ROBUST MODEL PREDICTIVE CONTROL FOR LINEAR PARAMETER VARYING SYSTEMS ALONG WITH EXPLORATION OF ITS APPLICATION IN MEDICAL MOBILE ROBOTS

A Thesis Submitted to
the College of Graduate and Postdoctoral Studies
In Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy
In the Division of Biomedical Engineering
University of Saskatchewan
Saskatoon

By
Mohsen Hadian

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Chair of the Division of Biomedical Engineering
University of Saskatchewan
57 Campus Drive
Saskatoon, Saskatchewan, S7N 5A9, Canada

OR

Dean of the College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan, S7N 5C9, Canada

ABSTRACT

This thesis seeks to develop a robust model predictive controller (MPC) for Linear Parameter Varying (LPV) systems. LPV models based on input-output display are employed. We aim to improve robust MPC methods for LPV systems with an input-output display. This improvement will be examined from two perspectives. First, the system must be stable in conditions of uncertainty (in signal scheduling or due to disturbance) and perform well in both tracking and regulation problems. Secondly, the proposed method should be practical, i.e., it should have a reasonable computational load and not be conservative.

Firstly, an interpolation approach is utilized to minimize the conservativeness of the MPC. The controller is calculated as a linear combination of a set of offline predefined control laws. The coefficients of these offline controllers are derived from a real-time optimization problem. The control gains are determined to ensure stability and increase the terminal set.

Secondly, in order to test the system's robustness to external disturbances, a free control move was added to the control law. Also, a Recurrent Neural Network (RNN) algorithm is applied for online optimization, showing that this optimization method has better speed and accuracy than traditional algorithms. The proposed controller was compared with two methods (robust MPC and MPC with LPV model based on input-output) in reference tracking and disturbance rejection scenarios. It was shown that the proposed method works well in both parts. However, two other methods could not deal with the disturbance.

Thirdly, a support vector machine was introduced to identify the input-output LPV model to estimate the output. The estimated model was compared with the actual nonlinear system outputs, and the identification was shown to be effective. As a consequence, the controller can accurately follow the reference.

Finally, an interpolation-based MPC with free control moves is implemented for a wheeled mobile robot in a hospital setting, where an RNN solves the online optimization problem. The controller was compared with a robust MPC and MPC-LPV in reference tracking, disturbance rejection, online computational load, and region of attraction. The results indicate that our proposed method surpasses and can navigate quickly and reliably while avoiding obstacles.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Chris Zhang. Your unwavering support, guidance, and patience have been invaluable throughout my academic journey. Your dedication to excellence and commitment to mentoring me has had a profound impact on my personal and professional growth. I am forever indebted to you for your exceptional guidance, insightful feedback, and meticulous editing, which have been instrumental in shaping this thesis.

I would also like to extend my appreciation to my committee members, Dr. Francis Bui, Dr. Fang Xiang Wu, and Dr. Daniel Chen, for their invaluable feedback and guidance. Their expertise, encouragement, and insightful comments have been pivotal in the development of my thesis. I am truly grateful for their unwavering support, and I am fortunate to have had such a knowledgeable and dedicated committee to guide me throughout the process.

I would like to thank my parents for their unwavering love and support. Your encouragement, motivation, and sacrifices have been the driving force behind my academic achievements. I am forever grateful for the values you instilled in me, and I am honoured to have had such amazing role models throughout my life.

Finally, I would like to thank all of my colleagues, friends, and family members who have supported me throughout this journey. Your encouragement and support have been a source of inspiration for me, and I am grateful for your presence in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviations	Definition
ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial Neural Network
FBL	Feedback Linearization
FOV	Field Of View
IBVS	Image-Based Visual Servoing
IO	Input-Output
LMI	linear matrix inequality
LPV	linear parameter varying
MAS	Multi-Agent Systems
MIMO	multiple-input multiple-output
MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control
PBVS	Position-Based Visual Servoing
PID	Proportional Integral Derivative
PSO	Particle Swarm Optimization
QP	Quadratic Programming
RNN	recurrent neural network
SMC	Sliding Mode Control
SS	State Space
TMPC	Tube-based MPC
WMR	Wheeled Mobile Robot

CHAPTER 1

INTRODUCTION

This chapter reviews the basic concepts of MPC and LPV systems and then studies research on MPC for LPV systems. In the end, we explain the objectives.

1.1 Model Predictive Control (MPC)

This section describes the MPC in detail, its elements, and its different types. In addition, we explain the advantages and challenges of this method. Finally, we describe the nonlinear MPC.

1.1.1 Principles

The primary purpose of MPC design is to calculate a sequence of future system inputs at each sampling time to optimize its future behaviour. This optimization is carried out over a specific time period and with the system information available at the start of that time period. The algorithm used in this controller can be seen in Figure 1.1.

i. At any sampling time t, the future system outputs on the horizon N_p are estimated using the system model. To show the predicted output values at the moment $t + k$ using the information up to moment t, we use the following display:

$$y(t + k|t) \qquad k = 1, \dots, N_p \tag{1.1}$$

These values will be obtained based on the past values of input and output up to the moment t as well as the future values of inputs.

ii. We show the sequence of future input signals as follows:

$$u(t + k|t) \qquad k = 1, \dots, N_c \tag{1.2}$$

where N_c is called the control horizon. In Figure 1.1, $N_p = N$ and $N_c < N$.

Control signals are optimized by a cost function to approximate future outputs to be as close to the reference signal $w(t+k|t)$ as possible. A quadratic function of the error between the predicted and desired outputs along the prediction horizon is defined. In many cases, this cost function also considers the amount of control effort (amplitude or amount of control signal changes). There is an explicit solution to the quadratic cost function for an unconstrained linear system; otherwise, implicit methods will be used. In most applications, the sequence of control inputs is considered to be fixed from a certain point onwards. In fact, the prediction horizon is always considered to be greater than or equal to the control horizon.

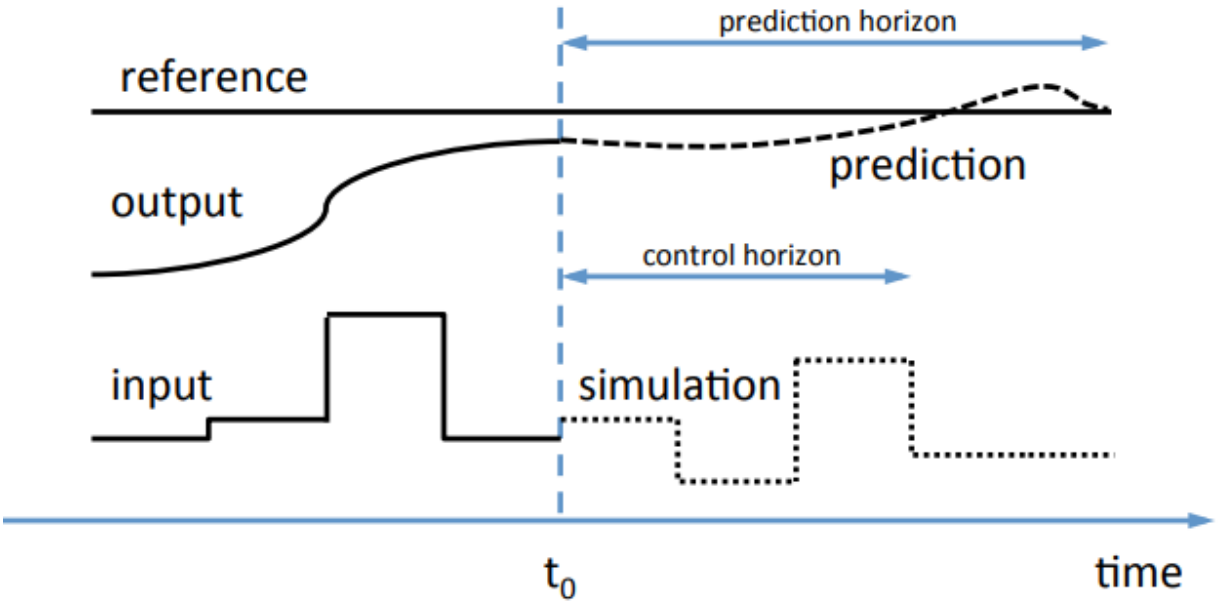


Figure 1.1: MPC strategy [1]

iii. Eventually, only the control signal $u(t|t)$ will be applied to the system, and the rest of the calculated control signals will not be used. The value of $y\ (t\ +\ 1)$ is known at the next sampling, and step 1 will be repeated with this new value to give a more accurate answer. As a result, $u(t+1|t+1)$is calculated that may be different from $u(t+1|t)$ due to new information.

The main structure of MPC is shown in Figure 1.2. The model is used to predict the future outputs of the system based on past and present values of outputs and future control actions. These actions are determined by the optimizer, considering the cost function (in which the tracking error is considered) and the constraints [2]:

1. Explicit process model used to predict future system outputs over a prediction horizon.

2. Calculate future control signals by minimizing a cost function based on the difference between the future outputs of the system (derived from the prediction model) and the desired values.

3. Apply the first optimally calculated control signal to the system and repeat the entire prediction cycle and optimization.

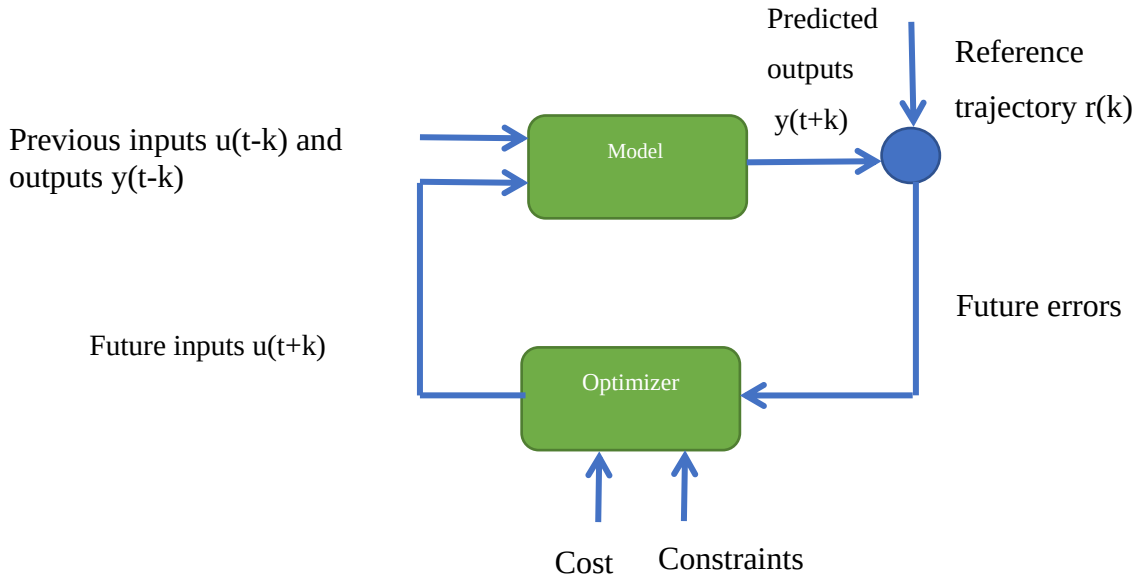


Figure 1.2: MPC structure

1.1.1 Prediction model block

The process prediction model is presented in Figure 1.2, which is responsible for predicting output over the prediction horizon. According to the process model, linear MPC can be divided into four categories, as indicated in Table 1.1.

In the first group, h is the output of the process when the input is an impulse signal. Likewise, s is the output of the system when the input is a step signal. In transfer function models, z^{-1} is a backward shift operator such that

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n_a} z^{-n_a} \tag{1.3}$$

$$B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_b} z^{-n_b}$$

In addition to the models described in Table 1.1, the system can be characterized by nonlinear models (principle-based or data-based). One of the conventional principle-based nonlinear models is the state-space presentation. Artificial neural networks, support vector

machines, and fuzzy logic can be employed for data-based classes. A disturbance model can also be added to the process model to deal with noise, unmodeled parameters, and model mismatch. This model is generally described by $n(k) = \frac{1}{1-z^{-1}} e(k)$, where e denotes the error.

Table 1.1: MPC classification according to the prediction model [2]

form	model	upside	downside
Impulse response	$y(k) = \sum_{i=1}^{N} h_i \, u(k-i)$	No previous information A process with delay and nonminimum phase	Stable process
Step response	$y(k) = y_0 + \sum_{i=1}^{N} s_i \, \Delta u(k-i)$	No previous information A process with delay and nonminimum phase	Stable process
Transfer function	$A(z^{-1})y(k) = B(z^{-1})u(k)$	unstable process	Previous information
State-space	$x(k) = Ax(k-1) + Bu(k-1)$ $y(k) = Cx(k)$	MIMO process	Unavailable states

1.1.2 Optimizer block

The optimizer block determines the optimal control input at a given control horizon. After selecting the appropriate process model in the prediction block, the system's future outputs are calculated and provided for this block. Inputs of one or more steps ahead ($u(k+j)$) or a fitting of them, as stated in (1.4), based on a series of basis functions, are regarded as inputs to the system.

$$u(k+j) = r_0 + r_1 C(k) + r_2 C(k)^2 + \cdots \quad (1.4)$$

Where C are basis functions. Calculating the optimal input sequence is mainly done by defining a cost function. Different criteria can be considered, such as the output error, the extent of the input and output signals, the input error, and the deviations from the input/output constraints.

1.1.2.1 Cost function

The ultimate goal of the cost function is that future states (X) on a given horizon track the specific reference signal W, while minimizing the control effort (Δu) required to do so. The general expansion for such an objective function would be as follows:

$$J(N_c, N_p) = \sum_{i=1}^{N_p} \delta(t)[X(k+i|t) - W(k+i)]^2 + \sum_{i=1}^{N_c} \lambda(t)\Delta u(k+i-1)^2 \quad (1.5)$$

In some cases, the second sentence of the above equation (which represents the control effort) is overlooked or has a small coefficient relative to the reference signal tracking, resulting in an unrestricted (or hardly limited) extent of the control effort signal. The components of the cost function are defined as follows:

- Parameters: N_p is the prediction horizon, specifying the samples on which the output tracks the reference signal, N_c is the control horizon, and $\delta(t)$ and $\lambda(t)$ are weighting matrixes. Various design parameters such as prediction horizons, control horizons, and weight matrices in the standard function should be adjusted according to the dynamics and behaviour of the system and control objectives. For example, instead of constant values for $\delta(t)$, it can change exponentially like 2^{N_p-i}.
- Reference path W: One of the benefits of MPC is that if future reference changes are known in advance, the system can react before a change occurs in the reference, thus eliminating the effects of process response delays. Even if the reference varies, a significant improvement in controller performance can be achieved by knowing the moment its value changes. The reference path $w\ (t\ +\ k)$ can differ from the original reference. It can be a soft approximation of the current output value $y(t)$ to a known reference using the first-order system:

$$\begin{gathered} w(t) = y(t) \\ w(t+k) = \alpha w(t+k-1) + (1-\alpha) r(t+N) \qquad k = 1, \dots, N \end{gathered} \quad (1.6)$$

α is a parameter between 0 and 1, which is an adjustable value that affects the dynamic response of the system. Figure 1.3 shows a path in which the reference $r\ (t\ +\ k)$ is constant and plotted for two different values of α. The small values of this parameter provide fast tracking response (w_1), and for larger α, the reference path is w_2,which makes the response smoother.

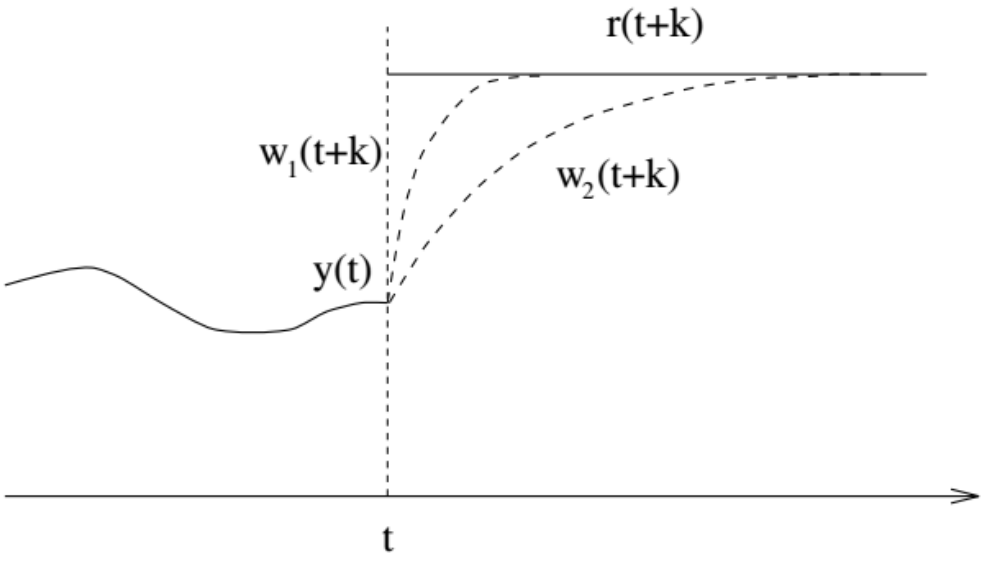


Figure 1.3: Reference path [2]

- Constraints: In practice, all systems have limitations, namely the extent and rate of change of control signal and output are considered as follows:

$$u_{min} \leq u(t) \leq u_{max} \tag{1.7}$$
$$\Delta u_{min} \leq u(t) - u(t-1) \leq \Delta u_{max}$$
$$y_{min} \leq y(t) \leq y_{max}$$

Adding these constraints to the objective function will make the optimization problem more complicated than the case without constraints [3].

1.1.2 Advantages and disadvantages

MPC has several advantages over other controllers, as follows:

- Using this controller usually does not require much control information. The controller is straightforward to set up.
- This controller can control a wide range of systems, from systems with simple dynamics to systems with complex structures, long latencies, non-minimum phases, and instability.
- This controller can be easily used for MIMO systems.
- Intrinsically introduce a leading path controller to compensate for unmeasured disturbances.
- This controller can be easily generalized to meet system constraints.

- It is convenient and effective when the desired future values are specific (robots or batch processes)

However, this controller has some limitations:

- If the dynamics change during the process, the computational volume increases significantly.
- Applying constraints complicates optimization operations.
- The basis of this algorithm is based on an appropriate model of the actual process; the difference between the model and process can affect the final accuracy.

1.1.3 Nonlinear MPC

In general, industrial processes are nonlinear, but many MPC applications use linear models for two reasons; linear modelling is accessible and gives good results in the vicinity of operation points. In a linear model with a square cost function, the QP convex optimization problem will be simple to solve and has been reviewed in many references. In fact, finding the optimal answer in less time than the sampling time is essential because the number of variables is high.

However, linear controllers do not produce good results in dynamic processes where the system is highly nonlinear. The law of linear control is not very practical for these processes. In many cases, the process is stable in the neighbourhood; therefore, a linear representation will suffice. On the other hand, there are some critical circumstances in which this does not happen. For example, there are processes for which nonlinear properties are powerful (even in the vicinity of stable states), and a linear model is insufficient for closed-loop stability. Moreover, some processes experience constant changes (startup, shutdown, etc.) and take a long time away from a steady-state operating area or even processes that are never in a steady-state condition; As in the case of batch processes, where the entire operation is performed in a transient state. The linear control law will not be very effective for these processes, so nonlinear controllers will be necessary to improve or stabilize performance. The principles of nonlinear predictive control are the same as regular predictive control, but they also have several issues:

Finding nonlinear models from experimental data is an unsolved problem. Nonlinear process identification techniques are still inadequate. Using neural networks or the Volterra series

does not generally solve the problem. On the other hand, achieving the model from the first principles (mass and energy balance) is not always possible.

- It is a non-convex optimization problem that is much more difficult to solve than the QP problem.
- Issues with local optimization appear, affecting the quality of control, and instability occurs.
- The optimization time increases, limiting the use of this technique to slow processes.
- Studying essential issues such as stability and robustness in nonlinear systems is more complex.

Some of these problems are being addressed to some extent, and NMPC is becoming an important area of research. Nonlinear models are more complicated to obtain than linear systems, and no suitable model exists for nonlinear processes.

1.2 Linear Parameter Varying (LPV) systems

In practice, most existing systems are nonlinear, change over time, and operate in a variable (non-fixed) operating condition. Researchers are constantly exploring new methods to improve the identification and control of nonlinear systems. One standard method is linearizing the system around a predetermined working point or path. Although working with these methods is simple and has been used in many cases so far, these models have a local functionality (around the work point) and do not consider the system's general nature. In other words, in cases where the system has nonlinear dynamics or working conditions that are changeable with time, the performance and accuracy of these systems are reduced.

Despite the disadvantages, engineers usually prefer to work with such systems over nonlinear systems with many complexities. At the same time, there are many attractive methods for optimal and robust control of linear systems. LPV systems have been introduced to consider both nonlinear and time-varying properties of systems and to use linear representations. LPV is a bridge between linear systems and nonlinear or time-varying systems. These systems try to represent nonlinear systems in the form of linear systems with variable parameters. These systems allow us to apply linear theories while also considering the nonlinear nature of the system.

Much research has been done to find the LPV system model in recent years. Generally, these methods can be divided into two categories: analytical and experimental methods. The first laws of physics, chemistry, thermodynamics, etc., are used to find the model in analytical

procedures. This modelling method usually requires deep knowledge of the system and is typically time-consuming and costly. Also, these methods are generally not directly applicable for control purposes and need modifications or simplifications. The second method obtains the system model using the measured input and output data called system identification. This method is more automatic than the previous method and therefore requires less time and cost, and at the same time, does not require much knowledge of the system. So far, this technique has been used to identify different systems, including airplanes, vehicles, wind turbines, servo systems, induction motors, Internet servers, CD players, and many others.

1.2.1 LPV systems representation

There are many ways to display LPV systems. In these models, the system parameters usually depend on the scheduling signal (θ), including their nonlinear and time-varying properties.

1.2.1.1 State-space form

The discrete-time state-space LPV system is as follows:

$$x_{t+1} = A(\theta_t)x_t + B(\theta_t)u_t + G(\theta_t)v_t^0 \tag{1.8}$$

$$y_t = \mathrm{C}(\theta_t)x_t + D(\theta_t)u_t + H(\theta_t)e_t^0 \tag{1.9}$$

Where x is the state variable, y is the output, u is the input, θ is the scheduling variable, the subset t represents time, and $v_t^0$and e_t^0 are process noises. For most control applications, the arrays A, B, C, D, G, H are defined as a linear combination of basis functions:

$$A(\theta_t) = A_0 + \sum_{i=1}^{n_\psi} A_i \psi^{[i]}(\theta_t) \tag{1.10}$$

where $\psi^{[i]}$ are finite scalar functions on θ and $\{A_i\}_{i=0}^{n_\psi}$ are fixed matrices of appropriate order. In (1.8) and (1.9), two noises are usually equal, and the matrix H is often an identity matrix (in these cases, the matrix G is traditionally denoted by K).

The display of state space can also be written in the form of Fractional Linear Representation (LFR) [4]:

$$\begin{bmatrix} x_{t+1} \\ \omega_t \\ y_t \end{bmatrix} = \begin{bmatrix} A_0 & B_{11} & B_0 & G_0 & 0 \\ C_{11} & 0 & D_{12} & D_{13} & D_{14} \\ C_0 & D_{21} & D_0 & 0 & H_0 \end{bmatrix} \begin{bmatrix} x_t \\ z_t \\ u_t \\ v_t^0 \\ e_t^0 \end{bmatrix}, z_t = \Delta(\theta_t)w_t \tag{1.11}$$

In this display mode, the scheduling signal is placed in the feedback path, and the rest of the system will be LTI. Readers can refer to [3] and [5] for more information about how to find matrices A_0, B_{11}, B_0, G_0, C_{11}, D_{12}, D_{13}, D_{14}, C_0, D_{21}, D_0, H_0 . The relations between (1.8) and (1.9) with (1.11) can be determined as follows:

$$A(\theta) = A_0 + B_{11}\Delta(\theta)C_{11}, B(\theta) = B_0 + B_{11}\Delta(\theta)D_{12}, C(\theta) = C_0 + D_{21}\Delta(\theta)C_{11} \quad (1.12)$$
$$D(\theta) = D_0 + D_{21}\Delta(\theta)D_{12}, G(\theta) = G_0 + B_{11}\Delta(\theta)D_{13}, H(\theta) = H_0 + D_{21}\Delta(\theta)D_{14}$$

It should also be noted that the signals u, θ, x, y are equivalent. Some of the particular modes of display (1.10) are:

- Polynomial parameter dependence:

$$A(\theta_t) = A_0 + \sum_{i=1}^{n_\psi} A_i {\theta_t}^i \quad (1.13)$$

- Affine parameter dependence:

$$A(\theta) = A_0 + \sum_{i=1}^{n_\theta} A_i \theta_i \quad (1.14)$$

In this display, $\theta_i, i = 1, \dots, n_\theta$ means different scheduling signals, and n_θ means the number of scheduling signals.

- Polytopic models:

$$A(\theta) = \sum_{i=1}^{z} a_i(\theta) * A_i \,, \sum a_i(\theta) = 1, a_i(\theta) > 0 \quad (1.15)$$

This representation assumes that the scheduling signal is bounded and inside a polytope as follows:

$$\theta \epsilon \, Co\{w_1, \dots w_z\} \quad (1.16)$$

where the w_i are vertex, and the scheduling signal can be overwritten as follows:

$$\theta = \sum_{i=1}^{z} a_i w_i, a_i \geq 0, \sum_{i=1}^{z} a_i = 1 \quad (1.17)$$

1.2.1.2 Input-Output form

The Input-Output display is as below [3]:

$$\begin{matrix} A(\theta_t)\check{y}_t = B(\theta_t)u_t \\ D(\theta_t)v_t^0 = C(\theta_t)e_t^0 \end{matrix} \quad (1.18)$$

$$y_t = \breve{y}_t + v_t^0$$

where u, $e_t^0$and y are similar to the Equations (1.8) and (1.9), $\breve{y}$ is noise-free output, v_t^0 is additional noise, and matrices A, B, C, D are the model descriptive matrices, all of which are dependent on θ and are defined by the following polynomials:

$$A(\theta_t) = \sum_{i=0}^{n_a} a_i(\theta_t) q^{-i} \tag{1.19}$$

where n_a is a polynomial degree. The system displayed in the input-output form can be written as the equivalent of the Infinite Impulse Response (IIR):

$$y_t = \sum_{i=0}^{\infty} g_i(\theta_t) q^{-i} u_t + \sum_{i=0}^{\infty} h_i(\theta_t) q^{-i} e_t^0 \tag{1.20}$$

An extension of Equation (1.20) would be to extend to bases other than q^{-i}. A unique selection of these basis functions for LTI systems, leading to OBF[1] display, has been extended to LPV systems [5] and [6]. In the structure of LPV systems based on OBF, the following series should be identified:

$$y_t = \underbrace{\sum_{i=0}^{n_e} \sum_{j=1}^{n_g} ((\omega_{ij}(\theta_t)\, \phi_{ij})\, q^{-i} q^{-j}) u_t}_{\overbrace{W(\theta)}} + e_t^0 \tag{1.21}$$

where $\omega_{ij}(\theta)$ is similar to $a_i(\theta)$ and $\phi_{n_g}^{n_e} = \{\phi_{ij}(q)\}_{j=1,\dots,n_g}^{i=0,\dots,n_e}$ is a sequence. The positive basis functions used in Equation (1.20) are a particular type of this basis. Thus, Equation (1.21) is an extension for Equations (1.20).

1.3 Research objectives

The purpose of this dissertation is to address the following problems:

1) Reduce the conservatism of MPC-LPV-IO controllers.
2) Demonstrate the stability of the MPC-LPV-IO controller under bounded disturbances while also reducing the computational load of the online optimization problem
3) Identification of LPV-IO systems through non-parametric methods.
4) Employ the developed controllers for a robot.

[1] Orthonormal basis function

1.4 Research contributions

The research contributions include the following:

1) Developed an interpolation-based MPC that reduces conservatism and online computation load.
2) Ensured the stability of the LPV-IO system in the presence of disturbances with ISPs' approach.
3) Used a recurrent neural network to reduce the time required to solve online optimization problems.
4) Utilized MPC control with LPV-IO representation for a wheeled mobile robot.
5) Improved the robot's performance when facing an object or disturbance regarding speed, stability, and conservatism.

The remainder of this thesis is structured as follows. Chapter 2 reviews the medical robots and control techniques for medical mobile robots, along with research on MPC-LPV. Chapter 3 investigates interpolation-based MPC for LPV systems with scheduling signal uncertainty and without external disturbance. Chapter 4 extends the analysis to include the presence of bounded disturbances while reducing computational burden using a recursive neural network. Meanwhile, a recursive neural network has been used to reduce computations. Both chapters 3 and 4 assume that the LPV input-output model exists or is obtained using a polynomial interpolation of several scheduling variables. Chapter 5 uses a method based on the support vector machine to find better accuracy. Chapter 6 evaluates an interpolation-based MPC with free control moves and RNN for a wheeled mobile robot, assessing its effectiveness in reference tracking, obstacle avoidance, computational load, and conservatism. Finally, the last chapter summarizes the results and provides suggestions for future works.

CHAPTER 2

A REVIEW OF CONTROL TECHNIQUES FOR MEDICAL WHEELED MOBILE ROBOTS: TOWARDS APPLICATION OF RESILIENT MODEL PREDICTIVE CONTROL WITH LINEAR PARAMETER VARYING SYSTEMS

The work presented in this chapter is included in the following paper:

Hadian, M., & Zhang, W. (2022). A Review of Control Techniques for Medical Wheeled Mobile Robot: Towards Application of Resilient Model Predictive Control with Linear Parameter Varying Systems. Annual Reviews in Control. Under review.

Abstract

Mobile medical robots are employed for various medical functions, including surgery, cleaning, disinfection, and drug delivery. The demand for precise tracking control of medical robots has increased recently. The tracking problem of wheeled mobile robots (WMR) is complicated due to nonholonomic constraints, nonlinear dynamics, and uncertainties. This paper reviews popular control approaches to cope with these problems, such as feedback linearization control, sliding mode control, fuzzy logic control, and vision-based control. We conclude that a combination of model-based predictive control (MPC), Linear Parameter Varying (LPV) systems, and resilient control would be one of the most promising approaches for WMR in the future. This conclusion derives from three main reasons. First, MPC can naturally work with Multiple-Input Multiple-Output (MIMO) systems with operational constraints. Second, LPV has been shown to be effective for modelling nonlinear dynamic behaviours of processes. Third, resilience is the ability to tolerate the most severe risks and unfavourable conditions while remaining as close to normal as possible.

2.1 Introduction

Today, the science of robotics is rushing to help surgeons, physicians, nurses, and most importantly, patients, and has revolutionized the healthcare system [6-8]. The COVID-19 pandemic highlighted medical robots' overarching value and resulted in their usage in many hospitals and quarantine centers [9-11]. In an epidemic crisis such as COVID-19, which is highly contagious, robots may take care of patients or carry out several routine hospital activities to mitigate disease outbreaks. A perfect demonstration can be seen in China, where several robots cleaned and cooked food in quarantine zones [12, 13]. Medical robots can act as surgeons [9, 14], rehabilitation workers [15-17], nurses [18, 19], dentists [20, 21], pharmacists [22], medical receptionists [23], and elderly caregivers [24]. They also can help people with autism [25, 26], disinfect hospitals [27], enter the human body as a capsule to diagnose and treat diseases [28, 29], and carry in-hospital items such as medications, medical records, and food [30-32]. They are believed to improve productivity and patient satisfaction and reduce healthcare costs. Meanwhile, they do not become sick and can reliably, quickly, and accurately work around the clock.

Path planning and path tracking are fundamental challenges in medical robotics [33]. Path planning algorithms analyze and identify an obstacle-free path for a mobile robot to navigate within the environment [34]. Path tracking control, also called trajectory tracking control, is designed for a mobile robot to track the reference path precisely. The literature classifies path planning algorithms as global or local [35]. Global path planning algorithms, such as the Dijkstra algorithm, A* algorithm, and D* algorithm, are limited to static environments with static obstacles. These algorithms require the robot to completely understand its surroundings to plan a path [36]. In contrast, in local path planning, the path to traverse is constructed offline before the robot navigates the environment. It enables the robot to navigate an unexplored environment safely. Local path planning techniques allow the robot to build a new path in real-time in response to sensory data. This approach enables the robot to navigate an unexplored environment securely.

As shown in Figure 2.1, local path planning algorithms can be divided into two categories: conventional and intelligent techniques. The classical group’s primary downside is its high computational cost and inability to adjust to environmental uncertainty; as a result, it is not recommended for real-time implementations [37-39]. Intelligent strategies mainly come from metaheuristics algorithms such as genetic algorithms [40], particle swarm optimization [41], and cuckoo search algorithms [42].

This paper focused on the path tracking control problem of wheeled mobile robots (WMR). Figure 2.2 illustrates the mission of a WMR to traverse from location A to location B while avoiding obstacles. The tracking control problem of WMR is highly complicated, stemming from nonlinear dynamics, nonholonomic constraints, and significant uncertainties. Many control techniques have been proposed in the literature to address these issues, such as PID controller [43], feedback linearized control [44], sliding mode control [45], resilient control [46], intelligent control, vision-based control, and Model Predictive Control (MPC) [47]. There seem to be three key problems and potential solutions we could take to alleviate these problems.

The main problem is the real-world limitations, also known as constraints, which must be considered in controller design since it lessens the achievable paths. These constraints include speed response, mobility constraints, computational cost, field-of-view constraints, maneuverability, and control stability issues [48]. A large body of research on WMR has investigated holonomic and nonholonomic constraints. A holonomic constraint restricts the robot's configuration that may be achieved (positional variables), i.e., there are locations that the robot cannot navigate. In opposition, a nonholonomic imposes restrictions on the achievable velocities of the robot (derivate of positional variables), i.e., some paths or directions are not allowed. In contrast to holonomic constraints, nonholonomic constraints are not integrable to provide constraints regarding positional variables, so the transformation matrix should be in terms of velocities, not positions. According to Brockett's theorem, no continuous time-invariant feedback of state variables can be found that asymptotically stabilizes the non-holonomic system around the equilibrium point [48, 49].

The best solution for meeting constraints would be MPC. The first applications of MPC were in process industry control [50] due to slow dynamics. Using MPC in robotics was impossible due to the need for fast processors. With the rapid development of fast processors and numerical computing algorithms, MPC's use in semi-fast systems, like mobile robots, has become possible. MPC is now used to control various systems [51-53].

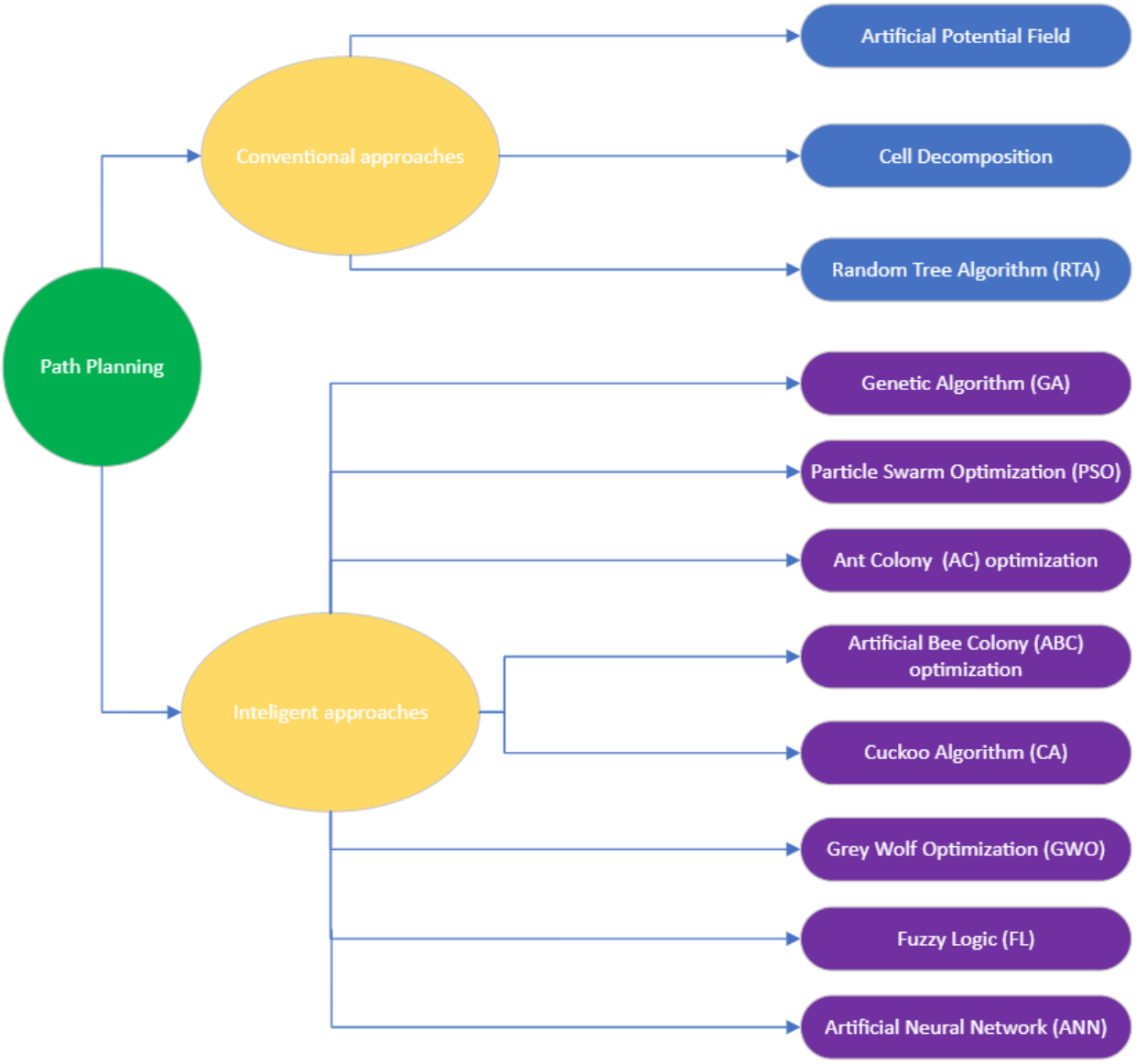


Figure 2.1: Path planning for WMR

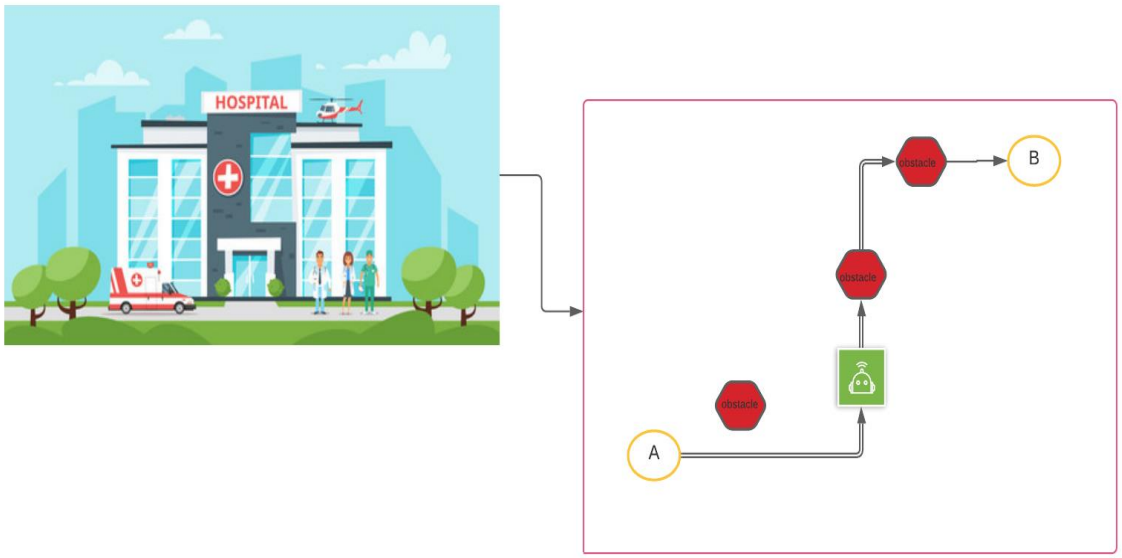


Figure 2.2: The trajectory tracking of WMR in a hospital

Unlike existing control methods, in which the control law is obtained based on the previous outputs of the system, the MPC is a model-based optimal method that obtains the control law with the prediction of the future outputs of the system [54-56]. How MPC works is similar to how human makes a decision. Humans generally plan to achieve the desired conditions and goals in the future by making decisions in the present and thinking of far-reaching consequences. MPC structure can be seen in Figure 2.3, where the prediction horizon (Np) is a time interval at which the system's future outputs are predicted. The control horizon (Nc) is the number of steps that the control input sequence calculations are performed. As shown in Figure 2.3, following the reference path is carried out using previous inputs-outputs and future estimated outputs. Then, with the help of this new information and the reference path and control strategy, a sequence of appropriate inputs over the control horizon is calculated. In other words, in MPC, the control signal in the next steps is determined as the system's output reaches the desired output value in the next steps [57, 58].

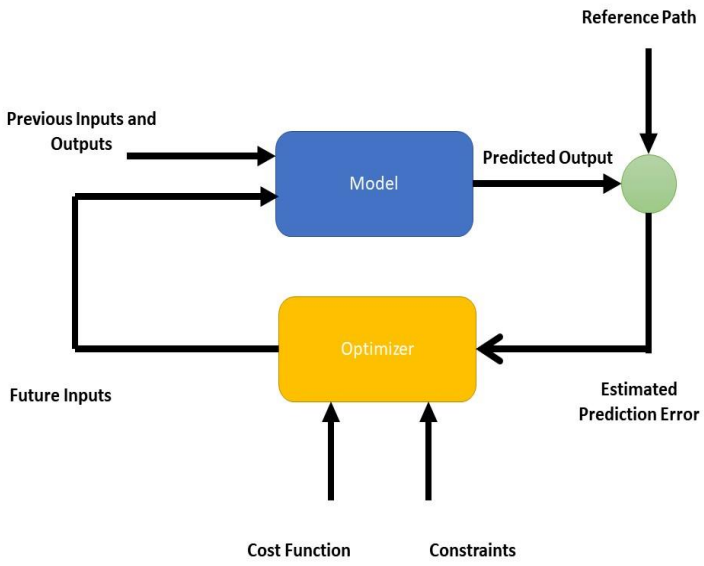


Figure 2.3: MPC framework

In this control method, at each time of sampling k, based on the measurements made up to the sample k, the controller predicts the system output for Np samples ahead and generates control input over N_c ($N_c \leq N_p$) samples to minimize the predetermined cost function. Also noteworthy is that only the first sentence of the control input sequence is applied to the system. Prediction and optimization are repeated for the next sample based on the new measurements.

According to Figure 2.3, the prediction model and cost function are the backbones of an MPC. The prediction model estimates the future outputs, usually described by step response, impulse response, transfer function, and state space. The cost function, defined as a sequence of control inputs and state deviation variables, is minimized to determine an optimal control input sequence. The type of the prediction model and cost function parameters are two fundamental principles of an MPC to find the optimal control laws.

The second problem is the nonlinear model of WMR. While influential theories exist in linear MPC, some systems are nonlinear and operate under extensive operating conditions, or external parameters can fundamentally change the process response [59]. For example, in a two wheels mobile robot, the process behaviour's perdition can be complicated, and an inaccurate model might cause the controller’s failure in practice. The rationale behind this failure is the nonlinear time-varying dynamics of WMR. This means that MPC’s desired performance crucially depends on how accurate the model is for coping effectively with mobile robots' nonlinear time-

varying features. On this account, a nonlinear model predictive controller (NMPC), which can predict the robot's behaviour up to several steps ahead and can optimally control and guide the robot, will be developed in several studies [47, 59, 60]. However, a higher level of algorithm complexity and excessive computational load in controller design are two main drawbacks of NMPC. Furthermore, modelling and output prediction for nonlinear systems are daunting compared to linear systems.

Because of NMPC restrictions, several researchers used linearization around the operational point. This method is simple to implement and has been tested in various settings. On the downside, linear-oriented MPCs perform well locally and remain valid around the operating point, but they perform poorly in other situations and cause instability. Besides, employing linear models might reduce prediction accuracy. Despite these flaws, industry experts favour linear frameworks over nonlinear ones that might result in excessive complexity. As a result, both linear and nonlinear MPC techniques offer advantages and disadvantages.

The most effective remedy that can be taken to deal with nonlinearities is LPV. MPC based on a linear parameter varying (LPV) system is brought to researchers' attention to attain a compromise between linear and nonlinear techniques [61-65]. The nonlinear/time-varying aspects of the system are taken into account in LPV models, which have a linear representation. By portraying nonlinear systems with a linear form with varying parameters, also known as scheduling variables, LPVs bridge the gap between linear and nonlinear/time-varying systems [61, 66]. The scheduling variables include nonlinear dynamic system characteristics, environmental conditions, and operating points. In a nutshell, LPV helps MPC deal with nonlinear/time-varying systems.

The last challenge is reaching a higher degree of robustness in controller design. The capacity to sustain operations amid a crisis is referred to as robustness. Robustness is the ability to maintain strength and effectiveness in the face of adversity. The more robust a system, the less impacted its performance is by disturbances or environmental changes. Because of parameter uncertainties and disturbances, implementing robust MPC approaches, such as min-max MPC [67] and Tube-based MPC (TMPC) [68], is unavoidable. Min-max MPC, also known as worst-case MPC, solves the optimization problem while considering all conceivable evolutions of uncertainties. However, in TMPC, the growth of uncertainty is restricted to a single tube. All robust controllers belong to resilient control systems, the most practical policy to mitigate uncertainties. Resilience is described as the capacity to restore regular operations after an interruption in a

reasonable amount of time. In contrast to robustness, which is proactive, resilience is reactive, occurring after an event. Resiliency is assessed by the time a system takes to recover to its original performance condition [69].

The rest of this paper is organized as follows. The significance and varieties of WMRs in healthcare environments will be discussed further in this section 2. Then we look at the different control strategies for WMR. Section 4 explains the fundamentals of MPC. This section also describes developing an MPC based on an LPV framework. Section 5 reviews research on MPC applications for WMR in terms of model, optimization, and stability. The conclusion and future works will be the final section.

2.2 Wheeled mobile robots in hospitals

Robots nowadays have become a handy tool to help medical staff, and their irreplaceable role in the medical field is continually growing. As illustrated in the introduction, they have been developed for surgery, rehabilitation, nursing, elderly care, and autism therapy. WMRs can be used in hospitals and medical centers to hold items, clean rooms, and help people with disabilities. Figure 2.4 shows the number of research articles on medical robots according to Scopus (documents with “medical” AND “robot” in the title, abstract, and keywords were searched). Likewise, the health robots trend is presented in Figure 2.5 (documents with “health” AND “robot” in the title, abstract, and keywords were searched). It is undeniable that medical robots have received significant interest in recent decades, and the number of research efforts is constantly growing.

