PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Lara Candido Alvim

Switching Receding-Horizon Approximate Estimation and Control of a Flexible Joint Robotic Manipulator

Dissertação de Mestrado

Dissertation presented to the programa de pós-Graduação em Engenharia Mecânica, do Departamento de Engenharia Mecânica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Mecânica.

Advisor: Prof. Helon Vicente Hultmann Ayala
Co-Advisor: Prof. Elias Dias Rossi Lopes

Rio de Janeiro
May 2023

Pontifícia Universidade Católica do Rio de Janeiro

Lara Candido Alvim

Switching Receding-Horizon Approximate Estimation and Control of a Flexible Joint Robotic Manipulator

Dissertation presented to the programa de Pós-graduação em Engenharia Mecânica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Mecânica. Approved by the Examination Committee:

Prof. Helon Vicente Hultmann Ayala
Advisor
Departamento de Engenharia Mecânica – PUC-Rio

Prof. Elias Dias Rossi Lopes
Co-Advisor
IME

Prof. Marco Antônio Meggiolaro
Departamento de Engenharia Mecânica – PUC-Rio

Prof. Roberto Zanetti Freire
UTFPR

Rio de Janeiro, May 10th, 2023

Lara Candido Alvim

Holds a bachelor’s degree in mechanical engineering (2019), from the Catholic University of Petrópolis.

Bibliographic data

Alvim, Lara Candido

Switching receding-horizon approximate estimation and control of a flexible joint robotic manipulator / Lara Candido Alvim; advisor: Helon Vicente Hultmann Ayala; co-advisor: Elias Dias Rossi Lopes. – 2023.

49 f. : il. color. ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Mecânica, 2023.

Inclui bibliografia

1. Engenharia Mecânica – Teses. 2. Controle preditivo baseado em modelo. 3. Sistemas híbridos. 4. Sistemas chaveados. 5. Sistemas não lineares. 6. Manipuladores robóticos. I. Ayala, Helon Vicente Hultmann. II. Lopes, Elias Dias Rossi. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Mecânica. IV. Título.

CDD:621

To my family,

for their trust and support.

Acknowledgments

To God, for the company, care, words, and opportunities during this journey.

To my Parents, Ana Cristina and Luís Flaviano, for encouraging me to move forward despite all the difficulties.

To my sisters Ana Carolina, Alessandra, Gabriela, and Vitória, and my brothers Luiz Otávio, Davi, Andrew, and Emanuel, for their support and understanding during this period.

To my niece Nathasha for her friendship and for making my life more colorful.

To all my family and brothers in Christ, for always being there and for the prayers!

To my friends Carlos Henrique, Leonardo, Lara Cristina, and Matheus Andrade for their help.

To my advisors, Helon Ayala and Elias Lopes, for their patience, advice, teachings, and support.

To PUC-Rio and all the staff of the Mechanical Engineering Department for their support.

"This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001"

Abstract

Alvim, Lara Candido. Ayala, Helon Vicente Hultmann (Advisor). **Switching Receding-Horizon Approximate Estimation and Control of a Flexible Joint Robotic Manipulator**. Rio de Janeiro, 2023. 49p. Dissertação de Mestrado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

The advances in Robotics in recent decades allow a growing range of robotic manipulator applications in various industry sectors. This directly impacts Human-Robot Interaction (HRI), increasing tasks that require a shared work environment, safety performance, and the contact detection ability of the robotic manipulator. Consequently, control methods capable of predicting contact, and controlling force or trajectory to avoid damage during collisions become increasingly necessary either for safety or performance reasons. Separating the dynamics of a single-link manipulator into two modes, namely position control mode (free mode) and torque control mode (contact mode), the first part of this dissertation deals with the estimation problem of states for active mode detection through the implementation of the Moving Horizon State Estimation with Neural Networks (NNMHSE) method. The effectiveness of the proposed estimation method is evaluated by comparing the states and modes generated by the MHSE and those estimated by the Neural Network. This method showed low RMSE values, high values of R^2, and a reduction in the processing time of the estimation algorithm. The second part of this dissertation deals with the position and force switching problem for a non-linear robotic manipulator, applying Model-Based Predictive Control (MPC). The implemented switched MPC algorithm effectively controlled both modes of the system, presenting low prediction error, approximately 2% in position control mode and 0.5% in torque control mode, even considering cyclical changes in the modes. Both methods prove to be suitable for controlling co-located robotic manipulators with humans or in unstructured environments through operation mode detection and position-torque switching control.

Keywords

Model predictive control; Hybrid systems; Switching systems; Nonlinear systems; Robotic manipulators.

Resumo

Alvim, Lara Candido. Ayala, Helon Vicente Hultmann. **Estimação de horizonte finito aproximada e controle preditivo de sistemas chaveados aplicados a manipuladores robóticos flexíveis**. Rio de Janeiro, 2023. 49p. Dissertação de Mestrado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Os avanços da Robótica nas últimas décadas permitem um aumento nas gamas de aplicações de manipuladores robóticos em diversos setores da indústria. Isto, impacta diretamente a interação Homem-Robô (HRI), resultando em um aumento de tarefas que requerem compartilhamento de ambiente de trabalho, desempenho de segurança e a habilidade de detecção de contato do manipulador robótico. Consequentemente, métodos de controle capazes de prever contato, controlar força ou trajetória para evitar danos durante colisões se tornam cada vez mais necessários seja por questões de segurança ou de desempenho. Separando a dinâmica de um manipulador de um único elo em dois modos, sendo eles modo de controle de posição (modo livre) e modo de controle de torque (modo de contato), a primeira parte desta dissertação, lida com o problema de estimação de estados para detecção do modo ativo através da implementação do método de Estimação de Estados de Horizonte móvel com Redes Neurais (NNMHSE). A efetividade do método de estimação proposto é avaliada através da comparação dos estados e modos gerados pelo MHSE e dos estimados pela Rede Neural. Este método apresentou baixos valores de RMSE, altos valores de R^2, e uma redução do tempo de processamento do algoritmo de estimação. A segunda parte desta dissertação lida com o problema de controle de posição e força chaveado para um manipulador robótico não linear, aplicando Controle Preditivo Baseado em Modelo (MPC). O algoritmo MPC chaveado implementado mostrou-se capaz de controlar efetivamente ambos os modos do sistema apresentando baixo erro na predição, aproximadamente 2% no modo de controle de posição e 0.5% no modo de controle de torque, mesmo considerando alterações cíclicas nos modos. Ambos os métodos provam ser adequados para controle de manipuladores robóticos colocalizados com seres humanos ou em ambientes desestruturados por meio da detecção do modo de operação e do controle chaveado posição-torque.

Palavras-chave

Controle preditivo baseado em modelo; Sistemas híbridos; Sistemas chaveados; Sistemas não lineares; Manipuladores robóticos.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANNs – Artificial Neural Networks

COP – Constrained Optimization Problem

DOF – Degrees of Freedom

HRC – Human-Robot collaboration

LMA – Levenberg-Marquardt

MHE – Moving-Horizon Estimation

MHSE – Moving-Horizon State Estimation

MPC – Model Predictive Control

NMPC – Nonlinear Model Predictive Control

NNMHE – Neural Network Moving-Horizon Estimation

NLP – Nonlinear Programming

ODE – Order Differential Equation

pHRI – Physical Human-Robot Interaction

R^2 – Coefficient of determination (R-Squared)

RMSE – Root mean square error

SSR – Space State Representation

Lean on, trust in, and be confident in the Lord with all your heart and mind, and do not rely on your own insight or understanding. In all your ways know, recognize, and acknowledge Him, and He will direct and make straight and plain your paths.

***Proverbs**, 3.5-6*

1
Introduction

Advances in robot recognition and manipulation skills have gradually increased over the last few decades, resulting in a wide range of applications for robotic manipulators inside and outside the industry. Among the crucial factors contributing to this advance is research on hardware and control technologies. The growing expectation for expanding the use of robotic manipulators increasingly motivates the development of research that considers robotics applications for different functionalities emphasizing how humans and robots can work together more effectively.

In recent decades, safety in physical Human-Robot interaction (pHRI) has become one of the biggest challenges and topics of study in the field of control due to advances in robotics and the increase in industrial activities that require Human-Robot collaboration (HRC) for successful task execution (Colgate et al., 2008; Haddadin, 2013; Nubert et al., 2013; Magrini et al., 2020; Zacharaki et al., 2020; Bi et al., 2021). The need for collaboration increases the complexity in the design of modern controllers, incorporating problems such as dynamic modeling (Bi et al., 2021; Xuan et al., 2020; Ibrahim et al., 2018), collision detection (Vorndamme et al., 2017; Haddadin et al., 2017), hybrid control of force and torque (Li et al., 2018; Wang et al., 2021).

1.1.
Contextualization

The study of controllers for robotic manipulators that perform tasks that require pHRI can be separated into two groups: trajectory controllers for activities in which the robot does not need to have contact with the environment (free motion) and controllers for activities that require the robot is in contact with its environment and controls the interaction force (Pankert and Hutter, 2020). When a manipulator performs tasks that require both types of controllers, two of the solutions are to apply hybrid control (Wang et al., 2021) or switching control (Li et al., 2018). According to Scharlau (2013), one of the motivations for studying switching systems is their application in the description of a significant number of processes and methods. Consequently, this has become a more discussed topic that generates varied research results in complex switched systems focused on controllability (Li et al., 2018), (Leica et al., 2015), observability (Du et al., 2021), (Yu et al., 2017), and stability (Vargas et al., 2019). Among a set of analysis properties, the observation of modes plays a crucial role in analyzing the response of multimode systems (Baglietto et al., 2012). To improve the accuracy of mode-observability of systems, some papers have investigated the Moving Horizon Estimation (MHE) approach (Guo and Huang, 2013; Alessandri et al., 2005). The MHE has emerged as a powerful technique for tackling the problem of estimating the state of a dynamic system in the presence of nonlinearities and disturbances (Alessandri et al., 2010). The main idea behind the MHE method is minimizing an estimation cost function defined on a sliding window, (Nocedal et al., 2006) state that the least-square problems measure the discrepancy between the model and the observed behavior of the system and select values for the parameters that best match the model to the data by minimizing the objective function.

1.2. Motivation

Advancing the state-of-the-art of (Li et al., 2018) which implemented switching control Position/Torque using a proportion-integral (PI) algorithm for position control and Sliding Mode Control (SMC) for torque control in a 1- DOF robotic manipulator, this dissertation proposes the implementation of the Model Predictive Control (MPC), which stands out as a control technique capable of dealing with restrictions and uncertainties, with time-varying dynamics and system models for anticipatory control actions. Model Predictive Control (MPC) is an advanced control method that generates control inputs to ensure a desired trajectory and force tracking by the manipulator. The MPC problem involves minimizing a cost function using a set of future control actions within a prediction horizon (Brunello, 2021). The MPC method has a preview capability that enables the prediction of the future behavior of the model and improves the controller performance by adding future references to the control. Moreover, it deals with system constraints and system input-output interactions. These characteristics are crucial to deal with the requirements of controllers which aim at pHRI problems.

Examples of MPC applications include path tracking based on machine learning (Liu et al., 2022), workspace reference tracing for mobile manipulators (Pankert and Hutter, 2020), switching systems with dwell-time restriction (Ong et al., 2016), and safe and fast tracking with neural networks (Nubert et al., 2020).

1.3. Objectives and Contributions

Firstly, this dissertation considers the application of an approximate filter of the mode observability properties for a robotic manipulator modeled as a nonlinear switching system, which has two dynamic tasks (which define the modes) to be performed. Afterward, it considers the application of switched MPC to control the position and force of the robotic manipulator, allowing the change of operation mode when necessary. That is, while the manipulator is in free movement, the activated controller is in position control mode, and if any contact with the end effector occurs, the controller switches to Torque Control Mode, ensuring that the exerted torque is not harmful to the end effector or the environment. The two main objectives are:

- Propose a neural network moving horizon estimator algorithm in the high-level programming language (Matlab).
- Propose an algorithm for implementing Switching Model Predictive Control (MPC) for position and torque control in the high-level programming language (Matlab).

1.4. Document Organization

The remainder of this dissertation is organized as follows: In section 2, the methods and the robotic manipulator used to simulate the proposed methodology are described. In Section 3, the formulation and simulation results of the Approximative NNMHE filter are present. In Section 4 the formulation and simulation results of the NMPC method are presented. Finally, section 5 concludes and suggests future experiments.

2
Methods

In this chapter, is described the theoretical background required to understand this dissertation. Firstly, the Moving Horizon State Estimation is presented, then the Model Predictive Control, and, lastly, the Case Study.

2.1.
Moving-horizon State Estimation (MHSE)

The Moving-Horizon State Estimation (MHSE) problem, also known as finite-memory, receding-horizon, or sliding-window estimation, consists of using a set of most recent information within a time interval to solve an optimization problem that estimates the dynamic states of a system, as presented in Figure 1. According to (Alessandri et al., 2005) two contributions are made up: the first one is a weighted term penalizing the distance of the current estimated state from its prediction (both computed at the beginning of the sliding window); the second is the usual prediction error computed based on the most recent measures.

