiles, et de manière plus rigide lorsqu'une grande précision ou l'application de forces importantes est requise. De même, l'impédance du robot maître peut être ajustée pour améliorer la perception de l'opérateur.

Cependant, une variation temporelle de l'impédance (en particulier une augmentation de la rigidité) peut injecter de l'énergie dans le système. Si cette injection d'énergie n'est pas correctement gérée, elle peut conduire à une violation de la condition de passivité du système de téléopération. La passivité est une propriété largement utilisée pour garantir la stabilité des systèmes téléopérés, surtout en présence d'un opérateur humain et d'un environnement dont les modèles sont incertains. La perte de passivité peut entraîner des oscillations instables et un comportement dangereux.

Cette troisième contribution propose un nouveau cadre de commande pour la téléopération bilatérale qui permet d'utiliser la VIC tout en garantissant rigoureusement la passivité du système, et donc sa stabilité. L'approche s'inspire et étend le concept des filtres de passivité, initialement proposés pour la robotique collaborative, au contexte spécifique de la téléopération bilatérale maître-esclave.

Le principe fondamental est de surveiller en continu le flux de puissance interne du système de téléopération. Ce flux peut être caractérisé par un terme de puissance dissipée, noté $\omega(t)$. Pour que le système soit passif, il est suffisant que $\omega(t)$ reste non négatif à tout instant. Lorsque les commandes VIC souhaitées par l'utilisateur (c'est-à-dire les profils d'impédance désirés pour le maître et l'esclave, et les consignes de position) risqueraient de rendre $\omega(t)$ négatif, le cadre de commande intervient de manière minimale.

Pour ce faire, plusieurs mécanismes de dissipation d'énergie sont intégrés :

1. **Filtrage du profil d'impédance désiré :** Les variations des matrices de rigidité (K_l, K_f) des robots maître et esclave sont filtrées (ralenties) si elles risquent d'injecter trop d'énergie.

2. **Filtrage des signaux de position de référence :** Les consignes de position pour le suivi entre le maître et l'esclave (x_l^r, x_f^r) sont également filtrées.

3. **Injection adaptative d'amortissement :** Un terme d'amortissement supplémentaire peut être injecté, principalement sur le robot maître.

Un problème d'optimisation linéaire est résolu en ligne à chaque pas de temps. Cet optimiseur détermine l'activation minimale de ces différents filtres et de l'injection d'amortissement nécessaire pour s'assurer que la condition $\omega(t) \geq 0$ est respectée, tout en essayant de suivre au mieux les commandes VIC désirées par l'utilisateur.

La Figure 6 illustre l'architecture de commande à impédance variable (VIC) passive proposée. Elle met en évidence les robots maître et esclave, leurs contrôleurs d'impédance respectifs, le générateur de profil d'impédance souhaité par l'utilisateur, et le module central d'optimisation de la passivité qui ajuste les filtres d'impédance et de référence de position pour garantir la stabilité du système.

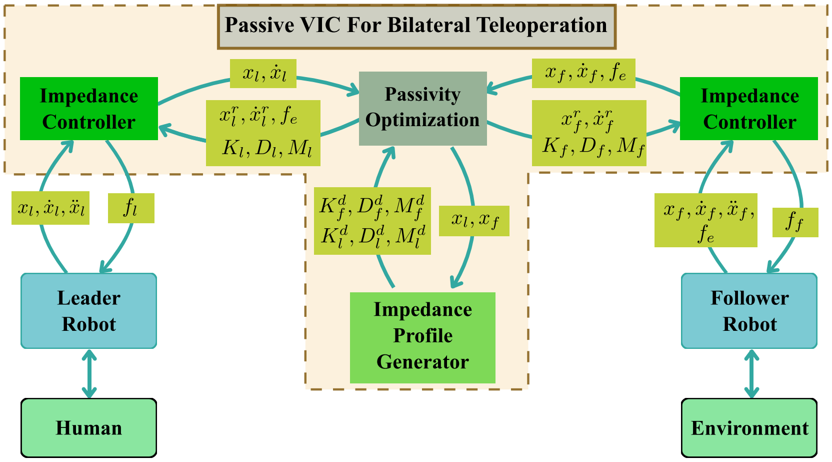

Figure C.6: Architecture de commande VIC passive. Le schéma de l'architecture proposée, montrant les robots, le générateur de profil d'impédance, les filtres et l'optimiseur.

La validation du cadre de commande à impédance variable (VIC) passive a été réalisée expérimentalement sur un banc d'essai de téléopération bilatérale composé de deux dispositifs haptiques Omega 3-DoF de Force Dimension (voire Figure 7). Le robot maître était un Omega7 et le robot esclave un Omega3, ce dernier étant équipé d'un capteur de force ATI Nano17 pour mesurer l'interaction avec l'environnement. L'environnement distant était constitué d'un bloc de mousse dans lequel un parcours (tunnel) avait été sculpté.

La tâche expérimentale consistait pour l'opérateur à guider le robot esclave pour interagir avec des zones cibles spécifiques (T1 et T2) situées aux extrémités du tunnel, puis à naviguer à travers le tunnel pour passer d'une cible à l'autre, et ce de manière répétée. Cette tâche a été conçue pour nécessiter des variations significatives de l'impédance du robot esclave : une rigidité élevée était souhaitée dans les zones cibles pour permettre l'application de forces précises, tandis qu'une faible rigidité était préférable dans le tunnel pour protéger ses parois contre des forces de contact excessives et pour faciliter un mouvement souple. L'impédance du robot maître était principalement configurée pour réfléchir les forces de l'environnement.

Les résultats expérimentaux ont confirmé la capacité du système à modifier dynamiquement les profils d'impédance du robot esclave en fonction de sa position dans l'environnement, conformément à la carte d'impédance prédéfinie. Plus important encore, l'optimiseur en ligne a réussi à maintenir la passivité du système tout au long de la tâche, comme en témoigne la puissance dissipée $\omega(t)$ qui est restée globalement non négative (aux erreurs numériques près). Lorsque les variations d'impédance souhaitées (par exemple, une augmentation rapide de la rigidité en entrant dans une zone cible) auraient pu violer la passivité, les filtres se sont activés (par exemple, le gain β_2 du filtre de rigidité de l'esclave a diminué), modérant temporairement les changements d'impédance pour préserver la stabilité. La Figure 7 ci-dessous illustre le banc d'essai et un exemple typique de l'évolution de la rigidité appliquée et de la puissance dissipée.

Ces expériences valident l'approche proposée comme une solution viable pour intégrer les avantages de la VIC en téléopération bilatérale tout en garantissant la stabilité grâce au maintien rigoureux de la passivité.

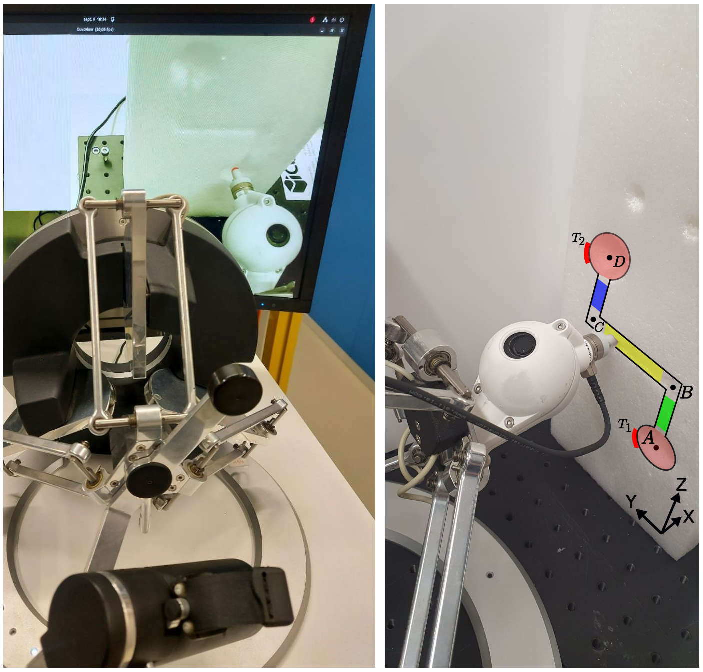

Figure C.7: Validation expérimentale de la VIC passive.