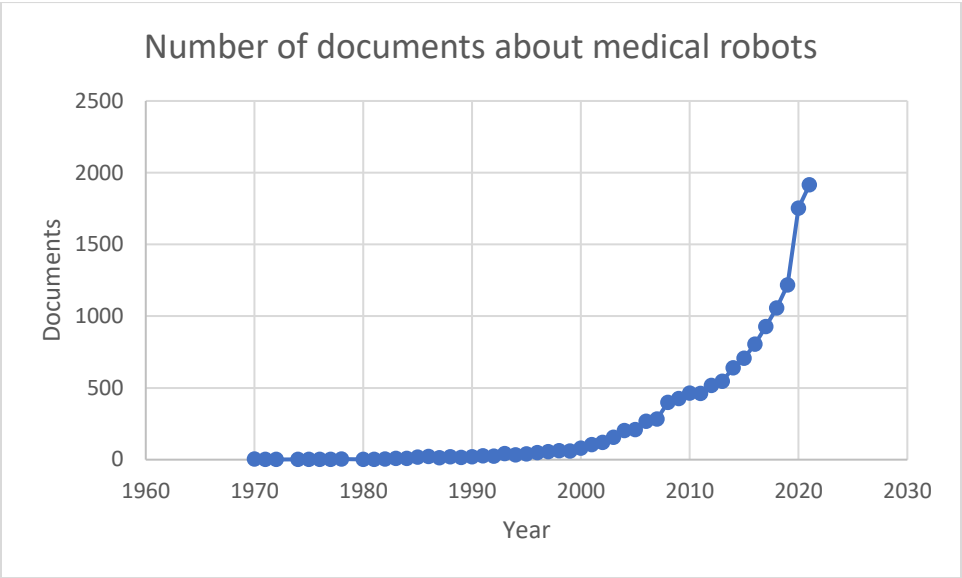


Figure 2.4: Documents per year for medical robots

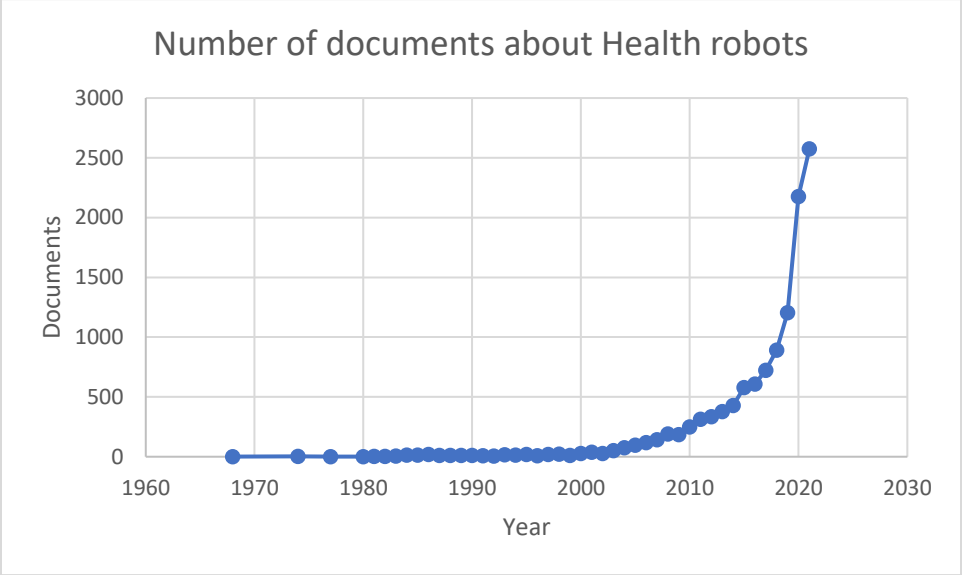


Figure 2.5: Documents per year for health robots

The robots in medical centers and hospitals can be stationary or mobile. Mobile robots are a group of medical robots that can move around and are not permanently attached to a place. They can be developed and designed according to the operation area, environment, and wheels. Firstly, the operation area can be land, air, or sea [70-72]. Land-based mobile robots are classified into wheeled and legged robots. The former is widely known for its simplicity in the mechanical structure, whereas legged robots are suitable for individual tasks in harsh environments such as stairs, rocks, and debris [73, 74]. Mobile robots can also be used with an arm to perform more

activities [75]. Second, the environment in which robots traverse should be contemplated in designing the controller. The environments can be structural, semi-structural, or non-structural, which might be static or dynamic. In a structured case, the path is predetermined, and the location, dimensions, size, and material (if required) of existing obstacles are known. The unstructured type is an unseen and unknown site, and the robot recognizes it by interacting with the environment. By comparison, the robot already knows parts of the space in a semi-structured environment. Lastly, regarding the number of wheels (one to six) and types of wheels (standard, castor, Swedish, and ball), there might be 17 different configurations for robots [76].

Robots need sensors to comprehend the environment and obstacles to determine where to go next. Choosing the right sensor based on the applications and operations is one of the most critical issues in the design and construction of mobile robots. Conventional sensors used in ground mobile robots are touch sensors, wheel encoders, Global Positioning System (GPS), heading sensors (including gyroscopes and compasses), Accelerometers, Inertial Measurement Units (IMUs), distance sensors, and digital cameras [77]. GPS sensors were hugely popular and widespread among robot specialists of all mentioned sensors. However, their applications have experienced a sharp decline in recent decades for four reasons. Firstly, they are proven unreliable in the indoor workspace of urban areas such as hospitals [78, 79]. Additionally, they lack the quality required for high-precision missions and systems with high-speed dynamics. Other drawbacks include huge costs and failure in building with intense illumination [80].

A number of works have shown that this problem can be overcome by using digital cameras. This field has gradually broadened as visual servoing or vision-based control. More information can be found in section 2.5. The rest of this section will be about mobile robot classification in terms of tasks in the hospital, namely object-carrying, cleaning, and service robots.

2.2.1 Object-carrying robots

Object-carrying robots could be employed in hospitals, homes, or elsewhere. They can deliver medicine, medical equipment, medical records, food, and even lift patients in hospitals. A study in a San Diego, USA, hospital reported that shift staff spends more than half of their working hours moving equipment or commuting between various hospital rooms and departments [6]. In this study, the Helpmate robot (Figure 2.6) moved heavy loads, drugs, food, and laboratory results, and it was estimated that just moving the drug by this robot can save around 60 thousand dollars a

year. Likewise, Transcar (Figure 2.7) and Nestor [6] robots can be examined for transporting medical equipment and records.

Figure 2.6: HelpMate Hospital Robot [6]

Figure 2.7: Transcar robot [6]

The Moxi robot (Figure 2.8) is an intelligent hospital robot that boosts clinical personnel's productivity [81, 82]. It can become a vital and reliable treatment team member by supporting members using social intelligence. The Moxi robot is equipped with a sophisticated arm to carry out mundane chores such as picking up medical items and placing them on a tray attached to the robot for taking items to specific locations [83]. It also has a monitor that reveals details about the missions performed. Another duty of this robot is to distribute medicine quickly to patients so that nurses have more time to care for patients. The designers insisted that the robot be friendly and welcoming since it is manufactured to function in a demanding environment. The manufacturer

describes Moxi's behaviour as pleasant, as evidenced by the robot's bright face, made up of an LED display with a subtle sound.

Figure 2.8: Moxi robot [81, 82]

TUG [7, 84] and HOSPI [8], as shown in Figure 2.9, are two robots that currently serve in hospitals using GPS systems. They can be controlled by a touchscreen interface and recovers energy after each mission to be called for the next mission. TUG is a versatile robot capable of carrying shelves, carts, and boxes containing medicine, laboratory samples, and other sensitive materials with a load of up to 453 kg. With this robot, fewer workers are expected to carry heavy loads, so they are protected from related physical injuries and have more time to monitor patients.

(a)

(b)

Figure 2.9: TUG [84] (a) and HOSPI [8] (b) robots

The Aircart robot is an intelligent wheelchair that moves slowly, needing low force for pushing and pulling, regardless of the user's weight [85]. This robot brakes sharply and automatically in critical situations. The Aircart robot is an enticing choice for those with little

mobility. The Ridgeback robot is a medium-sized platform with an omnidirectional propulsion system to lift heavy loads easily [86-88]. It can carry even more items than humans, suitable for indoor and outdoor applications.

2.2.2 Cleaning robots

Cleaning robots can be used to clean hallways, waste disposal, laundry, washroom cleaning, and sterilize medical devices and the environment. One may go to the hospital but return home with more new illnesses because hospitals are very contaminated. With this in mind, disinfectant robots are manufactured to fumigate patients' rooms independently for a few minutes with UV light [89]. The Xenex robot [90] (Figure 2.10) disinfects healthcare environments quickly and routinely. According to Westchester Medical Center, Xenex robots reduce Clostridioides difficile (an infectious disease) by 70% in the intensive care unit.

Figure 2.10: Xenex robot [90]

2.2.3 Service robots

Service robots include nursing, rehabilitation, social, and ambulance robots. Social robots are a group of robots that interact with patients, older adults, and kids. By way of illustration, the Mabu robot [91] benefits the elderly and the disabled through regular check-ins and signalling their medication time (Figure 2.11). It can use AI to make a personalized conversation with a human via voice, text, or mobile app. Robin robot [92, 93] is a friendly robot that can express all sorts of feelings, interact with children, and participate in conversations with them (Figure 2.12). By engaging children in play and conversation, Robin eliminates their feelings of isolation and tension during their hospital stay. In addition to robots used to connect with patients, robots can also assist in more critical scenarios. For example, ambulance robots, which often can fly, are for individuals with acute conditions like a heart attack or respiratory problems [94].

Figure 2.11: Mabu robot [91]

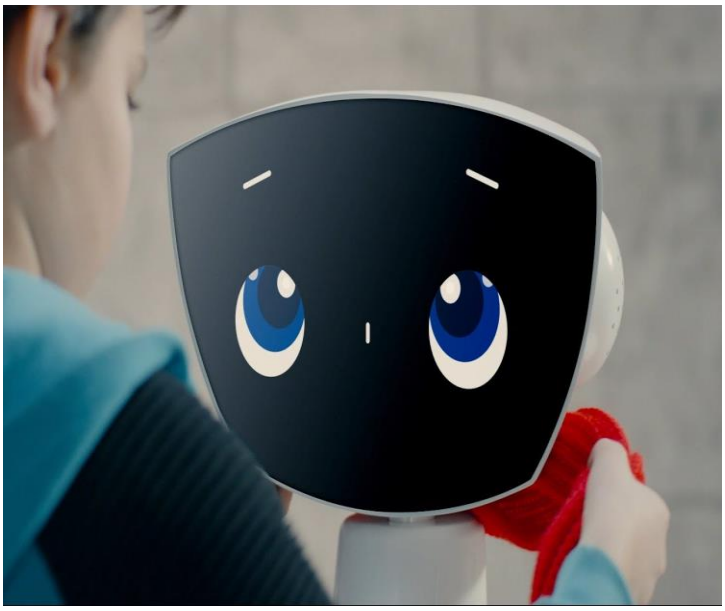


Figure 2.12: Robin robot [92]

2.3 Control of nonlinear medical robots

Unmanned Aerial Vehicles (UAV), automobiles, car-like vehicles, and wheeled mobile robots are examples of nonholonomic systems [79, 95-98]. The underlying geometry of nonholonomic systems makes control design difficult since the constraints are time variables[99]. Disturbances and parameter variations are other serious challenges that the controller should be met. Several typical solutions for controlling and stabilizing nonholonomic systems have been proposed [100-102]. The stabilization problem of nonholonomic WMRs has aroused the interest of many academics due to the increasing use of mobile robots and the inherent nonlinearities in their dynamics [103]. Because the wheels do not slip, nonholonomic constraints are applied to

wheeled mobile robotic systems, limiting their mobility. A controller of a WMR should be developed to deal with nonholonomic constraints, nonlinear dynamics, and uncertainties. A brief review of nonlinear control approaches has been outlined, including state feedback linearization control, sliding mode control, intelligent control, resilient control, vision-based control, and MPC for WMR.

2.3.1 Feedback linearization control

Feedback Linearization (FBL) is a typical nonlinear control strategy. The fundamental idea is to employ state feedback and nonlinear transformation to change a nonlinear system into a fully or partially decoupled linear system, allowing linear control strategies to be applied [104]. This technique linearizes the nonlinear system across a large operation range by using an appropriate nonlinear transformation $z = T(x)$ and a nonlinear state feedback variable $v = \alpha(x)+\beta(x)u$. No higher-order nonlinear factors are disregarded because the linear approximation is not used; this linearization has improved accuracy over classical linearization. Some implementations of FBL for WMR have been explored here, including PI-based FBL, MPC-based FBL, robust FBL, state-space FBL, and input-output FBL.

A PI controller for the motion control problem in a WMR is studied, where the dynamic model is simplified with a full-state FBL [105]. Experiments were used to validate the model and controllers. The robot's experimental trajectories for a circle presented reasonable accuracy. The nonlinear trajectory tracking problem for a nonholonomic car-like mobile robot with a constraint state is addressed [106]. The state constraint is highlighted in this work compared to previous efforts on nonholonomic car-like mobile robots. A nonlinear tracking controller is then achieved using the dynamic feedback linearization technique. Simulation findings show that the proposed control law is effective for trajectory tracking control. The construction of a control law employing the input-output feedback linearization approach to drive a nonholonomic WMR to track a particular trajectory when longitudinal and lateral slip exists is shown [44]. The system's asymptotical stability is confirmed. The results of the Matlab-Simulink simulation reveal that the control law is valid and performs well. An omnidirectional mobile robot is given a hierarchical control technique for tracking predetermined paths [107]. Dynamic feedback linearization makes up the lower layer. The lower layer's linear model is employed on the upper layer for an MPC-based trajectory tracking controller. MPC provides accurate predictive actions for particular scenarios, such as turning on corners, because the trajectory to be monitored is predetermined.

2.3.2 Sliding mode control

Sliding mode control (SMC) is an algorithm that is inherently robust to parameter changes, external disturbances, and uncertainty. SMC plays a vital role when there are large uncertainties in nonlinear automobile applications[108]. In cases where the requirement for robustness is critical, SMC would be a good solution. Two key issues plague conventional SMC. Firstly, chattering and sensitivity to mismatched uncertainties limit its practical applications. From a practical point of view, chattering can be devastating and may necessitate a larger amount of energy. Secondly, whereas traditional SMC provides asymptotic stability, there is no assurance that it will occur in a finite time, especially in the presence of mismatched disturbances. A considerable amount of literature has been published to handle mismatched uncertainty and minimize chatterings, such as integral SMC, quasi-SMC, and fuzzy-based SMC [109]. Many concerns, however, remain unsolved and remain an open study area.

Two implementations of conventional SMC for WMR are demonstrated. The first SMC is proposed for WMR trajectory tracking while attaining robustness to external disturbances and parameter uncertainty [110]. The designed controller showed good trajectory tracking performance even with a large upper bound of uncertainty. The second example is an SMC for WMR in polar coordinates [101]. The controller stabilizes the position and heading direction asymptotically. In contrast to prior research based on kinematics given in polar coordinates, constraints on the required linear and angular velocities, as well as the posture of the mobile robot, are removed. As a result, arbitrary trajectories like a circle and a straight line can be tracked even with substantial initial tracking errors and bounded disturbances.

In contrast to the two above examples, recent SMC studies focused on eliminating chattering, coping with a group of matched and mismatched uncertainties, and convergence rate. For example, an integral sliding mode control (ISMC) is proposed for WMR trajectory tracking [111]. The controller is set up to reduce matched disturbances and mismatched ones. The proposed controller has two parts: a high-level controller for stabilizing the nominal system and a discontinuous controller for assessing trajectory tracking in the presence of disturbances. A WMR with two parallel wheels and an intrinsically unstable inverse pendulum is controlled by an ISMC [112]. The WMR was underactuated, as it only had one actuator to regulate the wheels' movement while balancing the pendulum around the upright position. The proposed controller could eliminate the matched uncertainties and reduce the impact of unmatched uncertainties. The goal of [45] is

to develop a near-optimal SMC for continuous-time nonlinear MIMO systems that reduces chattering while minimizing the cost function. This research first examines the presence of chattering in regular SMC and the quasi-SMC approach. A continuous saturation function in quasi-SMC substitutes the standard SMC's signum function. Then, integrated reinforcement learning is employed to reduce chattering. A fuzzy adaptive sliding mode controller (FASMC) for the trajectory tracking problem in the face of uncertainties and disturbances is presented in another study [113]. The FASMC ensures the closed-loop system's stability and convergence based on a nonlinear dynamic model. Furthermore, the developed controller assures the system's robustness against dynamic disturbances and uncertainties, smoothness against chattering phenomena, and convergence of optimal velocity and posture errors.

2.3.3 Intelligent control

The ideas of fuzzy logic and artificial neural networks are explained in this part, examining their applications in WMR. Professor Lotfi Asgarzadeh coined "fuzzy logic" in 1965. He concluded that systems cannot imitate the thoughts and ideas of the human mind and cannot think like humans since digital logic only has two states for each decision: "True" and "False." On the other hand, human thinking has a degree of right or wrong for a choice. In classical set theory, the concept of membership shows whether an item belongs to a set. However, in fuzzy logic, the belonging of each item to a set has a degree of membership with a value between 0 and 1. An object, for example, belongs to set A with a membership of 0.3 and set B with a membership of 0.7. Figure 2.13 describes the four basic components of a fuzzy logic system: fuzzifier, rules, inference engine, and defuzzifier.

Fuzzifier: This stage turns the inputs into fuzzy information (linguistic variables). The numbers and information to be handled will become fuzzy sets and membership functions. As a result, the input data from sensors in a control system is updated and ready for fuzzy logic processing.

Rules: This section comprises conditional statements, also called if-then rules that an expert specifies to regulate the decisions for a "decision-making system." According to recent fuzzy theory developments, altering and eliminating some rules and membership functions is feasible to get the greatest results with the fewest rules [114].

Inference engine or intelligence: The degree of compliance of the fuzzy inputs with the rule base is assessed in this unit, and fuzzy outputs are generated. Various decisions are made dependent on the percentage of compliance.

Defuzzifier: The outputs of the fuzzy inference engine, which are in the form of fuzzy sets, are turned into crisp values in the last phase. At this point, optimal values are selected based on the outputs. Typically, this decision will be based on the highest level of compliance.

Fuzzy logic control is a heuristic technique that readily embeds knowledge and important features of human reasoning in designing nonlinear controllers [115, 116]. Qualitative and heuristic factors, which cannot be addressed by standard control theory, can be systematically employed using fuzzy control concepts for control purposes. Fuzzy logic control does not require a perfect mathematical model and copes with imprecise inputs. It can deal with nonlinearity and disturbances. Fuzzy logic controllers typically outperform other controllers in complicated, nonlinear, or undefined systems when excellent practical knowledge exists. Figure 2.13 depicts a fuzzy logic controller for a robot.

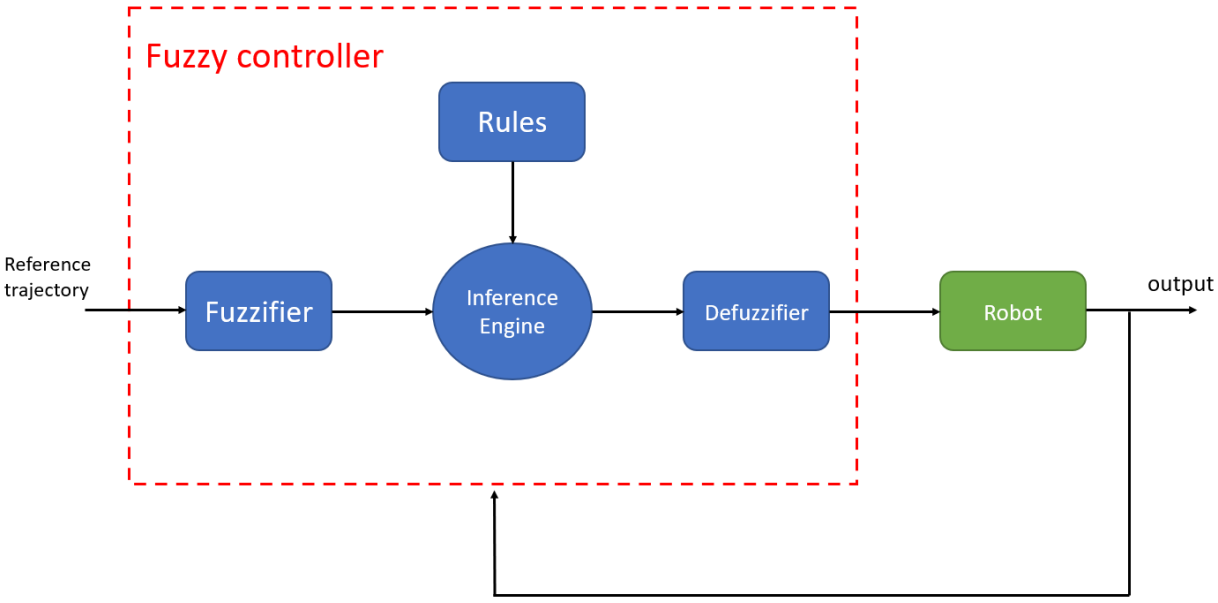


Figure 2.13: fuzzy logic controller

Like the fuzzy logic controller, researchers have expressed great interest in ANN over the last decades. ANNs are linked parallel structures that incorporate successive layers of processing units, known as neurons, and are inspired by the biological nervous system. Layers of parallel

neurons receive inputs and deliver outputs. Multi-Layer Perceptron (MLP) is the most famous type of ANN. In an MLP, there are three unique layers: the input layer, which receives the independent variables; the hidden layer, which serves as an intermediate layer; and the output layer, which sends the dependent variables. The size of the dependent and independent variables determines the number of neurons in the input and output layers. A weight, a bias, and an output transfer function are applied to each neuron. The learning process correlates the dependent and independent variables in the network (a process by which the weights and biases are modified to minimize network prediction errors) [117].

There are two types of ANN: feedforward and feedback. Feedforward ANNs are networks where the response channel is constantly processed forward and never returns to its predecessors' neurons. Signals can only transit through a one-way path in this network (from input to output). As a result, there is no feedback, which means that each layer's output simply impacts the next layer and has no effect on its own. In contrast, at least one return signal in feedback networks is sent from one neuron to neurons of the same or previous layers. The consequence is that the neuron's output at the present instant is determined by the input at that time and by previous information. Some studies on intelligent controllers have been summarized here, including type-1 fuzzy controller, adaptive fuzzy controller, type-2 fuzzy controller, fuzzy-PID controller, ANN-based PID controller, and Adaptive Neuro-Fuzzy Inference System (ANFIS) controller.

An adaptive fuzzy controller for trajectory tracking is designed [118]. The fuzzy logic system estimates the system uncertainty, including mobile robot parameter change and unknown nonlinearities. The Lyapunov stability theory demonstrates the system's stability and the convergence of tracking errors. Another study provides a fuzzy adaptive tracking control approach for wheeled mobile robots in the presence of unknown slippage, which violates the nonholonomic constraint [119]. These disturbances greatly reduce tracking performance and, as a result, should be compensated. Kinematics with state-dependent disturbances is derived from the general form of slippage in mobile robots. The Fuzzy adaptive observer estimates the state-dependent disturbances in kinematics and dynamics. Even when the nonholonomic requirement is violated, the controller can ensure that the trajectory tracking errors are finally restricted. The approach can be modified suitably for different types of trajectories in the presence of significant initial tracking errors and disturbances.

Two knowledge-based controllers are presented to solve the challenges of a calculated torque nonlinear controller (NC) in trajectory tracking of nonholonomic WMRs [120]. A fuzzy nonlinear controller (FNC) is created by substituting the proportional and differential parts of the NC with fuzzy functions. Due to the intricate dynamics of the WMR in which the center of mass does not correspond with the center of rotation, expert knowledge is extracted using fuzzy controllers with rotation angles and driving wheel speeds as input variables. Although the control torques are greatly reduced and smoothed, fuzzy tuning of the NC leads to improved tracking performance against measurement noises. Second, a fuzzy controller (FC) is created to track the WMR's position and orientation perfectly. Simulations are used to demonstrate the better performance of the proposed fuzzy controllers over the NCs.

The design and implementation of tracking and position control in omnidirectional mobile robots are discussed utilizing type-2 fuzzy systems [121]. Using a human expert's expertise in building a fuzzy rule base, the authors defined a relationship between the inductive voltage, the distance of objects in an unknown dynamic environment, and their linear and angular output velocities, making a breakthrough. They created a type-2 fuzzy controller and compared it to a type-1 fuzzy. The findings validated the omnidirectional robot's ability to navigate an unstructured environment with unknown obstacles.

A fuzzy-PID controller is presented for path tracking of a differential WMR [122], having two inputs and three outputs. Inputs are error and error rate, and outputs are PID coefficients. The simulation results demonstrate that the fuzzy-PID controller outperforms the traditional PID controller. The suggested controller has a higher convergence rate than the traditional PID controller for a WMR with any arbitrary initial state. It offers the benefits of quicker response, higher stability, and reduced tracking error. An optimal Mamdani-type fuzzy logic controller for trajectory tracking of a WMR is developed to deal with parametric and nonparametric uncertainties in a robot model [123]. The parameters of membership functions and the PID controller coefficients are simultaneously optimized using Random Inertia Weight Particle Swarm Optimization (RNW-PSO).

ANN-based PID is designed for a nonholonomic mobile robot's velocity and orientation tracking control [43]. The proposed nonlinear PID-based is a great combination of the traditional PID controller and a neural network, which has the remarkable capacity to constantly learn online, adapt, and deal with nonlinearity. The simulation experiment demonstrates the usefulness of the

proposed control algorithm and its improved performance, especially in disturbance rejection. In another study, the authors offer a path-planning system for a WMR navigating between fixed and moving obstacles [124]. The neural network inputs are the distance from the robot to the obstacles and the target. This system employs an ANN as a controller, and it is trained using a modification of the backpropagation through time approach that employs potential fields for obstacle avoidance. In [125], two ANNs guarantee that the mobile robot moves optimally from its present position to a predetermined position. One ANN specifies the position and size of the obstacle, while the other creates a continuous path to approach the target quickly and avoid the obstacles.

An ANFIS controller is presented in this paper for mobile robot navigation and obstacle avoidance in unknown static environments [126]. Several sensors, such as ultrasonic and infrared sensors, are utilized to identify forward obstacles in their surroundings. The ANFIS controller receives obstacle distances from sensors as inputs, and the controller output is a robot steering angle. The fundamental goal of this study is to use the ANFIS controller to steer the mobile robot across specified environments. Similarly, a Multiple Adaptive Neuro-Fuzzy Inference System (MANFIS) control strategy is designed and implemented for mobile robot navigation in various two-dimensional environments with static and moving obstacles [127]. Three infrared range sensors are installed on the front, left, and right sides of the robot, and they read the static and dynamic obstacles in the surroundings. This sensor data information is sent into the MANFIS architecture, generating suitable speed control orders for the robot's right and left motors. Two advanced ways for directing a non-holonomic mobile robot to travel in a cluttered environment with static obstacles are investigated [128]. First, a Fuzzy logic controller (FLC) with trapezoidal Membership functions was constructed. Second, the findings acquired from the trapezoidal fuzzy controller were optimized using an ANFIS controller.

A controller structure of a mobile robot that operates on two wheels powered by a dc motor is described [98]. For regulating the velocity of a DC engine, we initially considered a PID controller, a Fuzzy Logic Controller, and an ANFIS controller. This proposed approach has deduced that ANFIS is preferable to Fuzzy and PID regulators because it provides less overshoot and a shorter settling time of the output signal. For a two-wheeled differential mobile robot, a smart PID optimized by a neural networks-based controller (SNNPIDC) and a PD fuzzy logic controller (PDFLC) are constructed [129]. In the first, neural networks are employed to optimize the parameters of a PID controller, while in the second, a Mamdani-based fuzzy inference system is

used. The objective is to develop control algorithms for safe robot navigation while preventing motor damage. In these two control scenarios, the smart robot must complete tasks fast and adapt to changing environmental circumstances while maintaining stability and accuracy.

2.3.4 Resilient Control

Chris Zhang was the first to establish the notion of resilience [130]. All hazards and measures that jeopardize appropriate operation must be considered part of resilient design. Cyber and physical security, process efficiency and stability, and process compliance are examples of resilient control [131]. A resilient system must identify undesirable events and conditions, respond correctly, and quickly recover. Some resilience controls assist with diagnosis, while others aid reaction or recovery [132]. According to [133], resilience is the ability to withstand the utmost effect of an attack while operating as near to normal as practicable.

Resilience defines how systems maintain a reasonable level of normality in the face of disturbances or threats [134]. The authors first analyzed the interdependencies features in critical infrastructure systems and how resilience mitigates related risks. They then introduce the terms "agent" and "multi-agent systems" (MAS) to consider the distributed nature of critical infrastructure control systems and demonstrate the use of computational intelligence to manage policy and coordinate assets in MAS event-based dynamics (management, coordination), and time-based dynamics (execution). In addition, they analyze the MAS's optimal stabilization and propose that graph theory be applied to the MAS's execution layers. Robust control, analysis, recovery, and operation of mobile robot networks in time-varying tracking under global positioning attacks are studied [135]. Local and global tracking control methods provide redundancy to the mobile robot network and keep the intended functionality for improved resilience. The boundedness of the formation tracking error and the network's stability under multiple attack types are demonstrated using Lyapunov stability analysis.

WMR path-tracking control has attracted much research due to its wide range of applications, such as intelligent wheelchairs and exploration-assistant remote WMR [46]. With the rise in remote and autonomous operations/requirements for WMR, IoT devices are increasingly being used in the control loop. As a result, false data injection attacks (FDIA) provide interfaces for harmful activities. FDIAs have been proven to have devastating repercussions on feedback control systems. This work focuses on strengthening the robustness of dynamical observers against

FDIA since these attacks target the system measurement process. In particular, we present an attack-resistant pruning method to keep compromised channels out of the observer's processing.

Because vital infrastructure, such as control systems, has been computerized, they now interact with information technology to make them vulnerable to malicious attacks [136]. IT security emphasizes data correctness, which can be achieved by simple error correction techniques like packet re-transmission. On the other hand, control systems prioritize timely and accurate delivery of control signals, which can be harmed by delays or re-transmissions. This issue necessitates the development of resilient control algorithms that ensure the regular functioning of critical infrastructure in the face of malicious attacks and interruptions at both the physical and communication layers. A distributed formation control technique in which each agent relies on local knowledge and information from one neighbour to perform the group task cooperatively.

2.3.5 Vision-based control

Visual-based navigation marks a significant milestone in robotic engineering and has revolutionized how we control WMR. Vision is one of the most important senses of humans and animals, leading to better environmental comprehension. Likewise, digital cameras are ubiquitous for robots to understand and recognize the environment. Robots equipped with cameras can receive environmental information and correctly recognize changing circumstances. This information can be used as feedback to minimize errors. Some mobile robots are also designed for natural disasters; for instance, a rescue robot can find all injured people with a digital camera in case of an earthquake. With low prices, computer vision can be a valuable source of information without contact with the environment. The visual system is one of the most critical means of communication for receiving environmental information. The camera can have a fixed position or be mounted on the robot, called the eye-in-hand system [137].

Vision-based robot control (also known as visual-servoing) is an area of research that uses received information from a camera to control the robot [138, 139]. Position-Based Visual Servoing (PBVS) and Image-Based Visual Servoing (IBVS) are the two main visual servoing techniques, shown in Figure 2.14. In PBVS, the control law is constructed by an error between the current pose (position and orientation) and the desired pose in a 3D cartesian coordinate system [140]. The object’s current pose should be reconstructed from the image features with respect to the robot in each sampling time, prone to camera calibration errors and image noise. IBVS is more robust than PBVS since its error is the difference between features (namely straight lines, segments,

or points) of the current image and features of the desired image in a 2D plane [141]. Some factors severely affect IBVS performance, such as the singularity of the image Jacobian matrix, trapping in a local minimum, and the field of view (FOV) constraint violation [142]. Meanwhile, if the error between the current and desired feature is large, the loss of the features can occur or increase the Jacobian matrix's online calculation [143].

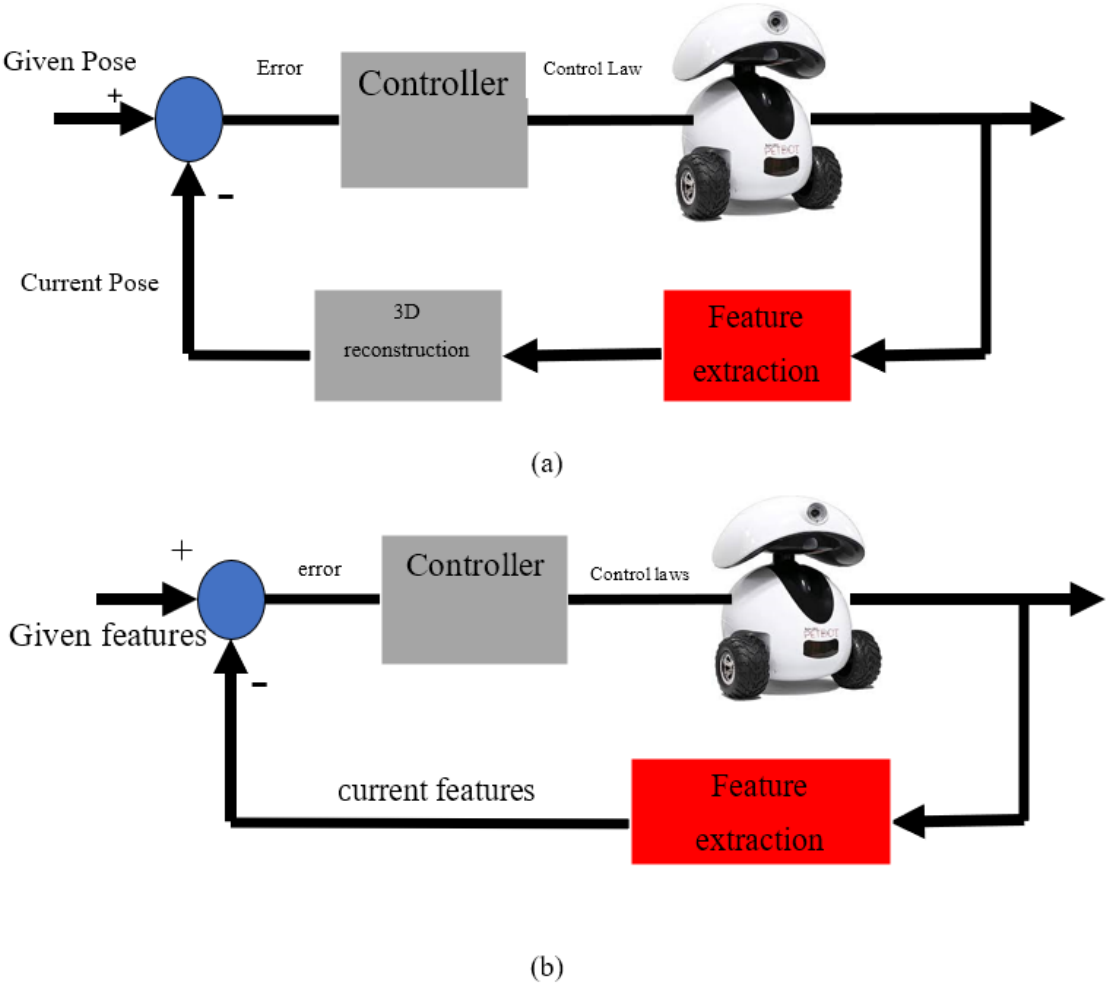


Figure 2.14: Visual servoing: a) PBVS framework b) IBVS framework

For a WMR with an onboard camera, a unified tracking and regulating visual servoing method is given. The 2-1/2-D visual servoing scheme aids in keeping targets in the camera's field of vision [144]. The proposed unified controller exhibits asymptotic stability even in the presence of uncertainties in the object model and depth information, as demonstrated by Lyapunov analysis.

A new vision-based controller is presented for wheeled mobile robots with a fixed monocular camera that can accurately execute autonomous navigation in agricultural regions [141]. The presence of uncertainties in the parameters of the robot-camera system and external

disturbances produced by high driving velocities, sparse plants, and terrain unevenness were studied in this research. The researchers then developed a resilient IBVS strategy for robot motion stabilization based on the SMC method, even in the face of such imperfections and disturbances. They used the Lyapunov stability theory to confirm the overall closed-loop system's stability and robustness.

A virtual-goal-guided rapidly exploring random tree (RRT)-based visual servoing strategy for nonholonomic mobile robots is suggested in research work to concurrently meet FOV constraint and velocity constraints during mobility toward the target pose [145]. The proposed method is divided into two sections: 1) trajectory planning in scaled Euclidean space, and 2) trajectory tracking control.

In another paper, a policy-based deep reinforcement learning (DRL) technique is used to investigate the image-based visual servoing (IBVS) topic for mobile robots with visibility constraints [146]. The conventional IBVS (C-IBVS) approach and associated feature-loss problems are presented first. Then, to overcome the feature-loss problem and enhance servo efficiency, a DRL-based IBVS approach is provided. To ensure analytical stability, the designed controller inherits the C-IBVS controller formulation. A policy-based DRL algorithm is proposed to design an adaptive law for tuning the controller gain in the continuous space, maintaining the feature in the camera's field of view while improving servo efficiency.

Another study examines the fixed-time tracking control issue for a nonholonomic wheeled mobile robot using visual servoing [139]. The robot system model with uncalibrated camera parameters is initially shown using the pinhole camera model. A tracking error system for the mobile robot and the target trajectory is provided. Finally, fixed-time tracking control laws for the mobile robot are suggested based on fixed-time control theory and Lyapunov stability analysis, allowing the robot to track the reference trajectory in a fixed amount of time. The fixed-time control settling time is governed only by the controller parameters and is independent of the system's initial conditions.

2.3.6 Discussions

The studied control approaches for WMR in the previous section were compared in facing nonlinear dynamics, constraints, robustness, and MIMO nature. Five controllers, namely FBL, SMC, fuzzy logic controller, ANN controller, and MPC, can be applied for a WMR independently, but the other two controllers, i.e., resilient control and vision-based control, are often integrated

with other control strategies. In controller design, real-world limitations (FOV and nonholonomic constraints) must be considered. MPC, one of the famous controllers for fulfilling input and output constraints, gives an adequate solution to this problem. A tentative conclusion is that MPC is the only technique to consider constraints in its design. When confronted with physical or cyber threats, MPC might be combined with a resilient strategy to stay steady and safe. Additionally, the LPV framework can help MPC be effective for nonlinear systems. The control techniques for WMR are briefly reviewed in Table 2.1.

Table 2.1: a brief review of control approaches for WMR

Technique	Examples	Pros and Cons
Feedback Linearized Control (FBL)	PI-based FBL MPC-based FBL Robust FBL State-space FBL Input-output FBL	It approximates the nonlinear model better than classical linearization. Linearization is not always possible. It is unable to handle constraints and robustness independently. MIMO systems can only be modelled using state-space FBL.
Sliding Mode Control (SMC)	Integral SMC Quasi-SMC Fuzzy-based SMC	It can be used for nonlinear systems. It can cope with uncertainties. The main issue is the chattering effect. Considering constraints complicates the controller design.
Fuzzy Logic Control	Type-1 fuzzy controller Adaptive fuzzy controller Type-2 fuzzy controller Fuzzy-PID controller	It does not require an accurate model of the system. It is suitable for MIMO and nonlinear systems. It necessitates in-depth knowledge of the system. It is easy to design and implement, especially for obstacle avoidance. Considering constraints complicates the controller design.
ANN-based Control	ANN-PID controller ANFIS controller	It can be used for MIMO and complex nonlinear systems. It does not need expert knowledge of the system. Considering constraints complicates the controller design.
Resilient Control	Resilient PID controller Resilient MPC	It is typically used in conjunction with other techniques to consider all kinds of uncertainties, especially cyber attacks.

Vision-based Control	PBVS IBVS	It is typically used with other techniques to help the robot better understand its surroundings, especially when both dynamic and static obstacles exist.
MPC	Min-max MPC Tube-based MPC LPV-based MPC Explicit MPC	It can deal with constrained MIMO systems. The computation load increases for nonlinear systems.

The first applications of MPC were in process industry control [50] due to slow dynamics. Using MPC in robotics was impossible due to the need for fast processors. With the rapid development of fast processors and numerical computing algorithms, MPC's use in semi-fast systems, like mobile robots, has become possible. MPC is now used to control various systems [51-53]. Figure 2.15 and Figure 2.16 indicate an upward trend in the number of research works for MPC for robots and MPC for WMRs, respectively. The search terms were ("control" AND "robot" AND "predictive" OR "mpc") and ("control" AND "wheeled" AND "robot"+ AND "predictive" OR "mpc").

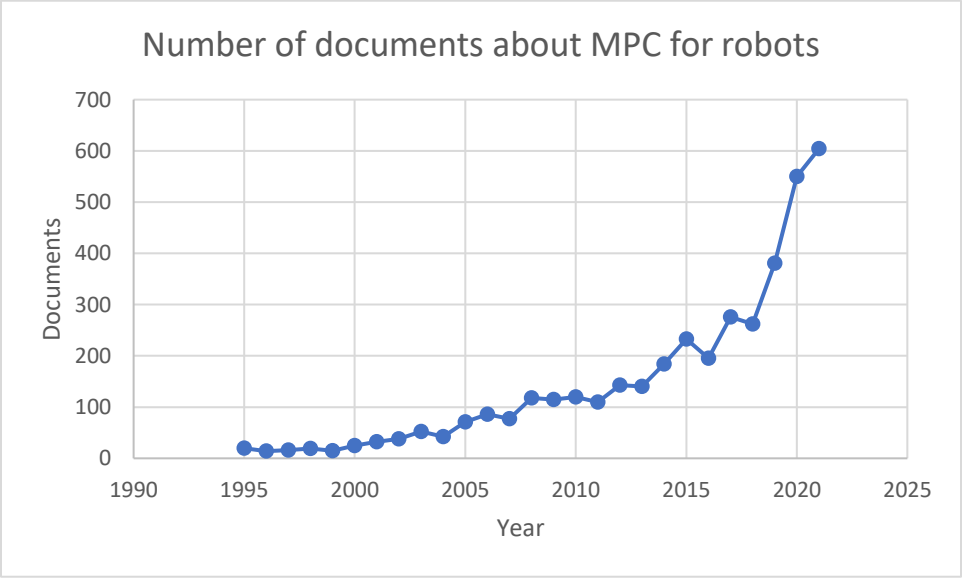


Figure 2.15: Documents per year for WMR controlled by MPC

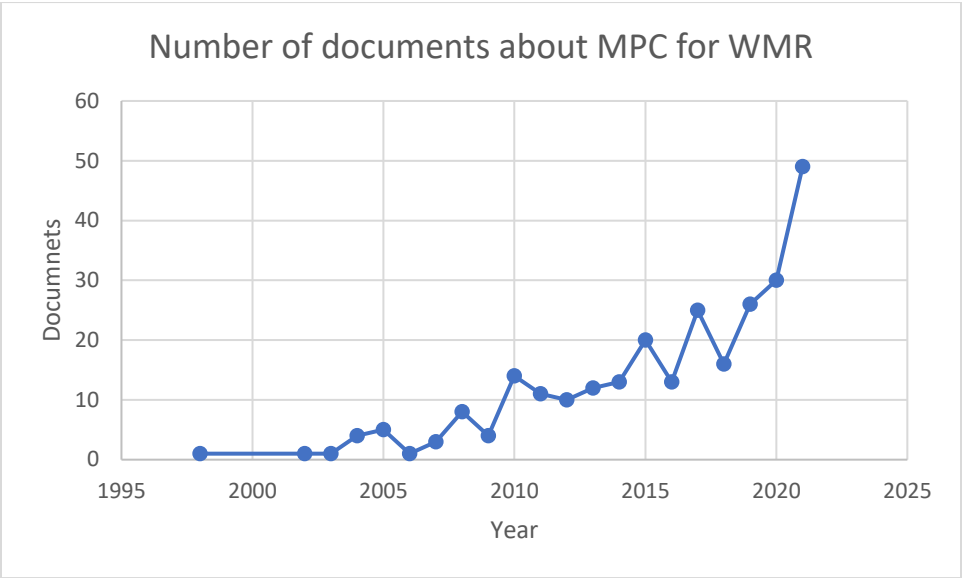


Figure 2.16: Documents per year for WMR controlled by MPC

The majority of contemporary MPC-based vehicle navigation techniques use linear kinematic models. While several path planning algorithms exist for vehicles with nonholonomic kinematics, it is often more difficult to demonstrate stability and robustness. A nonlinear trajectory tracking system can be employed to ensure that the real state converges to the nominal state.

Nonlinear MPC (NMPC) was created to cope with systems with nonlinear dynamics [147]. NMPC has two major drawbacks: greater algorithm complexity and an excessive computing burden. MPC based on an LPV system is brought to researchers' attention to attain a compromise between linear and nonlinear techniques [63, 66, 148]. The nonlinear/time-varying aspects of the system are taken into account in LPV models, which have a linear representation. By representing nonlinear systems with a linear form with dynamic parameters, also known as scheduling variables, LPVs bridge the gap between linear and nonlinear systems [149, 150]. In other words, scheduling variables, such as nonlinear dynamic system characteristics, environmental conditions, or operational points, might affect the linear model and controller. In brief, LPV helps MPC deal with nonlinear/time-varying systems. The principles of MPC, LPV and recent studies are reviewed in the next section.

2.4 Literature review for LPV-based MPC

The current LPV-based MPC techniques are studied here, including trackable feedback controllers, optimized feedback linearization controllers, interpolation-based controllers,

predictive dynamics controllers, and tube-based controllers. We also investigate the tracking problem for LPV-based MPC techniques. We finally summarized the findings in Table 2.2.

Computational softness in dynamic programming has been the source of inspiration for many studies to find a way to strike a good balance between achievable performance and computational complexity [151-153]. On the one hand, there are min-max methods with complex calculations and the best possible performance, and on the other hand, there are open-loop methods with low computations and poor performance. Tractable feedback methods look for something between computational complexity [154] and achievable control performance [155].

Concerning MPC, achievable control performance refers to the degree of conservatism. A non- conservatism controller will indeed find a solution if there is one. In contrast, a conservatize controller only looks for a solution in a particular class and may not find one. In addition, by limiting the solution class, the controller may not find the best possible answer, which is not optimal (for example, control inputs with more required power consumption). In the following, we will examine tractable feedback methods. This study will consider three features: 1) applicability of the method for LPV systems, 2) computational features, and 3) ability to include information about future changes in scheduling signals. Some methods we will explain in the following sections have not been used directly for LPV systems but have been developed for other systems, such as linear systems with disturbance, uncertainty, or linear parametrically uncertain (LPU).

2.4.1 Optimized feedback linearization strategies

One of the first works in robust MPC that has impressed many researchers is replacing nonlinear feedback with a linear, stable square robust feedback strategy. This feedback is calculated online to solve linear matrix inequalities (LMIs). Meeting constraints is obtained by finding the system state variables within an appropriate level of a square Lyapunov function corresponding to the optimized state feedback. This method can be used for systems with parametric uncertainty, which can be described as polytopic or with norm bounded.