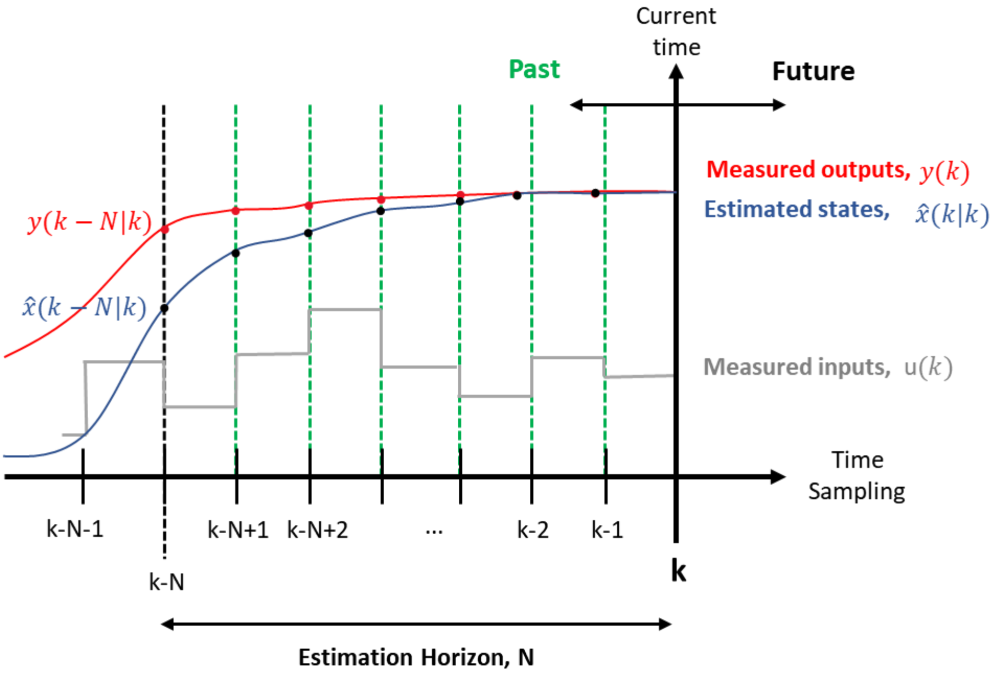


Figure 1 – Moving-Horizon State Estimation Method

Let the system be modeled by the following nonlinear discrete time system

$$\begin{aligned} x_{t+1} &= f(x_t, u_t) + \varepsilon_\mathrm{t} \\ y_\mathrm{t} &= h(x_t\,) + \eta_t \end{aligned} \qquad (1)$$

where, $t \in \mathbb{Z}_+ \ = \{0,1,\dots,\}$ is the time instant; $x_t \in\ X \subset \mathbb{R}^\mathrm{n}$ is the state vector (the initial state x_0 is unknown), $u_t \in\ U \subset \mathbb{R}^\mathrm{m}$ is the input vector; $\varepsilon_t \subset \mathbb{R}^\mathrm{n}$ is the system noise vector; $y\ \in \mathbb{R}^p$ is the output /observation vector; and $\eta_t \in \mathcal{V} \subset \mathbb{R}^\mathrm{n}$ is the random measurement noise vector. The statistics of the random variables $,\eta_\mathrm{t}$ are assumed to be unknown deterministic variables with unavailable statistics. Functions f and h are nonlinear.

The estimates are based on data obtained according to a receding-horizon strategy where the observation window is composed of $N+1$ measurements for a time interval $[t-N, t]$ with $t \geq N$. The objective of the MHSE is to find the state estimates $\hat{x}_{t|t}$ at time t on the basis of $I = [y_{t-N}, \dots, y_t, u_{t-N}, \dots, u_{t-1}]^T$ the information vector that contains all the information of the output and input vector within the observing horizon, as well as the state prediction $\bar{x}_{t-N}$ in the beginning of the window $[t-N]$.

The state estimates $[\hat{x}_{t-N|t}, \dots \hat{x}_{t|t}]$ are calculated at time $t = N, N+1, \dots$ by minimizing a least-squares cost function as in Eq. (2).

$$J = \mu \parallel \hat{x}_{t-N|t} - \bar{x}_{t-N} \parallel^2 + \sum_{i=t-N}^{t} \parallel y_i - h\big(\hat{x}_{i|t}\,, u_i\big) \parallel^2 \qquad (2)$$

where the coefficient μ is a positive scalar that serves as a penalty parameter that influences the fit of the estimated and predicted continuous state estimation at the beginning of the window. The vector $\bar{x}$ denotes an a priori prediction of x_0. In this work, the function h does not depend on u_i. The optimization problem is described in eq. (3).

$$\begin{aligned} \hat{x}_{t-N|t} &= \arg\min J\ (\hat{x}, u_i) \\ s.t.\ \ \hat{x}_{i+1|t} &= f\big(\,\hat{x}_{i|t}, u_i\big) \\ y_{i+1|t} &= h\,\big(\hat{x}_{i|t}\big) \end{aligned} \qquad (3)$$

After obtaining the estimated states at the beginning of the window, it is possible to obtain the estimated state at the current time using the dynamic model equation as in eq. (4).

$$\hat{x}_{i+1|t} = f(\hat{x}_{i|t}, u_i) \quad i = t - N, \dots, t - 1 \tag{4}$$

2.2. Model Predictive Control (MPC)

Model Predictive Control is an advanced control method that uses a dynamic model to predict the future behavior of a system and select the most suitable control action. An MPC model is classified as nonlinear (or nonconvex) if the cost function is nonconvex or if the system or constraints are nonlinear. The idea of the Nonlinear Model Predictive Control (NMPC) is presented by (Grüne and Pannek, 2011) and is schemed as follows: at each sampling instant, we optimize the predicted future behavior of the nonlinear system over a finite time horizon $\mathrm{k} = 0, \dots, \mathrm{N} - 1$ of length $\mathrm{N} \geq 2$, and use the first result of the optimal control sequence u_k^* as a feedback control value for the next sampling interval. A graphical representation of the MPC method can be observed in Figure 2.

Therefore, based on the model predictions and current measured or estimated states of the system the NMPC solves an optimal control problem through minimizing a cost function. Accordingly, the cost function J is in eq. (5).

$$J(u_{k,k+N-1}) = \sum_{i=k}^{k+N-1} \|x_i - q_i\|_Q^2 + \sum_{i=k}^{k+N-1} \|u_i - r_i\|_R^2 + \sum_{i=k}^{k+N-1} \|\Delta u_i\|_S^2 \tag{5}$$

where q_k, r_k are the references for x_k, u_k, respectively. Q represents the weight coefficient of the outputs, R represents the weight coefficient of the inputs and S represents the weight coefficient of input variation, which are diagonal matrices. Finally, the control action variation is given by $\Delta u_i = u_k - u_{k-1}$. In the NMPC cost function, weights are included to illustrate the importance of each performance metric present in it and the tracking ability of the system can be improved by tuning them, (Liu et al., 2022).

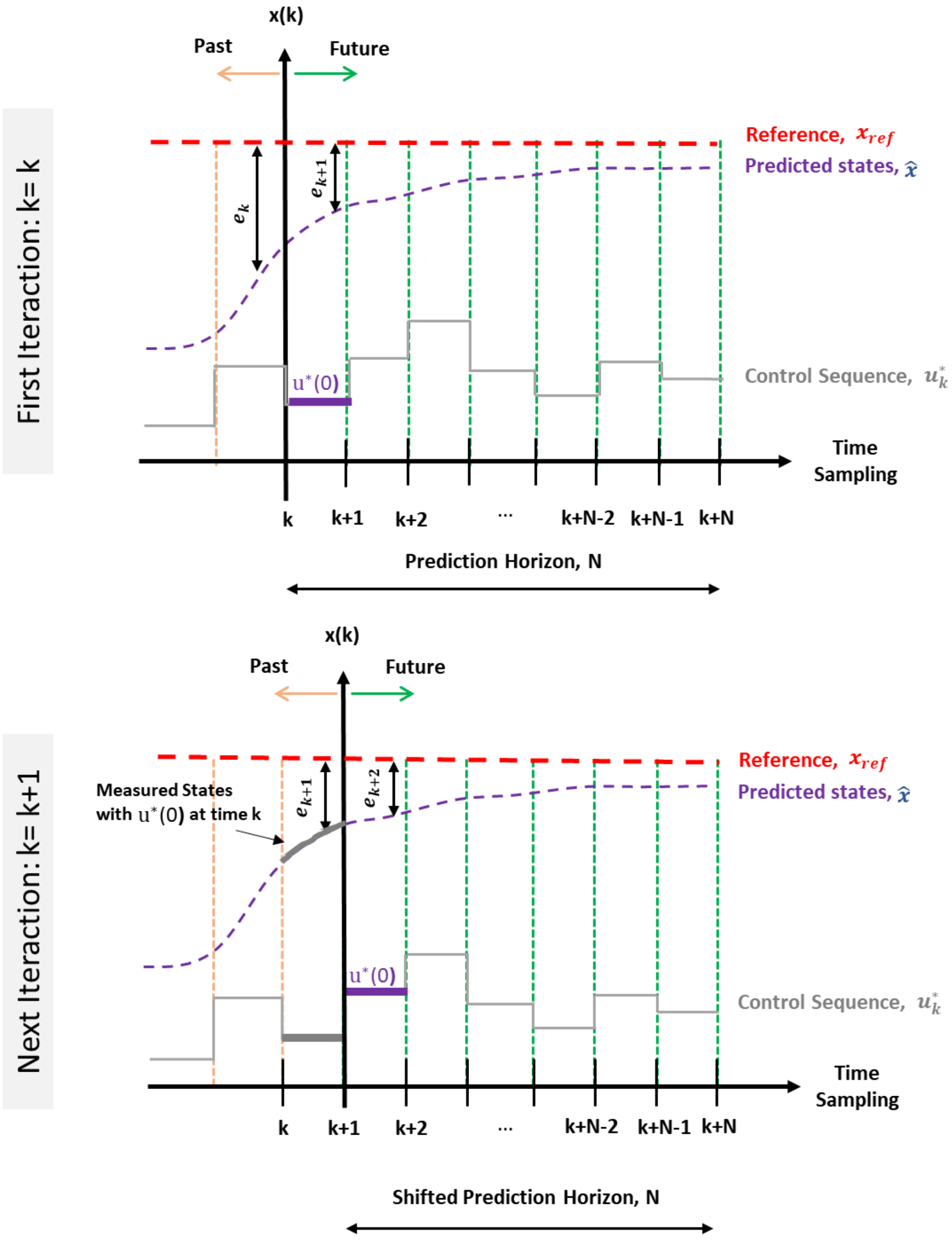


Figure 2 - MPC method execution over two consecutive time steps

The control law $u^*_{k,k+N-1}$ can be obtained by solving the optimization problem in eq. (6).

$$u^*_{k,k+N-1} = \arg\min J(u_{k,k+N-1}) \\ s.t.\ \ u_i \in U, x_i \in X \tag{6}$$

As indicated by (Rawlings et al., 2017) manipulated inputs, such as torque, are limited in most physical systems. But, for outputs or states, restrictions are imposed for safety, operability, and product quality. The ability to take constraints into account is one of the main reasons for the success of MPC in general (Grüne and Pannek, 2011). The Shooting-methods serve as a solution strategy for the constrained optimization problem of the NMPC, being possible to apply single or multiple shooting methods.

In this work, multiple shooting is being used, which according to (Katayama, 2023), is more suitable for NMPC in terms of computational speed and convergence. In the Multiple Shooting method, it is possible to add the states as variables of the optimization problem, imposing their constraints as arguments of the cost function. Therefore, the cost function is subject to dynamic system constraints and inequality constraints in the states and control inputs, as observed in eq. (7).

$$
\begin{aligned}
[u_k, x_k] = \arg\min J(u_k, x_k) & \\
s.t.\ \ x(i+1) = f\big(x(i), u(i)\big) & \\
y(i+1) = h\big(x(i)\big) & \\
x_k = 0 & \\
u_{min} \ \leq u_i \leq u_{max} & \\
x_{min} \ \leq x_i \leq x_{max} &
\end{aligned}
\tag{7}
$$

2.3. Constrained Optimization Problem

The Constrained Optimization Problem (COP) is part of the MHSE and MPC mathematical formulation. It is solved to find an optimal value, through the maximization or minimization of an objective function with equality or inequality constraints on the variables. The MHSE and MPC algorithms are iterative, they aim to generate a sequence of iterates that converges to a local solution to the optimization problem. A Constrained Optimization Problem is described in the general form as in eq. (8).

$$
\begin{array}{ll}
\min & f(x) \\
subject\ to & g_i(x) = 0\,, \quad i = 1\,\dots, n \\
 & h_j(x) \geq 0\,, \quad j = 1\,\dots, m
\end{array}
\tag{8}
$$

where x is the vector of decisions variable, $f(x)$ is the function to be optimized, g_i and h_i are the constraints that need to be satisfied. When the function or the constraints are nonlinear, we have a Nonlinear Programming (NLP) problem. Therefore, to solve the optimization problem, it is necessary to satisfy necessary (1st-order) and sufficient (2nd-order) conditions, the Karush-Kuhn-Tucker (KKT) conditions, and Lagrange multiplier conditions, respectively, as detailed by (Grüne, and Pannek 2011).