Conclusion Générale

Cette thèse a abordé plusieurs défis critiques liés à l'amélioration de la sécurité, de la transparence et de la robustesse des systèmes de téléopération bilatérale avec retour d'effort. L'objectif principal était de développer et de valider des stratégies de commande avancées permettant une manipulation à distance plus intuitive, plus sûre et plus efficace, en particulier face aux contraintes opérationnelles, aux incertitudes des modèles et à la nécessité d'un comportement d'interaction adaptable.

Trois contributions principales ont été présentées. Premièrement, un cadre d'**auto-réglage basé sur l'Optimisation Bayésienne** a été développé pour les contrôleurs MPC. Cette approche optimise de manière efficace les poids de la fonction de coût du MPC en minimisant directement une métrique de performance évaluée sur le système, améliorant significativement la transparence avec un effort de réglage minimal. Cette méthode a été validée en simulation et sur un banc d'essai 1-DoF.

Deuxièmement, pour répondre au problème crucial de la robustesse face aux incertitudes, un **contrôleur MPC robuste adaptatif (Tube-MPC)** a été conçu. Ce contrôleur garantit la satisfaction robuste des contraintes et la faisabilité récursive en présence d'incertitudes paramétriques bornées sur le modèle de l'environnement et de perturbations additives. Il intègre des mécanismes d'apprentissage en ligne (SML et LMS) qui affinent les bornes d'incertitude et estiment les paramètres de l'environnement, réduisant ainsi le conservatisme du contrôleur et améliorant ses performances au fil du temps. L'efficacité de cette approche a été validée par simulation.

Troisièmement, face au besoin d'adaptation dynamique du comportement d'interaction, un nouveau cadre de **commande à impédance variable (VIC) garantissant la passivité** a été proposé. La VIC offre des avantages significatifs en termes de flexibilité, mais peut compromettre la stabilité en violant la passivité. L'approche développée étend le concept des filtres de passivité et intègre, via une optimisation linéaire en ligne, de multiples stratégies de dissipation d'énergie (filtrage de profil d'impédance, filtrage de référence de position, injection d'amortissement). Cela permet de préserver la passivité, et donc la stabilité, tout en minimisant l'impact sur la performance. La faisabilité et l'efficacité de ce schéma VIC passif ont été validées expérimentalement sur un système 3-DoF.

En résumé, cette thèse propose des méthodes pratiques et théoriquement fondées pour faire progresser la commande en téléopération bilatérale. En développant des techniques pour le réglage efficace du MPC, la gestion robuste des contraintes sous incertitude, et la commande à impédance variable préservant la passivité, ces travaux ouvrent la voie à des systèmes de téléopération plus performants, fiables et sûrs, particulièrement pertinents pour des domaines exigeants tels que la chirurgie robotique.

Perspectives

Les recherches présentées dans cette thèse jettent les bases de systèmes de téléopération bilatérale plus sûrs et plus performants. Elles ouvrent également plusieurs pistes prometteuses pour des investigations futures visant à améliorer davantage la robustesse, l'adaptabilité, la performance et l'applicabilité pratique de ces systèmes.

1. Auto-réglage adaptatif pour des environnements multiples

Le cadre d'auto-réglage du MPC (Contribution 1) a été validé pour un environnement spécifique. Une extension naturelle serait de le rendre adaptatif aux variations de l'environnement. L'Optimisation Bayésienne Contextuelle (C-BO) pourrait être explorée pour modéliser la fonction de performance en fonction à la fois des poids du MPC et des paramètres contextuels de l'environnement (estimés en ligne). Cela permettrait au système de prédire et d'appliquer des réglages optimaux même pour des environnements non rencontrés lors de la phase d'apprentissage initiale.

2. Amélioration de l'apprentissage de l'incertitude (Contribution 2)

- **Gestion des paramètres d'environnement variant dans le temps :** La formulation actuelle suppose des paramètres d'environnement constants. Pour gérer des dérives lentes, un mécanisme d'inflation contrôlée de l'ensemble d'incertitude appris par SML pourrait être introduit, permettant de "rattraper" le vrai paramètre s'il sort de l'ensemble précédemment estimé.
- **Robustesse du SML aux mesures aberrantes (outliers) :** Le SML standard est sensible aux outliers. Des travaux futurs pourraient se concentrer sur l'amélioration de sa robustesse, par exemple en détectant et en rejetant les mesures qui contredisent fortement le modèle actuel d'incertitude.

3. Réduction du conservatisme et de la complexité de conception du MPC Robuste (Contribution 2)

La conception hors ligne du contrôleur de pré-stabilisation K, du coût terminal P, de l'ensemble terminal X_f et de la forme initiale du tube P_0 est complexe. L'application de la philosophie d'auto-réglage à ces éléments de conception eux-mêmes, en considérant l'équilibre entre la taille du tube, l'agressivité de K, et la complexité calculatoire, pourrait réduire le conservatisme et simplifier le déploiement.

4. Affinements de la Commande à Impédance Variable Passive (Contribution 3)

- **Contrôle de passivité spécifique à chaque direction cartésienne :** Actuellement, les gains des filtres de rigidité sont scalaires. Permettre des paramètres de filtrage indépendants pour chaque axe cartésien pourrait réduire le conservatisme en n'appliquant les interventions de passivité que le long des directions nécessaires.

- **Gestion de l'énergie et passivité intégrale :** Au lieu d'imposer $\omega(t) \geq 0$ (passivité instantanée), on pourrait n'exiger que l'intégrale de $\omega(t)$ soit positive, permettant de stocker temporairement l'énergie dissipée (comme dans les réservoirs d'énergie) pour des comportements non passifs ultérieurs, sous réserve d'une analyse de sécurité rigoureuse.

- **Atténuation du "chattering" :** L'injection d'amortissement (β_5) peut causer du "chattering". Des techniques comme le filtrage dynamique virtuel de la force d'amortissement ou un seuil de vitesse en deçà duquel l'amortissement est réduit pourraient être intégrées.

- **Auto-réglage des poids de l'optimisation de la passivité :** Les poids (p_j) qui gèrent l'équilibre entre les différents mécanismes de dissipation d'énergie sont actuellement réglés manuellement. Des méthodes d'apprentissage pourraient permettre de les ajuster automatiquement en fonction d'exigences de haut niveau ou des préférences de l'utilisateur.

Ces pistes de recherche visent à rendre les systèmes de téléopération encore plus intelligents, adaptatifs et transparents, élargissant ainsi leur champ d'application et leur acceptation dans des tâches complexes et critiques.

Publications Issues de la Thèse et Bibliographie Sélective

Publications du Doctorant Issues de la Thèse :

1. F. A. Almasalmah, H. Omran, C. Liu, T. Poignonec, and B. Bayle, "Auto-Tuning of Model Predictive Control for Bilateral Teleoperation with Bayesian Optimization," *IFAC-PapersOnLine*, vol. 58, pp. 85–90, Jan. 2024.
2. F. A. Almasalmah, H. Omran, C. Liu, and B. Bayle, "Adaptive Robust Model Predictive Control for Bilateral Teleoperation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7069–7074, Oct. 2023.
3. F. A. Almasalmah, T. Poignonec, H. Omran, C. Liu, and B. Bayle, "Passivity Filters for Bilateral Teleoperation with Variable Impedance Control," in *2025 IEEE International Conference on Robotics and Automation (ICRA)* — (Accepté).