This idea [156] has been improved in several ways, such as the quasi-min-max method [61], which generalizes polytopic LPV modes in which scheduling signal measurement is available. These measurements are extracted by extending the policy [156] with a free control step and optimizing the state feedback with dependent parameters. Another generalization, with two additional free control steps, has been developed by [157], which considers LPV systems with a limited rate of change in the scheduling signal.

The degree of freedom in [156] is constant and equal to zero; in [61], it is one, and in [157], it is two. These policies are computationally simple but place constraints on the structure of the relationship. In addition, small, fixed horizons give us little information about the future of scheduling paths.

Several studies [158, 159] extended the technique of [156] to an optional prediction horizon, performed by placing the N control step on the linear state feed. In these methods, similar to the [156] process, the end-state feedback is optimized at any time. In [160], this method has been extended to LPV systems.

All of the above methods must be squarely stable, i.e., linear state feedback must correspond to the quadratic Lyapunov function. This requirement is simplified in [161], and Lyapunov functions are considered with variable parameters. This method has an optional N prediction horizon. It suggests possibilities for extracting information about future scheduling paths. However, these algorithms suffer from high computational loads that increase exponentially with N. it is shown that this exponential growth in LPV systems can be prevented by assuming that the uncertainty is limited or taking a small boundary for scheduling variables [162].

Another study used parameter-dependent dynamic output feedback (PDDOF) laws to explore MPC for a quasi-LPV model with norm-bounded disturbance [163]. In contrast to the typical single-step strategy, a periodic approach is used in this article, in which a series of PDDOF rules is applied regularly in future projections. This periodic method has significantly bigger attraction zones and superior control performance than the single-step technique. The enhanced state and output/input convergence, as well as recursive feasibility, are ensured.

The challenges of robust output feedback MPC for constrained LPV systems exposed to bounded state and measurement disturbances are addressed in this paper [64]. The scheduling parameter vector is supposed to be an unmeasurable signal that takes values from a compact set. The proposed controller includes an interval observer, which uses available measurements to update state estimations, and an interval predictor utilized in the MPC algorithm's prediction phase. The MPC method assures recursive feasibility, constraint meeting, and input-to-state stability in the terminal set. In addition, this unique approach has low computational complexity and is simple to implement (similar to conventional MPC schemes).

A synthesis technique of dynamic output feedback robust MPC with input saturation is examined for LPV systems with uncertain scheduling parameters and bounded disturbance [164].

A key optimization challenge is tackled via convex optimization to lessen the online computing burden by pre-specifying controller parameters. The primary optimization issue ensures that the estimated state and estimation error converges inside the invariant sets, ensuring recursive feasibility and robust stability.

2.4.2 Interpolation

In interpolation-based MPC, control signals were derived from interpolation between several precalculated control gains. If this strategy is implemented, the online calculation will be moderated, and conservatism will be improved through an increased attraction region. The general method of interpolation MPC for LPV systems is investigated in [149, 165-167], in which several pairs of linear controllers corresponding to an ellipsoidal invariant set are calculated offline. This method contrasts with the previous form of "optimized feedback linearization strategies, in which the final sets and the corresponding controller were calculated online as part of the optimization. The current state variables are measured online, and then the control signal is generated by a suitable linear combination of the control inputs of the local controllers. Each controller has a control gain and a final set. The advantages of the interpolation method include reducing online computing and increasing the stability area. In these methods, the prediction horizon is zero, so there is little ability to predict future changes in the scheduling signal.

The interpolation MPC technique for nonlinear discrete-time systems described by the affine LPV model is investigated in this study [168]. The general nonlinear model is turned into the quasi-LPV model, which is then used to create the equivalent polytopic LPV model and the disturbed Linear time-invariant (LTI) model. An ellipsoidal invariant set based on a finite-horizon interpolation MPC technique is suggested. The feasible zone of the presented method is substantially more extensive than that of the zero-horizon interpolation MPC algorithm due to the finite-horizon approach.

To cope with constrained nonlinear MIMO systems, an LPV model was supplemented with an MPC to decrease computation and conservatism [149]. The LPV-MPC was built in an input-output architecture that eliminated the need for state measurement. Third, for conservatism, an interpolation-based MPC (IMPC) was included, in which control signals were created through interpolation between various precalculated control gains. If this method is followed, the online calculation will be tempered, and conservatism will improve due to a larger attraction zone. Two numerical examples were used to test the efficacy of the suggested strategy extensively. The results

demonstrated that the technique performed excellently in setpoint tracking and disturbance rejection, but it was also very tractable when online computation and conservatism were taken into account.

2.4.3 Predictive dynamics controller

LMIs were previously utilized to build a fixed state-feedback law that keeps the state vector inside feasible invariant sets. The state feedback law is considered to be constant at each time instant. This assumption imposes i) limits on achievable performance, ii) restrictions on the size of the allowable set of initial conditions, and iii) significant online computational demands [156]. The authors overcome these problems by using invariant sets that capture the system predictions, even during transients when the control is not a fixed state-feedback law. A new strategy is proposed in this paper that uses a constant state-feedback law but adds extra degrees of freedom by using perturbations on the fixed state-feedback law.

Unlike [156], who used a single quadratic Lyapunov function, [169] used several quadratic Lyapunov functions, each corresponding to a vertex of the polytope, to extend the result. To reduce conservativeness, [170] used nonlinear parameter-dependent Lyapunov functions.

Some studies have also been performed to optimize the offline design process [158], maximizing the size of the invariable ellipsoidal set, and resulting in a less conservative MPC. Other methods have been used to reduce conservatism, whose overall goal is to reduce the sensitivity of the closed-loop cost to model uncertainty [171]. In [172], the method [158] is modified using polygonal invariable sets instead of ellipsoidal sets. Because the state dimensions of the lifted system grow linearly with N, the complexity of the polygonal set in the worst case grows exponentially relative to N. In [172], it is shown that this computational complexity can be reduced under certain conditions, so that the complexity changes linearly with N.

An MPC for constrained nonlinear MIMO systems subjected to bounded disturbances was developed in another study. This study represents the nonlinear process by an LPV based on previous input-output data (LPV-IO). The primary goals of this study are to reduce the online computational load of MPC with an LPV-IO model compared to the existing literature and to confirm the controller's robustness in the presence of disturbance. A recurrent neural network (RNN) solves real-time optimization problems with less online computation for the first goal. A new control law is developed with a fixed control gain (K) and a free perturbation (C). Because of

a larger possible terminal region and free control moves, the proposed method has a reduced conservatism [150].

Another study presented output feedback robust MPC for an LPV model with norm-bounded unknown disturbance by parameterizing the infinite horizon control moves and estimated states into one free control move, one free estimated state, and a dynamic output feedback law [63]. This model first attempted to use the free control move and free estimated state in the output feedback MPC. The procedure was demonstrated to be recursively feasible, and the system state was guaranteed to converge to the vicinity of the equilibrium point.

The MPC technique often assumes that the control action can be determined instantly using the system information acquired at each sampling time [63]. However, the capacity of this technique to tackle the optimization issue in real-time limits its applicability. An enhanced online technique is provided in this study, in which the controller parameters optimized from the previous sampling interval are used to compute the current control action, i.e., the control action is executed one step ahead. This one-step forward method addresses the optimization issue during the sampling interval, implying that the controller and the real system operate concurrently. It is also important to note that fully lifted state vectors (including inputs) must be included in the invariable set, calculated offline, as part of the controller design. Hence, there is no straightforward method by which these algorithms can apply existing knowledge of possible future scheduling paths while the system is running. [173-175] explored more recent developments in predictive dynamics techniques for LPV systems.

2.4.4 Tube-based methods

The tube-based MPC (TMPC) paradigm [176] is based on the idea that processes with unknown (but bounded) uncertainty yield a finite range of different trajectories. These trajectories, sometimes called "tubes," relate to a specific representation of the uncertainty set. The controller must calculate a tube in the tube-based design paradigm so that all potential state trajectories remain inside this feature for all possible realizations of the (bounded) disturbances and uncertainties while retaining control performance. TMPC reduces dynamic scheduling complexity. These methods were initially used to control linear systems with disturbances. Despite the differences in the controller form, they all had the same complexity when estimating the future uncertain paths. By doing this, the complexity of the predicted paths will no longer grow exponentially. In tube-based methods, a range is set for future state variables that can be calculated online or offline.

TMPC techniques have been applied to linear systems with uncertainty [177, 178], where predicted paths with uncertainty are estimated using pre-designed multi-criteria versions in these works. Because the complexity of these sets is constant over the prediction horizon, the size of the overall optimization problem to be solved increases linearly with N. Because of this linear scale behaviour, tube-based frameworks can also include information about the future of scheduling paths.

MPC for currently available LPV systems presumes that the scheduling trajectory's future behaviour is unknown across the prediction horizon [179]. An anticipative TMPC method for polytopic LPV systems with complete state feedback is proposed in this study. Unlike previous approaches, the method explicitly considers expected future variations in the scheduling variable; its current value is measured precisely, while future values over the prediction horizon are defined as a series of sets describing expected deviations from a nominal trajectory.

For LPV systems with bounded disturbances and noises, this paper presents a solution to tube-based output feedback robust MPC (RMPC) [65]. The suggested method combines an offline optimization problem to design a look-up table and an online tube-based output feedback RMPC with tighter constraints and scaled terminal constraint sets. The proposed online optimization problem has a smaller online computational with fewer choice variables and constraints. The optimization problem will be recursive, and the controlled LPV system will be stable.

Another work offered a TMPC for polytopic LPV systems exposed to bounded additive disturbances [68]. The suggested technique described future scheduling evolutions by a known nominal trajectory and constant uncertainty sets around the nominal signals.

2.4.5 Tracking

All of the above techniques focus on stability, and this section investigates the tracking problem. Robust MPC tracking methods for LPU systems have been developed [180, 181], and it has been shown that offset tracking can be achieved if the uncertainty (corresponding to θ in LPV systems) is unchanged over time. In [182], the tracking of piecewise constant reference for LTI systems is considered. The exponential convergence of the outputs to the boundary region around the reference is ensured using tube-based methods. In [61], MPC is used for nonlinear systems modelled as LPVs (a family of linear systems around working points). It is shown that the closed-loop system is asymptotic stable if the optimization is viable. In [183], nonlinear systems are well-

modelled LPVs with discrete scheduling signals, and it is proved that if the problem can be optimized, the stable closed loop is asymptotic.

Because of changes in the scheduling signal, offset-free output tracking is often not achievable in MPC based on LPV state-space representations [184]. As a result, they attempted to ensure a pre-specified tracking error bound that is feasible for all permissible scheduling variable changes. A tube-based model predictive controller was constructed to bring the system's state into this set in a finite time. Similarly, how to employ quasi-LPV representations to efficiently construct Nonlinear MPC for reference tracking in the presence of nonlinear input and state constraints is described [62]. Standard QP solvers were employed for the online optimization, making the solution highly efficient and practical even for rapid plants.

Solar collector temperature control is another nonlinear problem described by an LPV model and controlled by an MPC [185]. At each instant, two QP are solved: the first considers a backward horizon of steps to find a virtual model-process tuning variable that defines the best LTI prediction model, taking into account the polytopic system's vertices; the second QP then optimizes performance along a forward horizon of steps using this LTI model.

Table 2.2: A literature review of MPC for LPV systems (SS: state-space, IO: input-output)

Method	Reference	System	Model	Uncertainty	Disturbance	Tracking	comments
Optimized feedback linearization strategies	[156]	Linear	SS	Polytopic Feedback	Yes	Constant	No free control moves
	[61]	Linear	SS	Polytopic	No	No	One free control moves No limitation on θ
	[157]	Linear	SS	Polytopic	No	No	Two free control moves limitation on θ
	[158]	Linear	SS	Polytopic	No	No	N free control move Ellipsoidal invariant set
	[159]	Nonlinear	SS	Polytopic	No	Constant	Constrained inputs and state No limitation on θ

	[160]	Nonlinear	SS	Polytopic	No	Constant	Constrained inputs limitation on $\Delta\theta$
	[161]	Linear	SS	Polytopic	No	No	Less conservative method Parameter-dependent control law and Lyapunov function
	[162]	Linear	SS	Norm bounded	No	No	Free prediction horizon The number of LMI linearly increases with the prediction horizon
	[165]	Nonlinear	LFR	Norm bounded	No	Constant	limitation on $\Delta\theta$
	[164]	Linear	SS	Polytopic Additive	Yes	No	Input saturation Dynamic output feedback
	[163]	Linear	SS	Polytopic Norm bounded	Yes	No	Periodic approach
	[64]	Linear	SS	Polytopic Additive	Yes	No	Interval observer for future state prediction
Interpolation	[186]	Linear	SS	Polytopic	No	No	Nonlinear control law (a combination of predetermined controllers) to lower computation load
	[187]	Linear	SS	Polytopic	No	No	Variable prediction

							horizon to reduce computational complexity
	[166]	Linear	SS	Polytopic	No	No	Considering four techniques for finding an Ellipsoidal invariant set and state feedback gain
	[168]	Nonlinear	SS	Polytopic Additive	Yes	No	ellipsoidal invariant set nonlinear systems
	[167]	Linear	SS	Polytopic Persistent disturbances	Yes	No	ellipsoidal invariant set optimal interpolation coefficients
	[149]	Nonlinear	IO	Polytopic	Yes	Yes	ellipsoidal invariant set reduced conservatism degree and computation load
Predictive dynamics	[172]	Linear	SS	Polytopic	No	No	
	[171]	Linear	SS	Polytopic	No	No	Less computation Less conservative
	[63]	Linear	SS	Polytopic	Yes	No	Norm-bounded unknown disturbance Output feedback
	[150]	Nonlinear	IO	Polytopic Additive	Yes	Yes	Less computation Robustness

Tube-based MPC	[188]	Linear	SS	Multicaptive	No	No	Less online parameters
	[178]	Linear	SS	Multicaptive Additive	Unmeasurable	No	Increased ellipsoidal invariant set
	[179]	Linear	SS	Polytopic	No	No	Anticipates future changes in the system dynamics
	[65]	Linear	SS	Polytopic Additive	Yes	No	Scaled terminal sets Tightened constraints
	[68]	Linear	SS	Polytopic Additive	Yes	No	future scheduling evolutions largest allowable disturbance
Tracking	[180]	Linear	SS	Polytopic	No	No	Constrained inputs
	[184]	Linear	SS	Polytopic	No	Constant	Tube-based control
	[62]	Linear	SS	Polytopic	No	Time variant reference	Fast tracking Nonlinear constraints
	[185]	Nonlinear	SS	Polytopic	Yes	Constant	Two QPs Reduced computation load

2.5 MPC for WMR

After introducing the principle of MPC and LPV and reviewing the literature on LPV-based MPC, we investigate the trajectory tracking problem in WMRs using MPC. This involves

examining closed-loop stability, online optimization problem, and modelling. Additionally, we highlight the significance of LPV-based MPC for WMR.

2.5.1 MPC

Control laws can effectively address the problem of path following or trajectory tracking in autonomous devices. However, nonlinear modes and inherent uncertainties (friction, terrain, and slippage) pose challenges that can make the path-tracking problem more complex, potentially leading to issues in the closed-loop system and requiring significant control signals. The ultimate aim is to develop a control law to guide the device along a specific path while considering factors such as response speed, accuracy, field of view, mobility constraints, computational costs, energy requirements, communication channels, and stability. To this end, various MPC algorithms have been developed based on the process description model, noise signal, disturbances, cost function, and optimization algorithm.

According to [189], the MPC problem can be expressed as an online optimal open-loop control problem with a finite horizon based on system dynamics and constraints, including states and control sequences. If there is no difference between the real system and the intended model and there is no disturbance, the input function found at time $t = 0$ can be used for the system at all times $t>0$. However, due to the difference between the real system and the considered model and disturbances, the real system's behaviour differs from the predicted behaviour. Suppose the criterion used is a cost function of the second-order form and a linear model is assumed, and there are no constraints on the system. In that case, the explicit solution can be applied online by implementing a lookup table (different gains for different operation points) or an explicit function of the past inputs, outputs, and path [2]. When nonlinear constraints are present, or the cost function does not have a specific form, online real-time optimization methods with numerical techniques are often necessary. Although an infinite predictive horizon can guarantee system stability, it is not a practical solution for control systems. To ensure the convergence of the optimization problem and achieve control signals with smooth transitions, it is common to include a term that penalizes the control law. This approach helps to ensure that the control signals remain within a certain range.

In an NMPC, the model, constraints, and cost function can be nonlinear, bringing new problems. Firstly, the identification of an accurate nonlinear model can be demanding. Additionally, using nonlinear models/cost functions/constraints makes the optimization problem more complicated and might result in a nonconvex optimization problem. The optimization

problem might be computationally ineffective, and there is no guarantee of finding a global optimum [190]. In short, in real-time control, where the optimum point must be achieved online, NMPC experiences difficulty finding optimum control action sequences, dealing with high computation load, and finding an accurate model that reliably predicts systems behaviour [2].

Regarding WMR, the constraints can be holonomic or nonholonomic. Holonomic constraints (related to positions such as x and y) restrict the robots' achievable poses. i.e., there are some banned areas where robots cannot enter that area. Nonholonomic constraints (related to velocity) do not limit accessible poses but restrict the path that can be followed to reach the destination. Dealing with nonholonomic constraints is troublesome and makes controller design more complex. According to Brockett's theorem, no continuous time-invariant feedback of state variables can be found asymptotically stabilizes the non-holonomic system around the equilibrium point [48, 49]. For the past 20 years, a great deal of research has been conducted into MPC theory for non-holonomic systems concerning this theorem. Briefly, MPC research works on non-holonomic robots can be divided into three categories:

1. Closed-loop stability
2. Optimization problem
3. Model

2.5.2 Closed-loop stability

The nonlinear model and nonholonomic constraints make the closed-loop stability of WMR challenging. It has been said that moving differential robots cannot be stabilized by a control law that does not change over time with feedback [191]. The WMR model is intimately linked to stability. Here, the stability problem has been investigated for linear MPC, modified linear MPC, and NMPC. In linear cases, asymptotical stability can be guaranteed by designing bounds for the cost function and finding the appropriate prediction horizon. So far, linear MPC theory has been fully developed [2], and essential issues such as stability have been studied for a broad spectrum of systems [192]. Ensuring stability will be more complicated if the nonlinear model depicts the WMR behaviour. A terminal cost function is typically added to the cost function in nonlinear conditions, corresponding to a terminal region [60]. The problem of finding terminal region (offline/online)[59], type of terminal regions (ellipsoidal invariant set/polyhedral invariant set) [193, 194], type of cost function (quadratic/nonquadratic) [195], and conservatism [196] are usually discussed in the second group (NMPC). Some of the selected studies are as follows:

[197] introduced one of the first implementations of NMPC for a WMR. The main idea was to use the device motion model and immediately calculate the optimal control sequence to minimize the projected motion tracking error. In the presence of an obstacle, the controller deviates from the original reference path and creates a new reference path with the help of the information derived from sensors. It is also claimed that the control horizon and weight matrix directly affect the position and velocity tracking accuracy and obstacle avoidance.

Concerning finding the terminal region, an MPC has been developed by [198] for tracking and regulating a non-holonomic mobile robot. The stability of the control is guaranteed by adding a terminal state penalty to the cost function and keeping the final state bounded. Circle curve tracking, eight-shaped curve tracking, and parallel parking line curve tracking have been studied. Meanwhile, a suboptimal solution has been provided to reduce computational load. In [199], a non-holonomic mobile robot's (four wheels with differential derive) regulation problem is developed with an MPC, stabilized by adding a terminal penalty to the cost function and corresponding terminal region. A suboptimal solution is regarded due to the computational complexity of the nonlinear system. An open-source toolkit software for WMR has been developed in this paper. The approach has been assessed for circle trajectory tracking and eight-shaped curve tracking.

Regarding the type of terminal region, an ellipsoidal invariant set is suggested by [193] for object manipulation of a nonholonomic differential-drive robot. A time-varying MPC controlled the robot to deal with the motion constraint that considered the friction between the robot and pushed object. In contrast, multiple offline polyhedral invariant sets corresponding to multiple controllers were developed to reduce online computation load [194]. The online controller was derived from a lookup table of predetermined controllers. In general, terminal constraints can be defined in such a way as to ensure asymptotic stability [200]. Many authors have suggested modifications in the terminal region [200-203] to reduce the degree of conservatism. [201] mentioned that an added terminal region usually ensures the stability of NMPC to the cost function. However, finding terminal region increase computation in online applications. The author took the effort out of the terminal region's online calculation since the attraction region might degrade in offline cases. A wheeled mobile robot's tracking and regulation problem is situationally tackled by a modified terminal region obtained from a T-S fuzzy model [202]. The controller has been tested in a circle reference trajectory. A randomized algorithm is proposed in [200] to design MPC. Making a trade-off between computation and closed-loop performance is the merit of this work. A

navigation function is used as the Lyapunov function, and state constraints are added to the cost function.

Next, we looked at studies focused on the cost function. An MPC approach with the LMI framework is presented in [204] for the tracking problem of a three-wheeled omnidirectional robot. Many convex optimization problems can be formulated as an LMI problem, particularly with uncertainties. The LMI problem is also called the worst-case or min-max optimization problem, where the upper bound of the cost function should be minimized. Considering input and output constraints, an upper bound to a quadratic objective function was defined to ensure stability. The control law would be in terms of $F = YQ^{-1}$ instead of $U = -Kx$ (for more details, refer to [204]). [195] offers a form of MPC based on a nonquadratic cost function to steer a unicycle non-holonomic robot to the desired position and orientation as follows:

$$l = q_1 x^4 + q_2 y^2 + q_3 \theta^4 + r_1 v^4 + r_2 \omega^4$$

Where (x, y) and θ are the position and orientation of the robot. $q_1, q_2, q_3, r_1, and r_2$ are the weighting matrix of inputs and outputs. The control signal $u = (v, \omega)$ includes the linear and angular speed of the robot. Without considering stabilizing constraints and additional costs, asymptotical stability is assured in this paper, which reduces the controller's conservatism rate. This paper has been extended by [205] without considering stabilizing the terminal condition. Similarly, this study also considered state constraints and a dynamic model in controller design. Meanwhile, a technique is designed to find the required length of the prediction horizon with sufficient stage cost and region of attraction.

Finally, we searched conservatism degree in controller design. A tube-based MPC was studied by [196], where an observer estimates outputs. The authors designed time-varying tubes to reduce conservatism. Meanwhile, computational complexity is moderately reduced due to tightened constraints. Likewise, a parameter-dependent control law is constructed for a WMR to reduce the degree of conservatism in an MPC. They employed an LPV framework to capture the nonlinear dynamics in the robot model.

2.5.3 Optimization problem

Various optimization algorithms can be used to solve the optimization problem of MPC for WMR regarding computation, cost function (linear, quadratic, nonlinear), and constraints diversity (mechanical, visibility). The optimization problem of MPC for WMR is traditionally solved by the sub-optimal method [206, 207] and genetic algorithm [103, 208].

Finding the global optimum in a standard MPC might be time-consuming. In contrast, a suboptimal MPC seeks a suboptimal solution with a lower iteration that declines calculation. A wide range of optimization methods for MPC is reviewed [209-212], including interior-point methods, active set methods, finite control set MPC, fast update method, and neural network methods. However, a few techniques have not been included in these review papers. For example, [213] proposed an NMPC for the trajectory tracking of a four-wheeled mobile robot, where the nonlinear cost function is online minimized by a conjugate gradient method, and argued that the result of this method was better than the steepest descent approach. It has been expressed that the optimization time would be less than 30 ms in each sampling time. The cost function consists of position error, orientation error, and control action. The same study examined a practical NMPC for a two-wheeled moving differential robot while preventing collisions with an obstacle [214]. With a time-varying reference path, the applied controller minimizes the cost function with discrete nonlinear online optimization using Lagrange multipliers on a predefined predictive horizon. The gradient descent method solves the QP problem, and input and output constraints are added to the cost function as a penalty term. An NMPC algorithm, based on Hamiltonian, has been proposed for online motion planning and obstacle avoidance of autonomous omnidirectional robots [215]. The cost function considered tracking error, control signal magnitude, and presence in areas including moving obstacles and lanes as a penalty. The ultimate goals of all optimization algorithms are to find the global minimum, deal with nonholonomic constraints, consider more realistic constraints, and have faster convergence while guaranteeing stability in the presence of an obstacle.

An RNN-based optimization is proposed in several works to reduce computation without deteriorating stability regarding the online computational burden. Thanks to global convergence and low computational burden, neural network-based optimizations have been applied for linear, quadratic [216, 217], nonlinear programming [218, 219], and variational inequality problems [220, 221]. The RNN has a convincing performance in either MPC or NMPC [222-224], in which nonlinear optimization can be converted to quadratic programming. In preliminary research on non-holonomic dynamic systems, neural network-based models were suggested to overcome the computational burden of prediction and minimization [225-227]. In this class of neural network-based MPC, the neural network was responsible for predicting the behaviour of WMR with real input/output data because they believed a better nonlinear model improves the optimization

problem. Others claimed that the first approach might suffer from nonconvex optimization or trapping in a local minimum and suggested a dual neural network for the optimization problem [228]. In this paper, the authors introduced an RNN approach to solving the real-time optimization problem in NMPC, rewritten as a second-order problem of two unknown parameters. A similar method has been presented in references [229] and [230].

Although various approaches have been proposed to solve the MPC's optimization problem along with stability for mobile robots, it is still challenging to design an optimal or near-optimal method for tracking in the presence of uncertainties and disturbances. A near-optimal tracking control method for a mobile robot based on the receding horizon (RH) dual heuristic programming has been introduced [231]. The main feature of this paper was considering uncertainties and disturbances in the QP. The authors employed a backpropagation controller to produce the desired velocity profile and an RH strategy for converting the infinite optimal control problem into a finite one. The closed-loop tracking equations are consecutively updated on each horizon. Another RH-based strategy is created to trace the reference path to reduce the effects of model errors on guide performance [232] by a sequence of control laws determined with Pontryagin's minimum principle. The control laws are designed to rectify path deviations due to the high device velocity when entering Mars, where the atmosphere is rarefied with high uncertainty. The nonlinear ordinary differential equations are rewritten in nonlinear algebraic equations through the bounded value problem (BVP), which lightens the significant computational load. Another example studied by [233], where a two-stage nonconvex nonlinear control approach for autonomous devices in the outdoor environment to track the road's center lines while preventing collisions with obstacles. The external loop creates a barrier-free path with inputs taken based on the device's simplified model, and the inner loop is responsible for path tracking. The moving robot's average speed was about 24 m/s, indicating a highly dynamic system. The recent development on robust MPC for WMR concentrated on parameter uncertainty [100], bounded time-varying additive uncertainty [234], and fast trajectory tracking in a noisy, uncertain environment [235].

2.5.4 Model

The MPC prediction model can be linear, nonlinear, or simplified to trade-off between complexity (computation, optimization) and precision. MPC is widely used for a group of problems in which the dynamics of the model under consideration are linear [196, 236]. However, there may be inherently nonlinear systems, like non-holonomic systems. Preliminary research assumed that

the model must be linearized to predict future behaviours [237]. The control law originated from a second-order cost function, which comprised system tracking error and control effort subjected to physical constraints. The apparent downside of linear MPC is that the model remains valid only around the operating point. Simply put, linear MPC depends on the system's operating point and can be used in a neighbourhood of the operating point.

Many researchers prefer to use modified MPCs with lower computation and trustable models. For example, in [238], a term was added to the cost function for obstacle avoidance by maximizing the distance between the robot and the obstacle. The cost function also considers the constraints as a penalty using the Karush-Kuhn-Tucker (KKT) condition. Some researchers have developed linear MPC for omnidirectional mobile robots [239-241] using linear time-varying systems and LMI constraints.

However, some researchers state that MPC is not a careful choice (even the time-varying linearized type) for the mobile robot problem due to the robot's uncontrollable linearized equations around the working point and the failure of stability hypotheses. They insist that the MMPC techniques need further examination [199]. Unlike MPC, the nonlinear controller does not depend on the system's operating point. Some consider nonholonomic constraints with a simplified robot model for state prediction, while the controller would be applied to real nonlinear WMR. For example, in [34], a nonlinear model is simplified by a bicycle model. The model is used in NMPC for state prediction. Two different methods of preventing obstacles are compared, and the controller is simulated in different scenarios, including static obstacles on restricted roads. Some studies insist on using nonlinear models; for example, an NMPC based on Gaussian model training has been described [242, 243] to model possible changes in system parameters. ANN is one of the most well-known algorithms for WMR modelling [102, 244-246]. For example, in [247], a car-like autonomous vehicle is guided by a GPC. A state-space model is used for the prediction of outputs without ANN. Noise removal and dealing with non-holonomic constraints are investigated in this paper. One of this study’s noticeable features is estimating position by an Extended Kalman Filter (EKF) extracting data from two different sensors. This study was improved in [102], where a new neural predictive control-based path tracking technique was developed. A multi-layer back-propagation neural network is used instead of a linear regression estimator to simulate non-linear kinematics to adapt the robot to a wide operating range. The neural predictive control for path tracking is a neural network-based model-based predictive control that may create output in terms

of robot kinematics and the desired path. In a similar work, an MPC is designed based on neural approximations to steer a nonholonomic wheel-legged robot in complex environments with mechanical model uncertainty and unknown disturbances [248]. Effective ways for dependable tracking control should be studied, keeping disturbances in mind in order to ensure the tracking performance of wheeled robots in an unpredictable environment. An adaptive MPC is established for a self-driving car in which the environment is continuously changing, and robots are subjected to a wide range of risks and disturbances [249]. The authors devised a PSO for tuning the MPC parameters for external disturbances and changing operating circumstances. ANN and ANFIS were used to provide the prediction model of MPC. Classic controllers are ineffective in such vehicles due to these characteristics, particularly lateral control. ANFIS-MPC was shown to be smooth yet demanding, whereas ANN-MPC proved to be more accurate. Further research will be required to focus on optimizing and learning the MPC prediction model under mixed longitudinal and lateral control.

Most notably, LPV, which is thoroughly detailed in section 2.4, is one of the approaches that has lately gained considerable interest and has been utilized to offer an accurate model for MPC.

2.6 LPV-based MPC for WMR

LPV-based MPC can deal with nonholonomic constraints, delays, and nonlinear dynamics in WMR. Figure 2.17 and Figure 2.18 are the results for search terms (“control” AND “wheeled” AND “robot” AND “LPV”) and (“control” AND “wheeled” AND “robot”+ AND “LPV” AND “MPC”), respectively. Because of the novel nature of this research topic, there are few publications on the subject, and numerous fields of research are still unexplored.

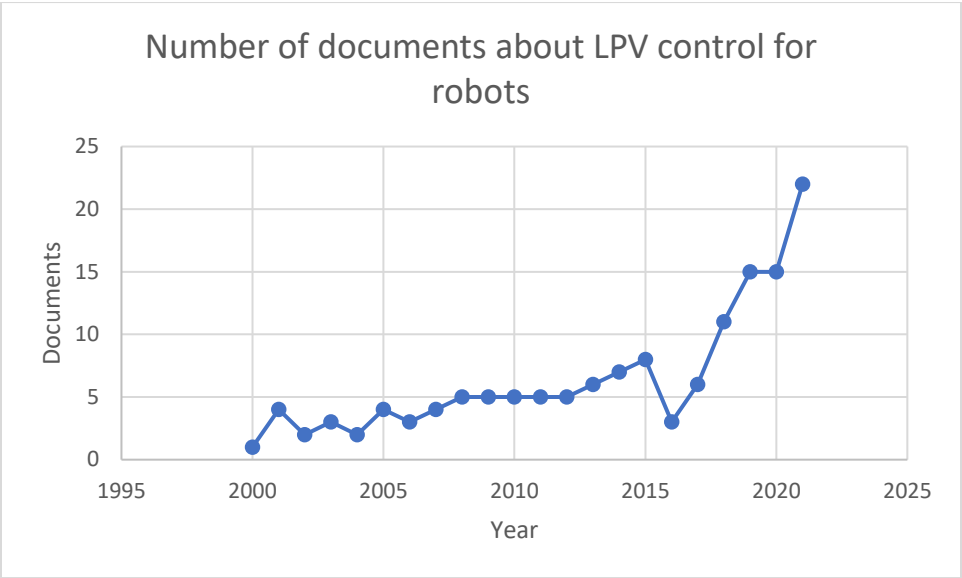


Figure 2.17: Documents per year for LPV control of robots

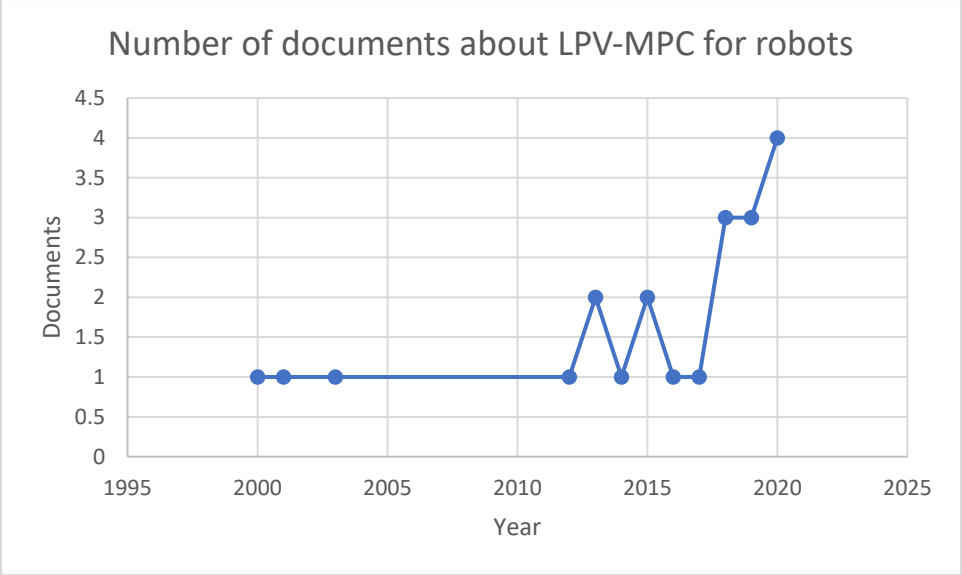


Figure 2.18: Documents per year for LPV-MPC of robots

Steering WMR using LPV-MPC is evolving gradually to solve problems of NMPC for WMR, namely computation load, conservatism, and stability.

We first investigated the online computation burden. For example, a cascade control is proposed to address the trajectory tracking problem for autonomous vehicles. The external loop solves position control using a novel LPV-MPC approach, and the internal loop dynamically controls the vehicle using an LPV-Linear Quadratic Regulator technique designed as an LMI problem called [250]. Both strategies use an LPV representation of the vehicle's kinematic and

dynamic models. The fundamental contribution of the LPV-MPC approach is its capacity to compute solutions that are extremely close to those found by the nonlinear version while greatly decreasing the computing cost and permitting the real-time operation. They presented a comparison of the NMPC formulation and the LPV-MPC technique to illustrate the potential of the LPV-MPC. Another cascade adaptive LPV-MPC technique was implemented for a three-wheeled omnidirectional mobile robot, subjected to incremental control input limitations and parameter uncertainties [100]. First, the intricate nonlinear dynamic model is converted into an LPV. About the control scheme, the outer loop was a kinematic-based control with PD feedback, and the inner loop was adaptive linear MPC. The controller was robust to parameter uncertainty, such as changes in the payload or ground conditions. The time-consuming parameter identification of the friction model is avoided in contrast to earlier investigations. Although the suggested control's stability and efficacy have been validated, the controller is dependent on the dynamic model, and parameter estimate errors remain. A technique of LPV control based on AI can be explored in the future to tackle this problem.

Concerning conservatism, a study described an MPC strategy for WMRs with various time scales and LPV [251]. The scheduling variable is believed to be able to be measured online and utilized for feedback. They developed a parameter-dependent control law deriving from a convex optimization problem with an LMI structure. The proposed approach minimizes conservatism and enhances performance, according to the findings. Likewise, regulating networked constrained polytopic LPV systems is discussed in [252]. The communication network is sensitive to latency effects, resulting in time-varying delays on both the control and measurement channels. This paper proposes a unique LPV adaptation of a recent Receding-Horizon control (RHC) approach for polytopic uncertain linear systems. It takes advantage of the availability of the LPV scheduling parameter for pre-computing nested families of one-step-ahead controlled ellipsoidal sets that are substantially less conservative than their robust counterparts. Finally, simulation results for constrained regulation of a mobile robot of the unicycle type demonstrate the efficacy and advantages of the suggested control strategy.

Regarding stability, a new path tracking approach for a nonholonomic mobile robot exposed to kinematic disturbances is given using a gain-scheduled control law [253]. Kinematic error in the tracking problem obtained the form of LPV with bounded disturbance. The target tracking is established for the environment with and without obstacles. The system stability has

been confirmed using a Lyapunov function in both structured and unstructured environments. Finding a control law for all routes despite linear and angular velocity constraints is the fundamental benefit of the proposed strategy. Similarly, a control technique for omnidirectional mobile vehicles with a high robustness level is offered [254]. It is demonstrated that the accompanying nonlinear control issue may be tractable by translating the complicated nonlinear and uncertain model into LPV form. Compared to those nonlinear control approaches, the suggested algorithm avoids the complicated designing procedure and minimizes the impacts of control constraints in fast dynamic environments. A sampled-data MPC tracking control approach is provided for mobile robots, represented as constrained continuous-time LPV systems, considering input saturation constraints [255]. The optimization problem is represented in an LMI format. A new Lyapunov function was generated to prove the stability in a sampling delay much larger than the literature. Finally, a numerical example is provided to show the efficiency of the strategy proposed. Another study improved LPV-MPC stability by adding a terminal cost and variable weight matrices [256]. Using an event-triggered mechanism, the controller solved a consensus problem on a nonholonomic model. The nonlinear dynamics of the nonholonomic model are reduced to a linear time-variant model. In order to eliminate offset due to modelling simplification, a quadratic cost function is derived by adding terminal state and variable weight matrices. Solving a quadratic programming issue yields the control signal for agents. During the event-triggered mechanism, a local optimum state controller is integrated with the LPV-MPC. The controller only optimizes when the trigger conditions are satisfied to decrease the computational burden.

2.7 Conclusion and future works

We have provided an extensive review of various strategies linked to the path tracking of WMR in hospitals in this research. Robots can transport medicine, medical equipment, medical records, food, and even lift patients in hospitals. Nonlinear controllers such as feedback linearization and fuzzy logic control have been frequently employed in the literature due to the nonlinear dynamics of WMR. Because uncertainties in model parameters and environment are unavoidable when the robot operates, sliding mode control, resilient control, and robust MPC have garnered much attention. Meanwhile, vision-based control allows robots to know their surroundings better, particularly dynamic obstacles.

Dealing with constraints is one of the main reasons that the use of MPC to regulate WMR has gained so much attention in recent years. However, there are still some unsolved research topics. Firstly, some issues of adopting MPC for robotics include ensuring stability and model type. LPV is one of the ways that has been lately employed to overcome this problem. The combination of MPC and LPV can be very beneficial for robot tracking problems. Much of the research done is based on state-space models. These techniques are difficult because state variables are often not measurable, and utilizing an observer makes the design problem more difficult. Hence, one topic that needs further attention is using the LPV-IO model. Secondly, a nonparametric LPV identification such as ANN and Support Vector Machine (SVM) can improve the model's accuracy. Another intriguing issue in trajectory tracking is determining the spots for the robot to begin functioning. This problem may be through an interpolation approach that increases the attraction region without burdening the online computation load. Finally, resilient control should be further investigated, which accounts for all types of physical and cyber constraints and allows the robot to recover in a critical and unstable scenario. The reason is that the current literature focuses on robustness to uncertainties, including input saturation, parameter changes, network delay, time-varying delay, bounded disturbance, time-varying disturbance, noises, and faults. Resilient control, however, can deal with all uncertainties in a more organized structure and would make a breakthrough in regulating WMR.

CHAPTER 3

AN INTERPOLATION-BASED MODEL PREDICTIVE CONTROLLER FOR INPUT-OUTPUT LINEAR PARAMETER VARYING SYSTEMS

Chapter 3 presents the study for Objective 1. The work presented in this chapter is included in the following paper:

Hadian, M., & Zhang, W., Ramezani, A. (2021). Interpolation-based Model predictive controller for input-output linear parameter varying systems. International Journal of Dynamics and Control. Published.

Abstract

This paper developed a controller for constrained nonlinear MIMO systems, which reduced computation and conservatism. Firstly, a linear parameter varying (LPV) model was chosen to capture the system's nonlinear/time-variant characteristics with scheduling variables, having less complexity than traditional nonlinear models. Secondly, this LPV model was accompanied by a model predictive controller (MPC) to deal with constrained MIMO systems. In contrast to nonlinear MPC, an LPV-MPC had a linear model in each sampling time, leading to a less complicated designing process and a lower online computation load. Moreover, the LPV- MPC was constructed in an input-output (IO) configuration not to require the state measurement. Thirdly, an interpolation-based MPC (IMPC) was supplemented for the sake of conservatism, where control signals were derived from interpolation between several precalculated control gains. If this strategy is implemented, the online calculation will be moderated, and conservatism will be improved through an increased attraction region. The efficiency of the proposed method was thoroughly examined in two numerical examples. The findings showed that the approach performed brilliantly in setpoint tracking and disturbance rejection and was deeply tractable considering online computation and conservatism.

3.1 Introduction

The model predictive controller (MPC) has been widely adopted in industry and academia over the past few decades, including robots, wind and steam turbines, autopilot, metal rolling, cement manufacturing, and petrochemical companies [47, 257-259]. The chief reason is that MPC is tailored to meet constraints imposed on inputs/outputs. Making a future prediction of the system's behaviour plays a pivotal role in MPCs. The more accurate the mathematical model is, the fewer errors the MPC will make. A vast number of studies pertaining to linear MPC have been carried out for more than half of a century ranging from linear adaptive MPC [260-263] to robust linear MPC [156, 264] and intelligent MPC [53, 265-267].

While influential theories exist in the field of linear MPC, some systems are nonlinear and operate under extensive operating conditions, or external parameters can fundamentally change the process response [268]. Nonlinear MPC (NMPC) has been developed to deal with systems with nonlinear dynamics. A higher level of algorithm complexity and excessive computational load are two main drawbacks of NMPC.

Moreover, modelling and output prediction for nonlinear systems are formidable compared to linear systems. Owing to these limitations, some employed linearization around the operating point. This strategy can be easily operated and tested successfully in many applications [2]. On the downside, linear-oriented MPCs locally work well and remain valid around the operating point, resulting in poor performance in certain circumstances and causing instability. Besides, using linear models can impair the accuracy of prediction. Despite these imperfections, industry specialists prefer linear frameworks to nonlinear ones that might arise sheer complexity. Consequently, both linear and nonlinear MPC strategies have upsides and downsides.

To reach a compromise between linear and nonlinear approaches, MPC based on a linear parameter varying (LPV) system is brought to researchers' attention [63, 66, 148, 157]. LPV models have a linear representation while taking account of the system's nonlinear/time-varying features. LPVs bridge the gap between linear systems and nonlinear/time-varying systems by depicting nonlinear systems with a linear form with variable parameters, also called scheduling variables [269, 270]. In other words, the linear model and controller can vary according to scheduling variables, which can be nonlinear dynamic parameters of the system, environmental conditions, or operating points. In short, LPV assists MPC in tackling nonlinear/time-varying systems.

LPV-MPCs could be divided into two broad classes of state-space and transfer-function forms. The overwhelming majority of research on LPV-MPC is constructed from LPV-SS models. These models often exacerbate computational complexity and barely model uncertainties well [271-273]. Moreover, they assume that state variables are measurable, which is hardly likely in practical terms. Input-output models can describe the behaviour of industrial systems more appropriately [274]. On these grounds, an LPV-IO-based predictive controller is studied in this paper. Only a few MPC have been documented for LPV systems under the I/O formulation [268].

In [275], MPC-LPV with SS and IO presentations are compared for a Control Moment Gyroscope (CMG). The results showed that the execution time would reduce using the IO framework. Another research [28] introduces an MPC with LPV-IO representation, where a terminal cost is added to grant stability. The optimization problem of MPC is written in Bilinear Matrix Inequality (BMI), which is non-convex and challenging to solve. This approach was improved in [276] in order to reduce the complexity level markedly. The authors designed a controller based on a Linear Matrix inequality (LMI) convex problem, successfully assessed in an ideal continuous stirred tank reactor (CSTR). This work employed offline controllers and terminal regions to moderately reduce the online computational burden. However, the controller suffers from huge online computational capacity, stemming from solving LMI equations and calculating the Markov coefficient of the LPV system [277]. Another weakness of this study is that scheduling variables are assumed to be fixed over the prediction horizon with unknown future values [278]. Consequently, the controller is highly conservative and computationally ineffective, which impedes the real-world application of the current MPCs with LPV-IO presentation [276, 279]. [29] sought an extension of [277], where computational complexity was addressed for an MPC with IO form. The authors guaranteed stability and setpoint tracking, while the problem of conservatism remains unsolved. In addition, some of the assumptions of this work (such as considering scheduling variables in terms of outputs ignoring states and inputs, and using an iterative method for estimating future scheduling variables) might make the controller even more conservative than [276]. For the first time, an IMPC with an LPV-IO structure is developed in this paper to accomplish a less conservative and more computationally efficient design. The IMPC framework uses a set of offline controllers to increase the area of attraction and bring down the online computational cost.

The process of finding a control signal in the MPC usually involves an optimization problem calculated online, whereby substantial computational resources are required, at the cost of MPC's ability to manage constraints. The controller shoulders an excessive computational load because the system's dynamics are nonlinear. Accordingly, IMPC is presented in the literature to guarantee the fulfillment of constraints with reasonable computations, in which the control signal is obtained by interpolating a set of predefined offline signals (u_i), which brings high-speed online calculations and increases attraction, i.e., IMPC handles online computational load and conservatism. IMPC was first introduced by [19-22], in which the authors employed an ellipsoidal invariant set to meet stability and recursive feasibility. Later research attempts to enlarge the region of attraction (ROA) by replacing the ellipsoidal invariant set with a polyhedral invariant set and introducing the Maximal Admissible Set (MAS). A body of studies has been conducted to widen ROA or MAS [167, 280]. The chief impediment to the existing IMPC is that they only expand ROA/MAS, resulting in a lower conservatism in simple processes. This paper enlarges the terminal region of states alongside ROA/MAS, ensuring a less conservative controller. More precisely, the offline state feedback controller is designed to use a broader possible terminal region in addition to increased ROA/MAS without violating constraints. To the best of our knowledge, this is the first time that an efficient IMPC with an LPV-IO model has been developed by adding a terminal region in the design step of the control gain. The results showed that the proposed method is noticeably more tractable than the traditional ones in the computation load and conservatism degree.