The NMPC algorithm is considered an NLP and solved by numerical optimization techniques (Katayama, 2023), which can be solved using CasADi (Computer algebra system with Automatic Differentiation), a symbolic optimization framework proposed by (Andersson et al., 2019) which provides the values of the derivatives for the numerical solution of the differential equations.

The optimization solver interfaced to CasADi used is IPOPT (Interior Point Optimizer), that "is an open-source software package for large-scale nonlinear optimization" Wachter (2002, 2009), designed as Interior-points methods (also called barrier methods).

2.4. Case study: 1-DOF flexible robotic manipulator

This section presents the model used to evaluate the performance of the estimation and control of the nonlinear switching system. The case study concerns a single-link flexible joint robotic manipulator analyzed as a switching system with two operation modes, the first when the link is free and the second when the link is in contact.

2.4.1. Dynamical Model

In this dissertation, we use only the mechanical system of a single-link flexible joint robotic manipulator (SLFJM) for the simulations. The basic schematic diagram is presented in Figure 3.

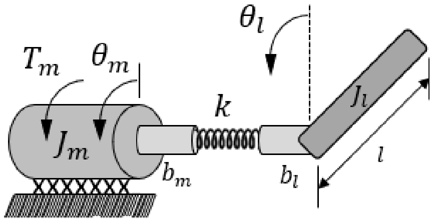


Figure 3 - Single-Link Flexible Joint Manipulator (SLFJM)

where, T_m is the motor torque, T_b is the friction torque, T_k is the torsional torque, T_L is the link load torque, J_m is the motor inertia, J_l is the link inertia, b_m is the motor viscous friction coefficient, b_l is the link viscous friction coefficient, θ_m is the angular position of the motor, θ_l is the angular position of the link, k is the stiffness constant of the spring which represent the flexible joint, m is the link mass, g is gravity acceleration, l is the link length. According to (Zhang et al., 2010) and (Fan and Arcak, 2003), suitable values for the system parameters are presented in Table 1.

Table 1 – System Parameters

Parameters	Values	Units
Link Inertia - J_l	9.3×10^{-3}	$kg \cdot m^2$
Motor Inertia - J_m	3.7×10^{-3}	$kg \cdot m^2$
Torsional Spring Constant - k	1.8×10^{-1}	$N \cdot m/rad$
Link Mass – m	2.1×10^{-1}	Kg
Link Length – l	0.15	m
Viscous Friction Coefficient -b_m	4.6×10^{-2}	$N \cdot m/V$
Gravity Constant – g	9.8	m/s^2
Amplifier Gain - k_τ	8×10^{-2}	Nm/V

According to (Kluever, 2015), the mathematical model can be obtained by applying Newton's second law for rotational systems, which states that the sum of the torques T acting on a body is equal to the product of inertia J and acceleration $\ddot{\theta}$, as presented in eq. (9).

$$\sum T = J \cdot \ddot{\theta} \tag{9}$$

Figure 4 presents the Free-Body diagrams of each inertial element of the SLFJM. Each arrow indicates an external torque acting on the inertial element, and the angular displacement is considered positive in the counterclockwise direction.

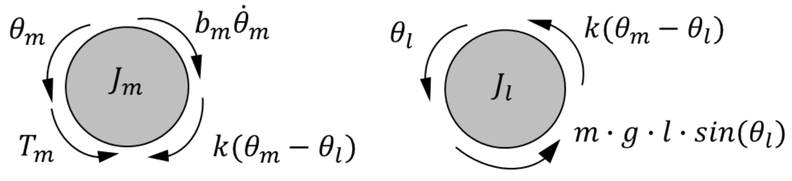


Figure 4 – Free-Body Diagram of the inertial elements J_m and J_l

$$T_m - T_b - T_k = J_m \cdot \ddot{\theta}_m \tag{10}$$

$$T_k + T_L = J_l \cdot \ddot{\theta}_e \tag{11}$$

$$T_m = k_t \cdot u \tag{12}$$

$$T_b = b_m \cdot \dot{\theta}_m \tag{13}$$

$$T_k = k(\theta_m - \theta_l) \tag{14}$$

$$T_L = mgl \sin \theta_l \tag{15}$$

Substituting eq. (12-14) into eq. (10) and eq. (11) and isolating the acceleration, we get the equations of motion.

$$\ddot{\theta}_m = \frac{1}{J_m}\left[T_m - b_m\dot{\theta}_m - k(\theta_m - \theta_l)\right] \tag{16}$$

$$\ddot{\theta}_l = \frac{1}{J_l}[k(\theta_m - \theta_l) - mgl \sin \theta_l] \tag{17}$$

The gravity term introduces a nonlinearity to the model, so, the equations of motion can be represented in the nonlinear space state representation (SSR), as present in eq. (18).

$$\dot{x} = f(x, u) \tag{18}$$

where x is a vector of the state variables, $\dot{x}$ is the derivative of the state variables, u is the input, and f is a nonlinear function. State variables x are the minimum number of elements that describes all the system dynamics, and the number of states that describes the system is relative to the order of the equation. For each Second Order Differential Equation (ODE), we have two states, then being necessary four states for the mechanical system. So, we define the angular position and angular velocity for the motor and link as continuous state variables. The input is defined as Torque T_m. The state and input vector are presented in eq. (19).

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \theta_m \\ \dot{\theta}_m \\ \theta_l \\ \dot{\theta}_l \end{bmatrix} \tag{19}$$
$$u = [u_1] = [T_m]$$

where, $\dot{\theta}_m$ is the motor angular velocity, and $\dot{\theta}_m$ is the link angular velocity. Deriving the continuous states variables, we have

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_m \\ \ddot{\theta}_m \\ \dot{\theta}_l \\ \ddot{\theta}_l \end{bmatrix} \tag{20}$$

Replacing eq. (16-17) in eq. (18), we have the dynamic model in a state space form.

$$\dot{x} = \begin{bmatrix} \dot{\theta}_m \\ \frac{1}{J_m}\left[T_m - b_m\dot{\theta}_m - k(\theta_m - \theta_l)\right] \\ \dot{\theta}_l \\ \frac{1}{J_l}[k(\theta_m - \theta_l) - mgl\sin\theta_l)] \end{bmatrix} \tag{21}$$

Replacing the state variables, eq. (20), in the State-Space model, eq. (21), we have the following set of equations.

$$f(x,u) = \begin{bmatrix} x_2 \\ \frac{1}{J_m}[u_1 - b_m x_2 - K(x_1 - x_3)] \\ x_4 \\ \frac{1}{J_l}[k(x_1 - x_3) - mgl\sin x_3)] \end{bmatrix} \tag{22}$$

2.4.2. Contact Modeling

Considering the presence of a fault in the system described in subsection 2.3.1, due to a force exerted against the link that turns the system unable to perform normal functions for a time or due to extra abnormal friction in the contact between the link and a surface, both capable of preventing the link displacement, as presented in Figure 5, the faults are modeled with constant states then the derivative states for both possible faults considered is equal zero. In addition to gravity, switching between free mode and contact mode also represents a non-linearity in the system.

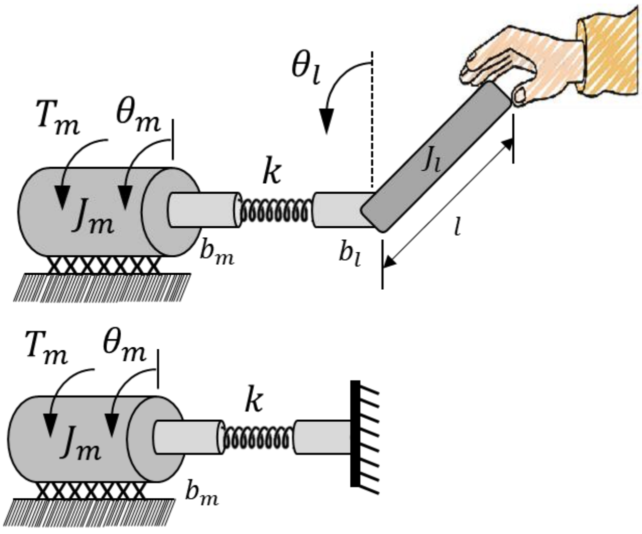


Figure 5 – Single-Link Flexible Joint Manipulator: Contact Model

By modeling the fault as a nonlinear system according to eq. (17), in section 2.3.1, we may consider that the robotic manipulator has two discrete states represented by the system modes $\sigma_t \in \{1,2\}$, where $\sigma_t = 1$ represents the system working under normal conditions with the link moving freely (free mode) and $\sigma_t = 2$ represents the system under fault condition with the link stopped (contact mode). The switching system equation may be represented by eq. (23) and eq. (24).

$$f_1(x_t, u_t) = \begin{bmatrix} x_2 \\ \frac{1}{J_m}[u_1 - b_m x_2 - K(x_1 - x_3)] \\ x_4 \\ \frac{1}{J_l}[k(x_1 - x_3) - mglsin(x_3)] \end{bmatrix} \tag{23}$$

$$f_2(x_t, u_t) = \begin{bmatrix} x_2 \\ \frac{1}{J_m}[u_1 - b_m x_2 - K(x_1 - x_3)] \\ 0 \\ 0 \end{bmatrix} \tag{24}$$

To discretize the continuous states and obtain an approximate solution for the system's responses by solving the nonlinear differential equation, was used the numerical method Runge-Kutta fourth order (RK4). The equation that describes the RK4 method is presented by (Brunello, 2021), as presented in eq. (25).

$$k1 \ = f\big(u(i), x(i)\big)$$
$$k2 \ = \ f(u(i), x(i) + 0{,}5\Delta_t k1)$$

$$k3 \;=\; f(u(i), x(i) \;+\; 0{,}5\mathit{\Delta}_t k2) \qquad (25)$$

$$k4 \;=\; f(u(i), x(i) \;+\; \mathit{\Delta}_t k3)$$

$$x(i+1) = x(i) + \frac{\mathit{\Delta}_t}{6}(k1 \;+\; 2\cdot k2 \;+\; 2\cdot k3 \;+\; k4)$$

where $\mathit{\Delta}_t$ is the sampling time, $u(i)$ is the input, $x(i)$ is de estimated states of the system, and k represents the interactions. The response of the switching system can be observed in Figure 6 for a time of 20s, being (0-10s) the free mode and (10-20) the contact mode, with a sampling time defined as 0.05s, an input $T_m = k_t \cdot u$, where $u = 2\sin\,(2t)$ and, the state initial condition $x_0 = [\pi, 0,\ \pi, 0]^T$.

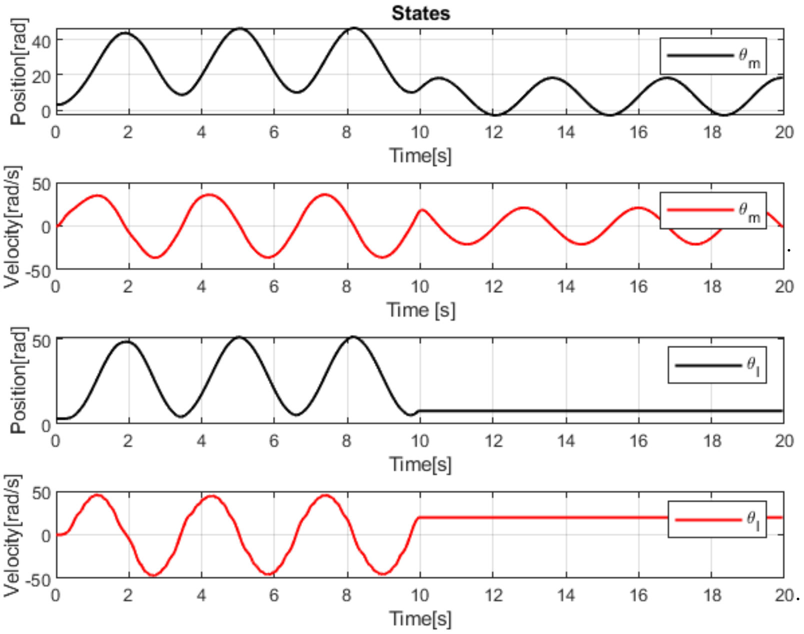


Figure 6 – Switching dynamics response of the Single-Link Flexible Joint Manipulator

3
Moving Horizon State Estimation for Contact Detection with Neural Networks

In this chapter, concepts related to Moving Horizon State Estimation for switching systems and related to Approximative filters with ANNs will be presented. Subsequently, the implementation results of the approximative filter with ANNs for the Moving-Horizon State Estimation Method, considering the switching system.

3.1.
Moving-horizon State Estimation for Switching Systems

The dynamics of hybrid systems incorporate continuous and discrete states of a system. To understand the behavior of these systems it is necessary to consider the variability of the subsystems or modes. For this, it is possible to isolate the discrete states of the system in switched events and consider that we have a switching system.