Communications Scientifiques :

- Présentation à la conférence IROS, 2023 (Detroit, USA) pour l'article [2].
- Présentation à la conférence IFAC CPHS24, 2024 (Antalya, Turquie) pour l'article [1].
- Présentation prévue à la conférence ICRA, 2025 (Atlanta, USA) pour l'article [3].
- Présentation aux Journées CAMI (Computer-Assisted Medical Interventions), 2022 (Montpellier, France).
- Présentation aux Journées CAMI, 2023 (Strasbourg, France).

Bibliography

[1] K. Hashtrudi-Zaad, *Design, Implementation and Evaluation of Stable Bilateral Teleoperation Control Architectures for Enhanced Telepresence*. PhD thesis, University of British Columbia, 2000.

[2] J. Colan, A. Davila, and Y. Hasegawa, "Tactile Feedback in Robot-Assisted Minimally Invasive Surgery: A Systematic Review," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 20, no. 6, p. e70019, 2024.

[3] A. M. Okamura, "Haptic Feedback in Robot-Assisted Minimally Invasive Surgery," *Current opinion in urology*, vol. 19, pp. 102–107, Jan. 2009.

[4] L. Peternel and A. Ajoudani, "After a Decade of Teleimpedance: A Survey," *IEEE Transactions on Human-Machine Systems*, vol. 53, pp. 401–416, Apr. 2023.

[5] N. Piccinelli and R. Muradore, "A bilateral teleoperation with interaction force constraint in unknown environment using non linear model predictive control," *European Journal of Control*, vol. 62, pp. 185–191, Nov. 2021.

[6] R. V. Patel, S. F. Atashzar, and M. Tavakoli, "Haptic Feedback and Force-Based Teleoperation in Surgical Robotics," *Proceedings of the IEEE*, vol. 110, pp. 1012–1027, July 2022.

[7] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Berlin: Springer, 2008.

[8] R. C. Goertz, "FUNDAMENTALS OF GENERAL-PURPOSE REMOTE MANIPULATORS," *Nucleonics (U.S.) Ceased publication*, vol. Vol: 10, No. 11, Nov. 1952.

[9] R. C. Goertz, "MECHANICAL MASTER-SLAVE MANIPULATOR," *Nucleonics (U.S.) Ceased publication*, vol. Vol: 12, No. 11, Nov. 1954.

[10] R. C. Goertz and F. Bevilacqua, "A FORCE-REFLECTING POSITIONAL SERVOMECHANISM," *Nucleonics (U.S.) Ceased publication*, vol. Vol: 10, No. 11, Nov. 1952.

[11] R. C. Goertz and W. M. Thompson, "ELECTRONICALLY CONTROLLED MANIPULATOR," *Nucleonics (U.S.) Ceased publication*, vol. Vol: 12, No. 11, Nov. 1954.

[12] T. B. Sheridan, "Telerobotics," *Automatica*, vol. 25, pp. 487–507, July 1989.

[13] T. Sheridan and W. Ferrell, "Remote Manipulative Control with Transmission Delay," *IEEE Transactions on Human Factors in Electronics*, vol. HFE-4, pp. 25–29, Sept. 1963.

[14] T. Sheridan, "Space teleoperation through time delay: review and prognosis," *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 592–606, Oct. 1993.

[15] G. Raju, G. Verghese, and T. Sheridan, "Design issues in 2-port network models of bilateral remote manipulation," in *1989 International Conference on Robotics and Automation Proceedings*, pp. 1316–1321 vol.3, May 1989.

[16] R. J. Anderson and M. W. Spong, "Asymptotic Stability for Force Reflecting Teleoperators with Time Delay," *The International Journal of Robotics Research*, vol. 11, pp. 135–149, Apr. 1992.

[17] G. Niemeyer and J.-J. Slotine, "Stable adaptive teleoperation," *IEEE Journal of Oceanic Engineering*, vol. 16, pp. 152–162, Jan. 1991.

[18] J. Vertut and P. Coiffet, *Teleoperations and Robotics: Evolution and Development.* USA: Prentice-Hall, Inc., July 1986.

[19] A. K. Bejczy, "Toward advanced teleoperation in space," in *Teleoperation and Robotics in Space* (S. B. Skaar and C. F. Ruoff, eds.), vol. 161 of *Progress in Astronautics and Aeronautics*, pp. 107–138, Reston, VA: American Institute of Aeronautics and Astronautics, 1994.

[20] J. Marescaux, J. Leroy, F. Rubino, M. Smith, M. Vix, M. Simone, and D. Mutter, "Transcontinental robot-assisted remote telesurgery: Feasibility and potential applications," *Annals of Surgery*, vol. 235, pp. 487–492, Apr. 2002.

[21] J.-H. Ryu, D.-S. Kwon, and B. Hannaford, "Stable teleoperation with time-domain passivity control," *IEEE Transactions on Robotics and Automation*, vol. 20, pp. 365–373, Apr. 2004.

[22] D. Lee and K. Huang, "Passive-Set-Position-Modulation Framework for Interactive Robotic Systems," *IEEE Transactions on Robotics*, vol. 26, pp. 354–369, Apr. 2010.

[23] M. Franken, S. Stramigioli, R. Reilink, C. Secchi, and A. Macchelli, "Bridging the gap between passivity and transparency," in *Proceedings of the Robotics: Science and Systems V*, MIT Press, 2010.

[24] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. MacChelli, "Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 741–756, 2011.

[25] S. F. Atashzar, M. Shahbazi, M. Tavakoli, and R. V. Patel, "A Passivity-Based Approach for Stable Patient–Robot Interaction in Haptics-Enabled Rehabilitation Systems: Modulated Time-Domain Passivity Control," *IEEE Transactions on Control Systems Technology*, vol. 25, pp. 991–1006, May 2017.

[26] S. F. Atashzar, M. Shahbazi, M. Tavakoli, and R. Patel, "A grasp-based passivity signature for haptics-enabled human-robot interaction: Application to design of a new safety mechanism for robotic rehabilitation," *The International Journal of Robotics Research*, Apr. 2017.

[27] S. F. Atashzar, I. G. Polushin, and R. V. Patel, "A Small-Gain Approach for Nonpassive Bilateral Telerobotic Rehabilitation: Stability Analysis and Controller Synthesis," *IEEE Transactions on Robotics*, vol. 33, pp. 49–66, Feb. 2017.

[28] P. Mitra and G. Niemeyer, "Model-mediated Telemanipulation," *The International Journal of Robotics Research*, vol. 27, pp. 253–262, Feb. 2008.

[29] D. A. Abbink, M. Mulder, and E. R. Boer, "Haptic shared control: Smoothly shifting control authority?," *Cognition, Technology & Work*, vol. 14, pp. 19–28, Mar. 2012.

[30] B. Bayle and L. Barbé, “Tele-manipulation,” in *Medical Robotics*, pp. 269–302, John Wiley & Sons, Ltd, 2013.

[31] U.S. Food and Drug Administration, Center for Devices and Radiological Health (CDRH) Office of Surveillance and Biometrics (OSB) Medical Product Safety Network (MedSun), “Medsun small sample survey – final report topic: da Vinci surgical system,” tech. rep., U.S. Food and Drug Administration, November 2013.

[32] “Intuitive reaches 10 million procedures performed using da Vinci Surgical Systems | Intuitive Surgical.” https://isrg.intuitive.com/news-releases/news-release-details/intuitive-reaches-10-million-procedures-performed-using-da-vinci/.