In summary, this paper seeks a solution for controlling nonlinear multi-input multi-output constrained systems with efficient online computation, no observer, massive enough initial/terminal region, and satisfactory closed-loop performance. The proposed Interpolation-based MPC controller with LPV-IO representation and expanded terminal region fulfill all these purposes. Similar to [276], the stability is confirmed using terminal cost and region. A larger terminal region and increased attraction area bring a less conservative controller. Meanwhile, using an interpolation of predefined controllers with the Ackerman formula will improve control performance. The paper is organized as follows. The LPV model is described in section II, followed by the controller architecture embodying MPC with LPV-IO form and IMPC structure. The case studies are explained in detail in section III, where the proposed methodology is examined, and the results are discussed. To sum up, the paper's findings are stated in the final section.

3.2 Problem formulation

In this section, the IO and SS configuration of the LPV model is first described. Subsequently, the optimization problem of MPC is solved, and control gain, the terminal region, and ROA are found. As previously discussed, although most studied MPC-LPV is based on the SS form, the IO form is far more desirable since this form only needs inputs and output measurement and has a smaller number of coefficients.

3.2.1 LPV representation

Assume the following discrete-time MIMO linear parameter-varying (LPV) transfer function:

$$A(p(k), q^{-1})y(k) = B(p(k), q^{-1})u(k) \tag{3.1}$$

This also can be shown as follows:

$$y(k) = -\sum_{i=1}^{n_a} a_i\big(p(k)\big)\, y(k-i) + \sum_{j=0}^{n_b} b_j\big(p(k)\big)\, u(k-j) \tag{3.2}$$

Subjected to the following constraints

$$u(k) \in U \equiv \{u \in \mathbb{R}^{n_u} |\ |u(k)| \le u_{max}\}$$
$$\Delta u(k) \in V \equiv \{\Delta u \in \mathbb{R}^{n_u} |\ |\Delta u(k)| \le \Delta u_{max}\}$$
$$y(k) \in Y \equiv \{y \in \mathbb{R}^{n_y} |\ |y(k)| \le y_{max}\}$$
$$p(k) \in P \equiv \{p \in \mathbb{R}^{n_p} |\ |p(k)| \le p_{max}\}$$

where $y(k)$ are outputs, $u(k)$ are inputs, $p(k)$ are scheduling variables, q^{-i} is a backshift operator, n_a is numerator order, n_b is dominator order, $n_u$is the number of inputs, n_y is the number of outputs, and $u_{max}, \Delta u_{max}$,and y_{max} are boundaries.

The state-space representation of the dynamic model given by the equation is:

$$x(k+1) = \underbrace{\begin{bmatrix} -a_1 & \dots & -a_{n_a-1} & -a_{n_a} & b_0+b_1 & \dots & b_{n_b-1} & b_{n_b} \\ I_{n_y} & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & & I_{n_y} & 0 & 0 & & 0 & 0 \\ 0 & \vdots & 0 & 0 & I_{n_u} & \vdots & 0 & 0 \\ 0 & \dots & 0 & 0 & I_{n_u} & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & 0 & 0 & I_{n_u} & 0 \end{bmatrix}}_{A(p(k))} x(k) + \underbrace{\begin{bmatrix} b_0 \\ 0 \\ \vdots \\ 0 \\ I_{n_y} \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{B(p(k))} u(k) \tag{3.3}$$

where $x(k) = [y(k-1) \dots y(k-n_a)\ \ u(k-1) \dots u(k-n_b)]^T$. For the sake of simplicity, $p(k)$ is removed from all coefficients a_i and b_i. We considered that $[A(k)\ \ B(k)] \in \Omega =$

$Co[[A_1\ B_1], \cdots, [A_m\ B_m]]$, where Ω is the polytope, Co is the convex hull, and $[A_j\ B_j]$ are vertices. The state variable is a combination of previous inputs and outputs by doing this. The controller uses the LPV-IO model to predict outputs and the LPV-SS model for stability provision. Also noteworthy is that the stability of the LPV-SS controller grants the equivalent LPV-IO controller stability [281].

3.2.2 Controller design

After introducing the cost function, the predicted outputs will be obtained using the step response and the Diophantine equation [2]. This is followed by finding a control signal from a set of precomputed control gains to have a larger attraction region. In last, the recursive feasibility and asymptotical stability will be proved. The controller aims at minimization of the cost function (J):

$$J = \underbrace{\sum_{i=0}^{N} e(k+i)^T\, Qe(k+i) + \Delta u(k+i-1)^T R \Delta u(k+i-1)}_{l(e,u)} + \Psi(k+N) \tag{3.4}$$

The first term $l(e,u)$ is called stage cost, being in charge of closed-loop achievements, where Q and R are positive definite state and input weighting matrixes, and e is the error, which is the difference between measured outputs (y) and reference (r). The term $\Psi(k+N)$is the terminal cost, which can be defined as $x(k+N)^T Q_p x(k+N)$.

3.2.3 The prediction model

For the prediction of error $e(k+j)$, finding $y(k+j)$ sentences is required, which can be calculated from step response or by solving a Diophantine equation. Given the step response of the LPV system (3.2), the future values of outputs over the horizon will be:

$$y(k,p(k)) = \sum_{i=1}^{\infty} g_i(p(k))\Delta u(k-i,p(k)) \tag{3.5}$$

$$y(k+j,p(k+j)) = \sum_{i=1}^{\infty} g_i(p(k+j))\Delta u(k+j-i,p(k)) + d(k+j,p(k))$$

where g_i denotes step response, and $d(k+j)$ is a disturbance, which can be derived from $y(k+j) - r(k+j)$. For a faster prediction, disturbances are assumed to be constant, which can be estimated by $y(k) - r(k)$. The scheduling variable $p(k)$ is removed from Equations (3.6) to (3.9) for the reasons of simplification. The Equation (3.5) can then be rewritten:

$$y(k+j) = (\sum_{i=1}^{k} g_i \Delta u(k+j-i) + \sum_{i=k+1}^{\infty} g_i \Delta u(k+j-i)) + (y(k) - r(k)) \tag{3.6}$$

Substituting $y(k) = \sum_{i=1}^{\infty} g_i \Delta u(k-i)$ in Equation (3.6) yields:

$$\begin{aligned} y(k+j) &= \sum_{i=1}^{k} g_i \Delta u(k+j-i) \\ &\quad + \left(\underbrace{\sum_{i=k+1}^{\infty} g_i \Delta u(k+j-i) + \sum_{i=1}^{\infty} g_i \Delta u(k-i) - r(k)}_{f(k+j)} \right) \\ &= \sum_{i=1}^{k} g_i \Delta u(k+j-i) + \sum_{i=1}^{\infty} (g_{k+i} - g_i) \Delta u(k-i) - r(k) \\ &= \sum_{i=1}^{k} g_i \Delta u(k+j-i) + f(k+j) \end{aligned} \tag{3.7}$$

Meanwhile, $f(k+j)$ can be abridged on the assumption that the system is asymptotically stable and after N sampling period g_{k+i} equals g_i.

$$f(k+j) = \sum_{i=1}^{N} (g_{k+i} - g_i) \Delta u(k-i) - r(k) \tag{3.8}$$

Using Equation (3.8), the sequence of predicted outputs over the prediction horizon can be obtained.

$$\begin{aligned} y(k+1) &= g_1 \Delta u(k) + f(k+1) \\ y(k+2) &= g_2 \Delta u(k) + g_1 \Delta u(k+1) + f(k+2) \\ &\vdots \\ y(k+N) &= \sum_{i=1}^{N} g_i \Delta u(k+N-i) + f(k+N) \end{aligned} \tag{3.9}$$

In short, the prediction relationship of the output vector is

$$Y(p) = G(p)U(p) + F(p) \tag{3.10}$$

where

(3.11)

$$G = \begin{bmatrix} g_1 & 0 & \dots & 0 \\ g_2 & g_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_2 \\ g_N & g_{N-1} & \cdots & g_1 \end{bmatrix}$$

$$Y = \begin{bmatrix} y(k+1) \\ \vdots \\ y(k+N) \end{bmatrix}, U = \begin{bmatrix} u(k) \\ \vdots \\ u(k+N-1) \end{bmatrix}, F = \begin{bmatrix} f(k+1) \\ \vdots \\ f(k+N) \end{bmatrix}$$

If the system is unstable, the outputs cannot be predicted by step response, and the outputs will be calculated by Diophantine equations given by:

$$1 = E_j(q^{-1}, p(k))\Delta A(q^{-1}, p(k)) + q^{-j}F_j(q^{-1}, p(k)) \tag{3.12}$$

By finding polynomials E_j and F_j , the output of the process can be predicted. The order of E_j and F_j polynomials are $j-1$ and n_a. These polynomials can be calculated when one is divided by $\Delta A(q^{-1}, p(k))$, where E_j is remainder and F_j is quotient. The Equation (3.1) then is multiplied by $\Delta = 1 - q^{-1}$:

$$\Delta A(q^{-1}, p(k))y(k) = B\big(q^{-1}, p(k)\big)\Delta u(k) \tag{3.13}$$

From Equation (3.12), ΔA can be expressed in terms of E_j and F_j:

$$\Delta A(q^{-1}, p(k)) = \frac{1 - q^{-j}F_j(q^{-1}, p(k))}{E_j(q^{-1}, p(k))} \tag{3.14}$$

By substituting (3.14) in (3.13) and multiplying by $q^{+j}E_j(q^{-1}, p(k))$, we have

$$(1 - q^{-j}F_j(q^{-1}, p(k)))y(k+j) = B\big(q^{-1}, p(k)\big)E_j(q^{-1}, p(k))\Delta u(k+j-1) \tag{3.15}$$

Subsequently, one can predict the outputs as follows:

$$y(k+j) = F_j\big(q^{-1}, p(k)\big)y(k) + B\big(q^{-1}, p(k)\big)E_j\big(q^{-1}, p(k)\big)\Delta u(k+j-1) \tag{3.16}$$

Or

$$y(k+j) = F_j\big(q^{-1}, \theta(k)\big)y(t) + G_j\big(q^{-1}, \theta(k)\big)\Delta u(k+j-1) \tag{3.17}$$

It is worth noting that E_j and F_j polynomials can be calculated recursively, according to [2]. The vectorized form of Equation (3.17) will be

$$Y = GU + F \tag{3.18}$$

where

$$G = \begin{bmatrix} G_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & G_N \end{bmatrix}, F = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_N \end{bmatrix}$$

3.2.4 Stability

After output prediction, the asymptotical stabilizing state feedback control can be described:

$$\begin{aligned} u &= -Kx \\ x(k) &\in S_I \\ x(k+N) &\in S_T \end{aligned} \tag{3.19}$$

S_I characterizes initial feasibility region, in which $S_I = \{x \in R^n | x \in X, -Kx \in U\}$, and S_T denotes the terminal feasibility region, $S_T = \left\{x \in R^n | x(k+N)^T Q_p x(k+N) < \frac{1}{\gamma}\right\}$. As previously discussed, this method aims to expand S_I and S_T to enjoy a progressive controller. The first step is to enlarge $S_I$through a general interpolation MPC. Secondly, the largest possible S_T that meet the constraints will be chosen by minimizing γ . The control law of IMPC can be described as follows:

$$\begin{aligned} x = \sum_{i=1}^{n} \lambda_i x_i \,, \sum_{i=1}^{n} \lambda_i = 1,\ \lambda_i \geq 0 \\ u = -\sum_{i=1}^{n} k_i \lambda_i x_i \end{aligned} \tag{3.20}$$

Where λ_i are coefficients, and n is the number of predefined feedback gains k_i that stem from Ackermann's formula as follows:

$$K_i = [0 \quad \cdots \quad 0 \quad 1] Q_{c,i}^{-1} \alpha_i(A) \tag{3.21}$$

where $\alpha_i(A)$ is the characteristic polynomial, and $Q_{c,i}$ is the controllability matrix.

$$\alpha_i(A) = \det(SI - (A_i - B_i K_i))$$
$$Q_{c,i} = [B \quad AB \quad A^2B \quad \dots \quad A^{n_x-1}B \ ] \tag{3.22}$$

Lemma1: Taking the state-space model, as stated in Equations (3.3), the proposed general IMPC method is recursively feasible.

Proof: Substituting Equation (3.20) in Equation (3.3) results in:

$$x(k+1) = \left(A(k)\sum_{i=1}^{n}\lambda_i x_i\right) - \left(B(k)\sum_{i=1}^{n}k_i\lambda_i x_i\right) = \sum_{i=1}^{n}(A - Bk_i)\lambda_i x_i \tag{3.23}$$
$$= \sum_{i=1}^{n} M_i \lambda_i x_i$$

If $x(k) \in S_I$, subsequently $(A - Bk_i)\lambda_i x_i \in S_I$, which yields $x(k+1) \in S_I$. Besides, it can easily be shown that $-Kx \in U$ as below:

$$|u| \le \sum_{i=1}^{n}\lambda_i |k_i x_i| \le (\sum_{i=1}^{n}\lambda_i)u_{max} \le u_{max} \tag{3.24}$$

According to Equation (3.23), the state trajectories are $\{-Kx, KM_i x, KM_i{}^2 x, \dots\}$. With this in mind, the enhanced feasible invariant set can be determined by Maximal Admissible Set (MAS) as:

$$s_i = \{x | \mathrm{H}_i x \le \alpha\} \tag{3.25}$$
$$S_I = Co\{S_1, S_2, \dots S_n\}$$

where $\mathrm{H}_i = \left[M_i, M_i{}^2, \dots, M_i{}^n\right]^T, and\ \alpha = [1, \dots, 1]^T$

After computing K_i and proving the recursive feasibility, the stability is required to be guaranteed with a Lyapunov function.

Lemma2: The system, described in Equation (3.1), is asymptotically stable under the interpolated control law (3.20) if there is $Q_p = Q_p{}^T > 0$, which implies Equation (3.26).

$$M_i{}^T Q_{p_i} M_i - Q_{p_i} + Q_i + K_i{}^T R_i K_i \le 0, \qquad i = 1, \dots n \tag{3.26}$$

Proof: Considering $V = x(k)^T Q_p x(k) > 0$ as the candidate Lyapunov function in order to ensure terminal penalty, the controller $u = -Kx$ exists, if the equation below satisfies:

$$V\big(x(k+1)\big) - V\big(x(k)\big) < 0, k = 0, \dots, p-1, x(k) \in S_T \tag{3.27}$$
$$x(k+1)^T Q_p x(k+1) - x(k)^T Q_p x(k) \le -x(k)^T Q x(k) - u(k)^T R u(k)$$

By substituting $u = -Kx$, we get

$$M^T Q_p M - Q_p + Q + K^T R K \leq 0 \tag{3.28}$$

$$M = diag(M_i)$$

After showing the recursive feasibility and asymptotically stability of the proposed method, the terminal region must be determined. While region S_T is desired to be as large as possible to reduce conservatism, increasing this area might cause violating the constraints. As a result, the minimum value of γ that satisfies constraints can be calculated by an optimization problem. It is proved that γ can be derived from the following optimization problem [276]:

$$\min_{\gamma} \gamma \tag{3.29}$$

$$\frac{1}{\gamma} A_T P^{-1} {A_T}^T \leq B_T {B_T}^T$$

where $A_T = [-I_n \;\; K \; I_n - K]^T, B_T = [x_{max} - x_s \;\; \Delta u_{max} \; x_{max} - x_s \;\; \Delta u_{max}]^T$, and x_s is the desired state.

It is verified that the proposed method with the IO form is asymptotically stable with the least possible conservatism through increased initial and final feasible regions. At last, the LPV-IMPC procedure can be summarized in six steps:

Step 1: Identify LPV input-output model

Step 2: Find K_i from Equation (3.21).

Step 3: Adjust the controller parameters, i.e., $Q, P,$ and N

Step 4: Specify Q_p and γ by solving Equations (3.26) and (3.29).

Step 5: Solve the constrained quadratic optimization problem to find λ_i and x_i

$$\min_{\lambda_i, x_i} J$$

Subjected to constraints (3.2) and

$$x(k) \in S_I$$

$$x(k+N) \in S_T$$

Step 6: Implement the control signal to the plant and repeat the procedure

3.3 Results and discussions

The proposed method is compared with one of the most current research on MPC-LPV-IO [276] in two case studies. Two scenarios are constructed to show the studied procedure's functionality in both case studies, including setpoint tracking and disturbance rejection. The underlying reason behind the first case study was to underline the proposed method's superiority

in reducing conservatism through an increased attraction region while being cost-effective. The investigated controller in [276] is confirmed to be asymptotically stable with a large enough terminal region using an IO model while suffering from computational complexity and conservatism. More specifically, although [276] guaranteed stability with the largest possible terminal region, the proposed strategy in this paper increases both terminal and initial region, resulting in a more tractable controller. The first example mainly compared two methods in terms of tractability (conservatism and computation) in a simple case study, where the control performance might be similar (in terms of transient and steady-state factors). The first example was a simple MIMO process with two inputs, two outputs, and one scheduling variable. However, in stark contrast, the second case study was a complicated large-scale process with 25 state variables, five inputs, five outputs, and two scheduling variables. The key objective of the second case study was to assess two methods in terms of control performance. Example two aimed to show the sharp distinction between the two methods when the computational volume experienced a significant rise due to multi-scheduling variables and a larger process size. Moreover, unlike the pre-known LPV-IO model of the first example, the process needed to be identified with an LPV-IO model in the second example. According to the ultimate aim of the two examples, [276] the controller in the first example is named terminal region MPC (TRMPC), and in the second example, LPV-IO-MPC.

3.3.1 Case study 1

The conservatism and computational load of the two methods were investigated in this example. The computation load was measured per second (s), and a criterion was indicated for the first time to evaluate progressiveness per percentage. The conservatism rate (CR) is calculated by the multiplication of the surface area of the initial feasible region (A_I) and the surface area of the final feasible region (A_F), divided by the square of the total surface area (A_T):

$$CR = \frac{A_I * A_F}{{A_T}^2} * 100 \tag{3.30}$$

where A_T can be expressed by $\left(x_{1_{max}} - x_{1_{min}}\right) * \left(x_{2_{max}} - x_{2_{min}}\right)$. The more the CR is, the less conservative the controller is. The first scenario was reference tracking, and the second was a detailed assessment of the studied method in terms of bounded disturbance rejection. The following LPV-IO model is a good case in point to confirm the proposed method's superiority in conservatism rate and computation load (the elapsed time for running the program) [276].

$$a_1(p_k) = -0.2 + 0.7p(k), \quad a_2(p_k) = 0.7 + 0.4p(k) \tag{3.31}$$
$$b_1(p_k) = 1.6 + 2.8p(k), \quad b_2(p_k) = 3.4 + 1.2p(k)$$
$$x_{max} = 3, P = [0,1]$$

According to the given IO model, the system has one input, one output, and two state variables. The proposed system's performance has been compared with [276], named (terminal region MPC) TRMPC in Figures 3.1 to 3.3. In this section, the first method is LPV-IO-IMPC, and the second is TRMPC. Looking at Figure 3.1, despite changes in the scheduling signal, the input and output of both methods track the reference. Both systems' transient response is fast enough (rise time of the first and second methods were 3.45 and 3.67 seconds, respectively) and experiences low oscillations (the overshoot of the first and second methods were 15.30% and 10.44%, respectively). In the case of steady-state responses, either has a short settling time (It took 12.54s and 13.31s to reach the setpoint for the first and second methods, respectively). Figure 3.2 shows how vast the initial feasible region and final feasible region were. It is evident that the [276] method is immensely conservative compared to the proposed method since the area of the initial feasible region in the proposed method (S_1) is about six times that of the TRMPC method (S_2) (the area of the second method is 3.2, and the area of the first method is 14.8). In contrast, both methods' terminal feasible region is roughly equal (the former is 2.45, and the latter is 2.32). As stated in Equation (3.30), the CR for the first method is 9.67 percent and for the second is 42.39 percent. Put another way, the current strategy is nearly four times less conservative than the previous one. It should also be noted that the time required to run the program (RT) in the proposed method is 1.49 seconds, while it is 15.80 seconds in the TRMPC, which undermines the applicability of the procedure. Accordingly, both methods were asymptotically stable and tracked the given reference, but the proposed method is way more feasible because it is considerably less time-consuming and more progressive. In the second scenario, the effectiveness of two approaches in the face of disturbance is investigated. The mentioned disturbance has affected the process by the amplitude of 0.1 from 71s to 80s. Turning to Figure 3.3, the most striking feature is that both strategies could eliminate the disturbance, albeit the TRMPC eliminated the disturbance slowly (the time required to remove the disturbance in the second method was 11

seconds and in the first method was 4 seconds). The difference between the two methods are numerically summarized in Table 3.1, where T_r is 100% rise time, T_s is 2% settling time, PO is Percentage Overshoot, e_{ss} is steady-state error, and T_d is the time required to reject disturbance.

Table 3.1:Comparison of two controllers' performance in case study 1

Controllers	$\boldsymbol{T_r(s)}$	$\boldsymbol{T_s(s)}$	PO (%)	$\boldsymbol{T_d(s)}$	RT(s)	$\boldsymbol{A_I}$	$\boldsymbol{A_F}$	CR (%)
TRMPC	3.67	13.31	10.44	11	15.80	3.2	2.45	42.39
LPV-IO-IMPC	3.45	12.54	15.30	4	1.49	14.8	2.32	9.67

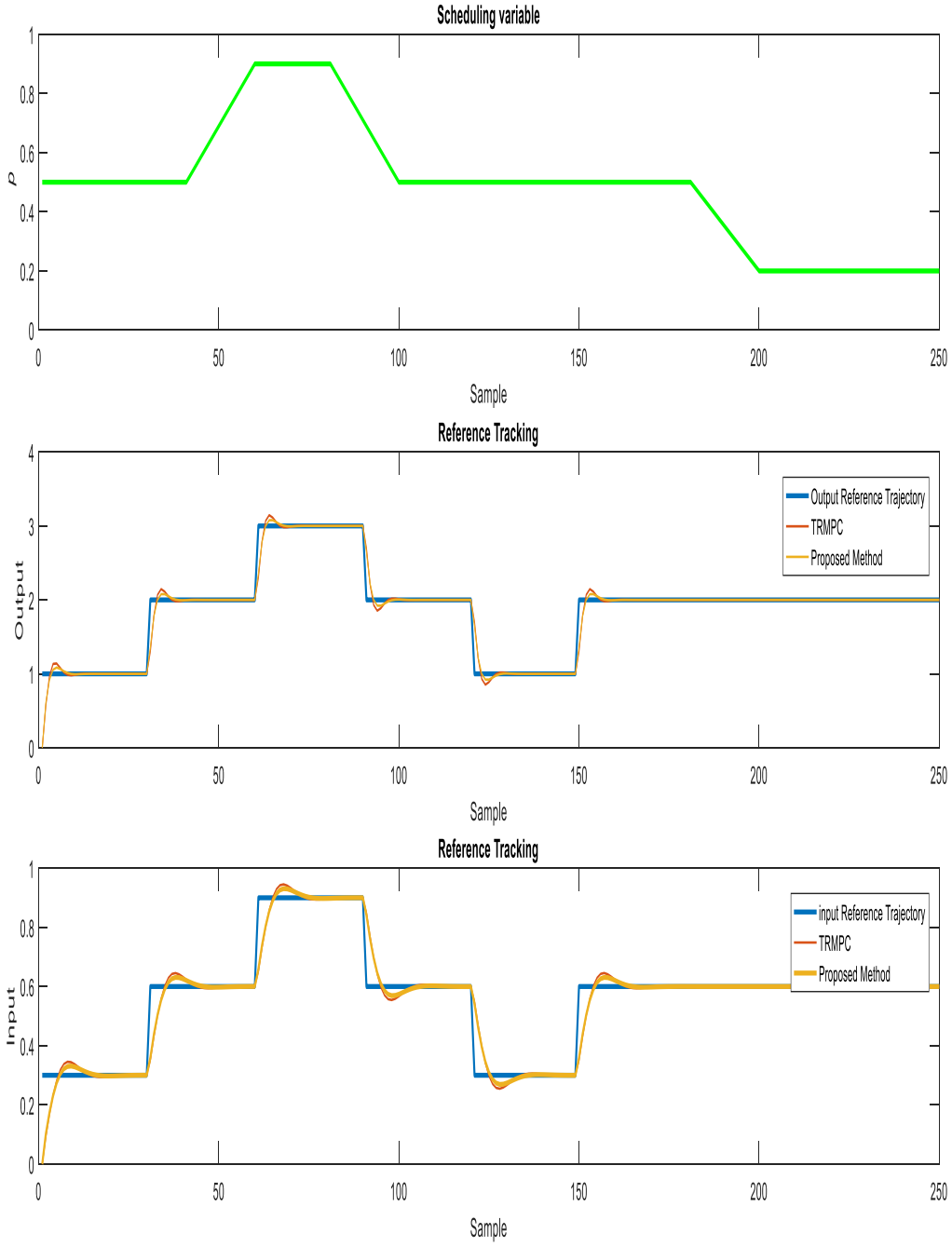


Figure 3.1: The first scenario (setpoint tracking): Scheduling variable (green), references (blue), TRMPC (Red), and the proposed method (orange)

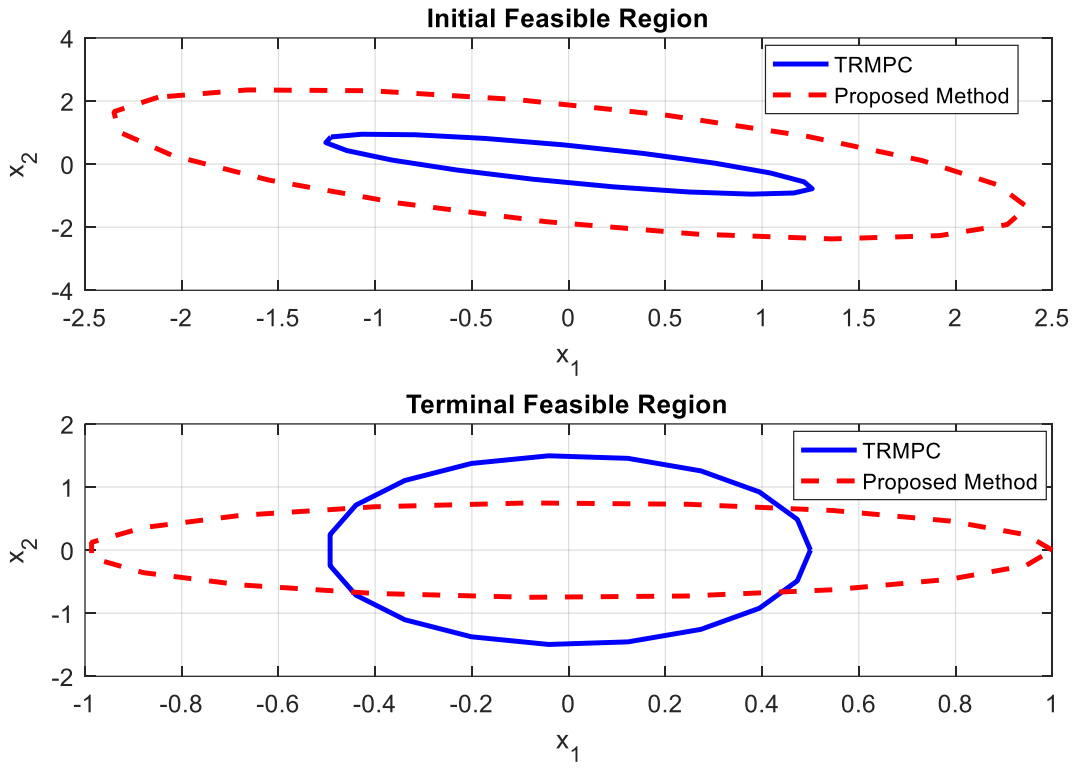


Figure 3.2: Initial and terminal feasible regions under the TRMPC (blue) and the proposed method (dashed-red)

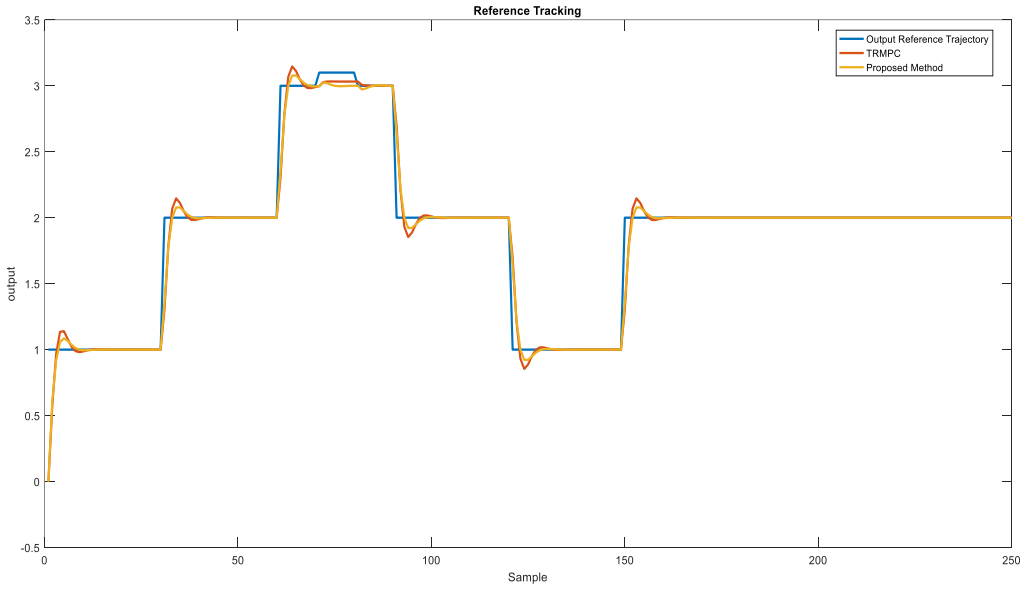


Figure 3.3:The second scenario: Reference tracking in the presence of disturbance under the TRMPC (red) and the proposed method (orange)

3.3.2 Case study 2

3.3.2.1 Process description

Alkylation of benzene with ethylene, a principal process in the petrochemical industry, produces ethylbenzene, widely used as a large-scale benchmark. As shown in Figure 3.4, the process studied in this work consists of four continuously stirred tank reactors (CSTRs) and a flash tank separator. A detailed description of the process is described in [52, 282]. The process's manipulated variables are heat inputs to five vessels shown by $Q1$, $Q2$, $Q3$, $Q4$, and $Q5$. The temperatures of five vessels $T1, T2, T3, T4, T5$ are considered the process outputs. Process outputs' dynamic behaviour is set out in detail in Equations 3.32 to 3.36, and all parameters are named in Table 3.2. For more information and fixed-parameter value, refer to [283].

Table 3.2: Process variables of the alkylation of benzene process

$\boldsymbol{C_{A1}, C_{B1}, C_{C1}, C_{D1}}$	Concentrations of A, B, C, D in CSTR-1
$\boldsymbol{C_{A2}, C_{B2}, C_{C2}, C_{D2}}$	Concentrations of A, B, C, D in CSTR-2
$\boldsymbol{C_{A3}, C_{B3}, C_{C3}, C_{D3}}$	Concentrations of A, B, C, D in CSTR-3
$\boldsymbol{C_{A4}, C_{B4}, C_{C4}, C_{D4}}$	Concentrations of A, B, C, D in Separator
$\boldsymbol{C_{A5}, C_{B5}, C_{C5}, C_{D5}}$	Concentrations of A, B, C, D in CSTR-1
$\boldsymbol{C_{Ar}, C_{Br}, C_{Cr}, C_{Dr}}$	Concentrations of A, B, C, D in F_r, F_{r1}, F_{r2}
$\boldsymbol{T_1, T_3, T_3, T_4, T_5}$	Temperatures in each vessel
$\boldsymbol{T_{ref}}$	Reference temperature
$\boldsymbol{F_3, F_5, F_7, F_8, F_9}$	Effluent flow rates from each vessel
$\boldsymbol{F_1, F_2, F_4, F_6, F_{10}}$	Feed flow rates to each vessel
$\boldsymbol{F_r, F_{r1}, F_{r2}}$	Recycle flow rates
$\boldsymbol{H_{vapA}, H_{vapB}, H_{vapC}, H_{vapD}}$	Enthalpies of vaporization of A, B, C, D
$\boldsymbol{H_{Aref}, H_{Bref}, H_{Cref}, H_{Dref}}$	Enthalpies of A, B, C, D at T_{ref}
$\boldsymbol{\Delta H_{r1}, \Delta H_{r2}, \Delta H_{r3}}$	Heat of reactions 1, 2 and 3
$\boldsymbol{V_1, V_2, V_3, V_4, V_5}$	Volume of each vessel
$\boldsymbol{Q_1, Q_2, Q_3, Q_4, Q_5}$	External heat/coolant inputs to each vessel
$\boldsymbol{C_{pA}, C_{pB}, C_{pC}, C_{pD}}$	Heat capacity of A, B, C, D at liquid phase
$\boldsymbol{\alpha_A, \alpha_B, \alpha_C, \alpha_D}$	Relative volatilities of A, B, C, D
$\boldsymbol{C_{A0}, C_{B0}, C_{C0}, C_{D0}}$	Molar densities of pure A, B, C, D
$\boldsymbol{T_{A0}, T_{B0}, T_{D0}}$	Feed temperatures of pure A, B, D
$\boldsymbol{k}$	Fraction of overhead flow recycled to the reactors

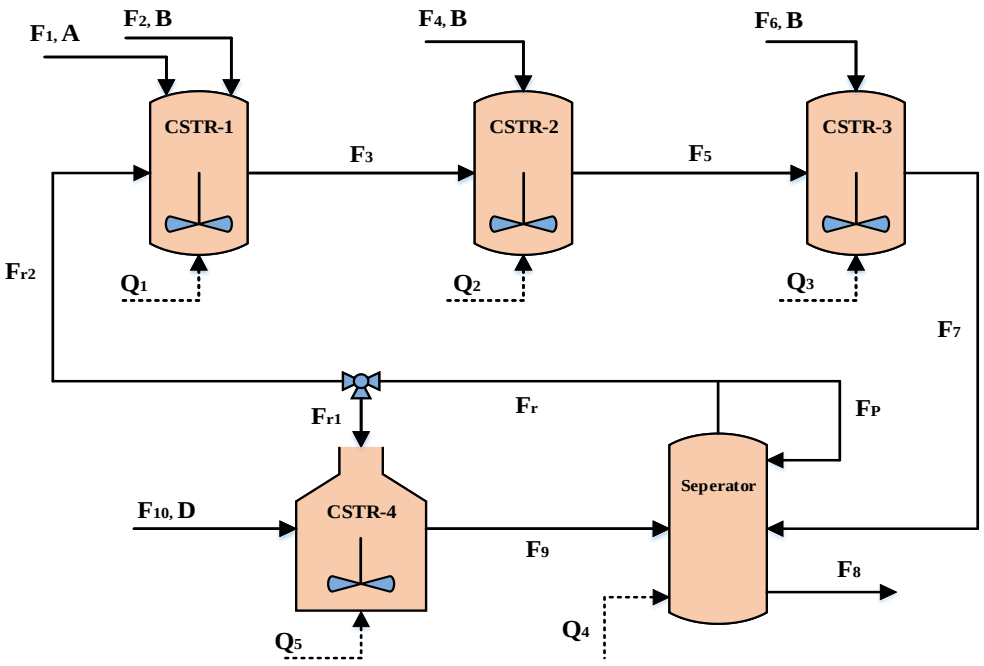


Figure 3.4: Flow diagram of alkylation of benzene process [52, 282]

$$\frac{dT_1}{dt} = \frac{Q_1 + F_1 C_{A0} H_A(T_{A0}) + F_2 C_{B0} H_B(T_{B0}) + \sum_i^{A,B,C,D} F_{r2} C_{ir} H_i(T_4) - F_3 C_{i1} H_i(T_1)}{\sum_i^{A,B,C,D} C_{i1}\, C_{pi} V_1} + \frac{-\Delta H_{r1} r_1(T_1, C_{A1}, C_{B1}) - \Delta H_{r2} r_2(T_1, C_{B1}, C_{C1})}{\sum_i^{A,B,C,D} C_{i1}\, C_{pi}} \tag{3.32}$$

$$\frac{dT_2}{dt} = \frac{Q_2 + F_4 C_{B0} H_B(T_{B0}) + \sum_i^{A,B,C,D} F_3 C_{i1} H_i(T_1) - F_5 C_{i2} H_i(T_2)}{\sum_i^{A,B,C,D} C_{i2}\, C_{pi} V_2} + \frac{-\Delta H_{r1} r_1(T_2, C_{A2}, C_{B2}) - \Delta H_{r2} r_2(T_2, C_{A2}, C_{B2})}{\sum_i^{A,B,C,D} C_{i2}\, C_{pi}} \tag{3.33}$$

$$\frac{dT_3}{dt} = \frac{Q_3 + F_6 C_{B0} H_B(T_{B0}) + \sum_i^{A,B,C,D} F_5 C_{i2} H_i(T_2) - F_7 C_{i3} H_i(T_3)}{\sum_i^{A,B,C,D} C_{i3}\, C_{pi} V_3} + \frac{-\Delta H_{r3} r_1(T_3, C_{A3}, C_{B3}) - \Delta H_{r3} r_2(T_3, C_{B3}, C_{C3})}{\sum_i^{A,B,C,D} C_{i3}\, C_{pi}} \tag{3.34}$$

$$\frac{dT_4}{dt} = \frac{Q_4 + \sum_i^{A,B,C,D} F_7 C_{i3} H_i(T_3) + F_9 C_{i5} H_i(T_5)}{\sum_i^{A,B,C,D} C_{i4}\, C_{pi} V_4} + \frac{\sum_i^{A,B,C,D} -M_i H_i(T_4) - F_8 C_{i4} H_i(T_4) - M_i H_{vapi}}{\sum_i^{A,B,C,D} C_{i4}\, C_{pi} V_4} \quad (3.35)$$

$$\frac{dT_5}{dt} = \frac{Q_5 + F_{10} C_{D0} H_D(T_{D0}) + \sum_i^{A,B,C,D} F_{r1} C_{ir} H_i(T_4) - F_9 C_{i5} H_i(T_5)}{\sum_i^{A,B,C,D} C_{i5}\, C_{pi} V_5} + \frac{-\Delta H_{r2} r_2(T_5, C_{B5}, C_{C5}) - \Delta H_{r3} r_3(T_5, C_{A5}, C_{D5})}{\sum_i^{A,B,C,D} C_{i5}\, C_{pi}} \quad (3.36)$$

The steady-state value of inputs and outputs and initial conditions are indicated in Table 3.3. The controller desire to steer the system from this initial condition to a steady-steady states condition while satisfying the constraints:

$$|Q_1 - Q_{1s}| < 7.5 \times 10^5, |Q_2 - Q_{2s}| < 5 \times 10^5, |Q_3 - Q_{3s}| < 5 \times 10^5$$
$$|Q_4 - Q_{4s}| < 6 \times 10^5, |Q_5 - Q_{5s}| < 5 \times 10^5$$

Table 3.3:the steady-state and initial values

Steady-state temperatures of vessels (K)	$\boldsymbol{T_{1s} = 477.24}$	$\boldsymbol{T_{2s} = 476.97}$	$\boldsymbol{T_{3s} = 473.47}$	$\boldsymbol{T_{4s} = 470.60}$	$\boldsymbol{T_{5s} = 478.28}$
Steady-state inputs (J/K)	$Q_{1s} = -4.4 \times 10^6$	$Q_{2s} = -4.6 \times 10^6$	$Q_{1s} = -4.7 \times 10^6$	$Q_{1s} = 9.2 \times 10^6$	$Q_{1s} = 5.9 \times 10^6$
Initial temperatures of vessels (K)	$T_{1,0} = 443.02$	$T_{2,0} = 437.12$	$T_{3,0} = 428.37$	$T_{4,0} = 433.15$	$T_{5,0} = 457.55$

3.3.2.2 Process identification and control

In contrast to previous works that developed a nonlinear MPC or noncentralized MPC, the MPC setup rests on an LPV model in this paper, having lower complexity and better performance. More significantly, F_r, C_{C0}, and C_{D0} are considered to be time-varying, for the first time, to the

best of the author's knowledge. From a practical point of view, these variables are selected to be scheduling variables with $\pm 10\%$ variation around their nominal values. Before applying the controller, the LPV-IO framework is required to be identified so that ten levels are defined within the range of 90% to 110% of the nominal value for each scheduling variable. The global LPV model originates from a polynomial interpolation of ten local LTI models, identified with 500 data in each level (operating condition). Each local model has five inputs and five outputs. All manipulated inputs are chosen to be pseudorandom binary signals to extract data with a sampling time of $\mathrm{T} = 10\,\mathrm{s}$. Afterward, the LTI models are identified using the MATLAB system identification Toolbox with the measured data. After estimating the output using polynomial interpolation, we realized that a third-degree interpolation produced a good fit for constructing the global LPV-IO model. At the same time, the order of local transfer functions varies from second to fifth. The scheduling variables and the responses from the global LPV-IO model and nonlinear model are presented in Figure 3.5 and Figure 3.6, respectively. The figures supplied clearly show that the identified LPV model can accurately predict the nonlinear system's behaviour. Moreover, the Mean Square Error (MSE) and Akaike's Final Prediction Error (FPE) for the global model were 0.032 and 0.058 in succession, and for the local LPV models, the range of MSE was between 0.004 and 0.062, and FPE changed from 0.040 to 0.081. Figure 3.6 and the numerical results show that the LPV-IO model is sufficiently reliable to estimate responses.

In the next step, the performance of the two controllers in respect of setpoint tracking and disturbance rejection is investigated, in which a step change at $t = 1500$ s with amplitude 2×10^{-3} is made in F_1 and F_2 as disturbances. The controller parameters were $N = 7$; $Q = 2000 * I_{5\times5}, R = 10^{-7} * I_{5\times5}$. The proposed approach is compared with the LPV controller studied in [276] in Figure 3.7, where the red line denotes the references, the dash-dot blue line is the proposed method, and the dashed black line is LPV-IO RMPC. Looking firstly at reference tracking, the proposed controller reached the setpoint at an acceptable time, while LPV-IO-RMPC suffers from large-amplitude oscillations. Although the LPV-IO-MPC has a faster response in some outputs, fluctuation might be detrimental. In general, regardless of the speed of response and oscillations, two methods followed the setpoints for T_1 to T_5 in the first scenario.

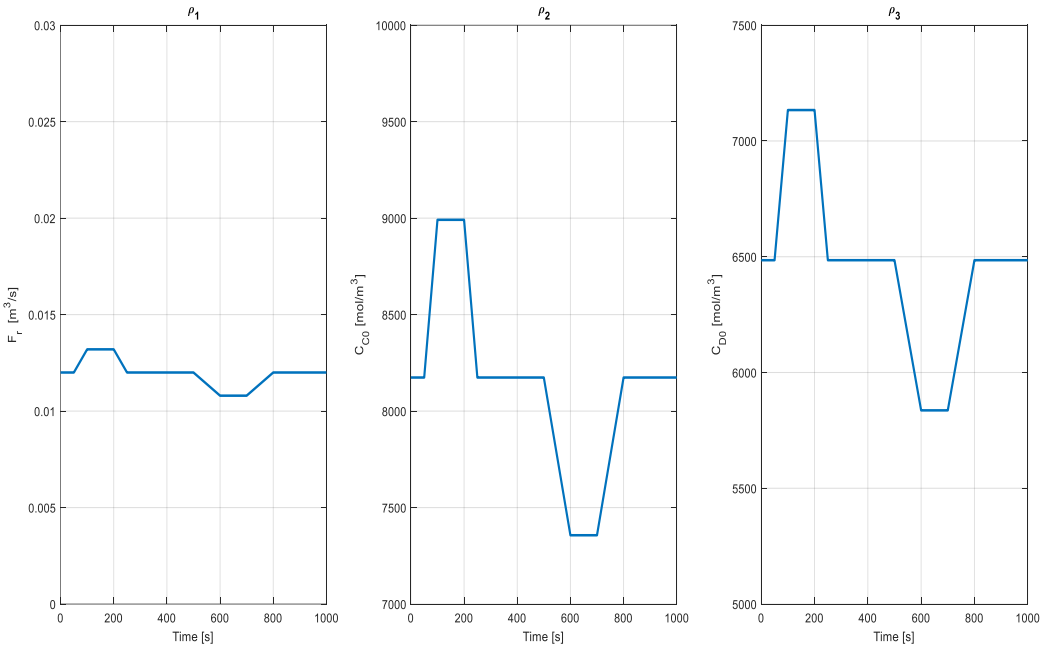


Figure 3.5: Scheduling variables from left to right: $\mathbf{F_r}$(Recycle flow rates), $\mathbf{C_{C0}}$(Molar densities of pure C), $\mathbf{C_{D0}}$(Molar densities of pure D)

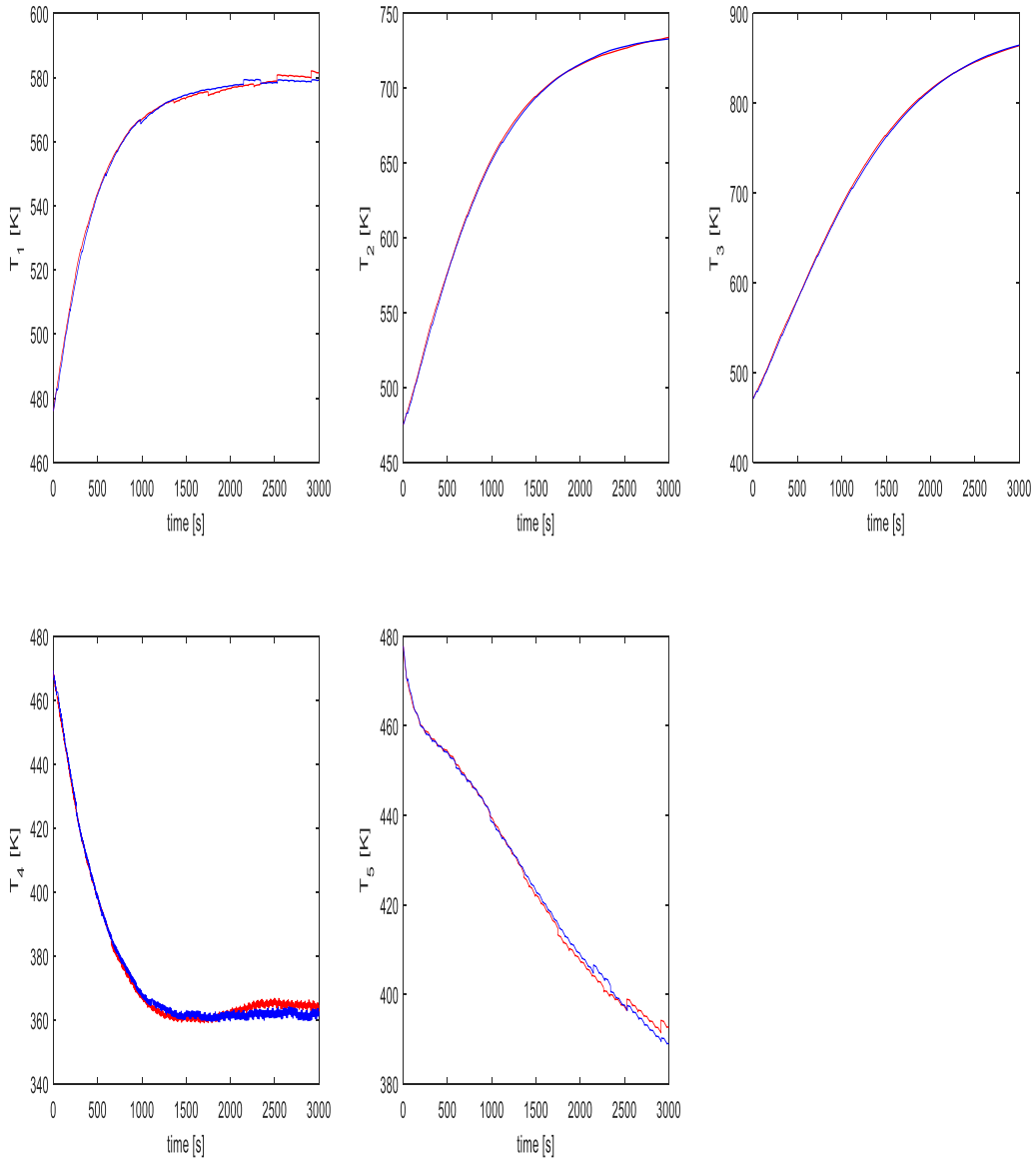


Figure 3.6: Temperatures in five vessels: blue and red lines indicate the original and simulated outputs from the LPV model, respectively.