One of the characteristics of the MHSE method is the mode observability, which can determine whether the output of a system can be identified and differentiated, (Baglietto et al., 2012), and can be used to evaluate switching systems. Mode estimation is a strategy that can be suitable for systems in which modes vary unpredictably over time due to the possibility of using a fixed observation window. This performance depends on an a priori estimation of all the continuous states of the systems. Based on the returned value of an objective function for each estimate within a moving horizon window, the discrete state, i.e., the active mode of the system, is estimated (Ayala, 2012).

Defining the switched nonlinear discrete-time equation as a combination of the switched nonlinear equation with the nonlinear discrete-time equations, we have

$$\begin{aligned} x_{t+1} &= f_{\sigma_t}(x_t, u_t) + \varepsilon_{\mathrm{t}} \\ y_{\mathrm{t}} &= h_{\sigma_t}(x_t) + \eta_t \end{aligned} \tag{26}$$

where $\sigma_t \in \{1,2 \ldots, n\}$ is a switching rule that has a finite set of subsystems.

The state estimation procedure of a nonlinear switching system within a moving horizon can be formulated as stated by (Ayala, 2012), at any instant of time, based on the recent information vector I_t and in a previously known prediction of the initial conditions $\bar{x}_{t-N}$, deduce the estimation of the mode $\hat{\sigma}$ and of the state vectors $\hat{x}_{t-N|t}$.

To infer mode and state estimations, based on I_t and $\bar{x}_{t-N}$, by resorting to a least-square approach, one can consider the following moving-horizon strategy (Baglietto et. al., 2012; Ayala, 2012), accordingly

$$\widehat{\sigma}, \hat{x}_{t-N|t} \in argmin_{\widehat{\sigma} \in \mathcal{M},\, \hat{x}_{t-N|t} \in X} \, J\left(\hat{x}_{t-N|t}, \bar{x}_{t-N}, \hat{\sigma}, I_t \right) \tag{27}$$

where the estimated states based on the dynamical model can be obtained as in eq. (28), and the predicted states as in eq. (29).

$$\hat{x}_{k+1|t} = f_{\widehat{\sigma}_t}\left(\hat{x}_{k|t}, u_k\right), k = t-N, \dots, t-1 \tag{28}$$

$$\bar{x}_{t-N|t} = f_{\widehat{\sigma}_{t-N-1}}\left(\hat{x}_{t-N-1|t-1}, u_{t-N-1}\right), t = N+1, N+2, \dots \tag{29}$$

The cost function is defined as eq. (30).

$$\begin{aligned} J\left(\hat{x}_{t-N|t}, \bar{x}_{t-N}, \hat{\sigma}_t, I_t \right) = \mu \parallel \hat{x}_{t-N|t} - \bar{x}_{t-N} \parallel^2 \\ \parallel y_{t-N|t} - F_{\widehat{\sigma}_t}\left(\hat{x}_{t-N|t}, u_{t-N,t-1}, N\right) \parallel^2 \end{aligned} \tag{30}$$

The optimal estimate made at time t for each sequence of modes $i \in \mathcal{M}$ is denoted by eq. (31), and the optimal associated cost is denoted by eq. (32).

$$\hat{x}^i_{t-N|t} \in argmin_{\hat{x}^i_{t-N|t} \in X} \, J\left(\hat{x}^i_{t-N|t}, \bar{x}_{t-N}, i, I_t \right), i = 1,2, \dots, m \tag{31}$$

$$\begin{aligned} J\left(\hat{x}^i_{t-N|t}, \bar{x}_{t-N}, i, I_t\right) = \mu \parallel \hat{x}^i_{t-N} - \bar{x}_{t-N} \parallel^2 + \\ \parallel y_{t-N|t} - F_{\widehat{\sigma}_t}\left(\hat{x}^i_{t-N}, u_{t-N,t-1}, N\right) \parallel^2 \end{aligned} \tag{32}$$

The MHE optimization problem for the states and the modes is summarized in Figure 7.

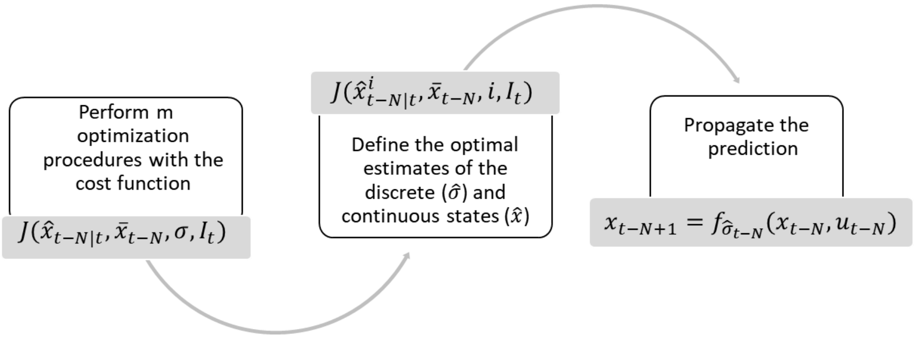

Perform m optimization procedures with the cost function
$J(\hat{x}_{t-N|t}, \bar{x}_{t-N}, \sigma, I_t)$
$J(\hat{x}^i_{t-N|t}, \bar{x}_{t-N}, i, I_t)$
Define the optimal estimates of the discrete ($\hat{\sigma}$) and continuous states ($\hat{x}$)
Propagate the prediction
$x_{t-N+1} = f_{\hat{\sigma}_{t-N}}(x_{t-N}, u_{t-N})$

Figure 7 – MHE Optimization Problem

3.2. Approximate filters with ANNs

Artificial Neural Networks (ANNs) are part of the set of supervised learning methods that seek to approximate a function to a dataset by minimizing the error between predicted and estimated outputs during the training process of a network (Ray,2019). For this, it is necessary to provide a set of observations data for the development of estimates that approximate the unknown function, also known as the target function. The Neural Networks that approximate the Moving-Horizon States and Mode Estimations, called NNMHE in this work, can be defined as

$$\hat{x}_t = NN(\,\bar{x}_{t-N}, I_t) \tag{33}$$

$$\hat{\sigma}_t = NN(\bar{x}_{t-N}, \hat{\sigma}, I_t) \tag{34}$$

where the input of such a network is the information vector I_t and the prediction $\bar{x}_{t-N}$ along the receding horizon. Among the various neural network architectures, this work uses a Feedforward Neural Network, due to its simplicity and applicability. The solution of the NNMHE may be performed through software toolboxes, therefore the training algorithm used was the Levenberg-Marquardt (LMA), available on MATLAB®. Figure 7 summarized the NNMHE problem.

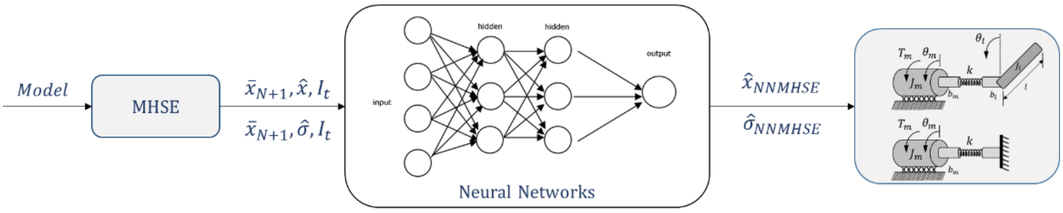


Figure 8 – NNMHE Problem

Although it is a metric for linear regression, the coefficient of determination R² is often used as a validation metric for neural network models, due to the ease of visualization of the model fit, however, according to Spiess and Neumeyer (2010) for some models this metric may be inappropriate, therefore, it should be analyzed along with other non-linear regression metrics. For this reason, this work the evaluation metrics for validation of the fit between the real and the estimated data are the coefficient of determination (R²) and the Root Mean Squared Error (RMSE) calculated as eq. (35) and eq. (36).

$$R^2 = 1 - \frac{\sum_{t=1}^{n}(x_i - \hat{x}_i)^2}{\sum_{t=1}^{n}(x_i - \bar{x}_i)^2} \tag{35}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{36}$$

For the R² metric, the results range from 0 to 1. R²=0 indicates that the model does not predict the output, and R²=1 that the model predicts the output perfectly. Regarding the RMSE, the smaller the error value, the better the fit of the model to the data set.

3.3. NNMHSE Results

For the MHSE, a window of $N = 10$ was used as simulation parameters, with the length of the estimation horizon equal $N + 1$, the weight of the cost function of μ =1e-3, and the stopping criterion of the optimization algorithm equal to $1e - 6$, respectively. Initially, a noiseless signal was used, equivalent to a sinusoidal function, with a sampling time of $t_s = 0.05s$ and a simulation time of $T = 120s$. Four states ($n_x = 4$), one input ($nu = 1$), and four outputs ($nz = 4$) were defined. Figure 9 presents the input signal used for the MHSE simulation.

The implementation of the approximate filter was carried out in MATLAB software for a simulation data set of 24,000 samples, resulting from the application of the MHSE method, as described in section 3.1, for the robotic manipulator with flexible joint described in section 2.3.2 of this dissertation.

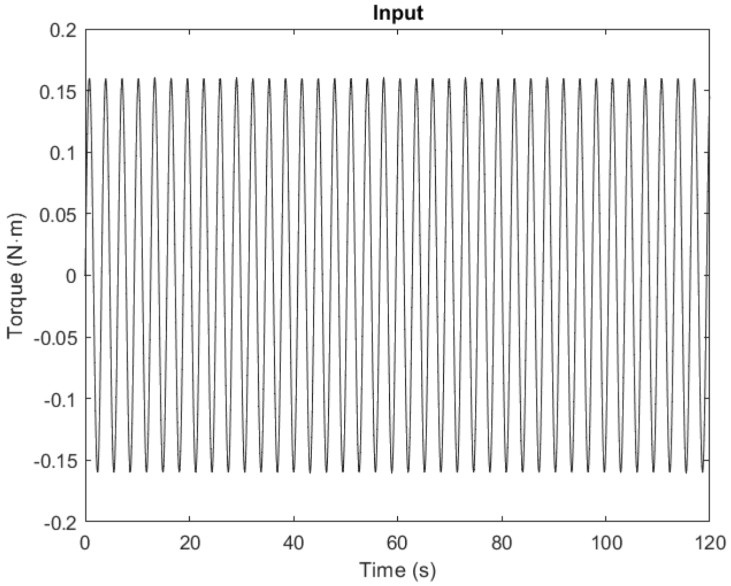


Figure 9 – Input Signal

Two networks were created, one for state estimation and the other for model estimation. The input parameters of the network are defined as the set of inputs U of the estimation window, the set of measured outputs Y within the estimation window, the states $\hat{X}_{t-N|t}$, and modes $\hat{\sigma}_{t-N|t}$. estimated at the beginning of the window.

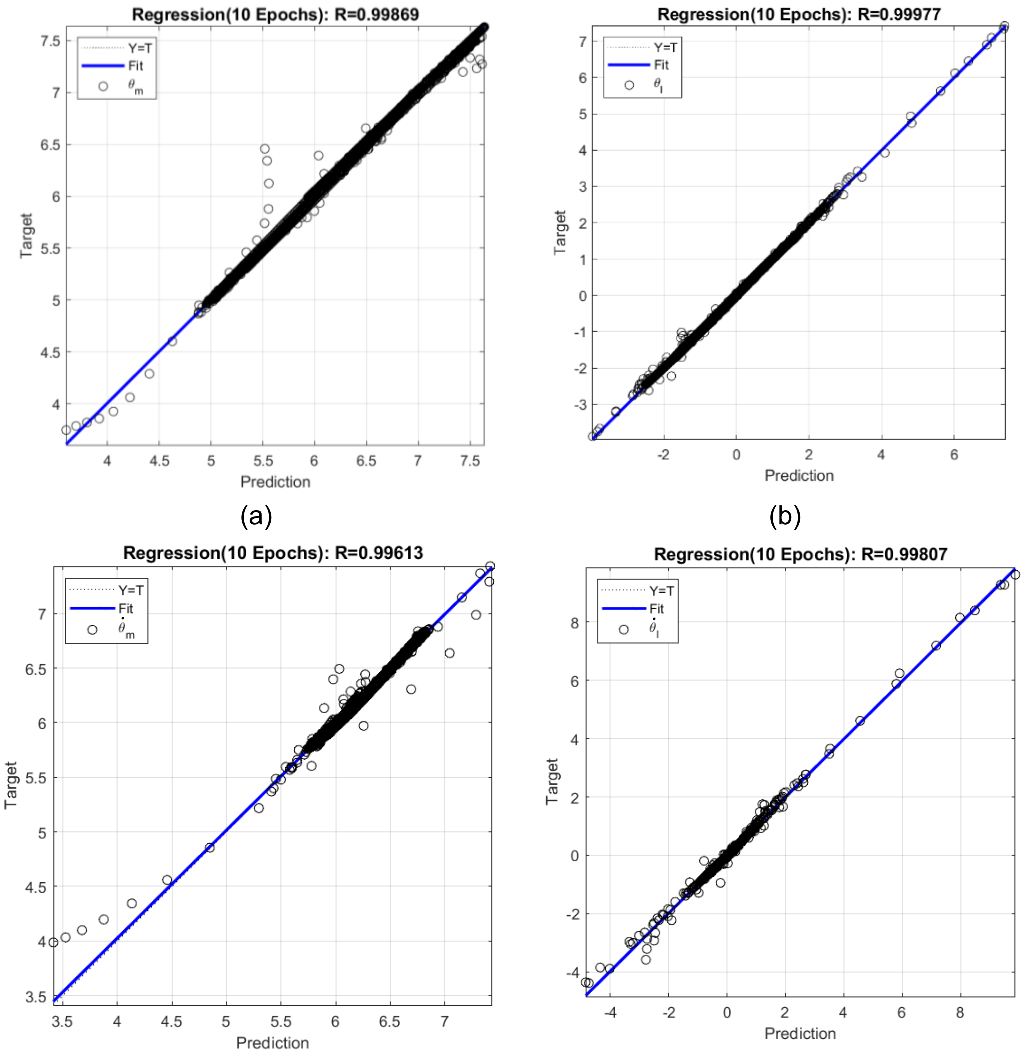
The ANN architecture is composed of 2 hidden layers and 10 neurons, and the tests were performed for different amounts of epochs. The purpose of varying the epochs is to spend more time training a network that can estimate the parameters well and can be tested later with noisy signals. As it is an approximation, the approximate filter tends to present a worsening in the estimation, therefore, to verify the accuracy of the adjustment of the Network to the estimation method, the coefficient of determination (R^2) and the Root Mean Squared Error (RMSE) is calculated. Table 2 presents the processing time for training the state estimation network and the RMSE values for each of the states numbered (1-4), being: 1 and 3 – Angular position and 2 and 4 – angular velocity, for the motor and link, respectively.