[33] N. Enayati, E. De Momi, and G. Ferrigno, “Haptics in Robot-Assisted Surgery: Challenges and Benefits,” *IEEE reviews in biomedical engineering*, vol. 9, pp. 49–65, 2016.

[34] G. R. Sutherland, S. Wolfsberger, S. Lama, and K. Zarei-nia, “The evolution of neuroArm,” *Neurosurgery*, vol. 72 Suppl 1, pp. 27–32, Jan. 2013.

[35] P. Picozzi, U. Nocco, C. Labate, I. Gambini, G. Puleo, F. Silvi, A. Pezzillo, R. Mantione, and V. Cimolin, “Advances in Robotic Surgery: A Review of New Surgical Platforms,” *Electronics*, vol. 13, p. 4675, Jan. 2024.

[36] D. Stephan, H. Sälzer, and F. Willeke, “First Experiences with the New Senhance® Telerobotic System in Visceral Surgery,” *Visceral Medicine*, vol. 34, pp. 31–36, Feb. 2018.

[37] P. Biswas, S. Sikander, and P. Kulkarni, “Recent advances in robot-assisted surgical systems,” *Biomedical Engineering Advances*, vol. 6, p. 100109, Nov. 2023.

[38] A. Brodie and N. Vasdev, “The future of robotic surgery,” *Annals of The Royal College of Surgeons of England*, vol. 100, pp. 4–13, Sept. 2018.

[39] L. Baldari, L. Boni, and E. Cassinotti, “New Robotic Platforms,” in *Robotic Surgery of Colon and Rectum*, pp. 199–206, Springer, Cham, 2024.

[40] Y. Ueda, S. Miyahara, K. Tokuishi, H. Nakajima, R. Waseda, T. Shiraishi, and T. Sato, “Impact of a pneumatic surgical robot with haptic feedback function on surgical manipulation,” *Scientific Reports*, vol. 13, Dec. 2023.

[41] M. Covas Moschovas, S. Saikali, A. Gamal, S. Reddy, T. Rogers, M. Chiara Sighinolfi, B. Rocco, and V. Patel, "First Impressions of the New da Vinci 5 Robotic Platform and Experience in Performing Robot-assisted Radical Prostatectomy," *European Urology Open Science*, vol. 69, pp. 1–4, Sept. 2024.

[42] H. Xin, J. S. Zelek, and H. Carnahan, "Laparoscopic surgery, perceptual limitations and force: A review," in *First Canadian Student Conference on Biomedical Computing*, vol. 144, Queen's University Kingston, ON, Canada, 2006.

[43] A. Okamura, "Methods for haptic feedback in teleoperated robot-assisted surgery," *The Industrial robot*, vol. 31, pp. 499–508, Dec. 2004.

[44] P. Joice, G. B. Hanna, and A. Cuschieri, "Errors enacted during endoscopic surgery–a human reliability analysis," *Applied Ergonomics*, vol. 29, pp. 409–414, Dec. 1998.

[45] B. T. Bethea, A. M. Okamura, M. Kitagawa, T. P. Fitton, S. M. Cattaneo, V. L. Gott, W. A. Baumgartner, and D. D. Yuh, "Application of haptic feedback to robotic surgery," *Journal of Laparoendoscopic & Advanced Surgical Techniques. Part A*, vol. 14, pp. 191–195, June 2004.

[46] B. Demi, T. Ortmaier, and U. Seibold, "The touch and feel in minimally invasive surgery," in *IEEE International Workshop on Haptic Audio Visual Environments and Their Applications*, pp. 6 pp.–, Oct. 2005.

[47] T. Ortmaier, B. Deml, B. Kübler, G. Passig, D. Reintsema, and U. Seibold, "Robot Assisted Force Feedback Surgery," in *Advances in Telerobotics*, pp. 361–379, Springer, Berlin, Heidelberg, 2007.

[48] O. Gerovichev, P. Marayong, and A. M. Okamura, "The Effect of Visual and Haptic Feedback on Manual and Teleoperated Needle Insertion," in *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2002*, pp. 147–154, Springer, Berlin, Heidelberg, 2002.

[49] M. Bergholz, M. Ferle, and B. M. Weber, "The benefits of haptic feedback in robot assisted surgery and their moderators: a meta-analysis," *Scientific Reports*, vol. 13, p. 19215, Nov. 2023.

[50] Y. Ueda, S. Miyahara, K. Tokuishi, H. Nakajima, R. Waseda, T. Shiraishi, and T. Sato, “First clinical application of a surgical robot with haptic force feedback function for thoracic surgery: a case report,” *Shanghai Chest*, vol. 8, July 2024.

[51] K. Iwatani, F. Urabe, S. Saito, S. Kawano, T. Yamasaki, S. Kimura, H. Otsuki, K. Fujio, T. Kimura, and J. Miki, “Initial experience of a novel surgical assist robot “Saroa” featuring tactile feedback and a roll-clutch system in radical prostatectomy,” *Scientific Reports*, vol. 14, p. 31727, Dec. 2024.

[52] E. L. Servais, L. Rashidi, P. Porwal, M. Garibaldi, and A. J. Hung, “Novel force feedback technology improves suturing in robotic-assisted surgery: a pre-clinical study,” *Surgical Endoscopy*, vol. 39, pp. 1217–1226, Feb. 2025.

[53] A. Gamal, M. C. Moschovas, S. Saikali, S. Reddy, Y. Ozawa, R. Sharma, A. Kunta, T. Rogers, and V. Patel, “Comparing the Technological and Intraoperative Performances of Da Vinci xi and DaVinci 5 Robotic Platforms in Patients Undergoing Robotic-Assisted Radical Prostatectomy,” *International braz j urol*, vol. 51, p. e20240569, Jan. 2025.

[54] D. Lawrence, “Stability and transparency in bilateral teleoperation,” *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 624–637, Oct. 1993.

[55] S. Tafazoli, S. Salcudean, K. Hashtrudi-Zaad, and P. Lawrence, “Impedance control of a teleoperated excavator,” *Control Systems Technology, IEEE Transactions on*, vol. 10, pp. 355–367, June 2002.

[56] R. Daniel and P. McAree, “Fundamental Limits of Performance for Force Reflecting Teleoperation,” *The International Journal of Robotics Research*, vol. 17, pp. 811–830, Aug. 1998.

[57] A. Alfi, A. Khosravi, and A. Lari, “Swarm-based structure-specified controller design for bilateral transparent teleoperation systems via mu synthesis,” *IMA Journal of Mathematical Control and Information*, vol. 31, pp. 111–136, Mar. 2014.

[58] R. Lin and T. Namerikawa, “Robust control of master-slave robot system considering environmental uncertainties,” in *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, pp. 1299–1304, July 2005.

[59] C. A. Lopez Martinez, I. Polat, R. Van De Molengraft, and M. Steinbuch, "Robust High Performance Bilateral Teleoperation Under Bounded Time-Varying Dynamics," *IEEE Transactions on Control Systems Technology*, vol. 23, pp. 206–218, Jan. 2015.

[60] E. Nuño, L. Basañez, and R. Ortega, "Passivity-based control for bilateral teleoperation: A tutorial," *Automatica*, vol. 47, pp. 485–495, Mar. 2011.

[61] N. Hogan, "Controlling impedance at the man/machine interface," in *1989 International Conference on Robotics and Automation Proceedings*, pp. 1626–1631 vol.3, May 1989.

[62] S. F. Atashzar, I. G. Polushin, and R. V. Patel, "Networked teleoperation with non-passive environment: Application to tele-rehabilitation," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5125–5130, Oct. 2012.

[63] I. G. Polushin, S. N. Dashkovskiy, A. Takhmar, and R. V. Patel, "A small gain framework for networked cooperative force-reflecting teleoperation," *Automatica*, vol. 49, pp. 338–348, Feb. 2013.