Concerning disturbance rejection, the studied approach outperforms with removing the disturbance in a short time ranging from 110s to 355s for T_1 to T_5. This method experienced insignificant undershoot around the setpoint, unlike other methods. After time $t = 1500s$, when the disturbance was applied to the process, the LPV-IO-MPC violated both inputs and outputs constraints and became highly unstable. This method diverged when facing a constant disturbance

and failed to track the setpoint. At the same time, the cost function of LPV-IO-MPC is considerably more significant than the proposed method's cost function, which is firmly rooted in the utter failure of the former in the presence of disturbances. The wasted resources by LPV-IO-MPC $4.53 * 10^8$ times greater than the proposed method, proving that the studied approach is significantly cost-effective.

The MSE, 100% rise time (T_r), 2% settling time (T_s), Percentage Overshoot (PO), steady-state error (e_{ss}), and time is required to reject disturbance (T_d) in different outputs for the two methods are reported in Table 3.4. In short, the proposed method had a high speed (rise time, settling time, and time required to remove disturbance) and lower MSE and cost. Not surprisingly, the MSE of the first method ranged between 4900.11 to 5410.02, and the proposed method's MSE changed from 43.68 to 287.90, which implies that using interpolation MPC can reduce the extent of the error and improve setpoint tracking. Regarding transient response factors, The LPV-IO-MPC had a fairly lower T_r and more massive PO, varying from 0% to 9.7 %, than IMPC. The proposed method had an acceptable range of T_r with negligible PO. Concerning the steady-state response factors, the proposed method for all temperatures had significantly smaller T_s, ranging from 186.00s to 418.81s. Meanwhile, the steady-state error of LPV-IO-MPC was enormous, although the proposed method reached the exact setpoint, and its error was zero. Taking disturbance removal into account, the disturbance can be successfully rejected in a time between 305s to 610s using IMPC. It should be mentioned that T_d has not been presented in Table 3.4 for the first method since it failed to remove disturbance.

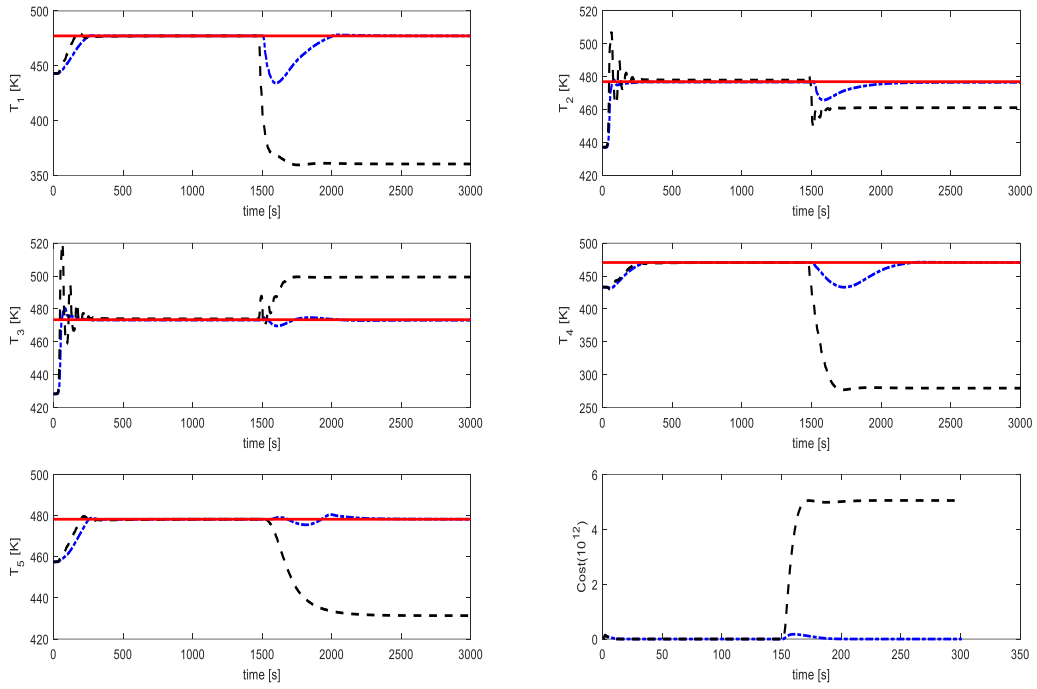


Figure 3.7: the temperatures of vessels and cost function using three studied methods (red line: references, the dash-dot blue line: proposed method, the dashed black line: LPV-IO RMPC)

Table 3.4: MSE, rise time, settling time, steady-state error, and disturbance rejection time of all temperatures

Methods	Criteria	T_1	T_2	T_3	T_4	T_5
LPV-IO-MPC	MSE	115.02	112.38	91.30	92.30	126.57
	T_r(s)	5334.06	5314.69	5076.86	4900.11	5410.02
	T_s(s)	651.33	763.51	268.00	460.60	312.60
	PO (%)	0.63	6.13	9.70	0	0.31
	e_{ss}	116.7	8.8	-25.9	190.9	49.9
	T_d(s)	---	---	---	---	---
Proposed method	MSE	158.82	45.72	43.68	248.20	198.4
	T_r(s)	254.1	69.57	57.58	287.90	248.2
	T_s(s)	262.1	418.81	186.64	295.80	296.60
	PO (%)	0.04	0.80	1.54	0	0.15

	e_{ss}	0	0	0	0	0
	T_d(s)	506	575	352	610	304

3.4 Conclusion

This paper interpolated MPC for LPV systems using input-output representation, requiring measured state-space variables. This paper's primary purpose was to ensure stability and applicability to real-world environments from the computational burden and prudence perspective—meanwhile, the method aimed to ensure acceptable transient and steady-state control performance. Firstly, the method has been proven to be asymptotically stable and recursive feasible. Subsequently, in two examples, the proposed method was compared with one of the up-to-the-minute research, called LPV-IO-MPC [29]. The first example showed that the proposed method markedly declines the computational load, dramatically grows the initial feasible region, and keeps the terminal region as large as possible, resulting in a less conservative controller. However, the other controller only improves the terminal region and suffers from conservatism despite convincing control performance. The second example pointed out that when the process is more complicated, large-scale, and has multiple scheduling variables, the controller's performance [276] is undermined in setpoint tracking and can be ruined by a constant disturbance. In contrast, the proposed method closely followed the reference and remained stable in the presence of disturbance.

CHAPTER 4

ROBUST MODEL PREDICTIVE CONTROLLER USING RECURRENT NEURAL NETWORKS FOR INPUT-OUTPUT LINEAR PARAMETER VARYING SYSTEMS

Chapter 4 presents the study for Objective 2. The work presented in this chapter is included in the following paper:

Hadian, M., Ramezani, A., & Zhang, W. (2021). Robust Model Predictive Controller Using Recurrent Neural Networks for Input–Output Linear Parameter Varying Systems. Electronics, 10(13), 1557. Published.

Abstract

This paper develops a Model Predictive Controller (MPC) for constrained nonlinear MIMO systems subjected to bounded disturbances. A Linear Parameter Varying (LPV) model assists MPC in dealing with nonlinear dynamics. While most research on MPC with LPV is established based on the state-space framework, some state variables might not be measurable in practice. This study represents the nonlinear process by an input-output LPV (LPV-IO) framework. Two primary objectives of this study are to reduce online computational load compared to the existing literature of MPC with an LPV-IO model and to confirm the robustness of the controller in the presence of disturbance. A Recurrent Neural Network (RNN) is employed to solve real-time optimization problems efficiently. Regarding robustness, a new control law is developed, which comprises a fixed control gain (K) and a free perturbation (C)—the former acts as stabilizing state feedback calculated offline with the terminal region. The free control moves (C), on the other hand, are determined online to ensure input-state stability in the presence of bounded disturbances. The proposed controller is proved to be stable and recursive feasible. The strategy is examined in an Alkylation of the Benzene Process and shows outstanding performance in both setpoint tracking and disturbance rejection problems. Moreover, the superiority of RNN over three conventional optimization algorithms is underlined in terms of MSE, the average time for solving the optimization problem, and the value of the cost function.

4.1 Introduction

Over the past few decades, MPC has attracted a great deal of attention in both industry and academia [2, 56, 284, 285]. This considerable prominence of MPC is rooted in several factors, including simple generalization to a MIMO system, applicability to non-minimum phase and unstable processes, capability to compensate for the delay, and imposing output/input constraints [2]. While influential theories exist in MPC for linear systems, some systems are nonlinear and operate under large operating conditions, or some external parameters can fundamentally change the process response [268]. Nonlinear MPC (NMPC) has been developed to deal with systems with nonlinear dynamics. A higher level of algorithm complexity and excessive computational load are two main drawbacks of NMPC.

An LPV framework is established in this paper to mitigate these problems by assisting the MPC in modelling dynamic and static nonlinearities [148, 273, 286, 287]. MPC founded on LPV has aroused enthusiasm among researchers to ensure practical implementation to bridge the gap between linear MPC and NMPC [63, 279, 288] (less complicated than nonlinear models and more accurate than linear models). With the help of LPV, MPC can be employed for nonlinear and/or time-varying systems [66, 148, 289, 290]. LPV offers MPC a linear model instead of nonlinear models, where the scheduling variable denotes nonlinear dynamic parameters, changing environmental conditions, and operating points. Also noteworthy is that in an LPV model, current values of scheduling variables can be measured or estimated, but the future values are unknown, so a robust MPC is required, or the future values can be estimated.

Most MPC-LPV literature centers on the state-space model, while states might not be measurable in many industrial applications. Moreover, using an observer can complicate the design. In [275], MPC-LPV with SS and IO presentations are compared for a Control Moment Gyroscope (CMG). The results showed that the execution time would reduce when using the IO framework. To this end, this paper establishes MPC based on an LPV-IO framework.

A few MPC have been documented for LPV systems under the I/O formulation, mainly focused on stability, computation, and conservatism [268]. One of the initial research on MPC with the LPV-IO model can be found at [279], where min-max (worse-case) cost functions are regarded. The cost function includes a stage cost for control performance and a terminal cost to grant stability. The optimization problem of MPC is written in Bilinear Matrix Inequality (BMI), which can be non-convex and challenging. This approach was improved in [276] in order to reduce the

complexity level markedly. The authors designed a controller based on a Linear Matrix inequality (LMI) convex problem, successfully assessed in an ideal continuous stirred tank reactor (CSTR). They also enlarge the terminal region to reduce conservatism moderately. This work employed offline controllers and terminal regions to moderately reduce the online computational burden. However, the controller still suffers from substantial online computational capacity, stemming from solving LMI equations and calculating the Markov coefficient of the LPV system [277]. Above all, it is assumed that the future values of scheduling variables can vary inside a convex polytope. This assumption is less conservative than a frozen or fixed scheduling variable over the prediction horizon, making the optimization problem nonlinear or unreachable [268, 276]. The online computational complexity of [276] has been slightly lessened in a sequence of quadratic programs. The main drawback of this work lacks provision for recursive feasibility.

The first novelty of our method is to make the controller, developed in [279], more computationally efficient. To the best of our knowledge, for the first time, a computationally efficient MPC with an LPV-IO model was presented in this study while keeping the conservatism degree as low as reasonably achievable. This paper proposes a recurrent neural network (RNN) to deal with the online optimization problem. Thanks to global convergence and low computational burden, neural network-based optimizations have been applied to linear, quadratic [216, 217], nonlinear programming [218, 219], and variational inequality problems [220, 221]. The RNN is pointed out to have a convincing performance in either MPC or NMPC [222-224], in which nonlinear optimization can be converted to quadratic programming. The motive behind selecting RNN is its promising optimality as well as less complexity, even in real-time nonlinear and large-scale optimization. That is why the studied method has been assessed in the Alkylation of the Benzene Process, a large-scale nonlinear process with 25 state variables, five inputs, and five outputs.

The second novelty of this paper is to guarantee the robustness of the system when facing bounded disturbance. The problem of disturbance rejection has not been studied so far for MPC with LPV-IO models. A stabilizing control law based on the input-output measurements is proved to ensure the superior closed-loop performance of the controller in the presence of additive disturbance. The control law includes a fixed control gain (K) and a free perturbation (C), the former designed offline with the largest possible terminal region to reduce conservatism. The free control moves are determined online to ensure input-state stability in the presence of bounded

disturbances. Adding control moves provide a less conservative framework than those studied in [276, 277], where only offline control gain is employed.

To sum up, the contributions of this method are as follows:

An RNN-based optimization algorithm is developed to offer global convergence and decline the online computational load.

Free control moves are added to the constant control gain to maintain the closed-loop stability when facing bounded disturbances.

Concerning previous studies for MPC with LPV, the proposed method inherently enjoys a shrunken conservatism degree due to finding the larger possible terminal region, using free control moves, and the global solution of the optimization problem.

The rest of the paper is organized as follows. The standard nonlinear model is reformulated as an LPV-IO model in Section 2. The proposed RMPC-LPV structure is developed in Section 3 and proved to be stable and feasible. The MPC-RNN is presented in section 4. Next, the state-space equations, inputs, outputs, and critical parameters of the Alkylation of the Benzene Process are wholly outlined in Section 5, followed by a collection of simulations to evaluate the proposed control methodology. Key findings are finally summarized in Section 6.

4.2 Problem statement

Assume the following discrete-time MIMO linear parameter-varying (LPV) transfer function with constraints:

$$y(k) = -\left(\sum_{i=1}^{n_a} A_i\big(p(k)\big)\, q^{-i}\right) y(k) + \left(\sum_{j=0}^{n_b} B_j\big(p(k)\big)\, q^{-j}\right) u(k) + d(p(k)) \tag{4.1}$$

Subject to the following constraints

$$\begin{aligned} u(k) \in U &\equiv \{u \in \mathbb{R}^{n_u} |\ |u(k)| \le u_{max}\} \\ \Delta u(k) \in V &\equiv \{\Delta u \in \mathbb{R}^{n_u} |\ |\Delta u(k)| \le \Delta u_{max}\} \\ y(k) \in Y &\equiv \{y \in \mathbb{R}^{n_y} |\ |y(k)| \le y_{max}\} \end{aligned} \tag{4.2}$$

where $y(k)$ are outputs, $u(k)$ are inputs, $p(k)$ are scheduling variables, q^{-i} is a backshift operator, n_a is the degree of the output polynomial, n_b is the degree of the input polynomial, n_u is the number of inputs, n_y is the number of outputs, and $u_{max}, \Delta u_{max}$, and y_{max} are boundaries. The state-space representation of the dynamic model given by Equation (4.1) is [279]:

$$x(k+1) = \underbrace{\begin{bmatrix} -a_1 & \cdots & -a_{n_a-1} & -a_{n_a} & b_0+b_1 & \cdots & b_{n_b-1} & b_{n_b} \\ I_{n_y} & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & & I_{n_y} & 0 & 0 & & 0 & 0 \\ 0 & \cdots & 0 & 0 & I_{n_u} & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & I_{n_u} & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & 0 & I_{n_u} & 0 \end{bmatrix}}_{A(p(k))} x(k) \quad (4.3)$$

$$+ \underbrace{\begin{bmatrix} b_0 \\ 0 \\ \vdots \\ 0 \\ I_{n_y} \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{B(p(k))} u(k) + d(k)$$

where $x(k) = [y(k-1) \dots y(k-n_a) \;\; u(k-1) \dots u(k-n_b)]^T$. For the sake of simplicity, $p(k)$ is removed from all coefficients a_i and b_i. The following assumptions are made:

A1: The parameter-varying matrix are $[A(k) \;\; B(k)] \in \Omega = Co[[A_1 \; B_1], \cdots, [A_m \; B_m]]$, where Ω is the polytope, Co is the convex hull, and $[A_i \; B_i]$ are vertices corresponding to scheduling variable p_i:

$$p = Co\{p_1, \dots p_m\} \quad (4.4)$$

A2: The profile of changes in scheduling variables is predetermined from a safety and economic point of view with the following upper and lower bounds:

$$p \in P \equiv \{p \in \mathbb{R}^{n_p} | \; |p(k)| \leq p_{max}\} \quad (4.5)$$

A3: The disturbance is bounded:

$$d(k) \in D \equiv \{d \in \mathbb{R}^{n_d} | \; |d(k)| \leq d_{max}\} \quad (4.6)$$

A4: The system (4.1) is controllable, i.e., the controllability matrix (M) has a full rank:

$$M = [B \; AB \; A^2B \dots A^{n-1}B] \quad (4.7)$$

4.3 Robust model predictive controller

This aims to define a control law that ensures the input-to-state practical stability (ISpS) of the system (4.1) with constraints (4.2). The control law is calculated by minimizing the cost function at any time instance k:

$$V_N(x_k, u_k, p_k) = \sum_{k=0}^{N} [e(k+i)^T Qe(k+i) + \Delta u(k+i-1)^T R \Delta u(k+i-1)] + [x(k+N)^T Q_p x(k+N)] \quad (4.8)$$

Subject to

$$y(k+j) = -\left(\sum_{i=1}^{n_a} A_i\, q^{-i}\right) y(k+j) + \left(\sum_{j=0}^{n_b} B_i\, q^{-i}\right) u(k+j) + d(k), \text{ j=1,2,…,N-1}$$

$$u(k+j) \in U$$

$$\Delta u(k+j) \in V$$

$$y(k+j) \in Y$$

$$x(k+N) \in X_f$$

where $e = r - y$ is the deviation of output from the reference trajectory, and Δu is the control increment. Q and R are positive definite state and input weighting matrices being in charge of closed-loop achievements, $Q_p = {Q_p}^T > 0$ is the terminal penalty matrix being designed to ensure stability, and X_f is the terminal region. The first and second parts of V_N is named stage cost (V_s) and terminal cost (V_T) respectively. Disturbances are supposed to be fixed over the prediction horizon.

According to [281], a state feedback controller can represent the system's stability with the LPV-IO form. The control law ($u(k)$) is comprised of a fixed state feedback K and a free control move (c), in which K is computed offline, and c is obtained from the minimization of (4.8):

$$u(k) = -Kx(k) + c(k) \quad (4.9)$$

The state-feedback gain accounts for maintaining the final state variable $x(k+N)$ in the terminal region while meeting constraints and keeping the controller as less conservative as possible. In fact, a constant state feedback controller confirms the stability when there is no disturbance. Otherwise, free control moves (c) will be determined online in a min-max problem to guarantee the ISpS. Thus, a part of the controller will be calculated offline for conditions that there

is not any disturbance, and a part of the controller will be set online to deal with existing disturbances or violations of $x(k+N) \in X_f$.

4.3.1 Offline controller

In this section, the stability of the system will be represented when $x(k) \in X_f$ through finding control gain K. After calculating the offline controller, a procedure to specify the terminal region is defined.

Theorem 1: Taking a disturbance-free state-space model of Equations (4.3), there exists a control law $u = -Kx$ that asymptotically stabilizes the system if V_T is a positive definite Lyapunov function such that

i. $V_T\big(x(k+1)\big) - V_T\big(x(k)\big) < 0$ for $\forall\, x \in X_f$ And $\forall\, p \in P$.
ii. If $x(k) \in X_f$ then $x(k+1) \in X_f$
iii. $|u| = |Kx| \leq u_{max}\ for\ \forall\, x \in X_f$.

Proof: Considering $V_T = x(k)^T Q_p x(k) > 0$ as the candidate Lyapunov function, the controller $u = -Kx$ exists, if the equation below satisfies:

$$V_T\big(x(k+1)\big) - V_T\big(x(k)\big) < 0 \tag{4.10}$$

$$x(k+1)^T Q_p x(k+1) - x(k)^T Q_p x(k) \leq -x(k)^T Q x(k) - \Delta u(k)^T R \Delta u(k)$$

It is assumed that there exits a K such that $A - BK$ is stable for all possible pairs of $[A(k)\ \ B(k)] \in \Omega$. By substituting $\Delta u = -Kx$, and $x(k+1) = (A-BK)x(k)$, it yields

$$\begin{aligned} &((A-BK)x(k))^T Q_p((A-BK)x(k)) - x(k)^T Q_p x(k) \\ &\leq -x(k)^T Q x(k) - \big(-Kx(k)\big)^T R(-Kx(k)) \end{aligned} \tag{4.11}$$

It can be rewritten in a more compact form:

$$M^T Q_p M - Q_p + Q + K^T R K \leq 0 \tag{4.12}$$

$$M = A - BK$$

Therefore V_T is a Lyapunov function, and the stabilizing state-feedback gain K can be found in (4.12). After showing the controller's stability, the terminal region is required to be specified, reducing the conservatism level. The terminal region X_f is considered to be an ellipsoidal invariant set:

$$X_f = \big\{x \in R^n | x(k+N)^T Q_p x(k+N) < \theta\big\} \tag{4.13}$$

While X_f is intended to be as broad as possible to reduce conservatism, it might lead to extensive control law and violation of input constraints. The maximum value of γ can be derived from an optimization problem such that input limitations are met:

$$\max \theta \tag{4.14}$$

Subject to

$$x(k+N)^T Q_p x(k+N) < \theta$$

$$|Kx| \leq u_{max}$$

In [279], it is proved that γ can equivalently be derived from the following optimization problem:

$$\min_{\gamma} \theta \tag{4.15}$$

$$\theta^2 A_T P^{-1} {A_T}^T \leq B_T {B_T}^T$$

where $A_T = [-I_n \;\; K \; I_n - K]^T, B_T = [x_{max} - x_s \;\; \Delta u_{max} \; x_{max} - x_s \;\; \Delta u_{max}]^T$, and x_s is the desired state. In this section, an offline state-feedback controller (K) and the corresponding terminal region (X_f) that stabilize the system are introduced. Unlike [276], where the future profile of scheduling variables $\{p(k+1), \dots, p(k+N)\}$ is assumed to be unknown, in this paper a least-square algorithm has been used to find scheduling variables over the prediction horizon. This means that the controller must not be robust against the possible uncertainties in the scheduling variables. For further information about the procedure for predicting the scheduling variables, please refer to [278].

4.3.2 Online controller

The free control moves will be determined online in case there is a disturbance or the condition $x(k) \in X_f$ does not meet. The online controller steers the states towards X_f, where offline control asymptotically stabilizes the system. Two Theorems are defined here to demonstrate the controller validity. Theorem 2 shows that the system remains ISpS when a bounded disturbance exists. Subsequently, the recursive feasibility of the system will be proved in Theorem 3. Because of additive disturbances, a min-max (worst-case) optimization problem is defined to cope with uncertainties. The optimization problem (4.8) can be rewritten considering the bounded disturbance:

$$\min_{c(k)} \max_{d(k) \in D} V_N(x(k), u(k), p(k), K, c(k)) \tag{4.16}$$

$$y(k+j) = -\left(\sum_{i=1}^{n_a} A_i\, q^{-i}\right) y(k+j) + \left(\sum_{j=1}^{n_b} B_i\, q^{-i}\right) u(k+j) + d(k),\ \text{j=1,2,…,N-1}$$

$$u(k+j) = -Kx(k+j) + c(k+j)$$

$$u(k+j) \in U$$

$$\Delta u(k+j) \in V$$

$$y(k+j) \in Y$$

$$d(k+j) \in D$$

$$x(k+N) \in X_f$$

The optimal solution to the problem (4.8), subjected to the system (4.1) with constraints (4.3), is the sequence $[c^*(k)\ c^*(k+1)\ \ldots c^*(k+N-1)]$ corresponding to $[u^*(k)\ u^*(k+1)\ \ldots u^*(k+N-1)]$. Two theorems are defined in this section; theorem 2 for showing stability and theorem 3 for showing recursive feasibility. To being with, a robust positively invariant (RPI) set is needed to be defined as below:

A set X is RPI for $x^+ = Ax + Bu + d$ with $u = -Kx$ if $x \in X\ for\ \forall\ x^+ \in X, d \in D$, $[A(k)\ \ B(k)] \in \Omega$

Theorem 2: There exists a sequence of optimal control input (c^*), and $\alpha, \beta, \gamma\ \lambda$ that ensures the ISpS of the system (4.1) with bounded disturbances (4.6) with the following assumptions:

i. $|u| = |Kx| \leq u_{max}\ for\ \forall\ x \in X_f$.
ii. X_f is an RPI set of the system (4.1) with $u = -Kx$.
iii. $\alpha\|x\|^\lambda \leq \|Px\| \leq \beta\|x\|^\lambda\ for\ \forall\ x \in X_f$.
iv. $\|Px^+\| - \|Px\| \leq -(\|Qx\| + \|Rkx\|) + \mu\|d\|$ for $x \in X, d \in D$, $[A(k)\ \ B(k)] \in \Omega$
v. $\|Qx\| + \|Ru\| \geq \gamma\|x\|^\lambda\ for\ \forall\ x \in X_f$

The underlying causes of these assumptions can be found in [291].

Proof: The definition of ISpS for the system (4.1) is

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|) + s_1 \tag{4.17}$$

$$V(x^+) - V(x) \leq -\alpha_3(\|x\|) + \alpha_4(\|d\|) + s_2$$

where x^+ RPI set in which $x^+ = Ax + Bu + d$ with $u = -Kx$, and α_1, α_2, and α_3 are $\mathcal{K}_\infty$-function, α_4 is $\mathcal{K}$-function, and s_1 and s_2 are positive numbers such that $x \in X$, $x^+ \in X$, and $d(k) \in D$. Also noteworthy is the fact that ISpS is equivalent to ISS for $s_1 = s_2 = 0$. The $V_N(x) > 0$ should be first proved to be bounded so that the Equation (4.17) can be expressed as:

$$V_i(x(k+i)) = \min_{c_k} \max_{d(k)\in D} [x(k+i)^T Q x(k+i) + u(k+i)^T R u(k+i) + V_{i+1}(x(k+i+1))] \quad (4.18)$$

Equation (4.18) is derived from a mathematical induction technique such that:

$$V_0(x(k)) = \min_{c_k} \max_{d(k)\in D} [x(k)^T Q x(k) + u(k)^T R u(k) + V_1(x(k+1))] \quad (4.19)$$

$$V_1(x(k+1)) = \min_{c_k} \max_{d(k)\in D} [x(k+1)^T Q x(k+1) + u(k+1)^T R u(k+1) + V_2(x(k+2))]$$

...

$$V_N(x(k+N)) = x(k+N)^T Q_p x(k+N)$$

The Equation (4.10), when the disturbance is regarded to be nonzero, results in:

$$x(k+1)^T Q_p x(k+1) - x(k)^T Q_p x(k) \le -x(k)^T Q x(k) - u(k)^T R u(k) + d_{max} \quad (4.20)$$

By developing this Equation for the next value of state variables, the following equations will be reached:

$$x(k+2)^T Q_p x(k+2) - x(k)^T Q_p x(k) \le -x(k+1)^T Q x(k+1) - u(k+1)^T R u(k+1) + d_{max} \quad (4.21)$$

…

$$x(k+N)^T Q_p x(k+N) - x(k)^T Q_p x(k) \le -\sum_{i=0}^{N-1} x(k+i)^T Q x(k+i) + u(k+i)^T R u(k+i) + d_{max}$$

From Equations (4.18) and (4.20), and substituting $u(k) = -Kx(k)$, the following is obtained.

$$V_N(x(k)) \le \max_{d(k)\in D} [x(k)^T Q x(k) - x(k)^T K R x(k)] + x(k+1)^T Q_p x(k+1) \quad (4.22)$$

where $V(x_{k+1}) \le x(k+N)^T Q_p x(k+N)$ and therefore

$$V_N(x) \le x(k)^T (Q - KR + Q_p) x(k) + d_{max} = (Q - KR + Q_p)\|x\| + d_{max} \quad (4.23)$$

It can be concluded that $V_N(x)$ has an upper bound and $\alpha_1 = Q - KR + Q_p$, $s_1 = d_{max}$ in the right-hand side of the first equation of (4.23). The difference between the two sequential Lyapunov functions is

$$V(x(k+1)) - V(x(k)) \leq \max_{d(k)\in D} -[x(k)^T Q x(k) + u(k)^T R u(k)] + x(k+N)^T Q_p x(k+N) \quad (4.24)$$

This can be expressed by

$$V(x(k+1)) - V(x(k)) \leq x(k)^T\big(-Q + KR + Q_p\big)x(k) + d_{max} = -(Q - KR - Q_p)\|x\| + d_{max} \quad (4.25)$$

Accordingly, the difference in the Lyapunov function is bunded such that $\alpha_3 = Q - KR - Q_p$, $s_2 = d_{max}$. It has been proved that the proposed controller is ISpS, and the optimal solution, satisfying input, output, and terminal constraints, can derive from a min-max problem (4.16). In the next stage, it has been shown that the solution is feasible.

Theorem 3: Given system (4.1) with constraints (4.2) and bounded disturbance (4.6), the closed-loop system is ISpS for $x_0 \in x_N$, and the system is feasible.

Proof: Suppose that there exists a terminal region, derived from (4.15), and a control gain K, calculated by Equation (4.12); then the system is stable and $x_0 \in x_N$. Consequently, if the optimization problem (4.8) is feasible at time instance k, it remains feasible in subsequent instances. After finding the optimal K, c^*, and the corresponding u^* at time instance k, the system of Equation (4.3) at the next instance can be rewritten.

$$x(k+1) = A\big(p(k)\big)x_k + B\big(p(k)\big){u_k}^* + d(k) \quad (4.26)$$

And

$$x(k+2) = A\big(p(k+1)\big)x_1 + B\big(p(k+1)\big){u_0}^* + d(k+1) \quad (4.27)$$

In general, the sequence of system states can be described as follow:

$$x(k+i+1) = A\big(p(k+i)\big)x_{k+i} + B\big(p(k+i)\big){u_{k+i}}^* + d(k+i) \quad (4.28)$$

By substituting $u(k) = -Kx(k) + c(k)$, the Equation (4.28) can be derived:

$$x(k+i+1) = \Big(A\big(p(k+i)\big) - B\big(p(k+i)\big)\Big) * x_{k+i} + B\big(p(k+i)\big){c_{k+i}}^* + d(k+i) \quad (4.29)$$

The above equation for $i = N$ results in

$$x(k+N+1) = \Big(A\big(p(k+N)\big) - B\big(p(k+N)\big)\Big) * x_{k+N} + B\big(p(k+N)\big){c_{k+N}}^* + d(k+N) \quad (4.30)$$

This implies that the optimal solution is feasible since

$$\begin{aligned} for\ i = 0, X_{N-1} \in X_N \\ for\ i = 1, X_{N-2} \in X_{N-1} \in X_N \\ for\ i = N, X_0 \in X_N\ and\ x(k+N+1) \in\ X_N \end{aligned} \tag{4.31}$$

According to theorems 1 to 3, the developed controller has two components. The first is determined offline, which included a control gain K and terminal region X_f. When $x \in X_f$, the offline controller asymptotically stabilizes the system. Otherwise, when there is uncertainty, an online controller verifies the ISpS. The online optimization problem is shown to be feasible. In the next section, a dynamic neural network is constructed to solve the online optimization problem for the first time.

4.4 Real-time optimization problem using RNN

The global convergence and low complexity of RNN for optimization of linear models, constrained linear models, linear models with uncertainty, and various nonlinear models are represented in the literature. In this study, RNN optimizes a real-time QP problem enjoying parallel computation. To begin with, the original optimization problem is required to transform into a standard form. The input-state relationship of Equation (4.1) can be expressed in the following form:

$$x(k+1) = A\big(p(k)\big)x(k) + B(p(k))u(k) + d(p(k)) \tag{4.32}$$

The vector of predicted outputs, inputs, and disturbance are:

$$\begin{aligned} X(k) &= [X(k+1) \dots X(k+N)]^T \in \mathcal{R}^{m*N} \\ U(k) &= [u(k) \dots u(k+N-1)]^T \in \mathcal{R}^{n*N} \\ D(k) &= [d(k+1) \dots d(k+N)]^T \in \mathcal{R}^{o*N} \end{aligned} \tag{4.33}$$

Where $m, n,$ and o are the number of states, inputs, and disturbances, respectively. The predicted states can be easily shown to be in the following form:

$$\begin{aligned} X(k+j) &= G\big(p(k+j-1)\big)X(k+j-1) + F\big(p(k+j-1)\big)U(k+j-1) \\ &\quad + D(k+j-1), \qquad j = 1, \dots, N \\ G\big(p(k)\big) &= [A\big(p(k)\big)\ A\big(p(k)\big)^2 \dots\ A\big(p(k)\big)^N]^T \end{aligned} \tag{4.34}$$

$$F\big(p(k)\big) = \begin{bmatrix} B\big(p(k)\big) \\ A\big(p(k)\big)B\big(p(k)\big) + B\big(p(k)\big) \\ \vdots \\ A\big(p(k)\big)^{N-1}B\big(p(k)\big) + \cdots + A\big(p(k)\big)B\big(p(k)\big) + B\big(p(k)\big) \end{bmatrix}$$

The constraints (4.2) can be represented as:

$$\begin{aligned} -u_{max} &\le u(k) \le u_{max} \\ -\Delta u_{max} &\le u(k) - u(k-1) \le \Delta u_{max} \\ -x_{max} &\le Gu + F + d \le x_{max} \end{aligned} \tag{4.35}$$

And the optimization problem (4.16) can be written as follows:

$$\begin{aligned} &\min_U X^T \bar{Q} X + U^T R U = \min_U (GX + FU + D)^T \bar{Q} (GX + FU + D) + U^T R U \\ &\min_U U^T * ((FU)^T \bar{Q} (FU) + R) * U + (GX + FU)^T \bar{Q} (GX + FU) \end{aligned} \tag{4.36}$$

where

$$\bar{Q} = \begin{bmatrix} Q & 0 \\ 0 & Q_p \end{bmatrix}$$

According to Equations (4.35) and (4.36), the standard form can be expressed as:

$$\begin{aligned} &\min_v \frac{1}{2} v^T H v + b^T v \\ &\text{Subject to } Tv \le q \end{aligned} \tag{4.37}$$

where

$$H = 2 * ((FU)^T \bar{Q} (FU) + R)$$

$$b = 2 * (GX + FU)^T \bar{Q} (GX + FU)$$

$$T = \begin{bmatrix} I_{n*n} \\ -I_{n*n} \\ I_{n*n} \\ -I_{n*n} \\ G \\ -G \end{bmatrix}, q = \begin{bmatrix} u_{max} \\ -u_{max} \\ \Delta u_{max} + u(k-1) \\ \Delta u_{max} - u(k-1) \\ X_{max} - F - d \\ -X_{max} + F + d \end{bmatrix}$$

Remark 1: The vector q is unknown since d is not measurable, while it can be estimated by the difference between the measured output and model output. In [228], a simple model to find unknown parameters is proposed.

Remark 2: In [222, 228], a simplified form of a dual neural network is described to ensure a lower computational burden by defining the dual form of the optimization problem (4.37) as:

- State equation:

$$\frac{d\omega}{dt} = \lambda(-TH^{-1}T^T\omega + M(TH^{-1}T^T\omega - TH^{-1}c - \omega) + TH^{-1}c) \tag{4.38}$$

Where ω is the state variable of the network, $\lambda > 0$ adjusts the convergence rate of RNN, and M is a piecewise linear function as:

$$M(z) = \begin{cases} q_{min} & z < q_{min} \\ z & q_{min} < z < q_{max} \\ q_{max} & z > q_{max} \end{cases} \tag{4.39}$$

- Output equation:

$$v = H^{-1}T^{T}\omega - H^{-1}c \tag{4.40}$$

In the dual form of the optimization problem, constraints are added to the main cost function as a penalty term. This means that if $Tv < q$, the cost function decreases by a factor of α; otherwise, for $Tv > q$, the cost function is penalized. The single-layer RNN is given in Figure 4.1. The global convergence of the proposed network is verified in [228].

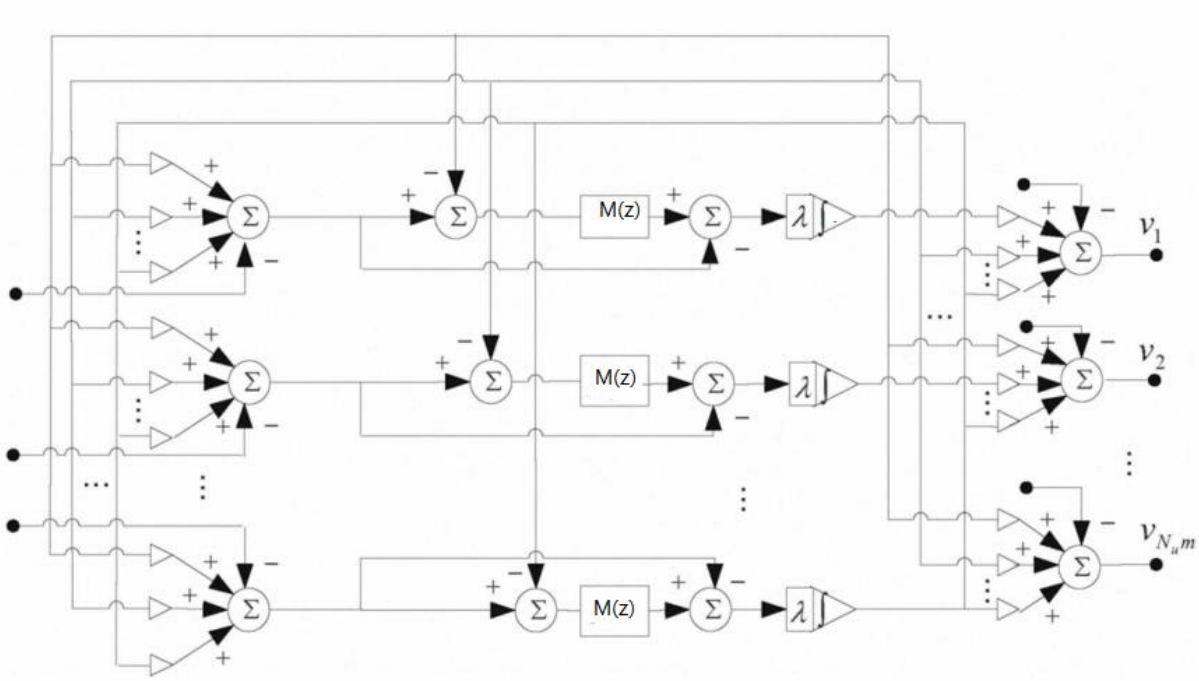


Figure 4.1: RNN framework [228]

Finally, the proposed MPC algorithm can be summarized in the following steps:

1) The value of $Q, R, \lambda, N, u_{max}, y_{max}, d_{max}, p$ and model(A, B)
2) Specify K using (4.12) and θ using (4.15).
3) Repeat the procedure of finding $u(k)$ by solving Equations (4.38) to (4.40).

The whole procedure is also depicted in Figure 4.2.

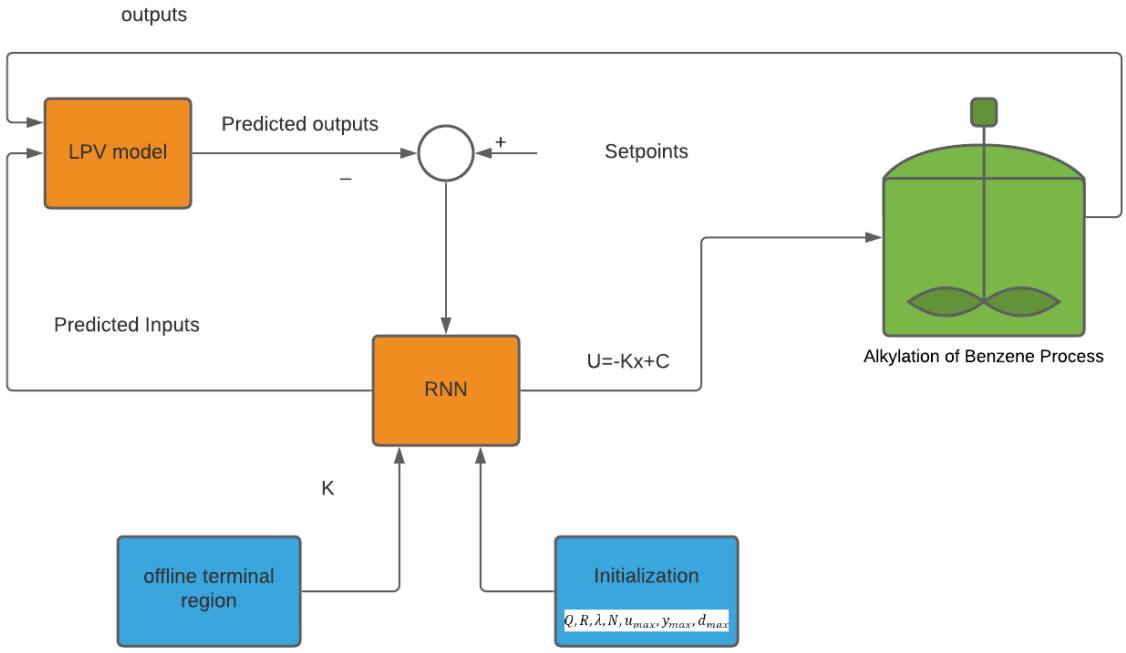


Figure 4.2:The proposed approach flowchart

4.5 Results and discussion

The performance of the controller in terms of setpoint tracking and disturbance rejection is investigated in the Alkylation of Benzene Process. For more information about this process, please refer to section 3.3.2.1. The disturbances are considered as a step change at $t = 1500$ s with an amplitude of 2×10^{-3} is made in F_1 and F_2. The controller parameters were $N = 6$; $Q = 1000 * I_{5\times5}, R = 10^{-7} * I_{5\times5}$ and $\lambda = 3.5$. All controllers employed in the performed comparison used the same parameters. The proposed approach is compared with linear RMPC [236], and the LPV controller studied in [276] in Figure 4.3, where the dashed red line denotes the references, the blue line is the proposed method, the green line is linear RMPC, and the black line is LPV-IO RMPC. Looking firstly at reference tracking, the proposed controller and LPV-RMPC (save T_5) reached the setpoint at an acceptable time, while LPV-RMPC suffers from large-amplitude oscillations. The reason behind the fluctuation in T_5 can be found in Equation (4.35), such that C_{D0} directly affect the output, and LRMPC failed to tackle the changes in Molar densities of pure D. In general, regardless of rise time, settling time, and overshoot, all three methods followed the setpoints for T_1 to T_4. Turning to disturbance rejection, the studied approach outperforms others by removing the

disturbance in a short time ranging from 120s to 355s for T_1 to T_5. After time t=1500s, when the disturbance was applied to the process, the LRMPC violated both inputs and outputs constraints, especially in T_1, although it can overcome the disturbances after a while; on the other hand, LPV-RMPC became highly unstable. Regarding the cost function, the wasted resources by LPV-RMPC are 5.04 times more than LRMPC and 1.59* 10 greater than that of the proposed method, which proved that the studied approach is significantly cost-effective. More importantly, the cost function of LRMPC and LPV-RMPC did not converge to zero, so they cannot deal with changes in three scheduling variables. The MSE of all three methods for different outputs is reported in Table 4.1. Using the proposed method leads to a sharp decline in error compared to other methods, as the MSE for LRMPC, LPV-RMPC, and the proposed method were 447.56, $5.09*10^3$, and 43.84, respectively. In short, the proposed method had an acceptable speed (rise time, settling time, and time required to remove disturbance) and lower MSE and cost.