Table 2 – RMSE for the States Approximation

Epochs	Time(s)	RMSE1	RMSE2	RMSE3	RMSE4
10	23.1589	4.53E-04	2.22E-03	9.74E-04	1.57E-03
20	34.3695	1.13E-03	1.24E-03	6.85E-04	1.08E-03
30	49.9881	5.23E-04	4.99E-04	5.32E-04	5.72E-04
40	63.7439	5.48E-04	6.21E-04	5.65E-04	5.78E-04
50	82.0597	2.16E-03	1.88E-03	1.93E-03	2.30E-03

60	105.3591	2.15E-04	2.32E-04	2.01E-04	3.27E-04
70	120.5277	5.37E-04	5.65E-04	5.35E-04	6.22E-04
80	130.2284	1.30E-03	1.35E-03	1.03E-03	1.26E-03
90	147.3986	4.03E-05	3.57E-05	4.35E-05	4.43E-05
100	202.9350	3.16E-04	2.35E-04	2.90E-04	3.53E-04

The low RMSE values for state estimation indicate a good model fit of the network to the data set. However, as the training and validation data belong to the same simulation and are in the same operating range, it is necessary to validate the network with other data sets. The R² graph was generated for each of the states of the systems, that is, the motor angular position of the (θ_m) and link angular position (θ_l), the motor angular velocity ($\dot{\theta}_m$) and link angular velocity ($\dot{\theta}_l$), and for each one of the epoch variations (10 - 100). Figure 10 presents the linear regression for the 10-epoch network that showed the worst R² values.



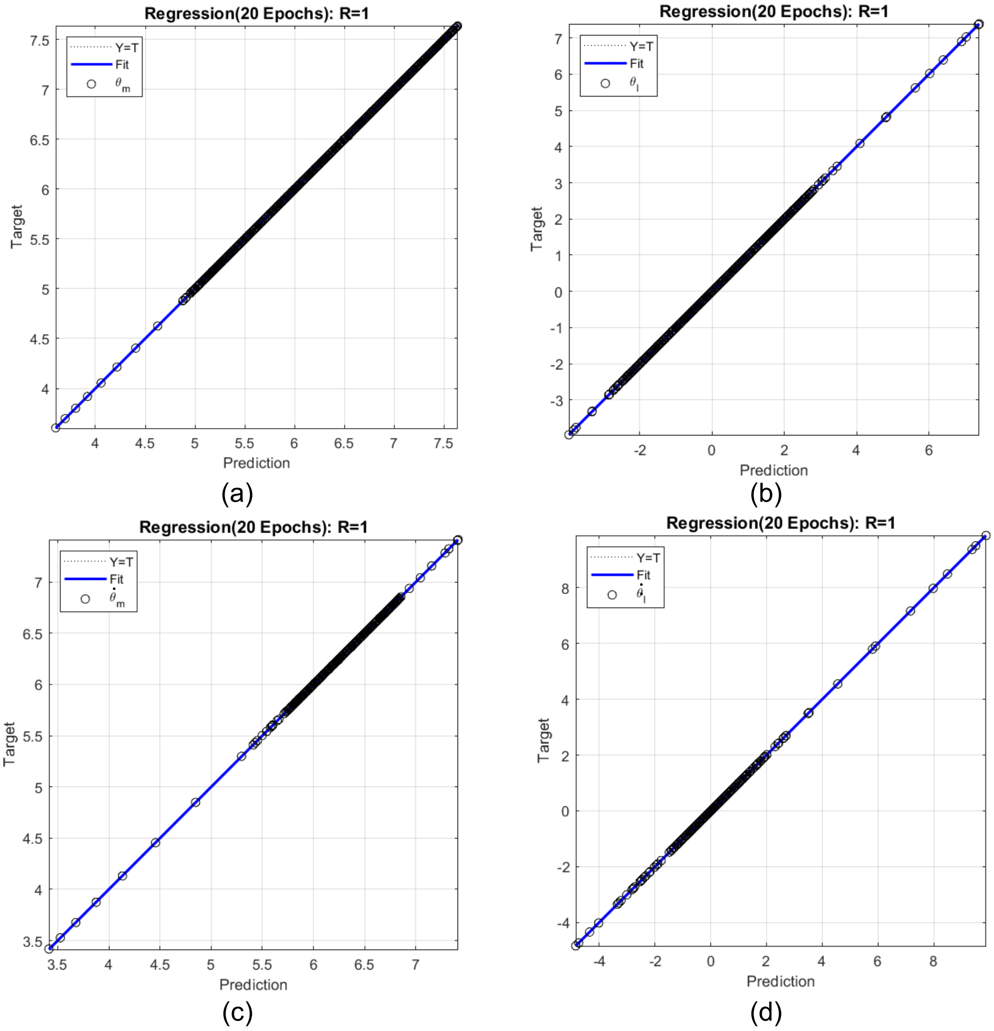
(a) (b)

(c) (d)

Figure 10 – R² for the estimated states in 10 Epochs for (a) Motor Position (b) Motor Velocity (c) Link Position (d) Link velocity

For the networks of 20 to 100 epochs, the R² values are equal to 1, and the regression graphs present slight variations in the estimated values. Therefore, Figure 11 presents the adjustment between the predicted values and those estimated by the network only for one of the cases in which the network has 20 epochs.

After analyzing the adjustment validation metrics, the network with 10 neurons and 20 epochs is chosen to demonstrate state estimation. Figures 12 and 13 compare the results of estimating the system states using the MHSE and the NNMHSE filter. The graph shows only 12 seconds of the 120 seconds of simulation for better data visualization, considering that are many samples for training the network. It is possible to observe that the NNMHSE effectively estimates the states, even considering the variations that occur every 2s, which represent the change in the manipulator's operation modes, always starting in the free operation mode and then passing to the contact operation mode.



(a) (b)

(c) (d)

Figure 11 – R² for the estimated states in 20 Epochs for (a) Motor Position (b) Motor Velocity (c) Link Position (d) Link velocity

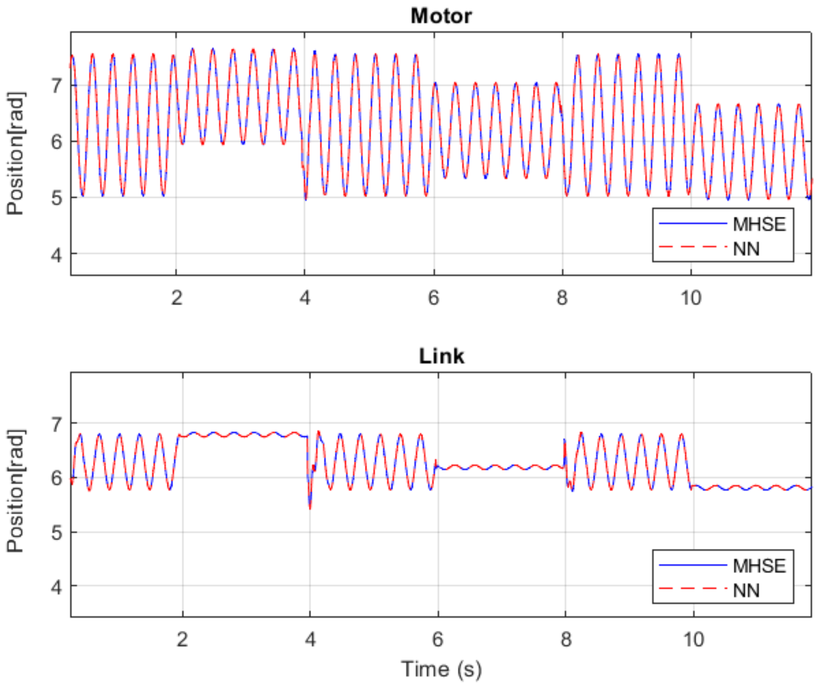


Figure 12 – Comparison between the State’s Estimation (Position) of the MHSE and NNMHSE methods.

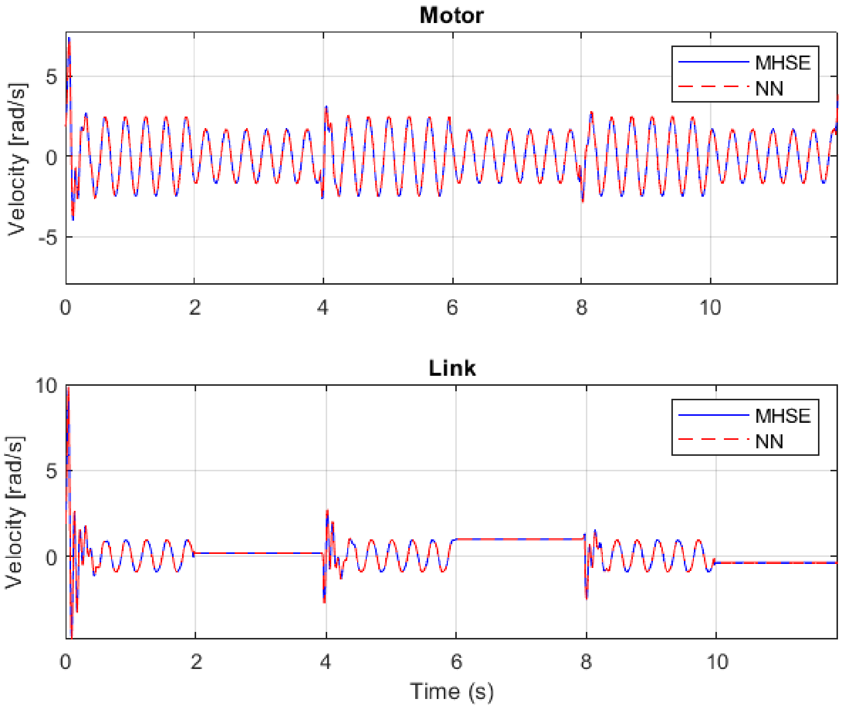


Figure 13 – Comparison between the State’s Estimation (Velocity) of the MHSE and NNMHSE methods.

Table 3 presents the network training processing time for estimating the mode with 10 neurons and 2 hidden layers. To verify the network adjustment for mode estimation R² value and the RMSE are also calculated for different epoch values. It is possible to observe that all estimates have a high R² value, above 0.9. Network training is fast because there are few variables to estimate the contact mode.

Table 3 – R² and RMSE for the Mode Approximation

Epochs	Time(s)	R²	RMSE
10	5.9454	0.9468	1.09E-01
20	10.8720	0.9798	6.84E-02
30	13.3603	0.9935	3.99E-02
40	17.0577	0.9995	1.07E-02
50	20.4367	0.9998	7.30E-03
60	24.4084	0.9986	1.89E-02
70	28.3328	0.9990	1.57E-02
80	31.4428	1.0000	2.80E-03
90	34.6771	1.0000	3.90E-03
100	38.5203	1.0000	1.30E-03

Figure 14 presents the R² value variation curve for each epoch used to train the mode estimation network.