[64] Y.-C. Lin, L.-H. Yuan, C.-S. Tseng, T.-Y. Hsieh, Y.-W. Huang, C.-Y. Huang, and S.-W. Huang, "Comparison of senhance and da vinci robotic radical prostatectomy: short-term outcomes, learning curve, and cost analysis," *Prostate Cancer and Prostatic Diseases*, vol. 27, pp. 116–121, Mar. 2024.

[65] P. Schleer, P. Kaiser, S. Drobinsky, and K. Radermacher, "Augmentation of haptic feedback for teleoperated robotic surgery," *International Journal of Computer Assisted Radiology and Surgery*, vol. 15, no. 3, pp. 515–529, 2020.

[66] M. Cavusoglu, A. Sherman, and F. Tendick, "Design of bilateral teleoperation controllers for haptic exploration and telemanipulation of soft environments," *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 641–647, Aug. 2002.

[67] Y. Yokokohji and T. Yoshikawa, "Bilateral control of master-slave manipulators for ideal kinesthetic coupling-formulation and experiment," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 605–620, Oct. 1994.

[68] J. Sheng and M. Spong, “Model predictive control for bilateral teleoperation systems with time delays,” in *Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No.04CH37513)*, vol. 4, pp. 1877–1880 Vol.4, May 2004.

[69] B. Hannaford, “Stability and performance tradeoffs in bi-lateral telemanipulation,” in *1989 International Conference on Robotics and Automation Proceedings*, pp. 1764–1767 vol.3, May 1989.

[70] P. F. Hokayem and M. W. Spong, “Bilateral teleoperation: An historical survey,” *Automatica*, vol. 42, pp. 2035–2057, Dec. 2006.

[71] P. Arcara and C. Melchiorri, “Control schemes for teleoperation with time delay: A comparative study,” *Robotics and Autonomous Systems*, vol. 38, pp. 49–64, Jan. 2002.

[72] S. Proia, R. Carli, G. Cavone, and M. Dotoli, “Control Techniques for Safe, Ergonomic, and Efficient Human-Robot Collaboration in the Digital Industry: A Survey,” *IEEE Transactions on Automation Science and Engineering*, vol. 19, pp. 1798–1819, July 2022.

[73] S. Haddadin, *Towards Safe Robots: Approaching Asimov’s 1st Law*, vol. 90 of *Springer Tracts in Advanced Robotics.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[74] F. Benzi, F. Ferraguti, and C. Secchi, “Energy Tank-based Control Framework for Satisfying the ISO/TS 15066 Constraint,” *IFAC-PapersOnLine*, vol. 56, pp. 1288–1293, Jan. 2023.

[75] “Medical electrical equipment — Part 2-77: Particular requirements for the basic safety and essential performance of robotically assisted surgical equipment, IEC 80601-2-77:2019,” 2019.

[76] F. Rydén and H. J. Chizeck, “Forbidden-region virtual fixtures from streaming point clouds: Remotely touching and protecting a beating heart,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3308–3313, Oct. 2012.

[77] F. J. Abu-Dakka and M. Saveriano, "Variable Impedance Control and Learning—A Review," *Frontiers in Robotics and AI*, vol. 7, Dec. 2020.

[78] R. Li, M. Cheng, and R. Ding, "Passivity-based bilateral shared variable impedance control for teleoperation compliant assembly," *Mechatronics*, vol. 95, p. 103057, Nov. 2023.

[79] Y. Michel, R. Rahal, C. Pacchierotti, P. R. Giordano, and D. Lee, "Bilateral Teleoperation With Adaptive Impedance Control for Contact Tasks," *IEEE Robotics and Automation Letters*, vol. 6, pp. 5429–5436, July 2021.

[80] K. Kronander and A. Billard, "Stability Considerations for Variable Impedance Control," *IEEE Transactions on Robotics*, vol. 32, pp. 1298–1305, Oct. 2016.

[81] F. A. Almasalmah, H. Omran, C. Liu, T. Poignonec, and B. Bayle, "Auto-Tuning of Model Predictive Control for Bilateral Teleoperation with Bayesian Optimization∗," *IFAC-PapersOnLine*, vol. 58, pp. 85–90, Jan. 2024.

[82] F. A. Almasalmah, H. Omran, C. Liu, and B. Bayle, "Adaptive Robust Model Predictive Control for Bilateral Teleoperation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7069–7074, Oct. 2023.

[83] F. A. Almasalmah, T. Poignonec, H. Omran, C. Liu, and B. Bayle, "Passivity Filters for Bilateral Teleoperation with Variable Impedance Control," in *2025 IEEE International Conference on Robotics and Automation (ICRA) - (Accepted)*.

[84] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for linear and Hybrid systems.* Cambridge University Press, 2017.

[85] A. Bemporad, D. Bernardini, R. Long, and J. Verdejo, "Model Predictive Control of Turbocharged Gasoline Engines for Mass Production," SAE Technical Paper 2018-01-0875, SAE International, Warrendale, PA, Apr. 2018.

[86] E. van Oort, Q. Chu, and J. Mulder, "Robust Model Predictive Control of a Feedback Linearized F-16/MATV Aircraft Model," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Guidance, Navigation, and Control and Co-located Conferences, American Institute of Aeronautics and Astronautics, Aug. 2006.

[87] A. K. Sampathirao, P. Sopasakis, A. Bemporad, and P. Patrinos, “GPU-Accelerated Stochastic Predictive Control of Drinking Water Networks,” *IEEE Transactions on Control Systems Technology*, vol. 26, pp. 551–562, Mar. 2017.

[88] T. Samad, M. Bauer, S. Bortoff, S. Di Cairano, L. Fagiano, P. F. Odgaard, R. R. Rhinehart, R. Sánchez-Peña, A. Serbezov, F. Ankersen, P. Goupil, B. Grosman, M. Heertjes, I. Mareels, and R. Sosseh, “Industry engagement with control research: Perspective and messages,” *Annual Reviews in Control*, vol. 49, pp. 1–14, 2020.

[89] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control Engineering Practice*, vol. 11, pp. 733–764, July 2003.

[90] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, “Review on model predictive control: an engineering perspective,” *The International Journal of Advanced Manufacturing Technology*, vol. 117, pp. 1327–1349, Nov. 2021.

[91] M. D. Killpack, A. Kapusta, and C. C. Kemp, “Model predictive control for fast reaching in clutter,” *Autonomous Robots*, vol. 40, pp. 537–560, Mar. 2016.

[92] K. Haninger, C. Hegeler, and L. Peternel, “Model Predictive Impedance Control with Gaussian Processes for Human and Environment Interaction,” Dec. 2022.

[93] T. Gold, A. Völz, and K. Graichen, “Model Predictive Interaction Control for Robotic Manipulation Tasks,” *IEEE Transactions on Robotics*, vol. 39, pp. 76–89, Feb. 2023.

[94] M. Bednarczyk, H. Omran, and B. Bayle, “Model Predictive Impedance Control,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4702–4708, May 2020.

[95] J. Matschek, J. Bethge, and R. Findeisen, “Safe Machine-Learning-Supported Model Predictive Force and Motion Control in Robotics,” *IEEE Transactions on Control Systems Technology*, vol. 31, pp. 2380–2392, Nov. 2023.

[96] M. Bowthorpe, M. Tavakoli, H. Becher, and R. Howe, “Smith predictor based control in teleoperated image-guided beating-heart surgery,” in *2013 IEEE International Conference on Robotics and Automation*, pp. 5825–5830, May 2013.

[97] S. Ganjefar, H. Momeni, F. Janabi Sharifi, and M. Hamidi Beheshti, "Behavior of Smith predictor in teleoperation systems with modeling and delay time errors," in *Proceedings of 2003 IEEE Conference on Control Applications, 2003. CCA 2003.*, vol. 2, pp. 1176–1180 vol.2, June 2003.