Table 4.1:MSE for five vessels temperatures in three studied techniques

MSE	T_1	T_2	T_3	T_4	T_5	total
LRMPC	391.40	387.25	346.72	331.76	408.73	347.56
LPV-RMPC	5.33e+03	5.31e+03	5.07e+03	4.90e+03	5.41e+03	5.09e+03
Proposed method	67.32	65.30	52.34	60.00	76.45	43.84

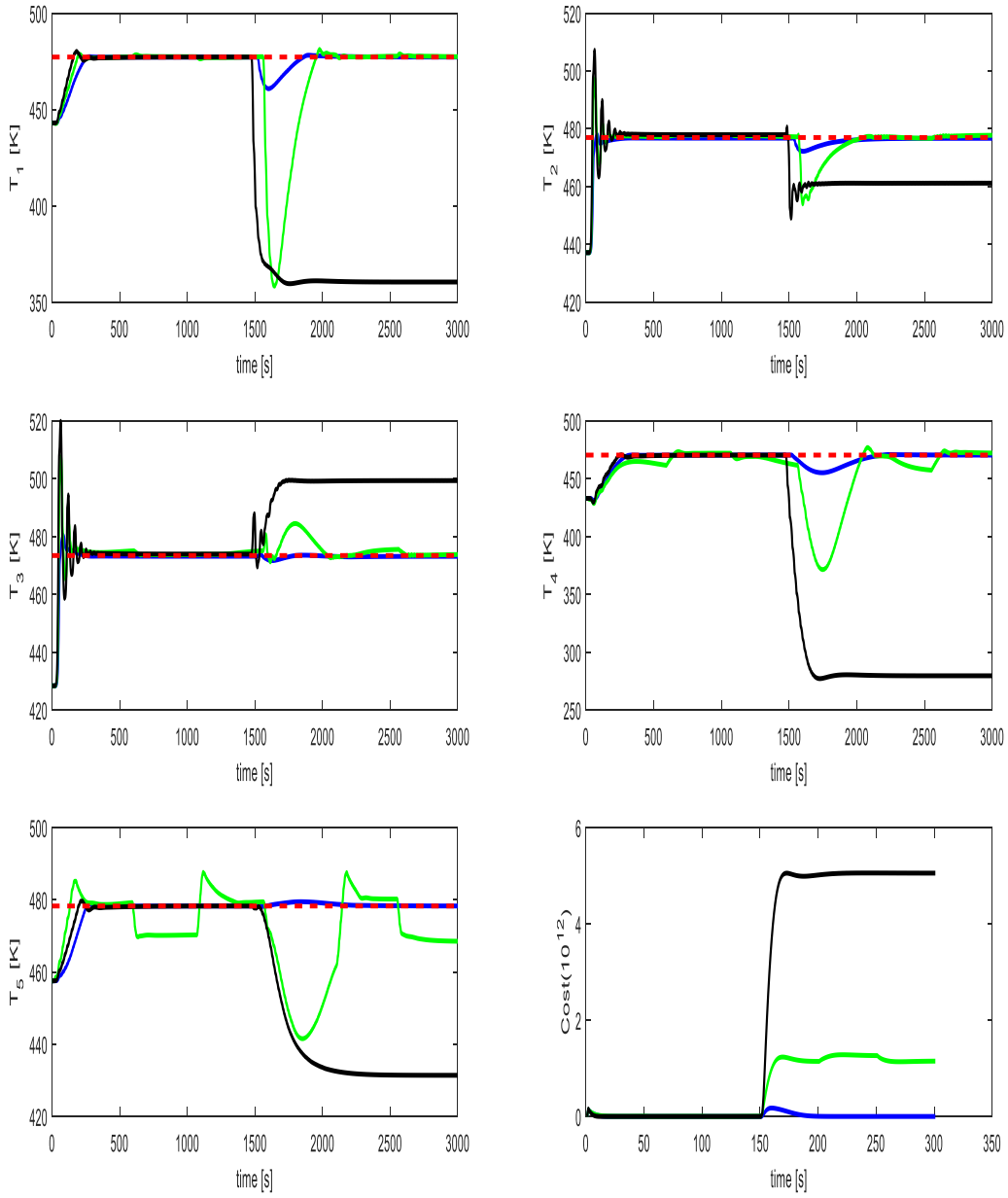


Figure 4.3: the temperatures of vessels and cost function using three studied methods (dashed red line: references, the blue line: proposed method, the green line: linear RMPC, the black line: LPV-IO RMPC)

At the same time, it has been proved that using RNN reduces the average time required for computing the control action in each sampling in contrast with NMPC based on Sequential Quadratic Programming (SQP), Genetic Algorithm (GA), and Singular Value Decomposition

(SVD). These methods' average run-time and MSE are reported in Table 4.2 and Figures 4.4 to 4.6. The RNN-based method experienced far less MSE and cost and quickly found the optimal control actions. GA ranked second in cost and MSE while slow, resulting in instability and significant fluctuation in some simulations. In stark contrast, however, SVD has faster responses, albeit it failed to converge the minimum cost function at times and saw monumental errors. SQP, on the other side, had a smaller MSE and cost than SVD and solved the optimization problem in a shorter time than GA did.

Table 4.2: A comparison of different optimization algorithms for finding the control signals

Optimization algorithm	MSE	Average time	Cost
RNN	43.84	0.033	2.03e+04
SQP	153.12	0.76	8.04e+06
GA	76.55	0.81	5.01e+04
SVD	241.22	0.013	2.05e+08

Figure 4.4:MSE of different optimization algorithms

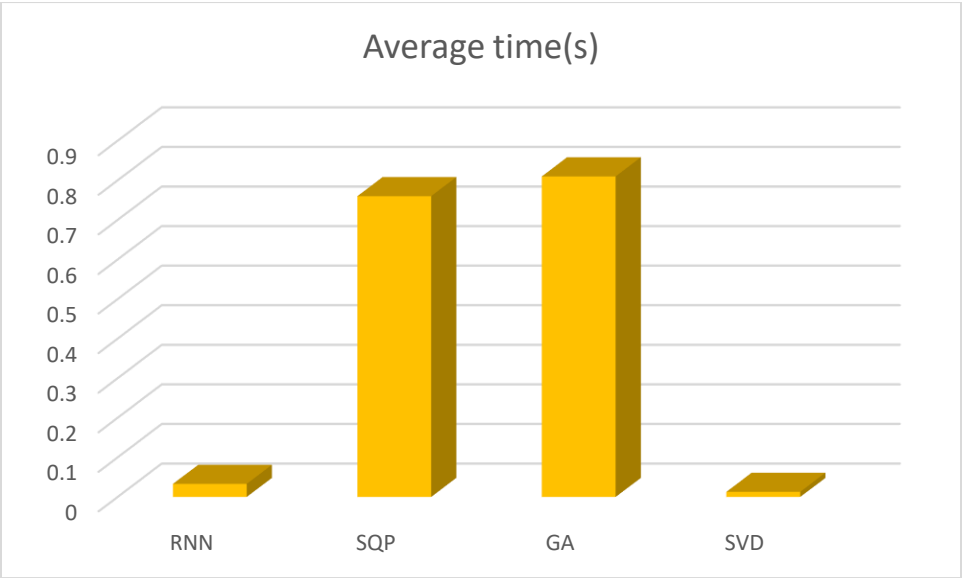


Figure 4.5: Average runtime of different optimization algorithms

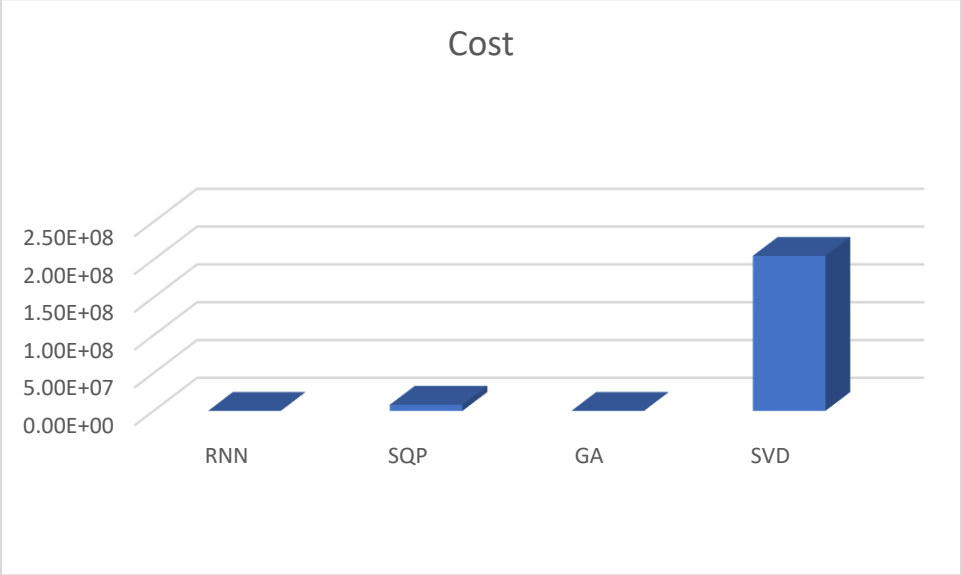


Figure 4.6: Cost function of different optimization algorithms

4.6 Conclusion

An RMPC with an LPV-IO model is investigated in this paper, in which an RNN algorithm solves the real-time optimization problem. This approach's effectiveness was monitored and compared with LRMPC [236] and LPV-RMPC [276] in an Alkylation of Benzene Process, which is nonlinear and large scale with three scheduling variables. The proposed method was astonishingly successful in both setpoint tracking and disturbance rejection having reasonable rise

time, settling time, MSE, and response amplitude. The LRPMC had a similar speed response while suffering from adverse oscillation. The LPV-RMPC can partially track the predefined reference outputs but fails to remain stable when facing disturbances.

Meanwhile, four optimization algorithms are utilized to solve the online optimization problem of the proposed controller. Results showed that RNN and GA outdid other algorithms in error reduction and finding the optimal solution; nonetheless, GA was sluggish and detrimental to stability. SVD had the fastest convergence rate with the highest MSE and cost value, and SQP, with a stable response, had the worst execution with a great average time and MSE.

CHAPTER 5

APPLICATION OF NONLINEAR MODEL PREDICTIVE CONTROL FOR AN ANAEROBIC DIGESTION PROCESS USING AN INPUT-OUTPUT LPV SYSTEM IDENTIFIED BY LS-SVM

Chapter 5 presents the study for Objective 3. The work presented in this chapter is included in the following paper:

Hadian, M., Zhang, W., & Ghanevati, A. (2022). Application of Nonlinear Model Predictive Control for an Anaerobic Digestion Process Using an Input-output LPV System Identified by LS-SVM. International Journal of dynamics and control, revised.

Abstract

Anaerobic Digestion (AD) process has been widely used as a green source of renewable energy for decades to scale back greenhouse gas emissions and conserve the environment. Having a nonlinear nature with a dramatic rate of variations in feed composition, designing an effective controller for this process has been surrounded by controversy. In this research, an MPC supported by a linear parameter varying (LPV) framework has been employed. The LPV model deals with the nonlinearities and changes in the parameters of this system. However, identifying the LPV model is a knotty problem, and an unsuitable selection of structural dependence and the order of coefficients of LPV might lead to closed-loop stability degradation. With this in mind, the LPV model's coefficients are identified by a Least Squares Support Vector Machine (LS-SVM) approach, where prior knowledge of the type of decadency and order is not required. The proposed strategy is tested on a virtual AD plant represented by complex Anaerobic Digestion Model No.1. The findings revealed the controller's superior performance in terms of output regulation when there are periodic changes in setpoint and the scheduling variable.

5.1 Introduction

Anaerobic digestion (AD) is an efficient biochemical degradation of organic waste, by which biogas is produced (a mixture of methane and carbon dioxide) as a green renewable energy supply. Other outstanding merits of this process would be reducing environmental pollution (caused by organic waste) and supplying nutrient fertilizer [1, 2]. Despite these striking features, the AD process consists of four complicated steps involving physicochemical and biological processes with highly nonlinear characteristics and uncertain kinetic parameters. Furthermore, the AD plants could be overloaded or inhibited by altering the form of input waste, leading to process instability. As a result, it is crucial for successful biogas production and keeping the AD stable to develop a proper control system [3-6]. Significant efforts have been put forwards in developing closed-loop control algorithms for methane production regulation ranging from simple PI [7, 8] to nonlinear and optimal ones [9-13].

The primary motive for selecting the model predictive controller (MPC) is seeking optimal control action subjected to input constraints, which is of vital importance from a practical point of view. Since predicting the system behaviour multi-step ahead, MPC can handle complex dynamics such as systems with a time delay or non-minimum phase and even unstable systems [14]. The previous studies of MPC on the AD process utilized the nonlinear model of AM2 or its modified version so as to control the produced methane flow rate [11, 13, 15]. The serious repercussion of using these nonlinear models is that the optimization problem can be nonconvex or computationally inefficient. On the other hand, using linear models is not as accurate as nonlinear models and only valid around the operating point. In light of this, an LPV framework is developed in this paper to bridge the gap between linear and nonlinear modelling without state estimation. This framework can trade off between computational load and the high precision of the model.

LPV models assist the MPC structure in modelling dynamic and static nonlinearities [16]. They straightforwardly identify the process by describing the process's operational knowledge as scheduling variables. LPV can model nonlinear dynamics without linearization by offering a linear structure, which depends on scheduling variables. In contrast to nonlinear identification methods such as nonlinear ARX [17] or neural networks [18], the LPV model with linear structure reduces the computational load and computation time. Input-output (IO) and state-space representations are two primary forms of LPV systems developed by researchers in the context of MPC. The inspiration for the current work comes from the fact that previous research rests on the assumption

that state-space variables are measurable, which is unattainable in practice. Even using an observer for estimating variables can deteriorate closed-loop performance [16, 19-21]. That is why an LPV-IO model is preferred to model the AD process.

In an LPV model, the coefficients are defined in terms of a function of unknown parameters related to the scheduling variable, the so-called basis function $\emptyset(\rho)$. The type and order of these functions can be challenging. An overwhelming majority of studies find the coefficients of the LPV model with parametric identification methods [22-24]. However, this strategy suffers from increased bias and variances due to faulty selection of the type of $\emptyset(\rho)$ and order of model, requiring professional knowledge of the process [25]. In some situations, the number of basic functions and their parameters increase to capture the great diversity of nonlinearities and uncertainties, resulting in an over-parametrized structure. A nonparametric structure based on a support vector machine (SVM) has been proposed to identify the LPV model and tackle the difficulties mentioned above.

SVM was originally a class of supervised learning techniques applied to a broad spectrum of fields, including data classification [26], function estimation [27], time series prediction[28], speech recognition [29], and disease diagnosis [30]. Thanks to the exceptional merits of SVM, this method has also been used to identify nonlinear models in various applications [19, 31]. The least-square SVM (LS-SVM) method [25, 32] is a variation of the original SVM approach. This method can identify the LPV model without prior knowledge of the coefficients' type/order of dependencies. In the LPV associated with SVM, parameters are estimated by solving a linear optimization problem, having a unique solution and cost-effective computational volume.

This research's main contribution is regulating methane flow rate with an MPC-LPV framework that reduces complexity over nonlinear MPC and increases accuracy over linear MPC. Furthermore, the LPV model coefficients are identified by a nonparametric LS-SVM tool for simplicity and accuracy compared with parametric LPV models. The proposed LPV_SVM_MPC strategy is implemented on Anaerobic Digestion Model No.1 (ADM1), and simulation results evaluate its performance. The remainder of the paper is organized as follows. The proposed MPC-LPV-SVM structure is first developed in Section 2. The AD process is presented in section 3, followed by a set of simulations to explore the proposed control methodology's performance. Key findings are finally summarized in Section 5.

5.2 Proposed strategy

This section describes the LPV-based predictive control method. First, how to calculate the coefficients of the input-output LPV with SVM is explained. This method can reasonably predict the behaviour of nonlinear systems without the need to measure state variables. Finally, the cost function and how to find the control signal from the constrained optimization problem are described.

5.2.1 Linear parameter varying system

One of the basic structures of the LPV model in discrete space is called the autoregressive model with exogenous input (ARX), which is shown as follows in a single-input and single-output mode:

$$\mathrm{y(k)} = -\sum_{\mathrm{i=1}}^{\mathrm{n_a}} \mathrm{a_i\big(p(k)\big)y(k-i)} + \sum_{\mathrm{j=0}}^{\mathrm{n_b}} \mathrm{b_j\big(p(k)\big)u(k-j)} + \mathrm{e(k)} \tag{5.1}$$

Where k denotes the sampling instant, and u and y are input and output signals, respectively. $p(k)$ is scheduling variable, and e is white noise. The set of $\mathrm{a_i}$ and $\mathrm{b_j}$ statically depend on p(k), which means every single coefficient only relies on the amount of $p(k)$ on that sampling time. One might express that the LPV system looks similar to the LTV system, while the coefficients are functions of $p(k)$ instead of time [33]; that is to say, LPV can be an extended version of LTV systems. With $p(k)$, an LPV model can describe nonlinear and time-varying features of the process.

5.2.2 Identification of input-output model with SVM

In this part, the LPV model (Equation (5.1)) is formulated by a computationally-efficient LS-SVM approach in order to estimate the coefficients based on N recorded data $\mathcal{D}_N = \{u(k), p(k), y(k)\}_{k=1}^{N}$ without the need to determine their dependence on $p(k)$ [25]. In this modelling approach, it is supposed that the dependence of the coefficients of Equation (5.1) upon signal $p(k)$ is undetermined. Therefore, the parameterized model of the system (5.1) is presented as follows:

$$\mathcal{M}_{\omega,\varphi}: y(k) = \sum_{i=1}^{n_g} \omega_i^T \phi_i\big(p(k)\big)\, x_i(k) + e(k) \tag{5.2}$$

In the rest of the paper, $\phi\big(p(k)\big)$ are indicated by $\phi(k)$ for the sake of simplicity. In Equation (5.2), $\phi_i: R \to R^{n_H}$ is undefined potentially infinite-dimensional feature map corresponding to the ith parameter vector and

$$x_i(k) = y(k-i), \quad i = 1, \dots, n_a \tag{5.3)(a}$$

$$x_{n_a+1+j}(k) = u(k-i), \quad j = 0, \dots, n_b \tag{5.3) (b}$$

where $n_g = n_a + n_b + 1$ and $w_i \epsilon R^{n_H}$ is parameter vector:

$$\omega = \left[\omega_1^T \ \omega_2^T \ \dots \ \omega_{n_g}^T\right]^T \tag{5.4}$$

$$\varphi(k) = \left[\phi_1^T\big(p(k)\big)x_1(k) \ \ \phi_2^T\big(p(k)\big)x_2(k) \dots \ \ \phi_{n_g}^T\big(p(k)\big)x_{n_g}(k)\right]^T \tag{5.5}$$

According to Equations (5.4) and (5.5), Equation (5.2) can be rewritten as follows:

$$y(k) = w^T \varphi(k) + e(k) \tag{5.6}$$

The ultimate goal of LS-SVM is the minimization of the following cost function:

$$J(\omega, e) = \frac{1}{2}\sum_{i=1}^{n_g} \omega_i^T \, \omega_i + \frac{\gamma}{2}\sum_{k=1}^{N} e^2(k) \tag{5.7}$$

In the above equation, γ is a positive real number called regularization factor, $e(k)$ indicates the error of the estimated model, and $\sum_{i=1}^{n_g} \omega_i^T \, \omega_i$ is account for the unknown dependence of LPV model. The $\mathcal{M}_{w,\varphi}$ can be estimated by solving the optimization problem in Equation (5.8).

$$\min_{\omega,\mathrm{e}} \ J(\omega, e) = \frac{1}{2}\sum_{i=1}^{n_g} \omega_i^T \, \omega_i + \frac{\gamma}{2}\sum_{k=1}^{N} e^2(k) \tag{5.8}$$

$$s.t. \ \ e(k) = y(k) - \sum_{i=1}^{n_g} \omega_i^T \phi_i(k) \, x_i(k)$$

The following lagrangian is constructed to solve the constrained optimization problem in Equation (5.8).

$$\mathcal{L}(\omega, e, \alpha) = J(\omega, \mathrm{e}) - \sum_{k=1}^{N} \alpha_k \left(\sum_{i=1}^{n_g} \omega_i^T \phi_i(k) \, x_i(k) - y(k) \right) \tag{5.9}$$

where $\alpha_k \epsilon R$ denotes lagrangian multipliers, and the optimum solution would be:

$$\frac{\partial \mathcal{L}}{\partial e} = 0 \;\rightarrow\; \alpha_k = \gamma e(k), \tag{5.10)(a}$$

$$\frac{\partial \mathcal{L}}{\partial \omega_i} = 0 \;\rightarrow\; \omega_i = \sum_{k=1}^{N} \alpha_k \phi_i(k)\, x_i(k), \tag{5.10)(b}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \;\rightarrow\; e(k) = y(k) - \sum_{i=1}^{n_g} \omega_i \phi_i(k)\, x_i(k), \tag{5.10) (c}$$

Substituting 5.10(a), 5.10(b), and 5.10(c) in Equation (5.6) yields

$$\sum_{i=1}^{n_g} \underbrace{\left(\sum_{k=1}^{N} \alpha_k^T x_i(k) \phi_i^T(k) \right)}_{\omega_i^T} \phi_i(k) x_i(k) + \underbrace{\gamma^{-1}\alpha_k}_{e(k)} \qquad for\ k \epsilon \{1, \dots, N\} \tag{5.11}$$

By applying this equation for N collected data, this equation can be reformulated as a compact form of N set of equations as below:

$$Y = (\Omega + \gamma^{-1} I_N)\alpha \tag{5.12}$$

where $Y = [y(1) \dots y(N)]^T$, $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_N]^T \epsilon\, R^N$, I_N is the identity matrix of size N, and Ω is a N by N matrix of kernel functions whose the (j,k)-th entry of this matrix is defined in Equation 5.13:

$$[\Omega]_{j,k} = \sum_{i=1}^{n_g} \left[\Omega^{\mathrm{i}}\right]_{j,k} \alpha \tag{5.13}$$

with

$$\begin{aligned} \left[\Omega^{\mathrm{i}}\right]_{j,k} &= x_i(j) \phi_i^T(j) \phi_i(k) x_i(k), \\ &= x_i(j) \langle \phi_i^T(j), \phi_i(k) \rangle x_i(k), \\ &= x_i(j) (K^i(p(j), p(k))) x_i(k) \end{aligned}$$

In Equation (5.13), K^i are positive-definite kernel functions, which derive from the inner product of $\phi_i^T(j)\phi_i(k)$. This strategy is named kernel trick and is capable of identifying a_i و b_j with a wide range of nonlinear dependencies without any explicit definition of the feature map $\{\phi_i\}_{i=1}^{n_g}$ [10, 11]. Among all kernel functions, the well-known Radial Bases Functions (RBF) are used in this study, in which σ is a hyper-parameter tuned by the user to control the width of RBF.

$$K^i(p(j)\,,p(k)) = \exp\left(-\frac{\big(p(j)-p(k)\big)^2}{\sigma_i^2}\right), \qquad i = 1, \dots, n_g \tag{5.14}$$

Having kernel functions, the matrix Ω can be computed.

$$\alpha = (\Omega + \gamma^{-1} I_N)^{-1} * Y \tag{5.15}$$

1.1 Model predictive controller

The cost function of applied MPC is expressed in Equation (5.16) [14]:

$$J\big(N_p, N_u\big) = \sum_{j=1}^{N_p} \delta[\hat{y}(k+j|k) - y_r(k+j)]^2 + \sum_{j=0}^{N_u-1} \lambda[\Delta u(k+j|k)]^2 \tag{5.16}$$

where $\hat{y}(k+j|k)$ is a j-step ahead prediction of output over the prediction horizon, $y_r(k+j)$ is the future reference trajectory, N_p and N_u are the prediction and control horizon and δ and λ are weighting factors. The upper and lower bounds on both process input and rate of change of input are considered as the system constraints:

$$u_{min} \le u \le u_{\max} \tag{5.17}$$
$$\Delta u_{min} \le \Delta u \le \Delta u_{\max}$$

By solving the following optimization problem, at each sampling time, the future control increments are calculated:

$$\min_{\Delta u(k|k),\Delta u(k+1|k)\dots\Delta u(k+N_u-1|k)} J(N_p\,, N_u) \tag{5.18}$$
$$s.t.\, u_{min} \le u \le u_{\max},$$
$$\Delta u_{min} \le \Delta u \le \Delta u_{\max}$$

It should be mentioned that only the first element of the computed input vector ($\Delta u(k|k)$) is applied to the process at each sampling time. The leading role of the LPV-SVM framework is estimating the output signal ($\hat{y}(k+j|k)$) over the prediction horizon N_P, which is required in the MPC setup. With no measurement noise, the SVM-LPV model in Equation (5.2) is written as below:

$$y(k) = \sum_{i=1}^{n_g} \omega_i \phi_i(k)\, x_i(k) \tag{5.19}$$

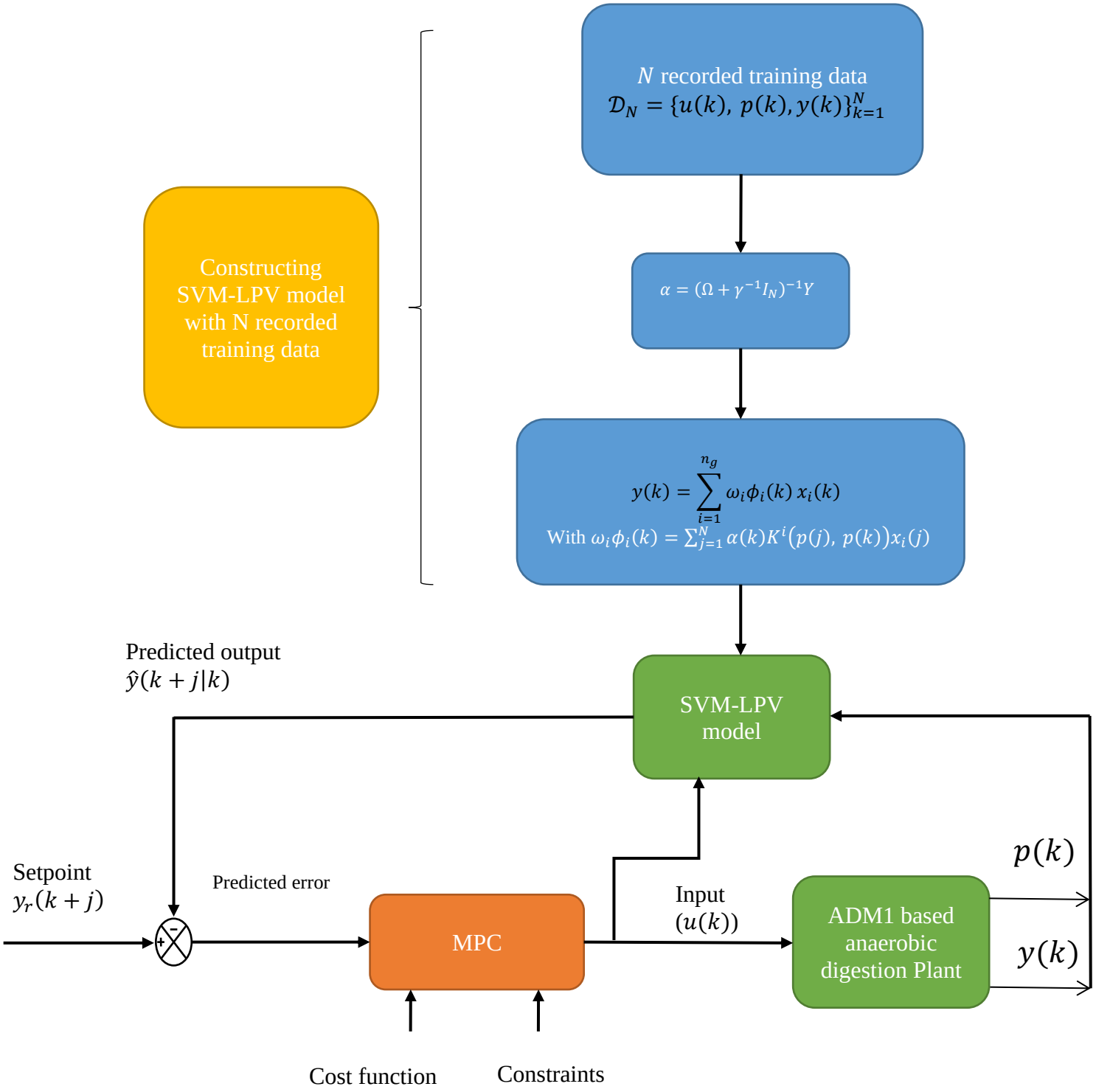


Figure 5.1: The flowchart of the proposed controller

With $\omega_i\phi_i(k) = \sum_{j=1}^{N} \alpha(k)K^i\big(p(j), p(k)\big)x_i(j)$. The values of n_a and n_b in Equations (5.3) (a) and (5.3)(b) are calculated based on the following performance index during the system identification procedure:

$$ISE = \sum_{k=1}^{N} \big(y(k) - \hat{y}(k)\big)^2 \tag{5.20}$$

These values are increased stepwise, and each time the ISE is calculated using Equation (5.20). For methane production rate as the process output, the minimum value of ISE was obtained, when $n_a = 3, n_b = 3$. No significant changes in ISE value were observed when the n_a were increased higher than 1 and n_b greater than 2. Hence, n_a and n_b were set to 1 and 2 respectively to reduce the computational load in the MPC algorithm.

$$x_1(k) = y(k-1), x_2(k) = \Delta u(k)\ x_3(k) = \Delta u(k-1)$$
$$y(k) = \omega_1\ \phi_1\big(p(k)\big)y(k-1) + \omega_2\ \phi_2\big(p(k)\big)\Delta u(k) + \omega_3\ \phi_3\big(p(k)\big)\Delta u(k-1) \tag{5.21}$$

The following flowchart depicts the procedure to find optimal control signals (Figure 5.1). The first step is to identify the LPV-IO model via SVM. To achieve this goal, we first collect N data sets ($\mathcal{D}_N = \{u(k), p(k), y(k)\}_{k=1}^{N}$). Then we specify the necessary parameters related to the model (such as n_a, n_b) and SVM (such as σ and γ). In the next step, we calculate the output vector (Y) and the kernel matrix (Ω) to find the α value. Finally, we find the output value. At the lower level, the MPC controls the system with this model. MPC finds the input signal ($u(k)$) value according to the predicted error and the predefined constraints and objective function. This input is then applied to the real system, and the system output is sent to the SVM-LPV model along with the scheduling signal. This model predicts the output at each sampling time and compares it with the reference value to construct the prediction error.

5.3 Anaerobic digestion process

The AD process comprises four complex stages, namely: hydrolysis, acidogenesis, acetogenesis, and methanogenesis. Complicated biological and physicochemical reactions, which occur in this process, make the dynamic modelling of AD a challenging task. A body of research was conducted to describe the AD process's dynamic model, such as AM2 [34] and AM2HN [35]. Of all proposed methods, ADM1[36] is recognized as the most comprehensive model, where

numerous complex mechanisms of the AD process are regarded. This model has been the standard benchmark of various studies, formed of 19 biochemical reactions, 6 acid-base reactions, 3 liquid gas transfers, and 35 differential equations, describing the dynamic behaviour of components concentration in the process.

For the uptake of substrate and growth of microorganisms, Monod type kinetics are considered, and inhibition effects of inhibitory substances in this process are considered by multiplying inhibition functions (I_j) in Monod kinetics [36]:

$$\rho_j = \frac{k_m S}{K_S + S} X . I_1 . I_2 \dots I_n \tag{5.22}$$

where parameters k_m and K_S are maximum uptake rate and half saturation value. S and X are dynamic variables corresponding to particulate and substrate concentrations,ρ_j denotes the jth biochemical kinetic rate equation, which is the primary nonlinearity behaviour source in this model, and more details are given in [37]. Depending on the inhibition level, the functions I_j can vary between 0 and 1. Material balances of components in the liquid phase (32 state variables) of the AD process, which are categorized as particulate substrate, soluble substrate, and acid-base ions, are presented in Equations (5.23) to (5.24), respectively:

$$\frac{dS_{liq,i}}{dt} = \frac{q_{in}}{V_{liq}} S_{in,i} - \frac{q_{out}}{V_{liq}} S_{liq,i} + \sum_{j=1}^{19} \rho_j \upsilon_{i,j} \tag{5.23}$$

$i = 1, \dots, 12; j = 25,26$

$$\frac{dX_{liq,i}}{dt} = \frac{q_{in}}{V_{liq}} X_{in,i} - \frac{q_{out}}{V_{liq}} X_{liq,i} + \sum_{j=1}^{19} \rho_j \upsilon_{i,j} \tag{5.24}$$

$i = 13,14, \dots, 24$

$$\frac{dS_i}{dt} = \sum_{j=A_1}^{A_6} \rho_j \upsilon_{i,j} \tag{5.25}$$

Where $i = 27, \dots, 32$ denote 6 acid-base components and A_1 *to* A_6 are 6 acid-base reactions.

V_{liq}, q_{in}, q_{out}, $S_{in,i}$, $X_{in,i}$ stand for liquid volume in the reactor, input flow rate, output flow rate, soluble concentration in input feed, and input particulate concentration in input feed, respectively. $\upsilon_{i,j}$ denotes rate coefficient of component i on process j. The following equation

gives the gas phase mass balances for the produced methane, carbon dioxide, and hydrogen concentration:

$$\frac{dS_{gas,i}}{dt} = -\frac{S_{gas,i}q_{gas}}{V_{gas}} + \rho_{T,i}\frac{V_{liq}}{V_{gas}} \tag{5.26}$$

where, S_{gas}, q_{gas}, V_{gas} and $\rho_{T,i}$ denote to ith component concentration, output biogas flow rate, gas volume in the reactor, and gas transfer rate from liquid to the gas phase, respectively. For more details about the kinetic equations, input feed concentration, and ADM1 parameters, the reader may refer to [37]. This deliberate system is single-input single-output (SISO). The input is the feed flow rate (q_{in}) and the output is methane gas flow rate (q_{ch_4}).

From a chemical composition perspective, organic wastes contain a diversity of substances that belong to biodegradable materials: carbohydrates, proteins, and lipids [2]. A change in the composition of these materials is an intrinsic feature of the AD process's organic waste feed, which can influence the AD process's dynamic behaviour and methane productivity. Waste Activated Sludge (WAS), a typical feed of the AD reactors, is considered in this study. The related feed composition and parameters are chosen according to [37]. The feed type contains a high protein value, and the concentration variation in this component is more than carbohydrate and lipid. Therefore, feed protein concentration ($X_{pr,in}$ in ADM1) is selected as scheduling variable, i.e. $p(k) = [X_{pr,in}]$.

5.4 Results and discussion

After introducing the proposed MPC setup and describing the AD process in detail, the simulation results are thoroughly discussed in this section. Firstly, the model is trained with a stepwise scheduling variable, a pseudo-random signal, followed by a test stage to validate the model. Subsequently, the MPC that regulates the actual process with this model is investigated when there is a stepwise change in Feed protein concentration.

5.4.1 System identification

Providing an accurate model plays a pivotal role in the performance of model-based controllers such as MPC. In addition to sufficient accuracy, this model should have minimal computational complexity. In this regard, an LPV model is proposed in this paper, whereby less computation load is required in MPC optimization problems compared to nonlinear MPC. Meanwhile, LPV model coefficients are determined by a nonparametric identification method

without prior knowledge of parameter dependency. Above all, in contrast to nonlinear MPC, where a nonconvex optimization problem or failure in finding the optimal solution is potentially viable, the proposed controller's optimization problem would be straightforward. System identification is performed using the LS-SVM-LPV input-output model structure in Equation 5.19.

Firstly, data required to identify the model from the ADM1-virtual plant simulated with sampling time T_s = 0.1 days (2.4 hours) (which derives from a good rule of thumb that sampling time is 15 times faster than the system output rise time subjected to a step input). In order to obtain rich input-output data to identify the LS-SVM-LPV model, the feed flow rate ($u(k)$) is supposed to be a pseudo-random signal, according to Figure 5.2 (a). Turning to Figure 5.2(b), p (k), i.e., the feed protein concentration ($X_{pr,in}$ in ADM1) experiences step changes in the range of -20% to +20% of nominal value ($20\ Kg(COD)^{-1}$) in a specific interval. The output of the process (methane gas flow rate) is measured at the specified sampling rate (Figure 5.2(c)). RBF kernel functions in Equation (5.14) with hyper-parameters $\sigma_1 = \sigma_2 = \sigma_3 = 0.6$ are considered to estimate the system output . It should be noted that both the magnitude and the duration of the input signal are randomly selected.

Secondly, the extracted data are needed to be trained by LS-SVM to identify the LPV model. Of the 3,000 data obtained for the identification phase, 75% of the data are used to train the model and the remaining (25%) for testing. Also noteworthy is that the test and train data were normalized before identifying the SVM-LPV model. The regularization parameter (in Equation (5.12)) has been tuned to $\gamma = 5000$. Figure 5.3 shows the estimated output of the SVM-LPV model in the train and test phases. The used model shows excellent accuracy in estimating the nonlinear output dynamics of the ADM1 model for both the train and test data groups .The estimated output by LPV-SVM can reliably predict the actual output in all operational areas. The cross-validation of the trained model is investigated in Figure 5.4, where the predicted data are plotted with respect to real data. There is a positive correlation between real and estimated data, and data are more widely scattered around a straight line. From the information supplied in Figures 5.3 and 5.4, it is evident that the output obtained by LS-SVM-LPV can reliably predict the dynamic behaviour of the AD process.

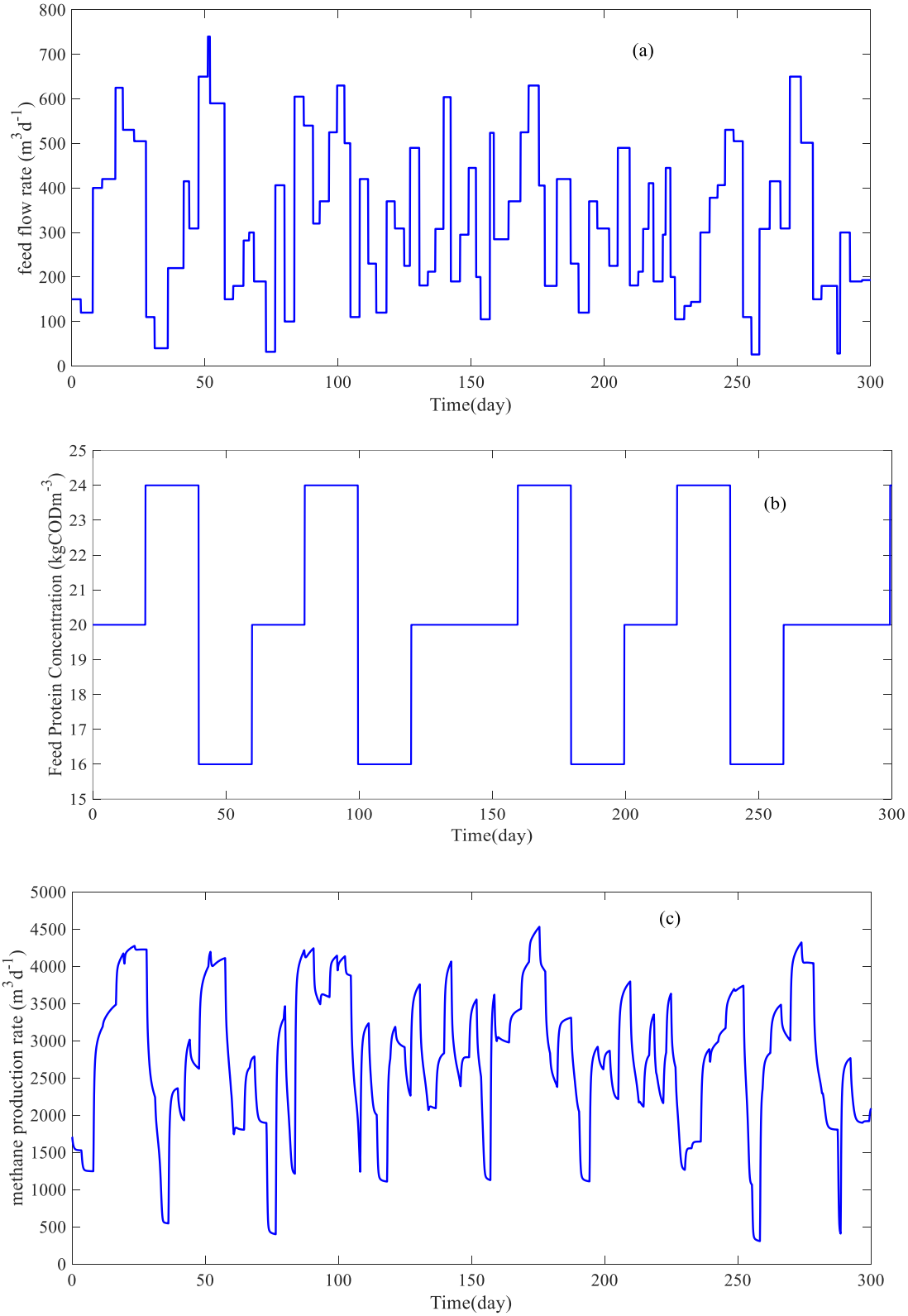


Figure 5.2:(a) The pseudo random input signal used for system identification. (b) The scheduling variable stepwise changes. (c) The output of the AD system

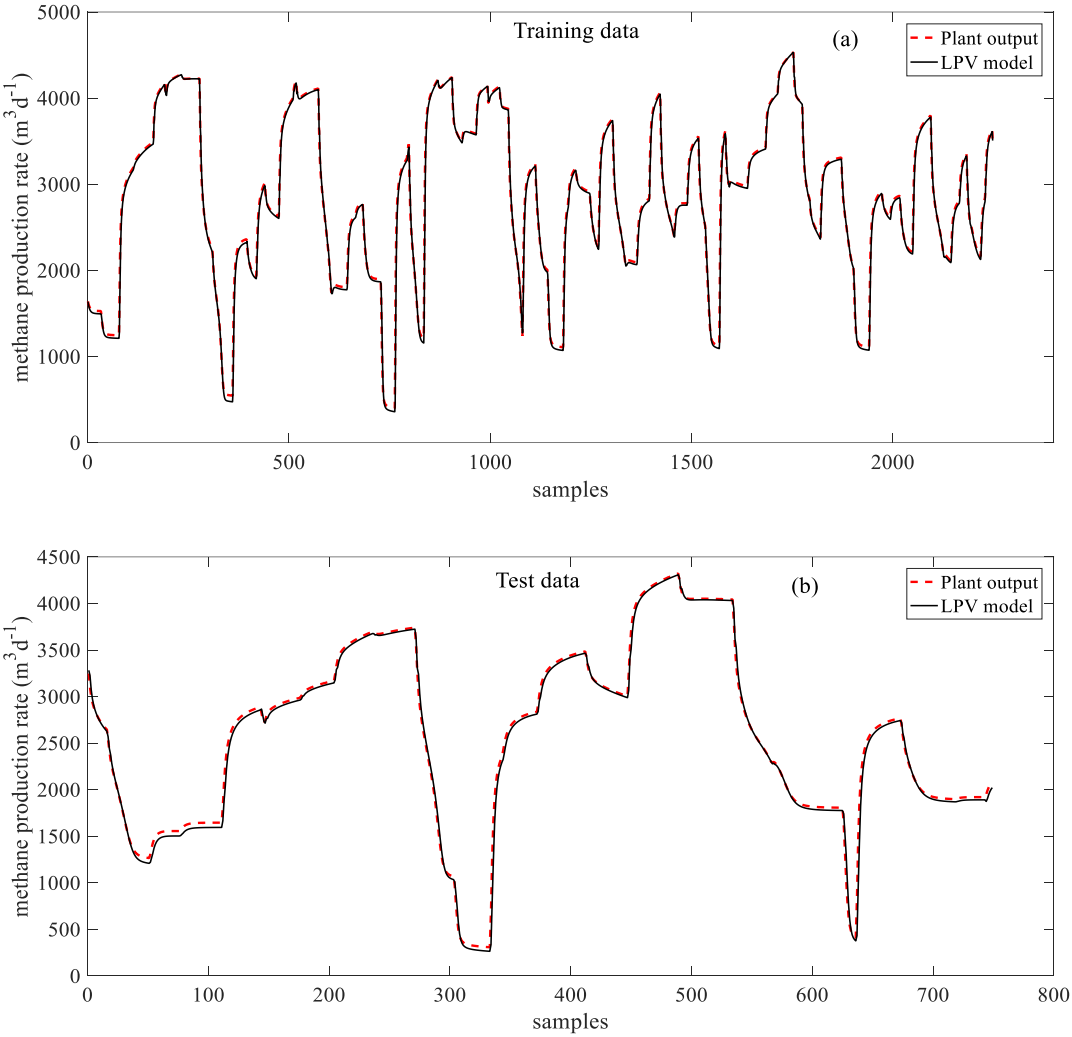
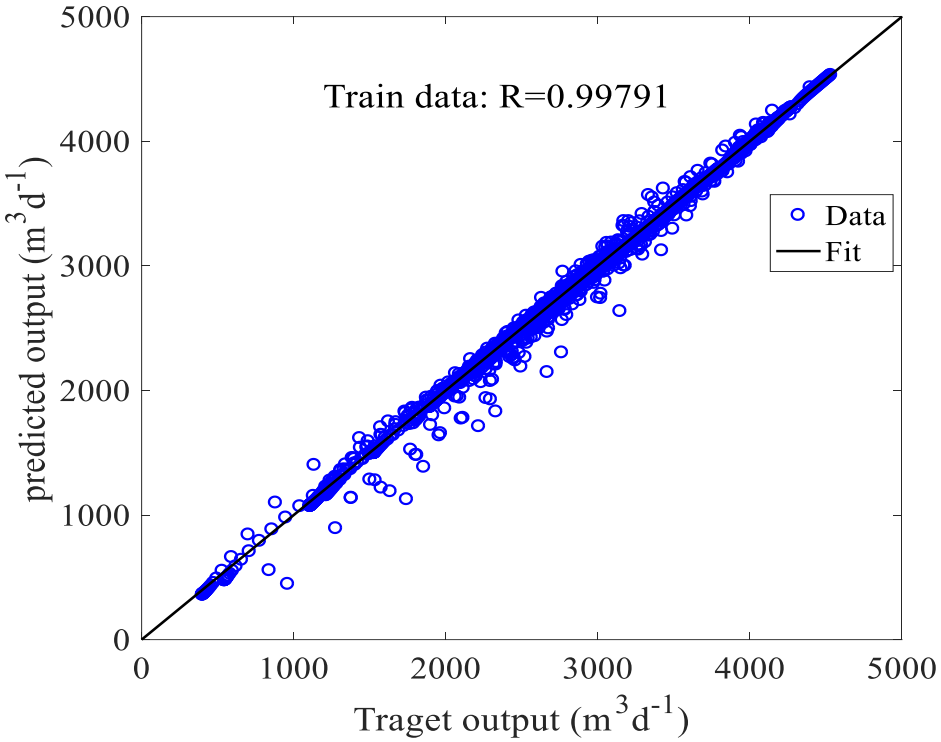


Figure 5.3: (a) The identification results for the training data. (b) The identification results for the test data.



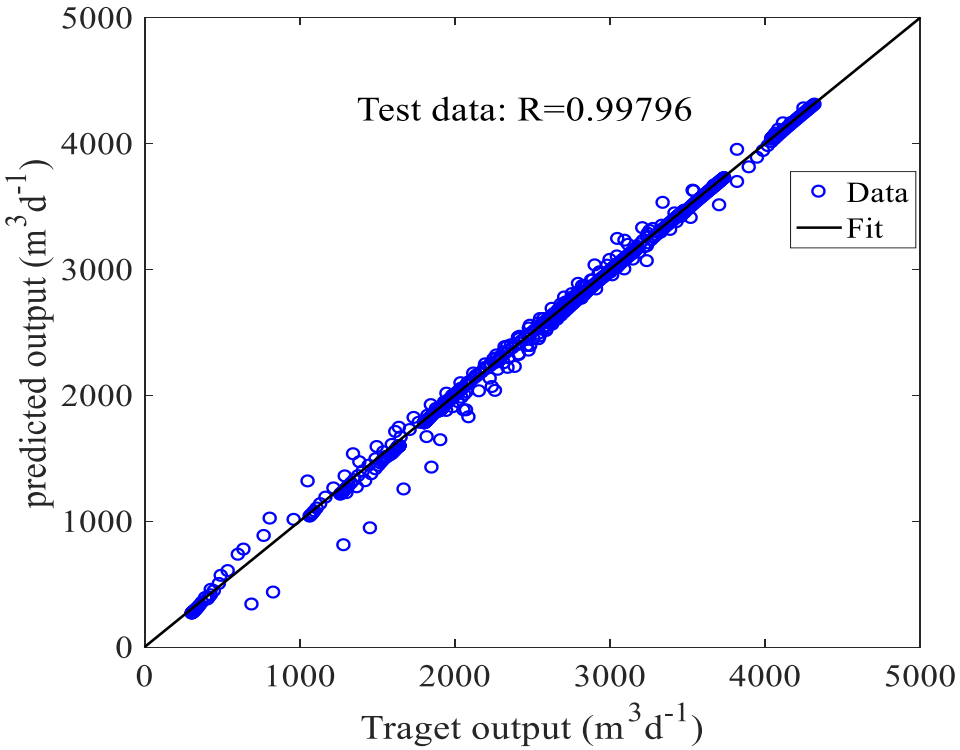


Figure 5.4: linear regression of predicted values relative to measured values for training and testing data set

Two numerical criteria are reported in Table 5.1 to validate the performance of the identified model. The NMSE of both train and test data are reasonably small, and R squares are just under one. Results show that LPV-SVM provides a reliable model for the controller.