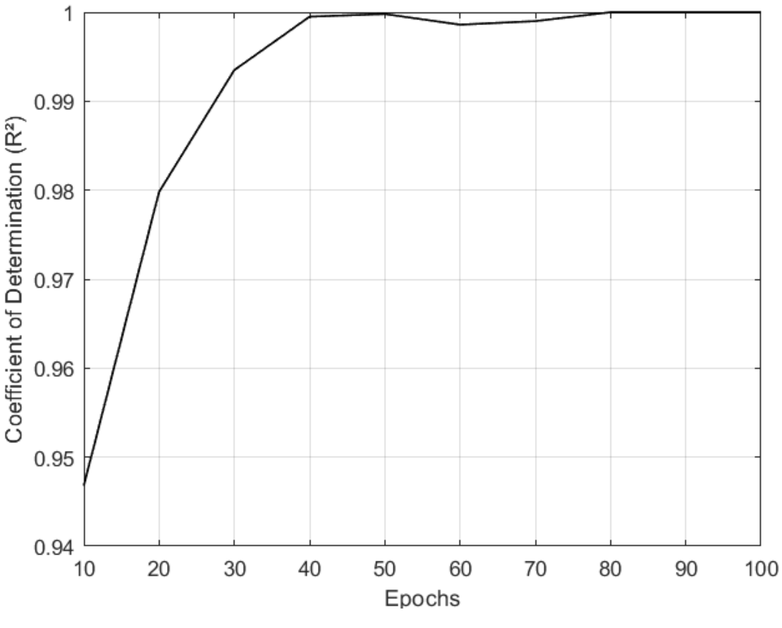


Figure 14 – R² for the Mode Approximation

After 80 Epochs, the Networks present a constant adjustment for the value of R² with a value equal to 1. Figure 15 presents the results of estimating the system modes, comparing the MHSE and the NNMHSE filter for this network. Mode 1 is equivalent to the manipulator in free movement, and mode 2 represents the manipulator in contact.

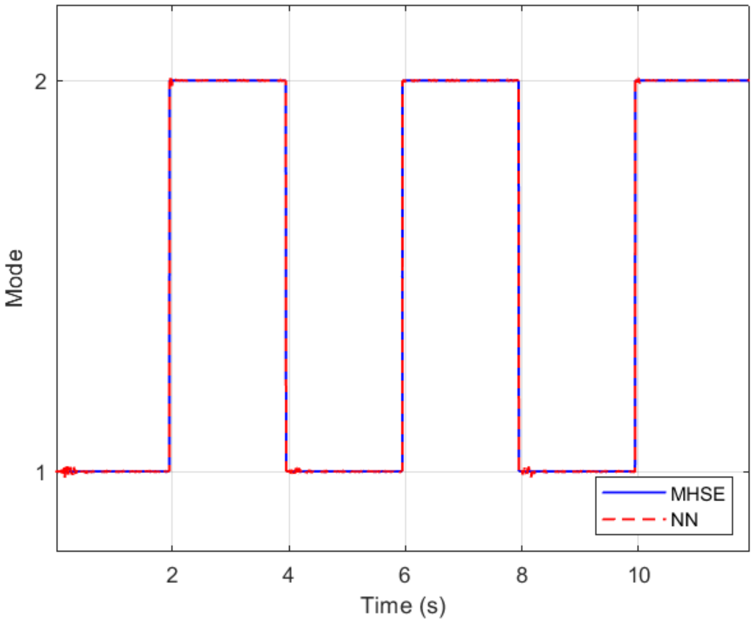


Figure 15 – Comparison between MHSE and the NNMHSE for the Mode Estimation

As this approximates the models, it is possible to observe that the estimation of the mode presents estimation failures, that is, because instead of the network finding only two values (1 and 2) referring to the system modes, the network finds approximate modes values and provides an efficient way to infer the modes. However, the main gain of applying the network is the reduction in the processing time of the estimation. Figure 16 compares the processing time per sample of the MHSE and the NNMHSE and shows that NNMHSE is on average almost four times faster than the MHSE.

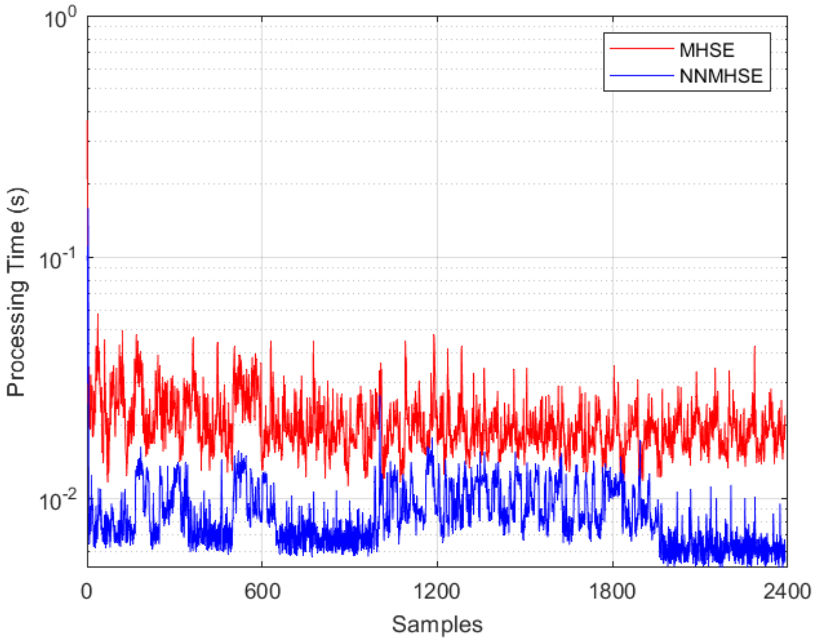


Figure 16 – Comparison between the MHSE and NNMHSE processing time.

4 Switching Nonlinear Model Predictive Position-Torque Control

In this chapter, concepts related to switching Model Predictive Control will be presented. Subsequently, the implementation results of the Nonlinear Model Predictive Control approach for the switching system.

4.1. Switching Model Predictive Control

According to (Liberzon, 2003), switched systems are continuous-time systems with isolated switched events that can be used to model plant systems with multiple modes and can be obtained from hybrid systems that use a family of controllers. They can be classified as state-dependent or time-dependent, autonomous (uncontrolled), or controlled, and can have various combinations of switching types.

In this dissertation the switching signal is classified as time-dependent (Liberzon,2003) and satisfies a minimal dwell time before switching for other control law, (Ong et al., 2016). So, the switching algorithm has a priori known of the switch in the mode over the simulation time. The equation for the nonlinear switched system is defined in eq. (37).

$$\begin{aligned} x_{\mathrm{t}+1} &= f_{\sigma}(x_t, u_t) \\ y_{\mathrm{t}} &= h_{\sigma}(x_t) \end{aligned} \tag{37}$$

where, x ∈ $\mathbb{R}^n$ is the input, $u \in \mathbb{R}^m$ is the control, and $y\ \in \mathbb{R}^p$ is the output, $t = 0,1, \ldots$ denotes discrete time and, $\sigma \in \{1,2 \ldots, n\}$ is a switching rule which takes its values in the finite set of subsystems. f and h are nonlinear functions of each subsystem. Unlike the MHSE, the MPC formulation in this dissertation does not consider the addition of noise (ε_t , η_t) to the system and measurements.

According to (Zhang et al., 2016), the switched MPC algorithm is configured for each subsystem. In this work, we consider two subsystems, corresponding to position and force control. The idea behind it is to use the MPC as a dual-mode controller that guarantees stability in the system when the switching between the control modes occurs. Figure 17 presents the switching MPC diagram.

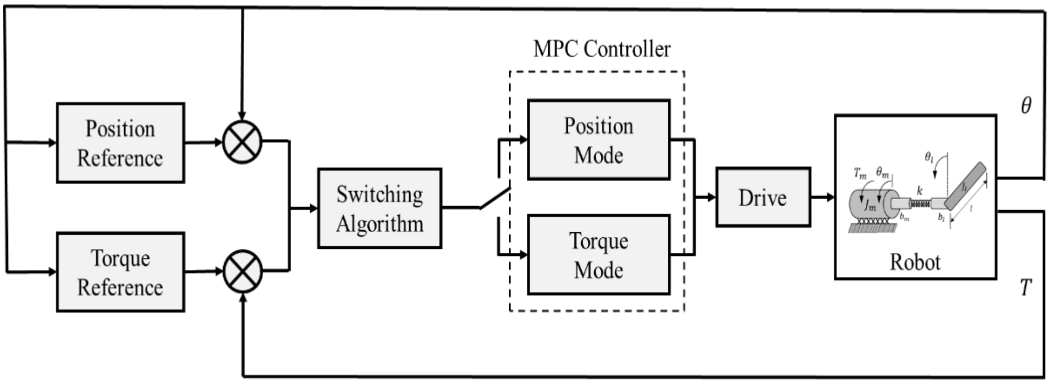


Figure 17 - Switching MPC diagram.

4.2. Control Scheme

The Model Predictive Control (MPC) uses a dynamic model of the system in the state-space form. The NMPC method was initially implemented for each control mode separately, making it possible, after trial and error, to adjust the weights for the cost function for each of the system modes. Subsequently, a single controller was developed, and a switching algorithm was created that considers the position and torque reference inputs and has as output the activation of the MPC controller to one of the modes based on the active states of the system. To facilitate the implementation of the NMPC, the optimization problem was solved using CasADi, which is an "open-source tool for nonlinear optimization, algorithmic differentiation, and optimal control" (Andersson et al., 2019). Summarizing the NMPC method, Figure 18 presents the nonlinear optimal control algorithm step by step.

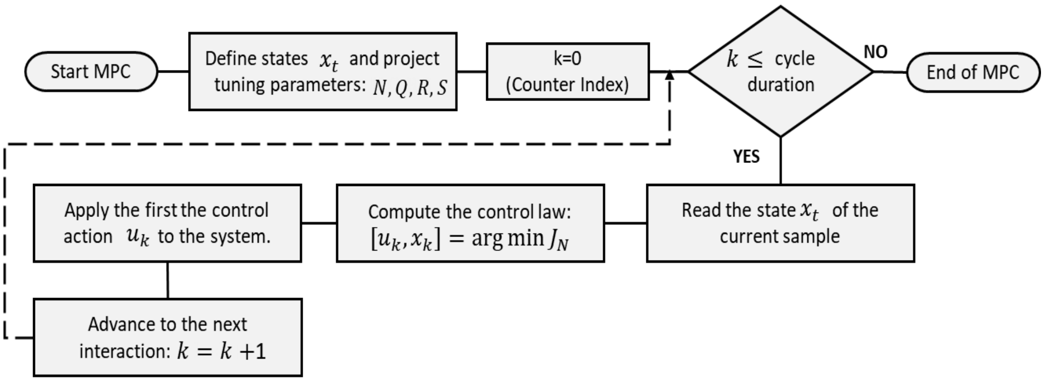

Start MPC
Define states x_t and project tuning parameters: N, Q, R, S
k=0 (Counter Index)
$k \leq$ cycle duration
NO
End of MPC
YES
Read the state x_t of the current sample
Compute the control law: $[u_k, x_k] = \arg\min J_N$
Apply the first the control action u_k to the system.
Advance to the next interaction: $k = k + 1$

Figure 18 - Step-by-step of the MPC implementation.

The control parameters are the physical boundaries of the states (x_{min}, x_{max}), the input boundaries ($u_{min}, u_{max}, \Delta u$), the prediction horizon (N), The weighting matrix of outputs (Q_1 e Q_2), the weighting matrix of inputs (R_1 e R_2), the weighting matrix of the input variation (S_1 e S_2), representing (1 and 2) each system mode. Table 4 presents the control parameters, and the system parameters are in subsection 2.3.1. The limits of the control actions are defined within the MPC algorithm according to Brunello (2021). The weighting matrix is defined through trials and errors, starting with the Q parameter, then the S parameter, and finally the R. In this work, a bigger penalization is given to the outputs (Q1 e Q2) to ponder the importance of the trajectory.

Table 4 – NMPC Control Parameters

Parameters	Values	Units
x_{min}	$-\mathrm{inf}$	rad
x_{max}	inf	rad
u_{min}	-50	$\mathrm{N \cdot m}$
u_{max}	50	$\mathrm{N \cdot m}$
N	10	$-$
Q_1	$1e9$	$-$
R_1	$1e1$	$-$
S_1	$1e3$	$-$
Q_2	$1e8$	$-$
R_2	0	$-$
S_2	$1e4$	$-$

4.3. NMPC Results

The NMPC problem was solved offline by using MATLAB software with a sampling time $T_S = 0.05s$ and a prediction horizon N=10. The simulation was running on a laptop computer with an Intel Core i5 at 2.4GHz with 8-GB RAM. The MPC optimization problem was solved through the CasADi symbolic toolbox, using the constrained interior-point algorithm with a tolerance of 10^{-5} for the stopping criteria. The control input signal is the torque by the motor $T_m = k_t * u$, where $u = 10 + \sin{(t)}$. Figure 19 presents the time evolution of the input torque and the spring torque over the simulation. It is observed that the necessary control action is below the maximum control limit defined in Table 4, which means that the MPC manages to find a solution without saturating the actuator.