[98] F. Buzan and T. Sheridan, "A model-based predictive operator aid for telemanipulators with time delay," in *Conference Proceedings., IEEE International Conference on Systems, Man and Cybernetics*, pp. 138–143 vol.1, Nov. 1989.

[99] R. Uddin, S. Park, S. Baek, and J. Ryu, "Model predictive energy-bounding approach for the perception of multiple degree-of-freedom objects in bilateral teleoperation with online geometry and parameter of remote environment: A feasibility test," in *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, pp. 1039–1041, Oct. 2013.

[100] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control—Part I. The basic algorithm," *Automatica*, vol. 23, pp. 137–148, Mar. 1987.

[101] D. Chen, N. Xi, Y. Wang, H. Li, and X. Tang, "Event-based predictive control strategy for teleoperation via Internet," in *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 359–364, July 2008.

[102] T. Slama, D. Aubry, R. Oboe, and F. Kratz, "Robust bilateral generalized predictive control for teleoperation systems," in *2007 Mediterranean Conference on Control & Automation*, pp. 1–6, June 2007.

[103] R. Ginhoux, J. Gangloff, M. de Mathelin, L. Soler, M. Sanchez, and J. Marescaux, "Active filtering of physiological motion in robotized surgery using predictive control," *IEEE Transactions on Robotics*, vol. 21, pp. 67–79, Feb. 2005.

[104] R. Uddin and J. Ryu, "Predictive control approaches for bilateral teleoperation," *Annual Reviews in Control*, vol. 42, pp. 82–99, 2016.

[105] M. Joinié-Maurin, B. Bayle, and J. Gangloff, "Force feedback teleoperation with periodical disturbance compensation," in *2011 IEEE International Conference on Robotics and Automation*, pp. 4828–4833, May 2011.

[106] T. Slama, A. Trevisani, D. Aubry, R. Oboe, and F. Kratz, "Experimental Analysis of an Internet-Based Bilateral Teleoperation System With Motion and Force

Scaling Using a Model Predictive Controller," *IEEE Transactions on Industrial Electronics*, vol. 55, pp. 3290–3299, Sept. 2008.

[107] M. Dominici, P. Poignet, and E. Dombre, "Compensation of physiological motion using linear predictive force control," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1173–1178, Sept. 2008.

[108] D. Chen, X. Tang, N. Xi, Y. Wang, and H. Li, "Stability analysis for Internet based teleoperated robot using prediction control," in *2011 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems*, pp. 138–143, Mar. 2011.

[109] M. Dominici and R. Cortesão, "Model predictive control architectures with force feedback for robotic-assisted beating heart surgery," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2276–2282, May 2014.

[110] S. V. Ghoushkhanehee and A. Alfi, "Model Predictive Control of transparent bilateral teleoperation systems under uncertain communication time-delay," in *2013 9th Asian Control Conference (ASCC)*, pp. 1–6, June 2013.

[111] Y. Yang, F. Yang, J. Hua, and H. Li, "Generalized predictive control for space teleoperation systems with long time-varying delays," in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3057–3062, Oct. 2012.

[112] N. Piccinelli and R. Muradore, "A passivity-based bilateral teleoperation architecture using distributed nonlinear model predictive control," *IEEE International Conference on Intelligent Robots and Systems*, pp. 11466–11472, 2020.

[113] A. Bemporad, "Predictive control of teleoperated constrained systems with unbounded communication delays," in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, vol. 2, pp. 2133–2138 vol.2, Dec. 1998.

[114] Z. Ghazi, A. A. Safavi, and M. Salimifard, "Improving MPC based teleoperation systems by employing intelligent optimization algorithms," in *Instrumentation and Automation The 2nd International Conference on Control*, pp. 670–675, Dec. 2011.

[115] Y. Hatori, H. Nagakura, and Y. Uchimura, “Teleoperation with variable and large time delay based on MPC and model error compensator,” in *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, (Kyoto, Japan), pp. 1–6, IEEE, June 2021.

[116] N. Piccinelli and R. Muradore, “Passivity-Based Teleoperation With Interaction Force Constraints Using Hybrid Linear Model Predictive Control,” in *2022 30th Mediterranean Conference on Control and Automation (MED)*, pp. 25–30, June 2022.

[117] S. Hu, E. Babaians, M. Karimi, and E. Steinbach, “NMPC-MP: Real-time Nonlinear Model Predictive Control for Safe Motion Planning in Manipulator Teleoperation,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8309–8316, Sept. 2021.

[118] M. Risiglione, J.-P. Sleiman, M. V. Minniti, B. Çizmeci, D. Dresscher, and M. Hutter, “Passivity-based control for haptic teleoperation of a legged manipulator in presence of time-delays,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5276–5281, Sept. 2021.

[119] J. Cheng, F. Abi-Farraj, F. Farshidian, and M. Hutter, “Haptic Teleoperation of High-dimensional Robotic Systems Using a Feedback MPC Framework,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6197–6204, Oct. 2022.

[120] C. Passenberg, A. Peer, and M. Buss, “A survey of environment-, operator-, and task-adapted controllers for teleoperation systems,” *Mechatronics*, vol. 20, pp. 787–801, Oct. 2010.

[121] M. Motaharifar, K. Hashtrudi-Zaad, S. F. Mohammadi, A. Lashay, and H. D. Taghirad, “A self-tuning dual impedance control architecture for collaborative haptic training,” *Mechatronics*, vol. 101, p. 103205, Aug. 2024.

[122] M. A. Soleimani, I. Sharifi, H. A. Talebi, A. A. Suratgar, and F. Koochaki, “Forbidden region virtual fixture for nonlinear teleoperation systems with variable time delays,” in *2014 Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, pp. 095–100, Oct. 2014.

[123] M. C. Priess, R. Conway, J. Choi, J. M. Popovich, and C. Radcliffe, "Solutions to the Inverse LQR Problem With Application to Biological Systems Analysis," *IEEE Transactions on Control Systems Technology*, vol. 23, pp. 770–777, Mar. 2015.

[124] S. Di Cairano and A. Bemporad, "Model Predictive Control Tuning by Controller Matching," *IEEE Transactions on Automatic Control*, vol. 55, pp. 185–190, Jan. 2010.

[125] R. Kress and J. Jansen, "Automatic tuning for a teleoperated arm controller," in *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*, (Tucson, AZ, USA), pp. 2692–2695, IEEE, 1992.

[126] H. Shokri-Ghaleh and A. Alfi, "A comparison between optimization algorithms applied to synchronization of bilateral teleoperation systems against time delay and modeling uncertainties," *Applied Soft Computing*, vol. 24, pp. 447–456, Nov. 2014.

[127] B.-Y. Kim and H.-S. Ahn, "Bilateral teleoperation systems using genetic algorithms," in *2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation - (CIRA)*, pp. 388–393, Dec. 2009.

[128] A. Said, A. Khalil, R. Petra, S. Yunus, A. S. Peng, and S. Khan, "PID Controller Optimization of Teleoperated 2DOF Robot Manipulator Using Artificial Bee Colony Algorithm," in *2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, pp. 1–6, Dec. 2019.

[129] H. A. Talavatifard, K. Razi, and M. B. Menhaj, "A Self-Tuning Controller for Teleoperation System using Evolutionary Learning Algorithms in Neural Networks," in *Computational Intelligence, Theory and Applications* (B. Reusch, ed.), (Berlin, Heidelberg), pp. 51–60, Springer, 2006.

[130] L. Barbé, B. Bayle, and M. de Mathelin, "Towards The Autotuning of Force-Feedback Teleoperators," *IFAC Proceedings Volumes*, vol. 39, pp. 482–487, Jan. 2006.