Table 5.1:Numerical validation of training and test data set

Identification results	NMSE	R
Train data	0.00042	0.99791
Test data	0.00050	0.99796

The following equations calculate the normalized MSE and correlation coefficient:

$$NMSE = \frac{\|y - \hat{y}\|^2}{\|y\|^2} \tag{5.27}$$

$$R^2 = 1 - \frac{SSE}{SST} \tag{5.28}$$

with

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

where $\|y\|$ denotes to Euclidian norm of y, y is system output, $\bar{y}$ is mean of output, $\hat{y}$ is predicted output, and n is the number of data.

5.4.2 LS-SVM-LPV-MPC closed-loop results

This section examines the proposed controller's functionality in regulation and disturbance (feed concentration) rejection by simulation based on the ADM1 model. The prediction horizon and the control horizon in this system are set equal to 5 and 3. Also, in the MPC objective function (Equation (5.16)), the parameters δ and λ are tuned to 1 and 0.05. To avoid sharp control action changes and to meet the system input constraints, the upper and lower limits of the constraints in Equation (5.17) are selected as follows:

$$0 \leq u \leq 740,$$

$$-316 \leq \Delta u \leq 316$$

The controller's behaviour is evaluated in the nonlinear AD process, modelled by LS-SVM-LPV, in two scenarios in terms of setpoint tracking and disturbance rejection. In both simulations, the scheduling variable is a pulse signal. In contrast to the first simulation's varying setpoint, the setpoint was fixed in the second simulation. The setpoint can be seen in Figure 5.5(b). It is chosen from a productivity point of view, which means in periods t = 0 to 15 (day) and t = 30 to 45 (day) when the scheduling variable increases from the nominal value (20 $Kg(COD)^{-1}$) to 23 $Kg(COD)^{-1}$, the corresponding methane productivity of the AD system is increased, and the setpoint reaches a maximum of $4000\frac{m^3}{d}$. By doing this, although the setpoint has experienced a stepwise change between $2000\frac{m^3}{d}$and $4000\frac{m^3}{d}$, productivity remained constant.

Figure 5.5 describes the first scenario results. Figure 5.5(a) (input) and Figure 5.5(c) (output) show that the proposed controller has superior performance against the simultaneous change in scheduling parameters and the set value. The output can correctly follow the setpoint such that there is a minor overshoot, and the response speed is acceptable. Regarding the input, the constraints are met, and sharp adverse changes have not been experienced. In the second scenario, the controller's performance under operating conditions of sharp change of input protein concentration and constant setpoint was also evaluated, the results of which are shown in Figure 5.6. According to Figure 5.6(b), the concentration of $X_{pr,in}$ in the ADM1 model at t = 0 (day) increased from 20 to 24, then decreased from 24 to 16 at t =7 (day), followed by an increase to 24 at t = 14 (day). As shown in Figures 5.6(a) and 5.6(c), while there was a dramatic change in feed concentration, the control remained stable and closely followed the setpoint variation. The controller removed the disturbance quickly without allowing it to ruin the output with high amplitudes.

In addition to Figures 5.5 and 5.6, four numerical measures are reported in Table 5.2 for two scenarios, including normalized MSE (NMSE), rise time (T_r), setting time (T_s), and disturbance rejection time (T_d). Because of three times changes of the setpoint and/or the scheduling variable, the average values of T_r, T_s, and T_d are indicated here. The MSE is less than 0.0041, which is practically acceptable. Based on the rise time, the process is about three times faster in the second scenario than in the first. The setting times of the process in the first and second simulations were 1.7167 seconds and 1.55 seconds. It took 0.9033 seconds for the controller to

remove the disturbance. In a nutshell, the proposed controller is remarkably fast with an acceptable setpoint tracking error and meets the process input's physical limitations.

Table 5.2: Performance of the proposed controller in terms of normalized MSE, rise time, settling time, and disturbance rejection time

Closed-loop results	NMSE	$\boldsymbol{T_r}$ (day)	$\boldsymbol{T_s}$(day)	$\boldsymbol{T_d}$ (day)
First scenario	0.0041	1.3500	1.7167	---
Second Scenario	0.0034	0.4500	1.55	0.9033

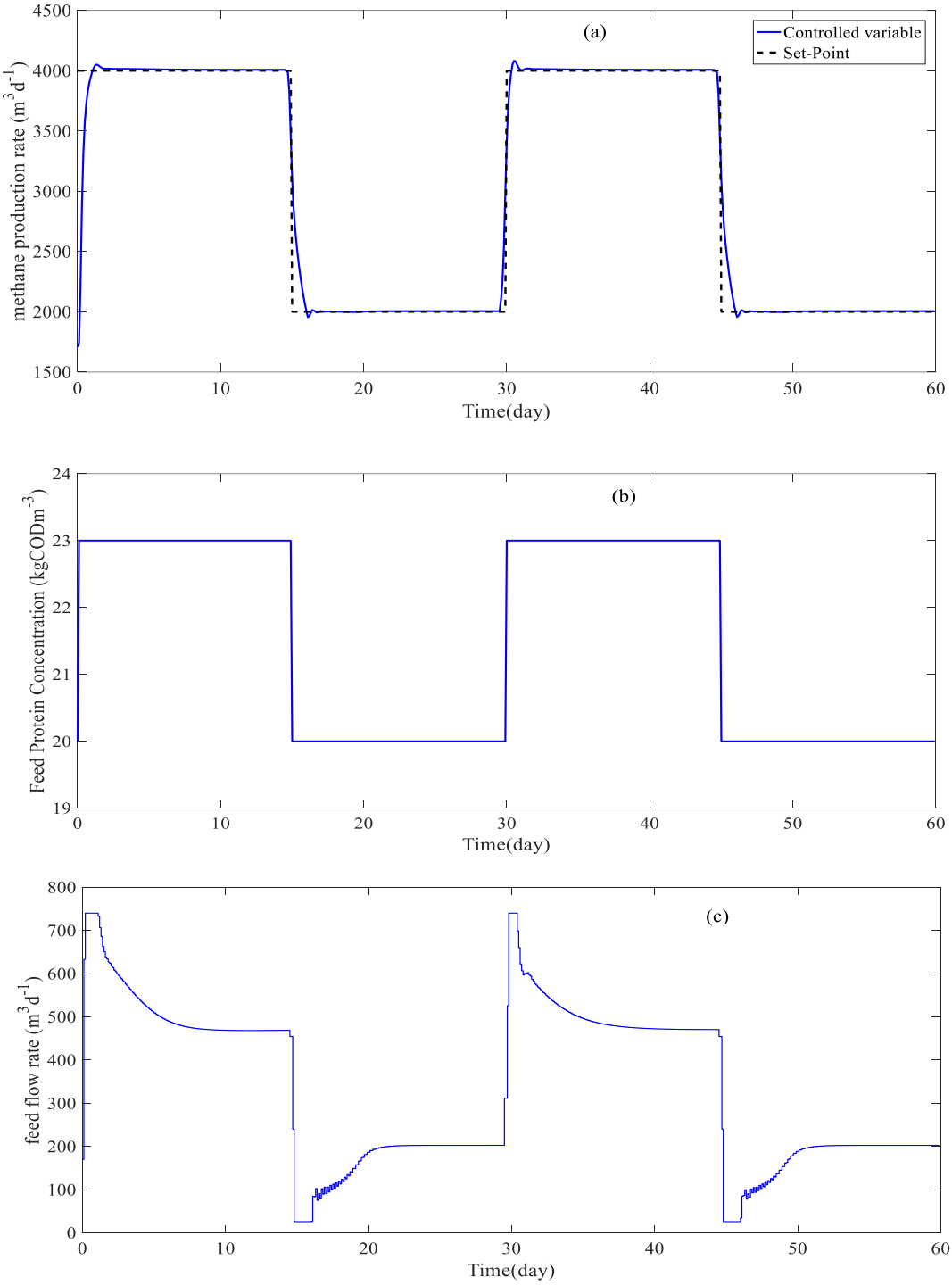


Figure 5.5: (a) output. (b) scheduling variable (c). input

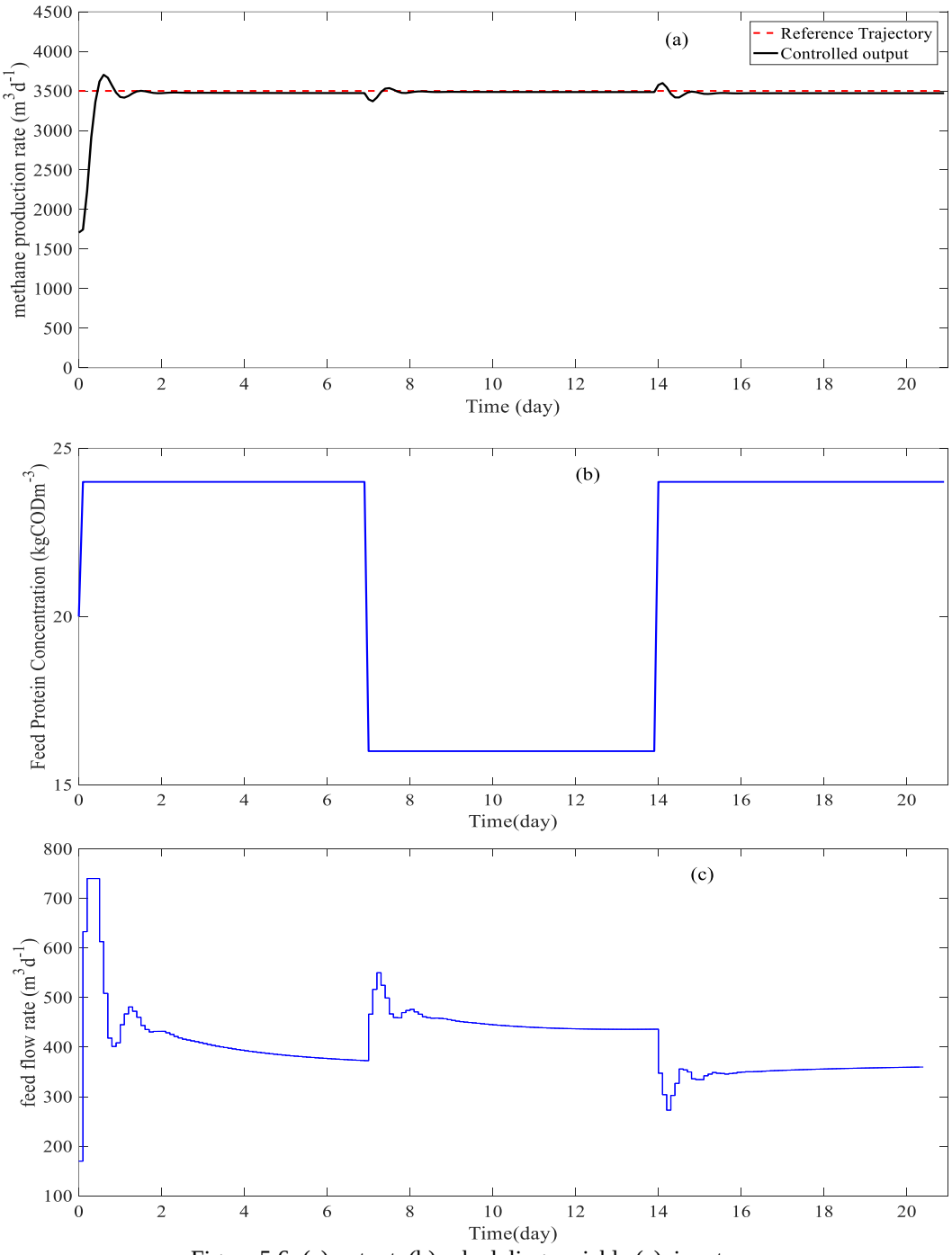


Figure 5.6: (a) output. (b) scheduling variable (c). input

5.5 Conclusion

This study proposes an MPC improved by LPV-IO for regulating the methane production rate in an anaerobic digestion system. A nonparametric LPV model is constructed with an LS-SVM strategy to predict the methane production rate based on input-output data before employing the model predictive controller structure for output regulation. The identification results showed that the LS-SVM-LPV could correctly predict the output of the system. Regarding train and test data, NMSE were 0.00042 and 0.00050, and R were 0.99791 and 0.99796. Furthermore, the controller results indicated that the designed MPC could regulate the methane production in two scenarios, whether when both the stepwise setpoint and scheduling variable changed or when the setpoint was unchanged, and the scheduling variable had a sharp variation. The proposed controller proved to be sturdy when the process faced a disturbance. The controller was fast enough with a setting time and rise time of fewer than two seconds. The tracking error was negligible, with MSE of 0.0041 and 0.0034 in the first and second tests

CHAPTER 6

AN IMPROVED INTERPOLATED MODEL PREDICTIVE CONTROL BASED ON RECURRENT NEURAL NETWORKS FOR A NONHOLONOMIC DIFFERENTIAL-DRIVE MOBILE ROBOT WITH QUASI-LPV REPRESENTATION: COMPUTATIONAL COMPLEXITY AND CONSERVATISM

Chapter 6 presents the study for Objective 4. The work presented in this chapter is included in the following paper:

Hadian, M., Zhang, W., & Etesami, D. (2022). An Improved Interpolated Model Predictive Control based on Recurrent Neural Networks for a nonholonomic differential-drive mobile robot with Quasi-LPV Representation: computational complexity and conservatism. International Journal of Robust and Nonlinear Control, under review.

Abstract

This paper presents an improved Model Predictive Control (MPC) for path tracking of a nonholonomic mobile robot with a differential drive. Nonlinear dynamics and nonholonomic constraints make the optimization problem of MPC for the robot challenging. Nonlinear dynamics of the robots are expressed by a Linear Parameter Varying (LPV), and a Recurrent Neural Network (RNN) solves the constrained optimization problem to find the optimal velocities. Moreover, an interpolation-based approach has been developed to improve the region of attraction. The algorithm's stability also has been guaranteed in the presence of bounded disturbances by adding free control moves to the control law. The controller efficiency has been evaluated in two scenarios in a hospital setting. The simulation results illustrate that the proposed method performs better than nonlinear MPC and standard LPV-based MPC in terms of computational cost, disturbance rejection, and region of attraction.

6.1 Introduction

The COVID-19 pandemic highlighted the overarching value of medical robots in hospitals and quarantine centers [9-11]. Mobile medical robots are employed for various medical functions, including surgery, cleaning, disinfection, and drug delivery [9, 17, 19, 27]. They are believed to improve productivity and patient satisfaction and reduce healthcare costs [6]. Meanwhile, they do not become sick and can work around the clock reliably, quickly, and accurately.

Path planning and path tracking are fundamental challenges in mobile medical robotics [33]. Path planning algorithms analyze and identify an obstacle-free path for a mobile robot to navigate the environment [34]. Path tracking control is designed for a mobile robot to track the reference path precisely. This paper focused on the path tracking control of Wheeled Mobile Robots (WMR). The tracking problem of WMR is complicated due to nonholonomic constraints, nonlinear dynamics, and uncertainties. Many control techniques have been proposed in the literature so far, such as PID controller [43], feedback linearized control [44], sliding mode control [45], resilient control [46], intelligent control [98], vision-based control [139], and Model Predictive Control (MPC) [47].

The first control difficulty is the real-world limitations, also known as constraints, which must be considered in controller design since it lessens the achievable paths. These constraints include speed response, mobility constraints, computational cost, field-of-view constraints, maneuverability, and control stability issues [48]. These constraints can be holonomic or nonholonomic. In contrast to holonomic constraints, nonholonomic constraints are not integrable to provide constraints in terms of positional variables, so the transformation matrix should be in terms of velocities, not positions. According to Brockett's theorem, no continuous time-invariant feedback of state variables can be found that asymptotically stabilizes the nonholonomic system around the equilibrium point [48, 49]. Model Predictive Control (MPC) is the most widely accepted solution for meeting constraints.

The second drawback of WMR is its nonlinear model. Several studies have been conducted to develop a nonlinear model predictive controller (NMPC). [32, 47, 59, 60]. NMPC, however, suffers from excessive computational load. The most effective remedy that can be taken to deal with nonlinearities is Linear Parameter Varying (LPV) [150]. Scheduling variables are used to describe nonlinear systems in a linear form. By portraying nonlinear systems with a linear form with varying parameters, we can bridge the gap between linear and nonlinear/time-varying

systems[61, 66]. MPC based on an LPV framework efficiently compromises linear MPC and NMPC [61-65].

Several studies have considered LPV-based MPC for WMR [292]. A new path tracking approach with a gain-scheduled control law has been developed for a nonholonomic mobile robot exposed to kinematic disturbances [253]. Kinematic error in the tracking problem is modelled as an LPV with bounded disturbance. The reference tracking is established for the environment with and without obstacles. The system stability has been confirmed using a Lyapunov function in structured and unstructured environments. Similarly, a control technique for omnidirectional mobile vehicles with a high robustness level is offered [254]. It is proven that the nonlinear controller is tractable by converting the uncertain nonlinear into LPV form. Compared to those nonlinear control approaches, the suggested algorithm avoids the complicated design procedure and minimizes control constraints' impacts in fast dynamic environments. A sampled-data MPC tracking control approach is provided for mobile robots, represented as continuous-time LPV systems with input saturation constraints [255]. The optimization problem is represented in an LMI format. A new Lyapunov function was generated to prove the stability in a long delay. In [250], it has been shown that LPV-based MPC had a lower online computation load than NMPC. They proposed a cascade control to address the trajectory tracking problem for autonomous vehicles. The external loop solves position control using a novel LPV-MPC approach, and the internal loop dynamically controls the vehicle using an LPV-Linear Quadratic Regulator technique designed as an LMI problem. Despite computational load reduction, the robustness of the controller when there are external disturbances has not been proven. An extension of the controller can be found in [293], where a tube-based MPC ensures robust stability. They also used the Zonotope theory to reduce the computational cost. In both studies, the proposed controller is designed with the largest possible terminal region, but the region of attraction has not been discussed. Similarly, [294] developed a cascade framework with an inner kinematic controller and an outer dynamic controller. They examined the importance of the kinematic and dynamic model and the effect of the bandwidth on closed-loop stability. Some researchers also developed tubed-based MPC with LPV form for WMR to suppress the disturbance effect while reducing the computational complexity. In [295] a primal-dual neural network approach solved the optimization problem, and in [68] some parts of controller design were offline to reduce the online computation load, but it can be conservative. A tube-based MPC for a lane-keeping system with high speed and highly curvy roads has been developed in

[296], where Semi Definite Programming (SQP) and LMI computed the control law. The designed controller is computationally efficient, yet highly conservative. The tracking control problem of WMR subjected to bounded kinematic disturbance is studied using LPV-based MPC in [253] while considering obstacle avoidance. Despite all efforts, three questions remain unanswered: how to use Input-Output (IO) LPV model, how to increase the region of attraction, and how to show robustness when the robot faces environmental disturbances.

To the best of our knowledge, all of the research on LPV-MPC for WMR is constructed with a state-space (SS) form, and no one has explored using an LPV-MPC with an IO model for WMR. The LPV with SS form is computationally intractable and does not accurately describe uncertainty [271, 273]. In addition, they use the highly improbable assumption that state variables are measurable. Some research has used an observer to estimate the states [297]. For example, observers are developed to estimate friction force, linear velocity, and angular velocity [292] and to estimate position, orientation, and wheel slip [298]. However, an observer can complicate the design and lead to closed-loop instability [276]. On these grounds, an LPV-IO-based MPC is studied in this paper.

Regarding the region of attraction, an interpolation-based MPC (IMPC) was supplemented. In an IMPC, control signals are generated by interpolating between several previously calculated control gains based on a kinematic model, resulting in a less conservative controller [167]. Some researchers employed an ellipsoidal invariant set to meet stability and recursive feasibility [63, 148]. Later research aims to broaden the region of attraction (ROA) by substituting a polyhedral invariant set for the ellipsoidal invariant set and introducing the Maximal Admissible Set (MAS). In our previous paper [149], an efficient IMPC with an LPV-IO model was developed by adding a terminal region and cost. The results showed that the proposed paper is noticeably more tractable than the traditional ones in the computation load and conservatism degree.

Above all, regarding disturbance rejection, some LPV-MPC studies on WMR relied on kinematic models, which are not always realistic when dynamic disturbances occur [299]. Others employed a cascade architecture with kinematic and dynamic models [250]. However, the cascade controller has certain drawbacks, including additional measurement, additional controller tuning, and a three to four times quicker inner loop. Given these disadvantages, working in a fast-paced environment with unexpected disturbances might be problematic. Therefore, we included free control moves to handle unknown disturbances. The free control moves are calculated online from

the MPC optimization problem using the WMR dynamic model. Simply put, our offline controller uses a kinematic model, and our online controller uses a dynamic controller. If the robot is subjected to external disturbances, the second controller is in charge of transferring the state variables into the terminal region, and the first controller is responsible for exponentially convergent the robot to the equilibrium point there. The offline controller is fixed and pre-calculated, whereas the online controller will be determined by solving a min-max optimization problem. An RNN will solve the online problem to deal with the nonlinear constrained min-max optimization problem. RNN is proven to have global convergence and low computational load [218, 223].

The main contributions of the paper are summarized as follows:

1. The controller is designed based on the LPV-IO model, so observer or state measurement is unnecessary. The controller relies on input-output measurements.
2. An interpolation-based strategy is implemented to find a set of offline controllers and increase the region of attraction, resulting in a less conservative design.
3. An RNN efficiently solves the quadratic optimization of MPC to find an online control law that removes bounded disturbance and ensures stability.
4. The control signal consists of two components: free control move, which is estimated from a dynamic model, and control gain, which is determined from a kinematic model. Asymptotic stability is considered in the first component, and disturbance robustness is guaranteed in the second.
5. The stability of the robot is proven when there are external disturbances

The rest of this paper is organized as follows. The kinematic and dynamic models and LPV forms are explained in section 2. Then we look at the robust IMPC for WMR and convert the quadratic optimization problem into RNN. Section 4 shows the simulation and results. The conclusions will be the final section.

6.2 Modelling

This section describes the kinematic and dynamic model of a typical WMR.

6.2.1 Kinematic model

The nonholonomic robot's kinematic model is defined as:

$$\begin{cases} \dot{x}(t) = v(t)\cos(\theta(t)) \\ \dot{y}(t) = v\,(\mathrm{t})\,\sin(\theta(t)) \\ \dot{\theta}(t) = \omega(t) \end{cases} \tag{6.1}$$

Where x, y, and θ are the position and orientation of the robot, v is linear velocity, and ω is the angular velocity. A trajectory tracking problem assumes that the pose of the robot (x, y, θ) should converge to the pose of the virtual reference robot (x_r, y_r, θ_r). The virtual reference path is defined as:

$$\begin{cases} \dot{x}_r(t) = v_r(t)\cos(\theta_r(t)) \\ \dot{y}_r(t) = v_r\,(\mathrm{t})\,\sin(\theta_r(t)) \\ \dot{\theta}_r(t) = \omega_r(t) \end{cases} \tag{6.2}$$

The error vector $[x_e \;\; y_e \;\; \theta_e]$is the difference between real-time pose and reference pose $[x - x_r \;\; y - y_r \;\; \theta - \theta_r]$. The errors, however, are stated in terms of the global inertial frame and should be transformed into the robot's local frame as follows:

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) & 0 \\ -\sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x - x_r \\ y - y_r \\ \theta - \theta_r \end{bmatrix} \tag{6.3}$$

The rear wheels have nonholonomic constraints, which must be considered before generating the error model:

$$\dot{x}\sin(\theta) = \dot{y}\cos(\theta) \tag{6.4}$$

Then, the kinematic error model has been constructed from 1 to 4:

$$\begin{aligned} \dot{x}_e(t) &= \omega(t)y_e(t) + v_r(t)\cos\big(\theta_e(t)\big) - v(t) \\ \dot{y}_e(t) &= -\omega(t)x_e(t) + v_r\,(\mathrm{t})\,\sin(\theta_e(t)) \\ \dot{\theta}_e(t) &= \omega_r(t) - \omega(t) \end{aligned} \tag{6.5}$$

6.2.2 Dynamic model

The Lagrangian dynamic equation of the WMR is defined as [300]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = E\tau - A^T(q)\lambda \tag{6.6}$$

where $q = [x\; y\; \theta]$ is the state vector, $\lambda = [\lambda_1\; \lambda_2\; \lambda_3]^T$ is the vector of constraint forces, $E = [0_{2\times3}\; I_{2\times2}]^T$ is the input matrix, $\tau = [\tau_r\; \tau_l]^T$ are the torques of the wheels (inputs), C is the Coriolis and centripetal forces, and M is the inertia matrix.

$$C(q,\dot{q}) = \begin{bmatrix} 0 & 0 & md\dot{\theta}cos(\theta) & 0 & 0 \\ 0 & 0 & md\dot{\theta}sin(\theta) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M(q) = \begin{bmatrix} m & 0 & mdsin(\theta) & 0 & 0 \\ 0 & m & -mdcos(\theta) & 0 & 0 \\ mdsin(\theta) & -mdcos(\theta) & I & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix}$$

$$m = m_c + 2m_w$$

$$I = I_c + 2m_w(d^2 + b^2) + 2I_m + m_c d^2$$

where m_w is the mass of a wheel and rotor together, m_c is the mass of the robot, I_c and I_w are the inertias of the rotor and wheels, I_w is the inertia related to the plane of the wheel.

6.2.3 LPV models

For the kinematic model, the scheduling variables are chosen to be $[\omega\ v_r\ \theta_e]$ and the kinematic-LPV model can be written as:

$$\dot{e}(t) = A(p_k(t))e(t) + Bu(t) - Bu_r(t) \tag{6.7}$$

$$A = \begin{bmatrix} 0 & \omega & 0 \\ -\omega & 0 & v_r * \dfrac{\sin(\theta_e)}{\theta_e} \\ 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}$$

Where e is the state error $[x_e\ \ y_e\ \ \theta_e]^T$, p_k is the scheduling variable, u is the control input $[v\ w]^T$ and u_r is the reference input $[v_r \cos(\theta_e)\ \ \omega_r]^T$.

For the dynamic model, the scheduling variables are chosen to be $[\omega_r\ v_r]$, and dynamic model can be written as:

$$\ddot{q} = M(q)^{-1}(E\tau - A^T(q)\lambda - C(q,\dot{q})\dot{q}) \tag{6.8}$$

It then can be converted into an LPV form as:

$$\dot{x}_d = \begin{bmatrix} 1 & 0 \\ 0 & -M^{-1}C \end{bmatrix} x_d + \begin{bmatrix} 0 \\ M^{-1}E \end{bmatrix} u - A^T\lambda \tag{6.9}$$

where $x_d = [q\ \ \dot{q}]^T$ and $u = \tau$.

6.3 Controller design

In this section, we first show how an LPV-SS model can be converted into an LPV-IO. We then define the cost function and develop the kinematic and dynamic controllers. Finally, we expressed the optimization problem in terms of RNN. Assume the following discrete-time MIMO linear parameter-varying (LPV) transfer function:

$$y(k) = -\sum_{i=1}^{n_a} a_i\big(p(k)\big)\, y(k-i) + \sum_{j=1}^{n_b} b_j\big(p(k)\big)\, u(k-j) \tag{6.10}$$

Subjected to the following constraints

$$u(k) \in U \equiv \{u \in \mathbb{R}^{n_u} |\, |u(k)| \leq u_{max}\}$$
$$\Delta u(k) \in V \equiv \{\Delta u \in \mathbb{R}^{n_u} |\, |\Delta u(k)| \leq \Delta u_{max}\}$$
$$y(k) \in Y \equiv \{y \in \mathbb{R}^{n_y} |\, |y(k)| \leq y_{max}\}$$
$$p(k) \in P \equiv \{p \in \mathbb{R}^{n_p} |\, |p(k)| \leq p_{max}\}$$

where $y(k)$ are outputs, $u(k)$ are inputs, $p(k)$ are scheduling variables, q^{-i} is a backshift operator, n_a is numerator order, n_b is dominator order, $n_u$is the number of inputs, n_y is the number of outputs, and $u_{max}, \Delta u_{max}$, and y_{max} are boundaries. The state-space representation of the dynamic model given by the equation is:

$$x(k+1) = \underbrace{\begin{bmatrix} -a_1 & \dots & -a_{n_a-1} & -a_{n_a} & b_0+b_1 & \dots & b_{n_b-1} & b_{n_b} \\ I_{n_y} & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & & I_{n_y} & 0 & 0 & & 0 & 0 \\ 0 & \vdots & 0 & 0 & I_{n_u} & \vdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & I_{n_u} & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & 0 & I_{n_u} & 0 \end{bmatrix}}_{A(p(k))} x(k) + \underbrace{\begin{bmatrix} b_0 \\ 0 \\ \vdots \\ 0 \\ I_{n_y} \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{B(p(k))} u(k) \tag{6.11}$$

where $x(k) = [y(k-1) \dots y(k-n_a)\ \ u(k-1) \dots u(k-n_b)]^T$. For the sake of simplicity, $p(k)$ is removed from all coefficients a_i and b_i. We considered that $[A(k)\ \ B(k)] \in \Omega = Co[[A_1\ B_1], \cdots, [A_m\ B_m]]$, where Ω is the polytope, Co is the convex hull, and $[A_j\ B_j]$ are vertices. The state variable is a combination of previous inputs and outputs by doing this. The controller uses the LPV-IO model to predict outputs and the LPV-SS model for stability provision. Also noteworthy is that the stability of the LPV-SS controller grants the equivalent LPV-IO controller stability [281].

6.3.1 The prediction model

Finding $y(k+j)$ sentences are required to predict the error $e(k+j)$. The future outputs can be calculated from a step response or by solving the Diophantine equation. Given the step response of the LPV system (6.10), the future values of outputs over the horizon will be:

$$y(k,p(k)) = \sum_{i=1}^{\infty} g_i(p(k))\Delta u(k-i,p(k)) \tag{6.12}$$

$$y(k+j,p(k+j)) = \sum_{i=1}^{\infty} g_i(p(k+j))\Delta u(k+j-i,p(k)) + d(k+j,p(k))$$

where g_i denotes step response, and $d(k+j)$ is a disturbance, which can be derived from $y(k+j) - r(k+j)$. For a faster prediction, disturbances are assumed to be constant, which can be estimated by $y(k) - r(k)$. The scheduling variable $p(k)$ is removed from the next equations for the reasons of simplification. The Equation (6.12) can then be rewritten:

$$y(k+j) = (\sum_{i=1}^{k} g_i\Delta u(k+j-i) + \sum_{i=k+1}^{\infty} g_i\Delta u(k+j-i)) + (y(k) - r(k)) \tag{6.13}$$

Substituting $y(k) = \sum_{i=1}^{\infty} g_i\Delta u(k-i)$ in Equation (6.13) yields:

$$\begin{aligned} y(k+j) &= \sum_{i=1}^{k} g_i\Delta u(k+j-i) + \sum_{i=1}^{\infty}(g_{k+i} - g_i)\Delta u(k-i) - r(k) \\ &= \sum_{i=1}^{k} g_i\Delta u(k+j-i) + f(k+j) \end{aligned} \tag{6.14}$$

Meanwhile, $f(k+j)$ can be abridged on the assumption that the system is asymptotically stable and after N sampling period g_{k+i} equals g_i.

$$f(k+j) = \sum_{i=1}^{N}(g_{k+i} - g_i)\Delta u(k-i) - r(k) \tag{6.15}$$

The prediction relationship of the output vector is

$$Y(p) = G(p)U(p) + F(p) \tag{6.16}$$

where

$$G = \begin{bmatrix} g_1 & 0 & \dots & 0 \\ g_2 & g_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_2 \\ g_N & g_{N-1} & \cdots & g_1 \end{bmatrix} \tag{6.17}$$

$$Y = \begin{bmatrix} y(k+1) \\ \vdots \\ y(k+N) \end{bmatrix}, U = \begin{bmatrix} u(k) \\ \vdots \\ u(k+N-1) \end{bmatrix}, F = \begin{bmatrix} f(k+1) \\ \vdots \\ f(k+N) \end{bmatrix}$$

6.3.2 Cost function

After finding the predicted outputs, the cost function should be defined to find optimal control signals. The controller aims to minimization of the cost function (J):

$$J = \underbrace{\sum_{i=0}^{N-1} e(k+i)^T Q e(k+i) + \Delta u(k+i-1)^T R \Delta u(k+i-1)}_{l(e,u)} + \Psi(k+N) \tag{6.18}$$

The first term $l(e,u)$, is called stage cost, being in charge of closed-loop stability, where Q and R are positive definite state and input weighting matrixes, and e is the error, which is the difference between measured outputs (y) and reference (r). The term $\Psi(k+N)$is the terminal cost, which can be defined as $x(k+N)^T Q_p x(k+N)$.

According to [281], a state feedback controller can represent the system's stability with the LPV-IO form. The control law ($u(k)$) is comprised of a fixed state feedback K and a free control move (c), in which K is computed offline, and c is obtained from the minimization of (6.18):

$$u(k) = -Kx(k) + c(k) \tag{6.19}$$

The state-feedback gain accounts for maintaining the final state variable $x(k+N)$ in the terminal region while meeting constraints and keeping the controller as less conservative as possible. In fact, a constant state feedback controller confirms the stability when there is no disturbance. Otherwise, free control moves (c) will be determined online in a min-max problem to guarantee the Input State practical stability (ISpS). Thus, a part of the controller will be calculated offline for conditions that there is not any disturbance, and a part of the controller will be set online to deal with existing disturbances or violations of $x(k+N) \in X_f$.

6.3.3 Kinematic controller

After output prediction and defining cost function, the asymptotical stabilizing state feedback control can be described:

$$\begin{aligned} u &= -Kx \\ x(k) &\in S_I \\ x(k+N) &\in S_T \end{aligned} \tag{6.20}$$

S_I characterizes the initial feasibility region, in which $S_I = \{x \in R^n | x \in X, -Kx \in U\}$, and S_T denotes the terminal feasibility region, $S_T = \left\{x \in R^n | x(k+N)^T Q_p x(k+N) < \frac{1}{\gamma}\right\}$. As previously discussed, this paper aims to expand S_I and S_T to enjoy a progressive controller. The first step is to enlarge $S_I$through a general interpolation MPC. Secondly, the largest possible S_T that meet the constraints will be chosen by minimizing γ. The control law of IMPC can be described as:

$$\begin{aligned} x = \sum_{i=1}^{n} \lambda_i x_i \,, \sum_{i=1}^{n} \lambda_i = 1,\ \lambda_i \geq 0 \\ u = -\sum_{i=1}^{n} k_i \lambda_i x_i \end{aligned} \tag{6.21}$$

where n is the number of predefined feedback gains k_i that stem from Ackermann's formula as follows:

$$K_i = [0 \quad \cdots \quad 0 \quad 1] Q_{c,i}^{-1} \alpha_i(A) \tag{6.22}$$

where $\alpha_i(A)$ is the characteristic polynomial, and $Q_{c,i}$ is the controllability matrix.

$$\begin{aligned} \alpha_i(A) = \det\big(SI - (A_i - B_i K_i)\big) = \det\,(SI - M_i) \\ Q_{c,i} = [B \quad AB \quad A^2B \quad \ldots \quad A^{n_x - 1}B\ ] \end{aligned} \tag{6.23}$$

Taking the state-space model, as stated in Equations (6.11), the proposed general IMPC method is recursively feasible [149]. According to Equation (6.23), the state trajectories are $\{-Kx, KM_i x, KM_i{}^2 x, \ldots\}$. With this in mind, the enhanced feasible invariant set can be determined by Maximal Admissible Set (MAS) as:

$$\begin{aligned} s_i = \{x | \mathrm{H}_i x \leq \alpha\} \\ \mathrm{S_I} = \mathrm{Co}\{\mathrm{s}_1, \ldots, \mathrm{s}_n\} \end{aligned} \tag{6.24}$$

where $\mathrm{H}_i = \left[M_i, M_i{}^2, \ldots, M_i{}^n\right]^T, and\ \alpha = [1, \ldots, 1]^T$

The system, described in Equation (6.10), is asymptotically stable under the interpolated control law (6.21) if there is $Q_p = Q_p{}^T > 0$ that implies Equation (6.25)[149].

$$M_i{}^T Q_{p_i} M_i - Q_{p_i} + Q_i + K_i{}^T R_i K_i \leq 0, \qquad i = 1, \dots n \tag{6.25}$$

After showing the proposed method's recursive feasibility and asymptotic stability, the terminal region must be determined. While region S_T is desired to be as large as possible to reduce conservatism, increasing this area might cause violating the constraints. As a result, the minimum value of γ that satisfies constraints

(6.10) can be calculated by an optimization problem. It is proven that γ can be derived from the following optimization problem [276]:

$$\min_{\gamma} \gamma \tag{6.26}$$

$$\frac{1}{\gamma} A_T P^{-1} A_T{}^T \leq B_T B_T{}^T$$

where $A_T = [-I_n \;\; K \; I_n - K]^T, B_T = [x_{max} - x_s \;\; \Delta u_{max} \; x_{max} - x_s \;\; \Delta u_{max}]^T$, and x_s is the desired state.

6.3.4 Dynamic controller

The free control moves will be determined online in case there is a disturbance or the condition $x(k) \in X_f$ does not meet. The online controller steers the states towards X_f, where offline control asymptotically stabilizes the system. Two Theorems are defined here to demonstrate the controller validity[150]. The first showed that the system remains ISpS when there is a bounded disturbance. The second showed that the system is recursively stable. Because of additive disturbances, a min-max (worst-case) optimization problem is defined to cope with uncertainties. The optimization problem (6.18) can be rewritten considering the bounded disturbance:

$$\min_{c(k)} \max_{d(k) \in D} V_N(x(k), u(k), p(k), K, c(k)) \tag{6.27}$$

$$y(k+j) = -\left(\sum_{i=1}^{n_a} A_i \, q^{-i}\right) y(k+j) + \left(\sum_{j=1}^{n_b} B_i \, q^{-i}\right) u(k+j) + d(k), \text{ j=1,2,…,N-1}$$

$$u(k+j) = -Kx(k+j) + c(k+j)$$

$$u(k+j) \in U$$

$$\Delta u(k+j) \in V$$

$$y(k+j) \in Y$$

$$d(k+j) \in D$$

$$x(k+N) \in X_f$$

The optimal solution to the problem (6.27), subjected to the system (6.10), is the sequence $[c^*(k) \; c^*(k+1) \dots c^*(k+N-1)]$ corresponding to $[u^*(k) \; u^*(k+1) \dots u^*(k+N-1)]$. [150]

provided two theorems for showing this optimization problem's stability and recursive feasibility. The designed controller has two components. The first is determined offline using the kinematic mode, which includes a control gain K and terminal region X_f. When $x \in X_f$, the offline controller asymptotically stabilizes the system. Otherwise, an online controller verifies the ISpS using a dynamic model when there is uncertainty. The online optimization problem is shown to be feasible. A dynamic neural network is constructed in the next section to solve the online optimization problem.

6.3.5 RNN

In this study, RNN optimizes a real-time QP problem enjoying parallel computation. To begin with, the original optimization problem is required to transform into a standard form [150].

$$X(k+j) = G\big(p(k+j-1)\big)X(k+j-1) + F\big(p(k+j-1)\big)U(k+j-1) + D(k+j-1), \qquad j = 1, \dots, N$$

$$G\big(p(k)\big) = [A\big(p(k)\big)\, A\big(p(k)\big)^2 \dots\, A\big(p(k)\big)^N]^T$$

$$F\big(p(k)\big) = \begin{bmatrix} B\big(p(k)\big) \\ A\big(p(k)\big)B\big(p(k)\big) + B\big(p(k)\big) \\ \vdots \\ A\big(p(k)\big)^{N-1}B\big(p(k)\big) + \cdots + A\big(p(k)\big)B\big(p(k)\big) + B\big(p(k)\big) \end{bmatrix} \tag{6.28}$$

And the optimization problem (6.27) can be written as follows:

$$\min_U X^T \bar{Q} X + U^T R U = \min_U (GX + FU + D)^T \bar{Q} (GX + FU + D) + U^T R U$$

$$\min_U U^T * ((FU)^T \bar{Q} (FU) + R) * U + (GX + FU)^T \bar{Q} (GX + FU) \tag{6.29}$$

where

$$\bar{Q} = \begin{bmatrix} Q & 0 \\ 0 & Q_p \end{bmatrix}$$

According to Equations (6.28) and (6.29), the standard form can be expressed as:

$$\min_v \frac{1}{2} v^T H v + b^T v$$

$$\text{Subject to } Tv \le q \tag{6.30}$$

where

$$H = 2 * ((FU)^T \bar{Q} (FU) + R)$$

$$b = 2 * (GX + FU)^T \bar{Q} (GX + FU)$$

$$T = \begin{bmatrix} I_{n*n} \\ -I_{n*n} \\ I_{n*n} \\ -I_{n*n} \\ G \\ -G \end{bmatrix}, q = \begin{bmatrix} u_{max} \\ -u_{max} \\ \Delta u_{max} + u(k-1) \\ \Delta u_{max} - u(k-1) \\ X_{max} - F - d \\ -X_{max} + F + d \end{bmatrix}$$

In [222, 228], a simplified form of a dual neural network is described to ensure a lower computational burden by defining the dual form of the optimization problem (6.30) as:

- State equation:

$$\frac{d\omega}{dt} = \lambda(-TH^{-1}T^T\omega + M(TH^{-1}T^T\omega - TH^{-1}c - \omega) + TH^{-1}c) \quad (6.31)$$

Where ω is the state variable of the network, $\lambda > 0$ adjusts the convergence rate of RNN, and M is a piecewise linear function as:

$$M(z) = \begin{cases} q_{min} & z < q_{min} \\ z & q_{min} < z < q_{max} \\ q_{max} & z > q_{max} \end{cases} \quad (6.32)$$

- Output equation:

$$v = H^{-1}T^T\omega - H^{-1}c \quad (6.33)$$

In the dual form of the optimization problem, constraints are added to the main cost function as a penalty term. This means that if $Tv < q$, the cost function decreases by a factor of α; otherwise, for $Tv > q$, the cost function is penalized. The global convergence of the proposed network is verified in [228].

6.4 Results and discussion

In this section, the performance of the proposed controller has been validated in two scenarios: 1) trajectory tracking and 2) disturbance rejection. The developed controller is compared with two previous controllers for WMR in the literature (NMPC and LPV-MPC [250])in a hospital environment with static obstacles. Firstly, we examine the closed-loop stability of these controllers in terms of response speed, convergence, and robustness when facing disturbances. Secondly, the tractability of controllers (online computation time and region of attraction) is examined. A waypoint path planning approach has been utilized in all simulations to generate a route between two points. The controller parameters are listed in Table 6.1.

Table 6.1: The designed controller parameters

Parameters	Values
Q	3*diag(1 1 2)
R	1 *diag(8 3)
Sampling time	0.1
$\boldsymbol{u_{max}}$	$v_{max} = [-12\ 12]\ mm/s$, $\omega_{max} = [-10\ 10] deg/s$
$\Delta \boldsymbol{u_{max}}$	[-0.5 0.5]
N	15
$\boldsymbol{\theta_e}$	[-0.1 0.1]

6.4.1 Scenario 1: trajectory tracking

This scenario has no disturbance, and the robot must follow a random path between certain fixed locations. Choosing a shorter path that avoids abrupt changes in velocities and passes through the designated locations is preferable. The hospital's map has several rooms and hallways, and the black areas show where the walls are. Figure 6.1 demonstrates that all techniques successfully pass through all points. While there are minor discrepancies in the pathways in certain places, particularly in the third room, none of the techniques exhibit excessive fluctuations or divergence. Figure 6.2 demonstrates that all three approaches adhered to the maximum speed limit; Nevertheless, it should be noted that the NMPC and LPV-MPC methods displayed substantial speed change rates in certain regions, causing instances where the robot came to a halt in several simulations.