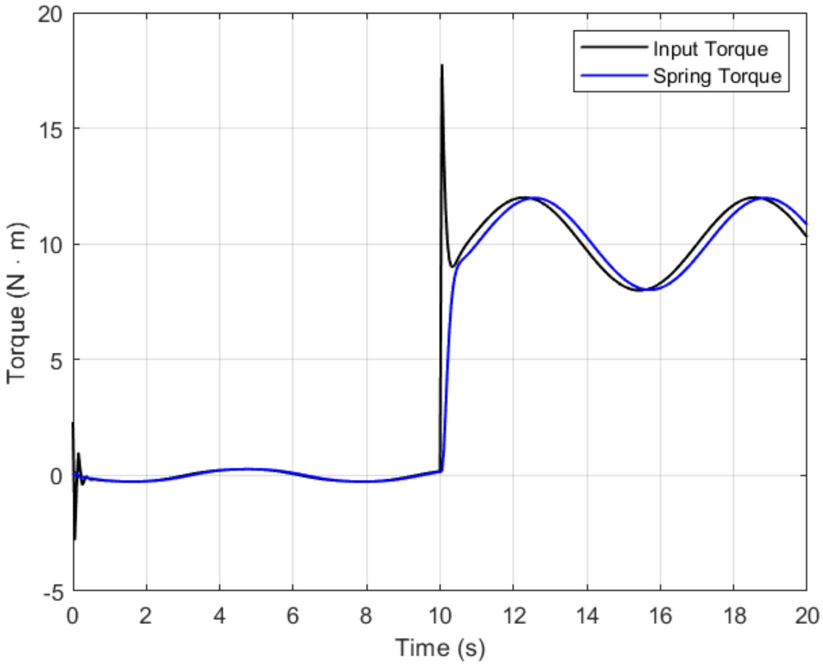


Figure 19 - Torque variation during switching control.

The state initial condition was defined as $x_0 = [\pi, 0, \pi, 0]^T$. The switched control considered two modes of simulation, (1) Position mode (0-10s) and (2) Contact Mode (10-20s). The switching variable has been chosen as time. In position mode, it is considered that MPC runs to control only position, and in contact mode, the torque is controlled to assure a desired force of contact. Figure 20 presents the switching of the control modes.

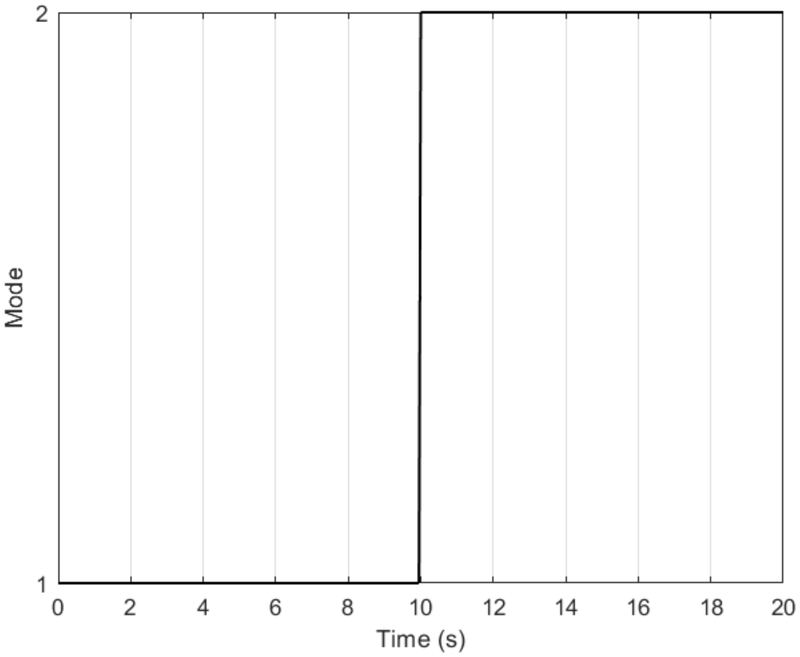


Figure 20 - Switching Control Modes

Figure 21 presents the path-tracking result for position and torque, which compares the desired reference trajectory and the obtained using the MPC framework. The error between the reference and the real prediction of the position and torque control over the simulation is also present in Figure 21.

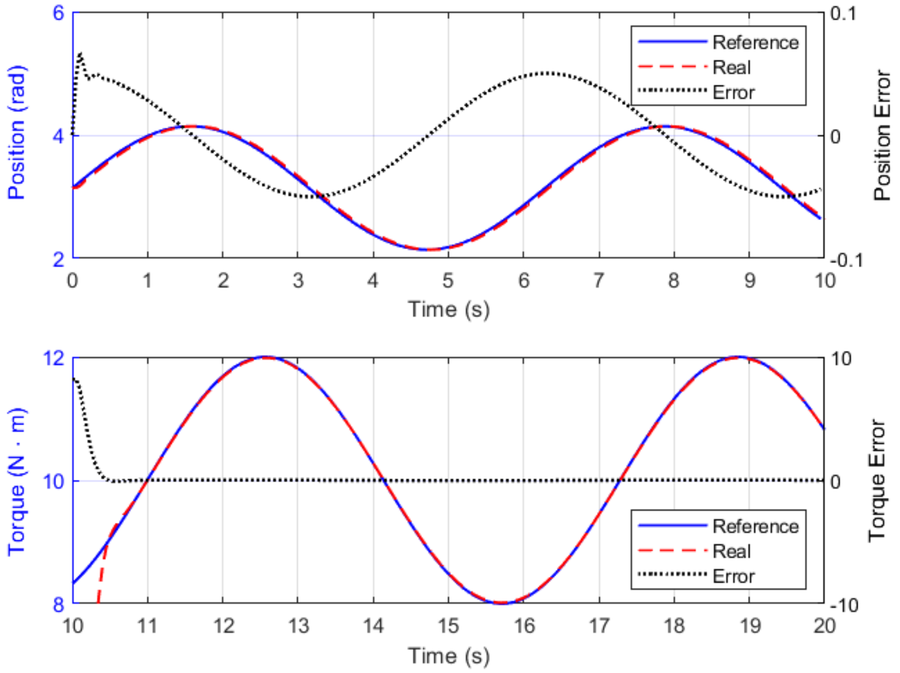


Figure 21 - Position and Torque control (Reference vs Real vs Error)

A very precise tracking can be observed in Figure 21 when the MPC control is applied. It is possible to observe that in the position mode, we have the spring torque following the motor input torque acting without much variation, but in the transition to the contact mode, this variation becomes much more significant. Despite this, the control quickly manages to reach the reference torque, which indicates the effectiveness of the controller.

To measure the error between the desired trajectory and the real trajectory of the controlled system, the error was calculated as in eq. (38).

$$Error = Z_{Ref} - Z_{Real} \tag{38}$$

where Z_{Ref} represent the reference position or reference torque, and Z_{Real} represent the real position or real torque. Despite the high error presented at the beginning of the Contact mode (10s), of almost 98% in the reference tracking for less than 1 second, which represents the transition between the position mode and the torque mode, analyzing the low percentage error over the simulation, approximately 2%

during positions control mode and 0.5% during the torque control mode, we may evaluate that the proposed controller is effective.

Figure 22 presents the algorithm processing time to solve the optimization problem and generate the control action during the simulation. The average value of the optimization sampling time is 0.0173s, almost three times faster than the simulation sampling time. Therefore, it is possible to conclude that the algorithm can be used on online applications.

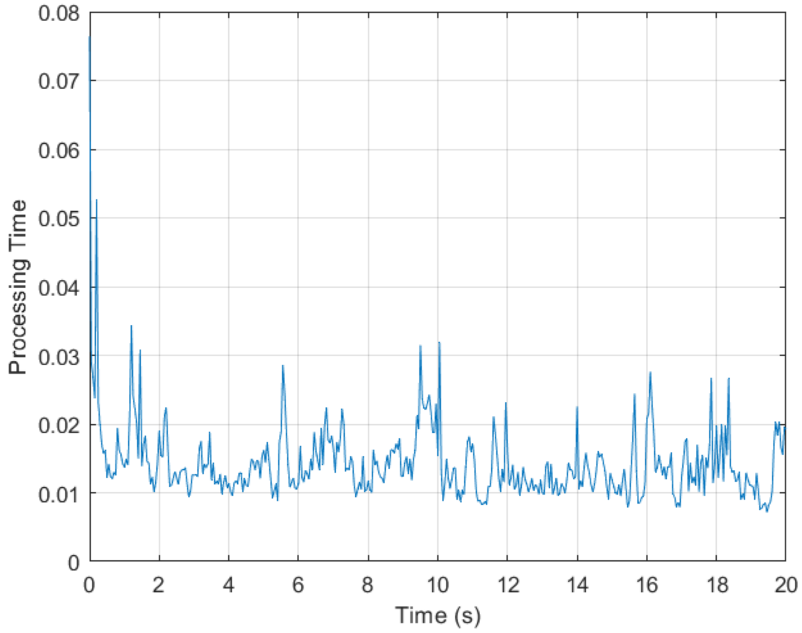


Figure 22– MPC optimization processing time

A second simulation was performed, with $x_0 = [0, 0, 0, 0]^T$, to check the controller tracking for the possible mode transitions (position to force; force to position; position to position), considering a cyclical movement of the robotic manipulator, that is, varying the link angles from 0-π/2 radians within a period of 15s, where the manipulator raises the link in 5s, remains stationary for 5s, then lowers the link in 5s. Figure 23 and Figure 24 present the variation of control modes and the variation of input torque and spring torque, respectively, for the cyclic simulation.

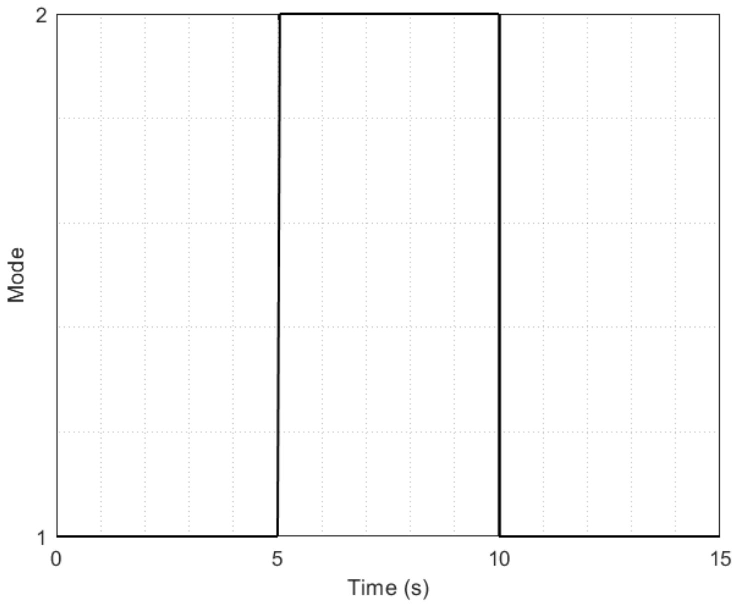


Figure 23 - Switching Control Modes in the cyclic simulation.

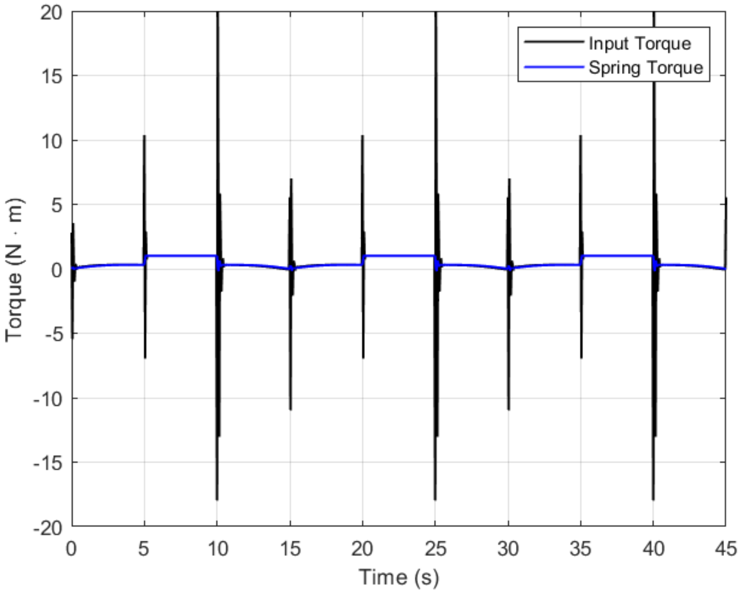


Figure 24 - Torque variation during switching control.

Figure 25 presents the MPC tracking for a simulation of three cycles of the robotic manipulator performing the same task. At each cycle, the manipulator starts the simulation in position control mode, raising the link until reaching the point where contact occurs (π/2 rad), then the controller switches to torque control mode so that the manipulator exerts a desired torque of $1\ N \cdot m$ over a period of 5s and finally returns to position control mode by lowering the link to the initial angle (0 rad).