[131] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the Human Out of the Loop: A Review of Bayesian Optimization," *Proceedings of the IEEE*, vol. 104, pp. 148–175, Jan. 2016.

[132] A. Marco, P. Hennig, J. Bohg, S. Schaal, and S. Trimpe, “Automatic LQR tuning based on Gaussian process global optimization,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 270–277, May 2016.

[133] P. Holzmann, M. Pfefferkorn, J. Peters, and R. Findeisen, “Learning Energy-Efficient Trajectory Planning for Robotic Manipulators Using Bayesian Optimization,” in *2024 European Control Conference (ECC)*, (Stockholm, Sweden), pp. 1374–1379, IEEE, June 2024.

[134] D. Dries, P. Englert, and M. Toussaint, “Constrained Bayesian optimization of combined interaction force/task space controllers for manipulations,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, (Singapore, Singapore), pp. 902–907, IEEE, May 2017.

[135] F. Zahedi, D. Chang, and H. Lee, “User-Adaptive Variable Damping Control Using Bayesian Optimization to Enhance Physical Human-Robot Interaction,” *IEEE Robotics and Automation Letters*, vol. 7, pp. 2724–2731, Apr. 2022.

[136] R. Garnett, *Bayesian Optimization*. Cambridge University Press, 1 ed., Jan. 2023.

[137] L. P. Fröhlich, C. Küttel, E. Arcari, L. Hewing, M. N. Zeilinger, and A. Carron, “Contextual Tuning of Model Predictive Control for Autonomous Racing,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10555–10562, Oct. 2022.

[138] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Adaptive computation and machine learning, Cambridge, Mass: MIT Press, 2006.

[139] B. Catkin and V. Patoglu, “Preference-Based Human-in-the-Loop Optimization for Perceived Realism of Haptic Rendering,” *IEEE Transactions on Haptics*, vol. 16, no. 4, pp. 470 – 476, 2023.

[140] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, “acados – a modular open-source framework for fast embedded optimal control,” *Mathematical Programming Computation*, Oct 2021.

[141] L. P. Fröhlich and A. Carron, “Bayesopt4ros: A Bayesian optimization package for the robot operating system.” `https://github.com/IntelligentControlSystems/bayesopt4ros`, 2021.

[142] L. G. García-Valdovinos, V. Parra-Vega, and M. A. Arteaga, "Observer-based sliding mode impedance control of bilateral teleoperation under constant unknown time delay," *Robotics and Autonomous Systems*, vol. 55, pp. 609–617, Aug. 2007.

[143] N. Chopra, M. Spong, S. Hirche, and M. Buss, "Bilateral teleoperation over the internet: the time varying delay problem," in *Proceedings of the 2003 American Control Conference, 2003.*, vol. 1, pp. 155–160, June 2003.

[144] Y.-H. Chang, C.-Y. Yang, and H.-W. Lin, "Robust Adaptive-Sliding-Mode Control for Teleoperation Systems with Time-Varying Delays and Uncertainties," *Robotics*, vol. 13, p. 89, June 2024.

[145] J. Yan and S. Salcudean, "Teleoperation controller design using H_∞-optimization with application to motion-scaling," *IEEE Transactions on Control Systems Technology*, vol. 4, pp. 244–258, May 1996.

[146] K. Kim, M. C. Cavusoglu, and W. K. Chung, "Quantitative Comparison of Bilateral Teleoperation Systems Using mu-Synthesis," *IEEE Transactions on Robotics*, vol. 23, pp. 776–789, Aug. 2007.

[147] G. Leung, B. Francis, and J. Apkarian, "Bilateral controller for teleoperators with time delay via /spl mu/-synthesis," *IEEE Transactions on Robotics and Automation*, vol. 11, pp. 105–116, Feb. 1995.

[148] A. César, L. Martínez, R. van de Molengraft, and M. Steinbuch, "Switching Robust Control for Transparent and Stable Teleoperation," *IFAC Proceedings Volumes*, vol. 46, pp. 530–536, Jan. 2013.

[149] C. A. L. Martinez, R. van de Molengraft, and M. Steinbuch, "Robust performance control design for bilateral teleoperation under time-varying bounded operator and environment dynamics," in *2012 American Control Conference (ACC)*, pp. 3465–3470, June 2012.

[150] J. Colgate, "Robust impedance shaping telemanipulation," *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 374–384, Aug. 1993.

[151] S. Khan, A. Sabanovic, and A. Nergiz, "Scaled Bilateral Teleoperation Using Discrete-Time Sliding-Mode Controller," *IEEE Transactions on Industrial Electronics*, vol. 56, pp. 3609–3618, Sept. 2009.

[152] A. Hace, M. Franc, and K. Jezernik, “Sliding mode control for scaled bilateral teleoperation,” *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, pp. 3430–3435, Nov. 2011.

[153] F. Huang, W. Zhang, Z. Chen, J. Tang, W. Song, and S. Zhu, “RBFNN-Based Adaptive Sliding Mode Control Design for Nonlinear Bilateral Teleoperation System Under Time-Varying Delays,” *IEEE Access*, vol. 7, pp. 11905–11912, 2019.

[154] A. Casavola, E. Mosca, and M. Papini, “Predictive teleoperation of constrained dynamic systems via Internet-like channels,” *IEEE Transactions on Control Systems Technology*, vol. 14, pp. 681–694, July 2006.

[155] Y. Yang, C. Hua, and X. Guan, “Finite Time Control Design for Bilateral Teleoperation System With Position Synchronization Error Constrained,” *IEEE Transactions on Cybernetics*, vol. 46, pp. 609–619, Mar. 2016.

[156] Y. Yang, H. Li, Y. Chen, and J. Yi, “Model Predictive Control for Space Teleoperation Systems Based on a Mixed-H2/H-inf Approach,” *Journal of Aerospace Engineering*, vol. 28, p. 04014133, July 2014.

[157] I. Kucukdemiral, H. Yazici, B. Gormus, and G. P. Bevan, “Data-driven H-infinity control of constrained systems: An application to bilateral teleoperation system,” *ISA Transactions*, vol. 137, pp. 23–34, June 2023.

[158] A. Bemporad and M. Morari, “Robust model predictive control: A survey,” in *Robustness in identification and control* (A. Garulli and A. Tesi, eds.), vol. 245, (London), pp. 207–226, Springer London, 1999.

[159] X. Ping, J. Hu, T. Lin, B. Ding, P. Wang, and Z. Li, “A Survey of Output Feedback Robust MPC for Linear Parameter Varying Systems,” *IEEE/CAA Journal of Automatica Sinica*, vol. 9, pp. 1717–1751, Oct. 2022.

[160] M. B. Saltık, L. Özkan, J. H. A. Ludlage, S. Weiland, and P. M. J. Van den Hof, “An outlook on robust model predictive control algorithms: Reflections on performance and computational aspects,” *Journal of Process Control*, vol. 61, pp. 77–102, Jan. 2018.

[161] J. Sieber, S. Bennani, and M. N. Zeilinger, “A System Level Approach to Tube-Based Model Predictive Control,” *IEEE Control Systems Letters*, vol. 6, pp. 776–781, 2022.

[162] D. Q. Mayne, M. M. Seron, and S. V. Raković, “Robust model predictive control of constrained linear systems with bounded disturbances,” *Automatica*, vol. 41, pp. 219–224, Feb. 2005.

[163] S. V. Raković, B. Kouvaritakis, R. Findeisen, and M. Cannon, “Homothetic tube model predictive control,” *Automatica*, vol. 48, pp. 1631–1638, Aug. 2012.