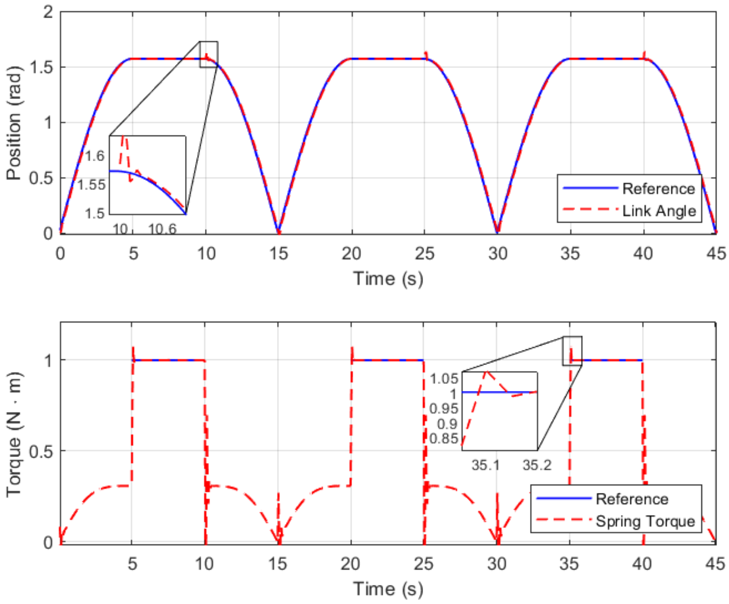


Figure 25 - Position and Torque control (Reference vs Real)

The control law obtained can perform trajectory control accurately even with the increase in the complexity of the trajectory reference. The trajectory performed by the MPC controller remains efficient for both control modes and their possible transitions, showing low error in transition instants, 0.1 rad above the position reference and, 0.05 N·m above the torque reference, for less than 1s. For the second case, the processing time of the MPC optimization algorithm, on average 0.0108s, almost five times faster than the simulation sampling time, also proves to be sufficient for online applications, as presented in Figure 26.

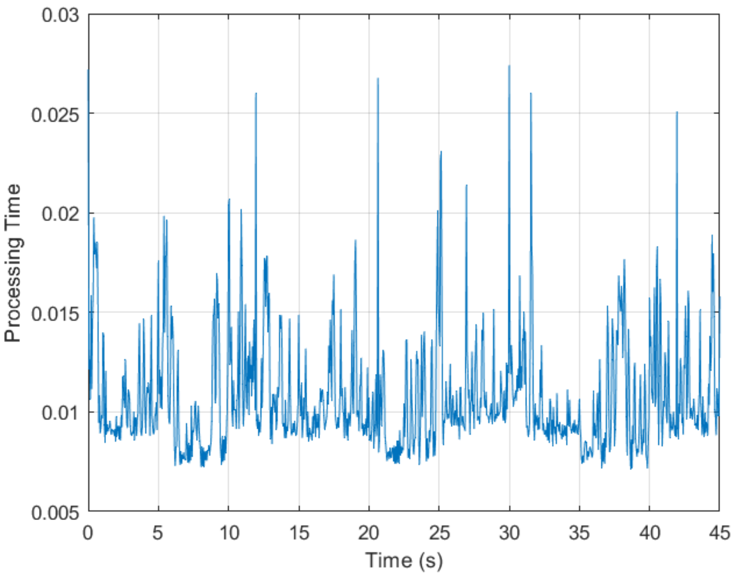


Figure 26 – MPC optimization processing time (cyclic simulation)

5
Conclusion and Future Work

In the present dissertation, we showed the application of switching nonlinear MPC by simulating an offline closed-loop system and a switching approximative filter for MHSE.

The NNMHE approximates well the estimation model presenting adequate results for the states and modes estimation and reduces the average processing time to four times faster than the MHSE processing time. Despite presenting good results, with low error values in the estimation of the states on average in the scale of 10^{-4} and, in the mode estimation 10^{-2}, an average fit of 99% for the NNMHSE network in the function of the model of the MHSE data, for evaluating the effectiveness of the method, it is necessary to test the functioning of the network for noisy data and compare the method with other estimators.

The switching NMPC controller was tested in two cases to illustrate the effectiveness of the proposed method. In both cases, the obtained controller ensures control precision for both position and torque control, proving effectiveness in following desired trajectory and torque references. The controller was also able to respect the established control action restrictions and with a processing time of the MPC algorithm, on average five times faster than the sampling time, indicating that the algorithm enables online applications. These results demonstrate that the controller has the potential to be applied to systems that require safety in physical Human-Robot interaction (pHRI).

In the future, we expect to advance the simulation by increasing the complexity of the simulation model by adding friction models and validating by testing the method in the test bench presented in Lopes (2021, 2022) and comparing the experimental results of the MPC control strategy for switched manipulators with the strategy presented by Li (2018).

We also expect to apply the MPC approach developed in the present work with the moving horizon estimation (MHE) approach by using the estimation concept to predict when the mode switching occurs and then activate the appropriate control mode in the MPC.

6
Bibliography

Alessandri, A., Baglietto, M., Battistelli, G., & Zavala, V. (2010, December). **Advances in moving horizon estimation for nonlinear systems**. In 49th IEEE Conference on Decision and Control (CDC) (pp. 5681-5688). IEEE.

Alessandri, A., Baglietto, M., & Battistelli, G. **Receding-horizon estimation for switching discrete-time linear systems**. IEEE Transactions on Automatic Control, 50(11), 1736–1748, 2005.

Andersson, J. A., Gillis, J., Horn, G., Rawlings, J. B., & Diehl, M. (2019). **CasADi: a software framework for nonlinear optimization and optimal control**. Mathematical Programming Computation, 11(1), 1-36. doi: 10.1007/s12532-018-0139-4

Ayala, H., Sampaio, R., Muñoz, D. M., Llanos, C., Coelho, L., & Jacobi, R. (2016, June). **Nonlinear model predictive control hardware implementation with custom-precision floating point operations**. In 2016 24th Mediterranean Conference on Control and Automation (MED) (pp. 135-140). IEEE.

Baglietto, M., Battistelli, G., Ayala, H. V. H., & Tesi, P. **Mode-observability conditions for linear and nonlinear systems**. In 2012 IEEE 51st IEEE Conference on Decision and Control (CDC) (pp.1941-1947). IEEE. 2012.

Bi, Z. M., Luo, C., Miao, Z., Zhang, B., Zhang, W. J., & Wang, L. (2021). **Safety assurance mechanisms of collaborative robotic systems in manufacturing**. Robotics and Computer-Integrated Manufacturing, 67, 102022.

Brunello, Rafael Koji Vatanabe. **Nonlinear moving-horizon state estimation for hardware implementation and a model predictive control application**. 2021. 69 f., il. Dissertação (Mestrado em Sistemas Mecatrônicos) — Universidade de Brasília, Brasília, (2021).

Colgate, E., Bicchi, A., Peshkin, M. A., & Colgate, J. E. **Safety for physical human-robot interaction**. In Springer Handbook of Robotics (pp. 1335-1348). Springer. (2008).

Du, Y., Liu, F., Qiu, J., & Buss, M. (2021). **A novel recursive approach for online identification of continuous-time switched nonlinear systems**. International Journal of Robust and Nonlinear Control, 31(15), 7546-7565.

Fan, X., & Arcak, M. **Observer design for systems with multivariable monotone nonlinearities**. Systems & Control Letters, 50(4), 319-330, 2003.

Grüne, L., Pannek, J. **Nonlinear Model Predictive Control. In: Nonlinear Model Predictive Control**. Communications and Control Engineering. Springer, London, (2011). https://doi.org/10.1007/978-0-85729-501-9_3.

Guo, Y., & Huang, B. (2013). **Moving horizon estimation for switching nonlinear systems**. Automatica, 49(11), 3270-3281.

Haddadin, S. **Towards safe robots: approaching Asimov's 1st law** (Vol. 90). Springer. (2013).

Haddadin, S., De Luca, A., & Albu-Schäffer, A. (2017). **Robot collisions: A survey on detection, isolation, and identification**. IEEE Transactions on Robotics, 33(6), 1292-1312.

Ibrahim, K., & Sharkawy, A. B. (2018). **A hybrid PID control scheme for flexible joint manipulators and a comparison with sliding mode control**. Ain Shams Engineering Journal, 9(4), 3451-3457.

Katayama, S., Murooka, M., & Tazaki, Y. **Model predictive control of legged and humanoid robots: models and algorithms**. Advanced Robotics, 1-18, (2023).

Kluever, C. A. **Dynamic Systems: Modeling, Simulation, and Control**. John Wiley & Sons, 2015.

Leica, P., Toibero, J.M., Roberti, F. et al. **Switched Control to Robot-Human Bilateral Interaction for Guiding People**. J Intell Robot Syst 77, 73–93 (2015). https://doi.org/10.1007/s10846-014-0098-6

Liberzon, D. **Switching in Systems and Control**. Birkhauser, 2003.

Li, Z. J., Wu, H. B., Yang, J. M., Wang, M. H., & Ye, J. H. **A position and torque switching control method for robot collision safety**. International Journal of Automation and Computing, 15(2), 156-168, (2018).

Liu, Y., He, N., He, L., Zhang, Y., Xi, K., & Zhang, M. **Self-Tuning of MPC Controller for Mobile Robot Path Tracking Based on Machine Learning**. Journal of Shanghai Jiaotong University (Science), 1-9, (2022).

Lopes, Felipe Rebelo; Meggiolaro, Marco Antonio. **Design of a Low-Cost Series Elastic Actuator for Application in Robotic Manipulators**. In: 26th internation Congress of Mechanical Engineering. ABCM. 2021.

Lopes, Felipe Rebelo; Meggiolaro, Marco Antonio (Advisor); Vicente Hultmann Ayala, Helon (Co-Advisor). **Design and control of an elastomer-based elastic coupling for SEA**. Rio de Janeiro, 2022. 126p. Tese de Doutorado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Magrini, E., Ferraguti, F., Ronga, A. J., Pini, F., De Luca, A., & Leali, F. (2020). **Human-robot coexistence and interaction in open industrial cells**. Robotics and Computer-Integrated Manufacturing, 61, 101846.

Nocedal, J., & Wright, S. (2006). **Springer series in operations research and financial engineering**. **In Numerical optimization**. New York: Springer.

Nubert, J., Köhler, J., Berenz, V., Allgöwer, F., & Trimpe, S. **Safe and fast tracking on a robot manipulator: Robust mpc and neural network control**. IEEE Robotics and Automation Letters, 5(2), 3050-3057. (2020).

Ong, C. J., Wang, Z., & Dehghan, M. **Model predictive control for switching systems with dwell-time restriction**. IEEE Transactions on Automatic Control, 61(12), 4189-4195, 2016.

Pankert, J., & Hutter, M. (2020). **Perceptive model predictive control for continuous mobile manipulation**. IEEE Robotics and Automation Letters, 5(4), 6177-6184.

Ray, S. "**A Quick Review of Machine Learning Algorithms**," *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Faridabad, India, 2019, pp. 35-39, doi: 10.1109/COMITCon.2019.8862451.

Spiess, AN., Neumeyer, N. **An evaluation of R2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: a Monte Carlo approach**. BMC Pharmacol 10, 6 (2010). https://doi.org/10.1186/1471-2210-10-6

Vargas, Alessandro N.; Carubtu, Constantin F.; ISHIHARA, João Y. **Stability of switching linear systems with switching signals driven by stochastic processes**. Journal of the Franklin Institute, v. 356, n. 1, p. 31-41, 2019.

Vorndamme, J., Schappler, M., & Haddadin, S. (2017, May). **Collision detection, isolation, and identification for humanoids**. In 2017 IEEE International Conference on Robotics and Automation (ICRA) (pp. 4754-4761). IEEE.

Wang, Z., Zou, L., Su, X., Luo, G., Li, R., & Huang, Y. (2021). **Hybrid force/position control in workspace of a robotic manipulator in**

uncertain environments based on adaptive fuzzy control. Robotics and Autonomous Systems, 145, 103870.

Wachter, A. **An interior point algorithm for large-scale nonlinear optimization with applications in process engineering**. 2002. 229. (Order No. 3040516) - Carnegie Mellon University, United States -- Pennsylvania, 2002.

Wachter, Andreas. **Short tutorial: Getting started with ipopt in 90 minutes**. In: Dagstuhl Seminar Proceedings. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2009.

Xuan, B. D. (2020). **Dynamics and control analysis of a single flexible link robot with translational joints**. VNUHCM Journal of Engineering and Technology, 3(4), 588-595.

Yu, L., Huang, J. & Fei, S. **Sliding Mode Switching Control of Manipulators Based on Disturbance Observer**. Circuits Syst Signal Process 36, 2574–2585 (2017). https://doi.org/10.1007/s00034-016-0421-5

Zacharaki, A., Kostavelis, I., Gasteratos, A., & Dokas, I. (2020). **Safety bounds in human-robot interaction: A survey**. Safety Science, 127, 104667.

Zavala, V. M., & Biegler, L. T. (2009). **The advanced-step NMPC controller: Optimality, stability, and robustness**. Automatica, 45(1), 86-93.

Zhang, L., Zhuang, S., & Braatz, R. D. **Switched model predictive control of switched linear systems: Feasibility, stability, and robustness**. Automatica, 67, 8-21, 2016.

Zhang, X., Polycarpou, M. M., Parisini, T. **Fault diagnosis of a class of nonlinear uncertain systems with Lipschitz nonlinearities using adaptive estimation**. Automatica, v. 46, n. 2, p. 290-299, 2010.