[164] S. V. Raković, W. S. Levine, and B. Açikmese, “Elastic tube model predictive control,” in *2016 American Control Conference (ACC)*, pp. 3594–3599, July 2016.

[165] M. E. Villanueva, M. A. Müller, and B. Houska, “Configuration-Constrained Tube MPC,” Aug. 2022.

[166] S. V. Raković, B. Kouvaritakis, M. Cannon, C. Panos, and R. Findeisen, “Fully Parameterized Tube MPC,” *IFAC Proceedings Volumes*, vol. 44, pp. 197–202, Jan. 2011.

[167] D. Saccani, G. Ferrari-Trecate, M. N. Zeilinger, and J. Köhler, “Homothetic Tube Model Predictive Control With Multi-Step Predictors,” *IEEE Control Systems Letters*, vol. 7, pp. 3561–3566, 2023.

[168] J. Köhler, E. Andina, R. Soloperto, M. A. Müller, and F. Allgöwer, “Linear robust adaptive model predictive control: Computational complexity and conservatism – extended version,” Mar. 2020.

[169] B. Pluymers, J. Rossiter, J. Suykens, and B. De Moor, “The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty,” in *Proceedings of the 2005, American Control Conference, 2005.*, pp. 804–809 vol. 2, June 2005.

[170] K. Kouramas, S. Rakovic, E. Kerrigan, J. Allwright, and D. Mayne, “On the Minimal Robust Positively Invariant Set for Linear Difference Inclusions,” in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 2296–2301, Dec. 2005.

[171] M. Tanaskovic, L. Fagiano, R. Smith, and M. Morari, “Adaptive receding horizon control for constrained MIMO systems,” *Automatica*, vol. 50, pp. 3019–3029, Dec. 2014.

[172] M. Lorenzen, M. Cannon, and F. Allgöwer, "Robust MPC with recursive model update," *Automatica*, vol. 103, pp. 461–471, May 2019.

[173] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference*, (Zürich, Switzerland), pp. 502–510, July 17–19 2013. `http://control.ee.ethz.ch/~mpt`.

[174] J. Lofberg, "YALMIP : a toolbox for modeling and optimization in MATLAB," in *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, pp. 284–289, Sept. 2004.

[175] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 10.1.*, 2024.

[176] R. Lundström and L. Burström, "Mechanical impedance of the human hand-arm system," *International Journal of Industrial Ergonomics*, vol. 3, pp. 235–242, Apr. 1989.

[177] J. Li, G. Li, Z. Chen, and J. Li, "A Novel EMG-Based Variable Impedance Control Method for a Tele-Operation System Under an Unstructured Environment," *IEEE Access*, vol. 10, pp. 89509–89518, 2022.

[178] D. S. Walker, R. P. Wilson, and G. Niemeyer, "User-controlled variable impedance teleoperation," in *2010 IEEE International Conference on Robotics and Automation*, pp. 5352–5357, May 2010.

[179] Z. Wang, X. Xu, D. Yang, B. Güleçyüz, F. Meng, and E. Steinbach, "Teleoperation with Haptic Sensor-aided Variable Impedance Control Based on Environment and Human Stiffness Estimation," *IEEE Sensors Journal*, pp. 1–1, 2024.

[180] M. Howard, D. J. Braun, and S. Vijayakumar, "Transferring Human Impedance Behavior to Heterogeneous Variable Impedance Actuators," *IEEE Transactions on Robotics*, vol. 29, pp. 847–862, Aug. 2013.

[181] G. Gourmelen, B. Navarro, A. Cherubini, and G. Ganesh, "Human guided trajectory and impedance adaptation for tele-operated physical assistance," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9276–9282, Sept. 2021.

[182] L. Peternel, T. Petrič, and J. Babič, “Human-in-the-loop approach for teaching robot assembly tasks using impedance control interface,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1497–1502, May 2015.

[183] J. Corredor, J. Sofrony, and A. Peer, “Decision-Making Model for Adaptive Impedance Control of Teleoperation Systems,” *IEEE Transactions on Haptics*, vol. 10, pp. 5–16, Jan. 2017.

[184] Costas S. Tzafestas, Spyros V. Velanas, and G. Fakiridis, “Adaptive impedance control in haptic teleoperation to improve transparency under time-delay,” *IEEE International Conference on Robotics and Automation*, pp. 212–219, May 2008.

[185] J.-H. Ryu, J. Artigas, and C. Preusche, “A passive bilateral control scheme for a teleoperator with time-varying communication delay,” *Mechatronics*, vol. 20, pp. 812–823, Oct. 2010.

[186] H. Choi, R. Balachandran, and J.-H. Ryu, “Chattering-Free Time Domain Passivity Approach,” *IEEE Transactions on Haptics*, vol. 15, pp. 572–581, July 2022.

[187] M. Panzirsch, J.-H. Ryu, and M. Ferre, “Reducing the conservatism of the time domain passivity approach through consideration of energy reflection in delayed coupled network systems,” *Mechatronics*, vol. 58, pp. 58–69, Apr. 2019.

[188] S. A. Schwarz and U. Thomas, “Variable Impedance Control for Safety and Usability in Telemanipulation,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6177–6182, Oct. 2022.

[189] M. Panzirsch, M. Sierotowicz, R. Prakash, H. Singh, and C. Ott, “Deflection-Domain Passivity Control of Variable Stiffnesses Based on Potential Energy Reference,” *IEEE Robotics and Automation Letters*, vol. 7, pp. 4440–4447, Apr. 2022.

[190] M. Bednarczyk, H. Omran, and B. Bayle, “Passivity Filter for Variable Impedance Control,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7159–7164, Oct. 2020.

[191] M. Tufail and C. W. de Silva, “Impedance control schemes for bilateral teleoperation,” in *2014 9th International Conference on Computer Science & Education*, pp. 44–49, Aug. 2014.

[192] A. Bambade, S. El-Kazdadi, A. Taylor, and J. Carpentier, “PROX-QP: Yet another Quadratic Programming Solver for Robotics and beyond,” in *Robotics: Science and Systems XVIII*, Robotics: Science and Systems Foundation, June 2022.

[193] A. Krause and C. Ong, “Contextual Gaussian Process Bandit Optimization,” in *Advances in Neural Information Processing Systems*, vol. 24, Curran Associates, Inc., 2011.

[194] R. Pinsler, P. Karkus, A. Kupcsik, D. Hsu, and W. S. Lee, “Factored Contextual Policy Search with Bayesian optimization,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7242–7248, May 2019.

[195] C. Yu, J. Cao, and A. Rosendo, “Learning to Climb: Constrained Contextual Bayesian Optimisation on a Multi-Modal Legged Robot,” *IEEE Robotics and Automation Letters*, vol. 7, pp. 9881–9888, Oct. 2022.

[196] A. Chen, T. Ngoc Dinh, T. Raïssi, and Y. Shen, “Outlier-robust set-membership estimation for discrete-time linear systems,” *International Journal of Robust and Nonlinear Control*, vol. 32, no. 4, pp. 2313–2329, 2022.

[197] N. Meslem and A. Hably, “Robust set-membership state estimator against outliers in data,” *IET Control Theory & Applications*, vol. 14, no. 13, pp. 1752–1761, 2020.

[198] F. Ferraguti, N. Preda, A. Manurung, M. Bonfe, O. Lambercy, R. Gassert, R. Muradore, P. Fiorini, and C. Secchi, “An Energy Tank-Based Interactive Control Architecture for Autonomous and Teleoperated Robotic Surgery,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1073–1088, 2015.

[199] E. Naerum, O. J. Elle, and B. Hannaford, “The Effect of Interaction Force Estimation on Performance in Bilateral Teleoperation,” *IEEE Transactions on Haptics*, vol. 5, pp. 160–171, Apr. 2012.