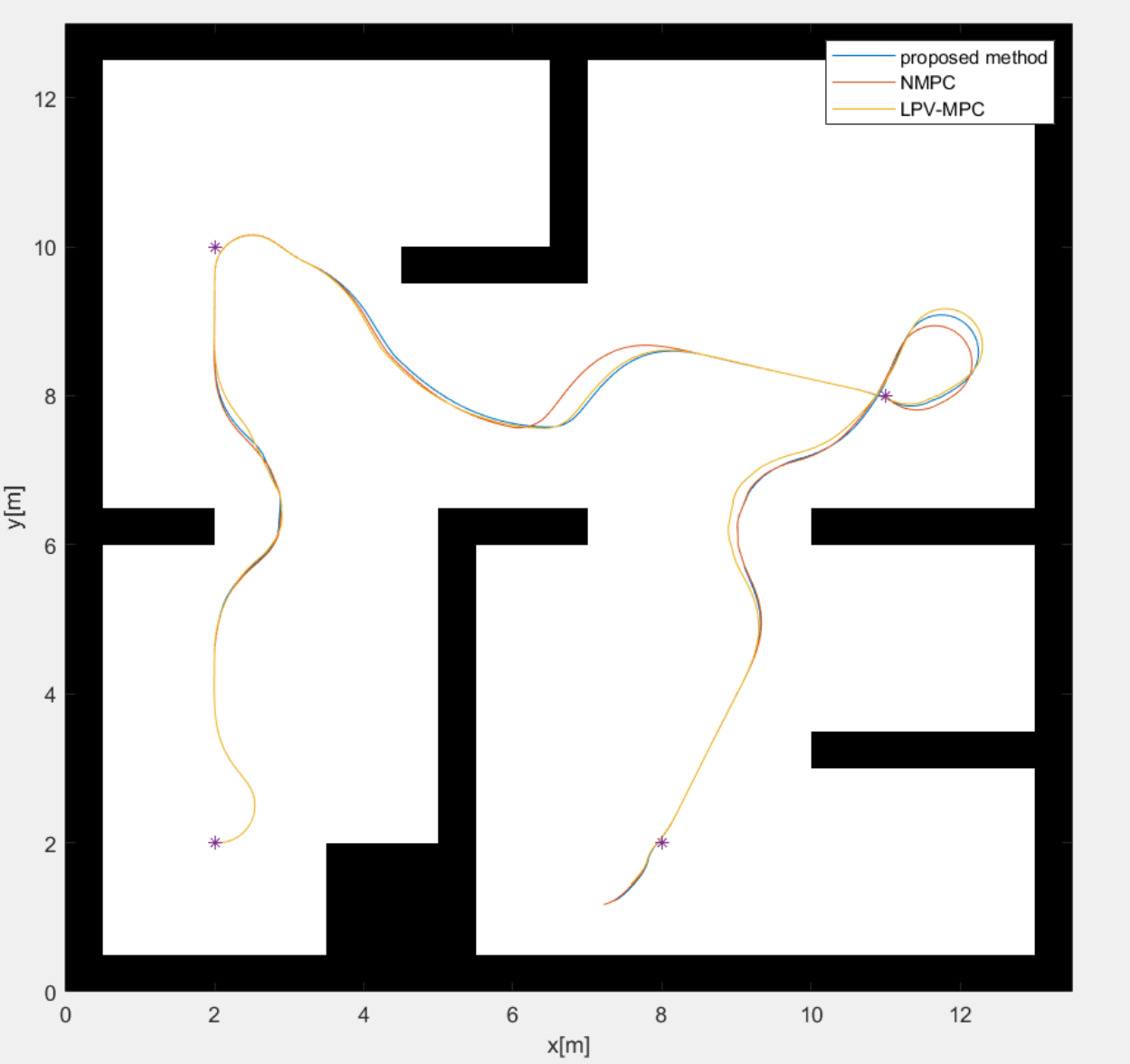


Figure 6.1: Simulation results of proposed control, NMPC, and LPV-MPC in the first scenario

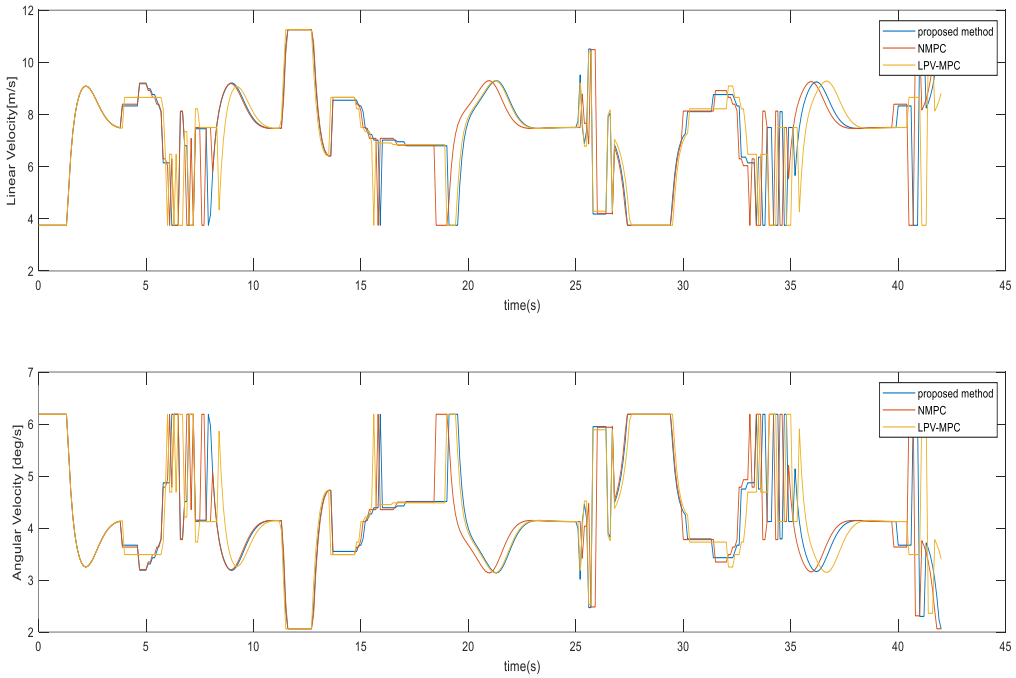


Figure 6.2: Linear and Angular velocities of the proposed controller, NMPC, and LPV-MPC in the first scenario

6.4.2 Scenario 2: disturbance rejection

A disturbance signal from 0 to 2 in 20 seconds and from 2 to 0 in 25 seconds has been applied to both speed inputs to evaluate the controllers' robustness. According to Figure 6.3, the proposed method is the only one that can retain its stability in the presence of disturbance while also passing through all spots with adequate speed and precision. The LPV-MPC technique is terminated at the third point because the robot has collided with a wall or has become unstable. Although the NMPC technique traverses all the spots, it is not particularly precise, and the path has significant shifts and sometimes crashes with the walls. At the end of the route, this strategy also achieves marginal stability. Figure 6.4 depicts two approaches (LPV-MPC and NMPC) in which the robot pauses after a certain period of time and does not continue to stay on the path. Furthermore, the speed variations in these two approaches are abrupt and harsh. The proposed technique includes logical adjustments and takes into account the speed constraints.

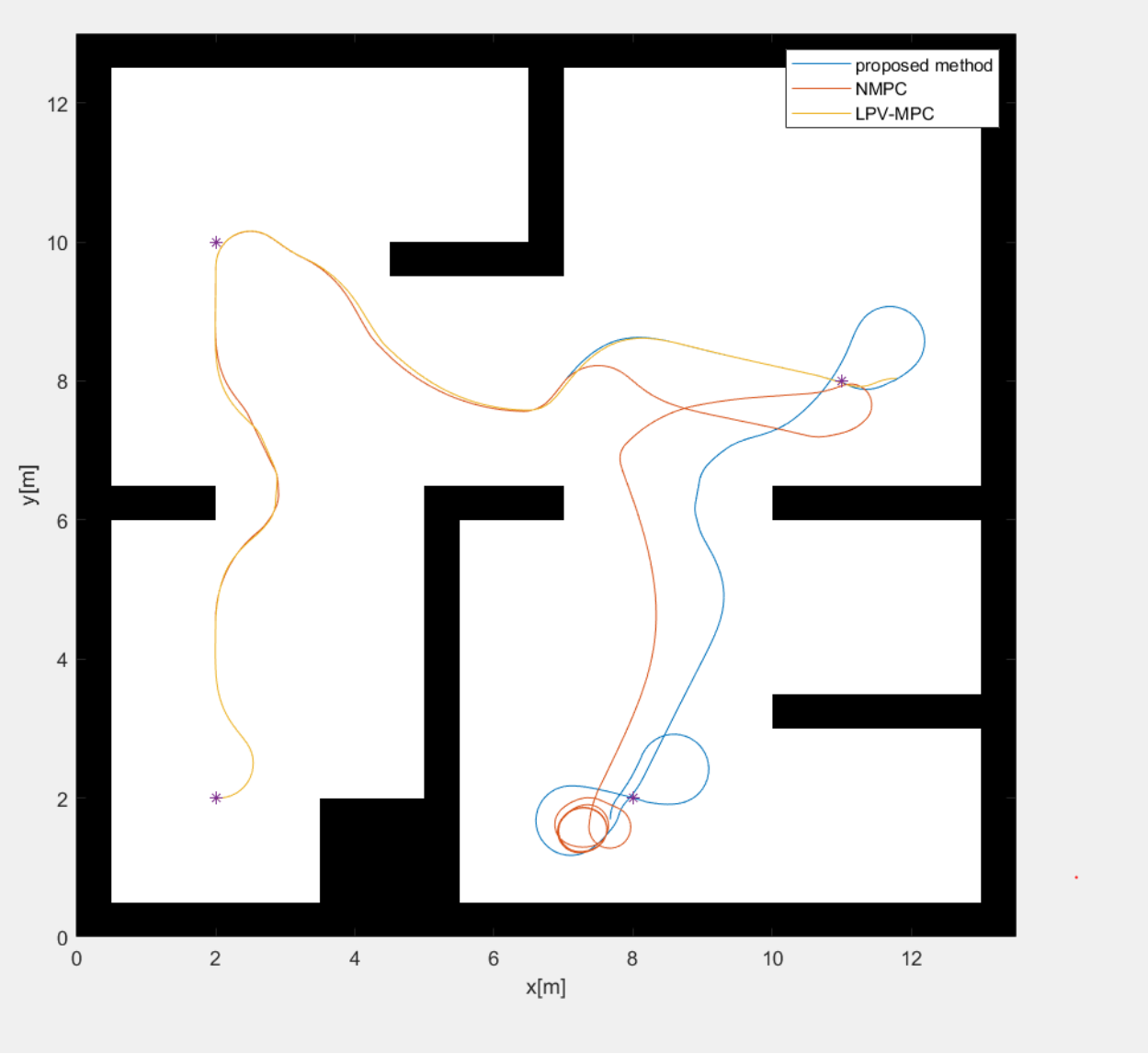


Figure 6.3: Simulation results of proposed control, NMPC, and LPV-MPC in the second scenario

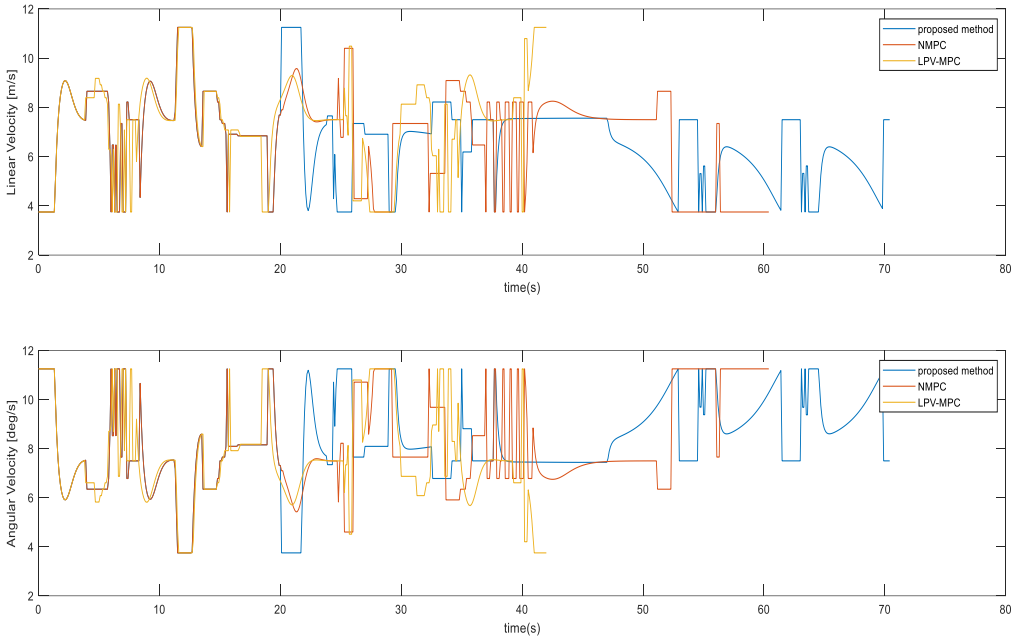


Figure 6.4: Linear and Angular velocities of the proposed controller, NMPC, and LPV-MPC in the second scenario

6.4.3 Tractability

Improved tractability is one of the primary strengths of the proposed approach, which is discussed from two viewpoints in this section: 1) online computational load; 2) region of attraction. Figure 6.5 shows the region of attraction for three controllers. It is evident that the controller created in this research clearly has the largest surface and has successfully increased the potential starting place for the robot. NMPC was ranked second because the robot model was not simplified compared to the LPV-MPC. Because of the conversion of a nonlinear model into an LPV model with a fixed controller, the method of [250] has the smallest region of attraction. A preliminary conclusion is that interpolating predetermined control gains can expand the initial feasible region (about two times bigger than NMPC and five times bigger than LPV-MPC). In addition, the proposed controller is less conservative because it finds the maximal final region using equation 6.26. This indicates that, in comparison to previous controllers, the proposed controller not only maximizes the terminal region but also has a significantly larger region of attraction.

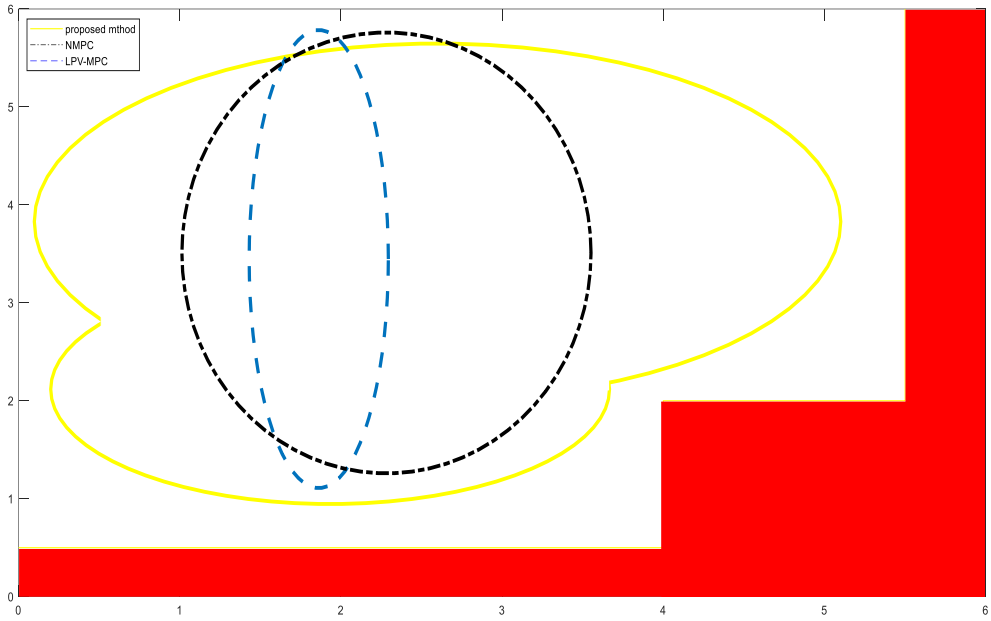


Figure 6.5: Region of attraction for the proposed controller, NMPC, and LPV-MPC

We also studied the average elapsed time per iteration in Table 6.2. The NMPC is the slowest controller since solving a nonlinear optimization problem is time-consuming and challenging. Although the proposed solution must solve a min-max optimization problem to find the optimal control laws, it is four times faster than LPV-MPC. The reason behind this rapid response is that some calculation has been done offline, and the online problem has been solved with RNN, which is far more efficient than the GUROBI method [250].

Table 6.2:The online elapsed time per iteration for three studied controllers

Controller	Elapsed time (ms)
Proposed approach	0.0071
NMPC	3.8
LPV-MPC	0.029

6.5 Conclusions

A nonlinear model predictive controller has been proposed in this paper, where an LPV framework describes the nonlinear nonholonomic model of the robot. The online optimization

problem is solved with a neural network strategy to ameliorate computational complexity. An interpolation technique was also added to increase the region of attraction. The controller is compared with an NMPC and LPV-MPC regarding controller performance and traceability. All controllers perform well in a hospital environment where there are no disturbances. When a bounded disturbance was applied to the velocities of the robot, the NMPC and LPV-MPC became unstable or followed the reference inefficiently. In contrast, our method maintained the robot's stability without harsh velocities changes. Above all, our controller is less conservative than NMPC and LPV-MPC, i.e., it has a bigger region of attraction with a lower online computational load.

CHAPTER 7

CONCLUSIONS AND FUTURE WORKS

In this thesis, we first discussed the various medical robots, followed by the controllers utilized for wheeled robots. The combination of MPC and LPV for controlling wheeled robots in the hospital environment is suitable due to the robot's nonlinear dynamics and nonholonomic constraints. However, MPC-LPV techniques in the literature had some fundamental issues: 1) heavy computation load; 2) low attraction region; 3) robustness. We theoretically proved that MPC-LPV could be improved to address these problems. We then tested the proposed controller for a robot in a hospital setting.

In Chapter 3, we interpolated a series of MPC controllers for LPV systems with input-output representation. The findings indicated that the suggested strategy significantly reduces the computational burden, considerably increases the initial feasible zone, and retains the terminal region as big as possible, resulting in a less conservative controller. The suggested technique is shown to closely follow the reference and remain stable in the presence of disturbance even when the process is more intricate, large-scale, and involves various scheduling variables,

Chapter 4 examines an RMPC with an LPV-IO model, and the real-time optimization problem is solved using an RNN algorithm. With suitable rising time, settling time, MSE, and response amplitude, the suggested technique successfully succeeded in both setpoint tracking and disturbance rejection. Meanwhile, four optimization techniques are compared to handle the suggested controller's online optimization problem. RNN outperformed GA, SVD, and SQP in speed, MSE, and cost value.

Chapter 5 uses an LS-SVM technique to build a nonparametric LPV model before utilizing the MPC. The identification results demonstrated that the LS-SVM-LPV could properly forecast the system output. When the process encountered a disturbance, the proposed controller proved robust. With a settling time and a rising time of fewer than two seconds, the controller was quick enough.

Chapter 6 suggests an improved MPC-LPV for a nonlinear nonholonomic robot model. The online optimization problem is handled using a neural network technique to reduce computational complexity. An interpolation approach was also used to expand the region of attraction. The controller function well in a hospital setting, with or without disturbances. Our solution maintained the robot's stability while avoiding abrupt velocity changes. Above all, our controller is less conservative than existing controllers, with a larger zone of attraction and a lower online computing burden.

Path planning is one of the topics that might be addressed in the controller's next design. Furthermore, because physicians, nurses, and personnel are always moving about in the hospital setting, multiple moving obstacles can be included in the simulations. It is also possible to think about the issue of adding an arm robot to a wheeled robot; in this scenario, both the wheeled robot and the arm robot need to be controlled. The suggested controller's application to leader-follower robot navigation is an intriguing subject. Regarding the controller, deep learning techniques can improve the estimation of scheduling variables over the prediction horizon to find a more reliable controller.

REFERENCES

[1] D. Djurdjanovic, L. Mears, F. Akhavan Niaki, A. Ul Haq, and L. Li, "Process and operations control in modern manufacturing," in *International Manufacturing Science and Engineering Conference*, 2017, vol. 50749, p. V003T04A057: American Society of Mechanical Engineers.

[2] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.

[3] A. Kutnetsov and D. Clarke, "Advances in Model Based Predictive Control. Chapter: Application of constrained GPC for improving performance of controlled plants," ed: Oxford University Press, 1994.

[4] K. Zhou, J. C. Doyle, and K. Glover, *Robust and optimal control*. Prentice hall New Jersey, 1996.

[5] Z. Alkhoury, M. Petreczky, and G. Mercère, "Structural properties of affine LPV to LFR transformation: Minimality, input-output behavior and identifiability," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 3781-3786: IEEE.

[6] L. Landers, C. F. Timoney, and R. A. Felder, "Medical Mobile Robotics: An Industry Update," *JALA: Journal of the Association for Laboratory Automation*, vol. 5, no. 3, pp. 26-29, 2000.

[7] R. Bloss, "Mobile hospital robots cure numerous logistic needs," *Industrial Robot: An International Journal*, 2011.

[8] K. Niechwiadowicz and Z. Khan, "Robot based logistics system for hospitals-survey," in *IDT Workshop on interesting results in computer science and engineering*, 2008.

[9] R. Kimmig, R. H. Verheijen, and M. Rudnicki, "Robot assisted surgery during the COVID-19 pandemic, especially for gynecological cancer: a statement of the Society of European Robotic Gynaecological Surgery (SERGS)," *Journal of Gynecologic Oncology*, vol. 31, no. 3, 2020.

[10] H. A. Hadi, "Line Follower Robot Arduino (using robot to control Patient bed who was infected with Covid-19 Virus)," in *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2020, pp. 1-3: IEEE.

[11] S.-Q. Li *et al.*, "Clinical application of an intelligent oropharyngeal swab robot: implication for the COVID-19 pandemic," *European Respiratory Journal*, vol. 56, no. 2, 2020.

[12] G.-Z. Yang *et al.*, "Combating COVID-19—The role of robotics in managing public health and infectious diseases," ed: Science Robotics, 2020.

[13] Z. Zeng, P.-J. Chen, and A. A. Lew, "From high-touch to high-tech: COVID-19 drives robotics adoption," *Tourism Geographies*, pp. 1-11, 2020.

[14] F. A. Matsen III, J. L. Garbini, J. A. Sidles, D. C. Baumgarten, and B. S. Pratt, "Robot-aided system for surgery," ed: Google Patents, 1990.

[15] T. Noritsugu and T. Tanaka, "Application of rubber artificial muscle manipulator as a rehabilitation robot," *IEEE/ASME Transactions On Mechatronics*, vol. 2, no. 4, pp. 259-267, 1997.

[16] D. J. Williams, H. I. Krebs, and N. Hogan, "A robot for wrist rehabilitation," in *2001 Conference Proceedings of the 23rd Annual International Conference of*

the IEEE Engineering in Medicine and Biology Society, 2001, vol. 2, pp. 1336-1339: IEEE.

[17] L. Tang, G. Liu, M. Yang, F. Li, F. Ye, and C. Li, "Joint design and torque feedback experiment of rehabilitation robot," *Advances in Mechanical Engineering*, vol. 12, no. 5, p. 1687814020924498, 2020.

[18] E. Palma and C. Bufarini, "Robot-assisted preparation of oncology drugs: The role of nurses," *International Journal of Pharmaceutics*, vol. 439, no. 1-2, pp. 286-288, 2012.

[19] A. Abutaleb, J. Alsabhani, S. Alkinani, S. Alkaydi, S. Alghamdi, and A. Bensenouci, "Design and Implementation of a Nurse Robot," in *International Conference on Industrial Engineering and Operations Management*, 2020.

[20] J. Brief *et al.*, "Robot assisted dental implantology," *International Poster Journal*, vol. 4, no. 1, p. 109, 2002.

[21] T. C. van Riet, K. T. C. J. Sem, J.-P. T. Ho, R. Spijker, J. Kober, and J. de Lange, "Robot technology in dentistry, part two of a systematic review: an overview of initiatives," *Dental Materials*, vol. 37, no. 8, pp. 1227-1236, 2021.

[22] A. W. Yaniv and S. J. Knoer, "Implementation of an iv-compounding robot in a hospital-based cancer center pharmacy," *American journal of health-system pharmacy*, vol. 70, no. 22, pp. 2030-2037, 2013.

[23] D. L. Johanson *et al.*, "Smiling and use of first-name by a healthcare receptionist robot: Effects on user perceptions, attitudes, and behaviours," *Paladyn, Journal of Behavioral Robotics*, vol. 11, no. 1, pp. 40-51, 2020.

[24] S. Koceski and N. Koceska, "Evaluation of an assistive telepresence robot for elderly healthcare," *Journal of medical systems*, vol. 40, no. 5, p. 121, 2016.

[25] D. J. Ricks and M. B. Colton, "Trends and considerations in robot-assisted autism therapy," in *2010 IEEE international conference on robotics and automation*, 2010, pp. 4354-4359: IEEE.

[26] M. A. Saleh, F. A. Hanapiah, and H. Hashim, "Robot applications for autism: A comprehensive review," *Disability and Rehabilitation: Assistive Technology*, vol. 16, no. 6, pp. 580-602, 2021.

[27] M. Guettari, I. Gharbi, and S. Hamza, "UVC disinfection robot," *Environmental Science and Pollution Research*, pp. 1-6, 2020.

[28] L. Yan, T. Wang, D. Liu, J. Peng, Z. Jiao, and C.-Y. Chen, "Capsule robot for obesity treatment with wireless powering and communication," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 2, pp. 1125-1133, 2014.

[29] M. Hu, X. Ge, X. Chen, W. Mao, X. Qian, and W.-E. Yuan, "Micro/nanorobot: A promising targeted drug delivery system," *Pharmaceutics*, vol. 12, no. 7, p. 665, 2020.

[30] E. Broadbent, R. Stafford, and B. MacDonald, "Acceptance of healthcare robots for the older population: Review and future directions," *International journal of social robotics*, vol. 1, no. 4, p. 319, 2009.

[31] R. H. Taylor, "A perspective on medical robotics," *Proceedings of the IEEE*, vol. 94, no. 9, pp. 1652-1664, 2006.

[32] Y. Sun, L. Guan, Z. Chang, C. Li, and Y. Gao, "Design of a low-cost indoor navigation system for food delivery robot based on multi-sensor information fusion," *Sensors*, vol. 19, no. 22, p. 4980, 2019.

[33] O. Martins, A. Adekunle, S. Adejuyigbe, O. Adeyemi, and M. Arowolo, "Wheeled Mobile Robot Path Planning and Path Tracking Controller Algorithms: A Review," *Journal of Engineering Science & Technology Review*, vol. 13, no. 3, 2020.

[34] M. A. Abbas, R. Milman, and J. M. Eklund, "Obstacle avoidance in real time with nonlinear model predictive control of autonomous vehicles," *Canadian journal of electrical and computer engineering*, vol. 40, no. 1, pp. 12-22, 2017.

[35] B. Patle, A. Pandey, D. Parhi, and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," *Defence Technology*, vol. 15, no. 4, pp. 582-606, 2019.

[36] H.-y. Zhang, W.-m. Lin, and A.-x. Chen, "Path planning for the mobile robot: A review," *Symmetry*, vol. 10, no. 10, p. 450, 2018.

[37] H. Adeli, M. Tabrizi, A. Mazloomian, E. Hajipour, and M. Jahed, "Path planning for mobile robots using iterative artificial potential field method," *International Journal of Computer Science Issues (IJCSI)*, vol. 8, no. 4, p. 28, 2011.

[38] F. Lingelbach, "Path planning using probabilistic cell decomposition," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, 2004, vol. 1, pp. 467-472: IEEE.

[39] X. Wang, X. Luo, B. Han, Y. Chen, G. Liang, and K. Zheng, "Collision-free path planning method for robots based on an improved rapidly-exploring random tree algorithm," *Applied Sciences*, vol. 10, no. 4, p. 1381, 2020.

[40] Y. Pan, Y. Yang, and W. Li, "A deep learning trained by genetic algorithm to improve the efficiency of path planning for data collection with multi-UAV," *Ieee Access*, vol. 9, pp. 7994-8005, 2021.

[41] B. Song, Z. Wang, and L. Zou, "An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve," *Applied Soft Computing*, vol. 100, p. 106960, 2021.

[42] H. Kundra, W. Khan, M. Malik, K. P. Rane, R. Neware, and V. Jain, "Quantum-Inspired Firefly Algorithm integrated with cuckoo search for optimal path planning," *International Journal of Modern Physics C*, vol. 33, no. 02, p. 2250018, 2022.

[43] J. Ye, "Adaptive control of nonlinear PID-based analog neural networks for a nonholonomic mobile robot," *Neurocomputing*, vol. 71, no. 7-9, pp. 1561-1565, 2008.

[44] N. V. Tinh, N. T. Linh, P. T. Cat, P. M. Tuan, M. N. Anh, and N. P. Anh, "Modeling and feedback linearization control of a nonholonomic wheeled mobile robot with longitudinal, lateral slips," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, 2016, pp. 996-1001: IEEE.

[45] L. Guo, H. Zhao, and Y. Song, "A Nearly Optimal Chattering Reduction Method of Sliding Mode Control With an Application to a Two-wheeled Mobile Robot," *arXiv preprint arXiv:2110.12706*, 2021.

[46] Y. Zheng and O. M. Anubi, "Attack-resilient observer pruning for path-tracking control of wheeled mobile robot," in *Dynamic Systems and Control Conference*, 2020, vol. 84287, p. V002T33A001: American Society of Mechanical Engineers.

[47] D. He, T. Qiu, and R. Luo, "Fuel efficiency-oriented platooning control of connected nonlinear vehicles: A distributed economic MPC approach," *Asian Journal of Control*, vol. 22, no. 4, pp. 1628-1638, 2020.
[48] T. P. Nascimento, C. E. Dórea, and L. M. G. Gonçalves, "Nonholonomic mobile robots' trajectory tracking model predictive control: a survey," *Robotica*, vol. 36, no. 5, p. 676, 2018.
[49] R. W. Brockett, "Asymptotic stability and feedback stabilization," *Differential geometric control theory*, vol. 27, no. 1, pp. 181-191, 1983.
[50] J. Rault, A. Richalet, J. Testud, and J. Papon, "Model predictive heuristic control: application to industrial processes," *Automatica*, vol. 14, no. 5, pp. 413-428, 1978.
[51] O. Calli, C. O. Colpan, and H. Gunerhan, "Energy, exergy and thermoeconomic analyses of biomass and solar powered organic Rankine cycles," *International Journal of Exergy*, vol. 29, no. 2-4, pp. 172-192, 2019.
[52] K. Salahshoor and M. Hadian, "A decentralized event-based model predictive controller design method for large-scale systems," 2014.
[53] M. Hadian, M. Mehrshadian, M. Karami, and A. Biglary Makvand, "Event-based neural network predictive controller application for a distillation column," *Asian Journal of Control*, 2019.
[54] F. Allgower, R. Findeisen, and Z. K. Nagy, "Nonlinear model predictive control: From theory to application," *Journal-Chinese Institute Of Chemical Engineers*, vol. 35, no. 3, pp. 299-316, 2004.
[55] R. Findeisen, L. Imsland, F. Allgower, and B. A. Foss, "State and output feedback nonlinear model predictive control: An overview," *European journal of control*, vol. 9, no. 2-3, pp. 190-206, 2003.
[56] R. Zhang, A. Xue, and F. Gao, *Model predictive control*. Springer, 2019.
[57] B. Kouvaritakis and M. Cannon, *Non-linear Predictive Control: theory and practice* (no. 61). Iet, 2001.
[58] M. Nikolaou, "Model predictive controllers: A critical synthesis of theory and industrial needs," 2001.
[59] H. Yang, H. Zhao, Y. Xia, and J. Zhang, "Nonlinear MPC with time-varying terminal cost for tracking unreachable periodic references," *Automatica*, vol. 123, p. 109337, 2021.
[60] H. R. Adriani, N. Y. Lademakhi, and A. Korayem, "Superiority of Nonlinear and Stable MPC in a Differential Mobile Robot: Accuracy and Solving Speed," in *2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM)*, 2021, pp. 13-17: IEEE.
[61] Y. Lu and Y. Arkun, "Quasi-min-max MPC algorithms for LPV systems," *Automatica*, vol. 36, no. 4, pp. 527-540, 2000.
[62] P. G. Cisneros and H. Werner, "Fast nonlinear MPC for reference tracking subject to nonlinear constraints via quasi-LPV representations," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11601-11606, 2017.
[63] B. Ding, P. Wang, and J. Hu, "Dynamic output feedback robust MPC with one free control move for LPV model with bounded disturbance," *Asian Journal of Control*, vol. 20, no. 2, pp. 755-767, 2018.

[64] A. Dosreisdesouza, D. Efimov, and T. Raïssi, "Robust output feedback MPC for LPV systems using interval observers," *IEEE Transactions on Automatic Control*, 2021.
[65] X. Ping, J. Yao, B. Ding, and Z. Li, "Tube-based output feedback robust MPC for LPV systems with scaled terminal constraint sets," *IEEE Transactions on Cybernetics*, 2021.
[66] D. Li and Y. Xi, "The feedback robust MPC for LPV systems with bounded rates of parameter changes," *IEEE Transactions on Automatic Control*, vol. 55, no. 2, pp. 503-507, 2010.
[67] T.-H. Kim and H.-W. Lee, "Quasi-min-max output-feedback model predictive control for LPV systems with input saturation," *International Journal of Control, Automation and Systems*, vol. 15, no. 3, pp. 1069-1076, 2017.
[68] R. Heydari and M. Farrokhi, "Robust tube-based model predictive control of LPV systems subject to adjustable additive disturbance set," *Automatica*, vol. 129, p. 109672, 2021.
[69] J. Wang, H. Wang, Y. Zhou, Y. Wang, and W. Zhang, "On an integrated approach to resilient transportation systems in emergency situations," *Natural Computing*, pp. 1-9, 2017.
[70] R. Bloss, "By air, land and sea, the unmanned vehicles are coming," *Industrial Robot: An International Journal*, vol. 34, no. 1, pp. 12-16, 2007.
[71] H. G. Nguyen *et al.*, "Land, sea, and air unmanned systems research and development at SPAWAR Systems Center Pacific," in *Unmanned Systems Technology XI*, 2009, vol. 7332, p. 73321I: International Society for Optics and Photonics.
[72] X. Chen, Y. Chen, and J. Chase, *Mobile robots-state of the art in land, sea, air, and collaborative missions*. 2009.
[73] S. G. Tzafestas, *Introduction to mobile robot control*. Elsevier, 2013.
[74] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.
[75] Y. Wang, H. Lang, and C. W. De Silva, "A hybrid visual servo controller for robust grasping by wheeled mobile robots," *IEEE/ASME transactions on Mechatronics*, vol. 15, no. 5, pp. 757-769, 2010.
[76] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419839596, 2019.
[77] J. Martinez-Gomez, A. Fernandez-Caballero, I. Garcia-Varea, L. Rodriguez, and C. Romero-Gonzalez, "A taxonomy of vision systems for ground mobile robots," *International Journal of Advanced Robotic Systems*, vol. 11, no. 7, p. 111, 2014.
[78] A. Soloviev and D. Venable, "Integration of GPS and vision measurements for navigation in GPS challenged environments," in *IEEE/ION Position, Location and Navigation Symposium*, 2010, pp. 826-833: IEEE.
[79] J. Courbon, Y. Mezouar, N. Guénard, and P. Martinet, "Vision-based navigation of unmanned aerial vehicles," *Control Engineering Practice*, vol. 18, no. 7, pp. 789-799, 2010.
[80] R. M. Martín, M. Lorbach, and O. Brock, "Deterioration of depth measurements due to interference of multiple RGB-D sensors," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4205-4212: IEEE.

[81] N. Chatterji, C. Allen, and S. Chernova, "Effectiveness of Robot Communication Level on Likeability, Understandability and Comfortability," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2019, pp. 1-7: IEEE.
[82] K. Fischer, O. Niebuhr, R. M. Langedijk, and S. Eisenberger, "I Shall Know you by your Voice–Melodic and Physical Dominance in the Design of Robot Voices," in *1st International Seminar on the Foundations of Speech*, 2019, pp. 88-90: Syddansk Universitet.
[83] M. Head and L. Hines, "Nursing Education Posts."
[84] B. Mutlu and J. Forlizzi, "Robots in organizations: the role of workflow, social, and environmental factors in human-robot interaction," in *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2008, pp. 287-294: IEEE.
[85] T. Stack, L. T. Ostrom, and C. A. Wilhelmsen, *Occupational ergonomics: A practical approach*. John Wiley & Sons, 2016.
[86] M. A. Roa, D. Berenson, and W. Huang, "Mobile manipulation: toward smart manufacturing [tc spotlight]," *IEEE Robotics & Automation Magazine*, vol. 22, no. 4, pp. 14-15, 2015.
[87] M. Khoramshahi and A. Billard, "A dynamical system approach to task-adaptation in physical human–robot interaction," *Autonomous Robots*, vol. 43, no. 4, pp. 927-946, 2019.
[88] M. Khoramshahi and A. Billard, "A dynamical system approach for detection and reaction to human guidance in physical human–robot interaction," *Autonomous Robots*, vol. 44, no. 8, pp. 1411-1429, 2020.
[89] A. Begić, "Application of service robots for disinfection in medical institutions," in *International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies*, 2017, pp. 1056-1065: Springer.
[90] M. Devetiyarova, E. Zueva, and S. Ostreiko, "Medical robots," 2017.
[91] C. Clabaugh and M. Matarić, "Robots for the people, by the people: Personalizing human-machine interaction," *Science Robotics*, vol. 3, no. 21, p. eaat7451, 2018.
[92] T. Belpaeme *et al.*, "Multimodal child-robot interaction: Building social bonds," *Journal of Human-Robot Interaction*, vol. 1, no. 2, 2012.
[93] L. Cañamero and M. Lewis, "Making new "New AI" friends: designing a social robot for diabetic children from an embodied AI perspective," *International Journal of Social Robotics*, vol. 8, no. 4, pp. 523-537, 2016.
[94] M. Arif and H. Samani, "AMBUBOT: Ambulance robot automated external defibrillator robotic ambulance," in *16th International Conference on Advanced Communication Technology*, 2014, pp. 58-66: IEEE.
[95] Y. Wang, L. He, and C. Huang, "Adaptive time-varying formation tracking control of unmanned aerial vehicles with quantized input," *ISA transactions*, vol. 85, pp. 76-83, 2019.
[96] W. B. Qin, Y. Zhang, D. Takács, G. Stépán, and G. Orosz, "Dynamics and control of automobiles using nonholonomic vehicle models," *arXiv preprint arXiv:2108.02230*, 2021.

[97] B. Li and Z. Shao, "Simultaneous dynamic optimization: A trajectory planning method for nonholonomic car-like robots," *Advances in Engineering Software,* vol. 87, pp. 30-42, 2015.
[98] H. Khan, S. Khatoon, P. Gaur, and S. A. Khan, "Speed control comparison of wheeled mobile robot by ANFIS, Fuzzy and PID controllers," *International Journal of Information Technology,* pp. 1-7, 2022.
[99] I. Kolmanovsky and N. H. McClamroch, "Developments in nonholonomic control problems," *IEEE Control systems magazine,* vol. 15, no. 6, pp. 20-36, 1995.
[100] C. Ren, C. Li, L. Hu, X. Li, and S. Ma, "Adaptive model predictive control for an omnidirectional mobile robot with friction compensation and incremental input constraints," *Transactions of the Institute of Measurement and Control,* vol. 44, no. 4, pp. 835-847, 2022.
[101] D. Chwa, "Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates," *IEEE transactions on control systems technology,* vol. 12, no. 4, pp. 637-644, 2004.
[102] D. Gu and H. Hu, "Neural predictive control for a car-like mobile robot," *Robotics and Autonomous Systems,* vol. 39, no. 2, pp. 73-86, 2002.
[103] D. R. Ramírez, D. Limón, J. Gomez-Ortega, and E. F. Camacho, "Nonlinear MBPC for mobile robot navigation using genetic algorithms," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, 1999, vol. 3, pp. 2452-2457: IEEE.
[104] C. Wu and F. Blaabjerg, "Advanced control of power electronic systems—An overview of methods," *Control of Power Electronic Converters and Systems,* pp. 1-33, 2021.
[105] D. Liu, H. Sun, Q. Jia, and L. Wang, "Motion control of a spherical mobile robot by feedback linearization," in *2008 7th World Congress on Intelligent Control and Automation*, 2008, pp. 965-970: IEEE.
[106] B. d'Andrea-Novel, G. Bastin, and G. Campion, "Dynamic feedback linearization of nonholonomic wheeled mobile robots," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, 1992, pp. 2527, 2528, 2529, 2530, 2531, 2532-2527, 2528, 2529, 2530, 2531, 2532: IEEE Computer Society.
[107] J. M. Ortiz and M. Olivares, "Trajectory tracking control of an omnidirectional mobile robot based on MPC," in *Proc. IEEE 4th Latin Amer. Robot. Symp.(Lars)*, 2007, pp. 8-9.
[108] C. M. Martinez and D. Cao, *iHorizon-Enabled Energy management for electrified vehicles*. Butterworth-Heinemann, 2018.
[109] S. Gambhire, D. R. Kishore, P. Londhe, and S. Pawar, "Review of sliding mode based control techniques for control system applications," *International Journal of dynamics and control,* vol. 9, no. 1, pp. 363-378, 2021.
[110] R. Solea, A. Filipescu, and U. Nunes, "Sliding-mode control for trajectory-tracking of a wheeled mobile robot in presence of uncertainties," in *2009 7th Asian control conference*, 2009, pp. 1701-1706: IEEE.
[111] A. Bessas, A. Benalia, and F. Boudjema, "Integral sliding mode control for trajectory tracking of wheeled mobile robot in presence of uncertainties," *Journal of Control Science and Engineering,* vol. 2016, 2016.

[112] J.-X. Xu, Z.-Q. Guo, and T. H. Lee, "Design and implementation of integral sliding-mode control on an underactuated two-wheeled mobile robot," *IEEE Transactions on industrial electronics,* vol. 61, no. 7, pp. 3671-3681, 2013.
[113] B. Moudoud, H. Aissaoui, and M. Diany, "Fuzzy adaptive sliding mode controller for electrically driven wheeled mobile robot for trajectory tracking task," *Journal of Control and Decision,* vol. 9, no. 1, pp. 71-79, 2022.
[114] A. Jain and A. Sharma, "Membership function formulation methods for fuzzy logic systems: A comprehensive review," *J. Crit. Rev,* vol. 7, no. 19, pp. 8717-8733, 2020.
[115] J. F. Silva and S. F. Pinto, "Linear and nonlinear control of switching power converters," in *Power Electronics Handbook*: Elsevier, 2018, pp. 1141-1220.
[116] A. N. Fard, M. Shahbazian, and M. Hadian, "Adaptive fuzzy controller based on cuckoo optimization algorithm for a distillation column," in *2016 International Conference on Computational Intelligence and Applications (ICCIA)*, 2016, pp. 93-97: IEEE.
[117] M. Jamshidian, M. Hadian, M. M. Zadeh, Z. Kazempoor, P. Bazargan, and H. Salehi, "Prediction of free flowing porosity and permeability based on conventional well logging data using artificial neural networks optimized by imperialist competitive algorithm–a case study in the South Pars Gas field," *Journal of Natural Gas Science and Engineering,* vol. 24, pp. 89-98, 2015.
[118] T. Das and I. N. Kar, "Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots," *IEEE Transactions on Control Systems Technology,* vol. 14, no. 3, pp. 501-510, 2006.
[119] D. Chwa, "Fuzzy adaptive tracking control of wheeled mobile robots with state-dependent kinematic and dynamic disturbances," *IEEE transactions on Fuzzy Systems,* vol. 20, no. 3, pp. 587-593, 2011.
[120] J. Keighobadi and M. B. Menhaj, "From nonlinear to fuzzy approaches in trajectory tracking control of wheeled mobile robots," *Asian Journal of Control,* vol. 14, no. 4, pp. 960-973, 2012.
[121] F. Cuevas, O. Castillo, and P. Cortes-Antonio, "Design of a Control Strategy Based on Type-2 Fuzzy Logic for Omnidirectional Mobile Robots," *Journal of Multiple-Valued Logic & Soft Computing,* vol. 37, 2021.
[122] K. Lee, D.-Y. Im, B. Kwak, and Y.-J. Ryoo, "Design of fuzzy-PID controller for path tracking of mobile robot with differential drive," *International Journal of Fuzzy Logic and Intelligent Systems,* vol. 18, no. 3, pp. 220-228, 2018.
[123] D. N. M. Abadi and M. H. Khooban, "Design of optimal Mamdani-type fuzzy controller for nonholonomic wheeled mobile robots," *Journal of King Saud University-Engineering Sciences,* vol. 27, no. 1, pp. 92-100, 2015.
[124] I. Engedy and G. Horváth, "Artificial neural network based local motion planning of a wheeled mobile robot," in *2010 11th International Symposium on Computational Intelligence and Informatics (CINTI)*, 2010, pp. 213-218: IEEE.
[125] P. Bozek, Y. L. Karavaev, A. A. Ardentov, and K. S. Yefremov, "Neural network control of a wheeled mobile robot based on optimal trajectories," *International Journal of Advanced Robotic Systems,* vol. 17, no. 2, p. 1729881420916077, 2020.

[126] A. Pandey, S. Kumar, K. K. Pandey, and D. R. Parhi, "Mobile robot navigation in unknown static environments using ANFIS controller," *Perspectives in Science,* vol. 8, pp. 421-423, 2016.
[127] A. Pandey, A. K. Kashyap, D. R. Parhi, and B. Patle, "Autonomous mobile robot navigation between static and dynamic obstacles using multiple ANFIS architecture," *World Journal of Engineering,* 2019.
[128] H. Batti, C. Ben Jabeur, and H. Seddik, "Autonomous smart robot for path predicting and finding in maze based on fuzzy and neuro-Fuzzy approaches," *Asian Journal of Control,* vol. 23, no. 1, pp. 3-12, 2021.
[129] C. Ben Jabeur and H. Seddik, "Design of a PID optimized neural networks and PD fuzzy logic controllers for a two-wheeled mobile robot," *Asian Journal of Control,* vol. 23, no. 1, pp. 23-41, 2021.
[130] W. Zhang and C. Van Luttervelt, "Toward a resilient manufacturing system," *CIRP annals,* vol. 60, no. 1, pp. 469-472, 2011.
[131] C. G. Rieger, D. I. Gertman, and M. A. McQueen, "Resilient control systems: Next generation design research," in *2009 2nd Conference on Human System Interactions*, 2009, pp. 632-636: IEEE.
[132] S. Zhang and S. D. Wolthusen, "Efficient control recovery for resilient control systems," in *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*, 2018, pp. 1-6: IEEE.
[133] C. Deng, Y. Wang, C. Wen, Y. Xu, and P. Lin, "Distributed resilient control for energy storage systems in cyber–physical microgrids," *IEEE Transactions on Industrial Informatics,* vol. 17, no. 2, pp. 1331-1341, 2020.
[134] C. G. Rieger, K. L. Moore, and T. L. Baldwin, "Resilient control systems: A multi-agent dynamic systems perspective," in *IEEE International Conference on Electro-Information Technology, EIT 2013*, 2013, pp. 1-16: IEEE.
[135] Y.-C. Liu, K.-Y. Liu, and Z. Song, "Resilient Time-Varying Formation Tracking for Mobile Robot Networks under Deception Attacks on Positioning," *arXiv preprint arXiv:2110.10678,* 2021.
[136] E. M. Amullen, S. Shetty, and L. H. Keel, "Model-based resilient control for a multi-agent system against Denial of Service attacks," in *2016 World Automation Congress (WAC)*, 2016, pp. 1-6: IEEE.
[137] E. Malis, "Survey of vision-based robot control," *ENSIETA European Naval Ship Design Short Course, Brest, France,* vol. 41, p. 46, 2002.
[138] Z. Li, C. Yang, C.-Y. Su, J. Deng, and W. Zhang, "Vision-based model predictive control for steering of a nonholonomic mobile robot," *IEEE Transactions on Control Systems Technology,* vol. 24, no. 2, pp. 553-564, 2015.
[139] O. Meiying, H. Sun, Z. Zhang, and S. Gu, "Fixed-Time Trajectory Tracking Control for Nonholonomic Mobile Robot Based on Visual Servoing," 2021.
[140] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice, "Position based visual servoing: keeping the object in the field of vision," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, 2002, vol. 2, pp. 1624-1629: IEEE.
[141] G. B. Barbosa, E. C. Da Silva, and A. C. Leite, "Robust Image-based Visual Servoing for Autonomous Row Crop Following with Wheeled Mobile Robots," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 2021, pp. 1047-1053: IEEE.

[142] F. Ke, Z. Li, H. Xiao, and X. Zhang, "Visual servoing of constrained mobile robots based on model predictive control," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1428-1438, 2016.
[143] G. Allibert, E. Courtial, and Y. Touré, "Real-time visual predictive controller for image-based trajectory tracking of a mobile robot," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 11244-11249, 2008.
[144] B. Li, Y. Fang, G. Hu, and X. Zhang, "Model-free unified tracking and regulation visual servoing of wheeled mobile robots," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1328-1339, 2015.
[145] R. Wang, X. Zhang, Y. Fang, and B. Li, "Virtual-goal-guided RRT for visual servoing of mobile robots with FOV constraint," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.
[146] Z. Jin, J. Wu, A. Liu, W.-A. Zhang, and L. Yu, "Policy-Based Deep Reinforcement Learning for Visual Servoing Control of Mobile Robots With Visibility Constraints," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 2, pp. 1898-1908, 2021.
[147] J. Zhang, K.-S. Chin, and M. Ławryńczuk, "Nonlinear model predictive control based on piecewise linear Hammerstein models," *Nonlinear Dynamics*, vol. 92, no. 3, pp. 1001-1021, 2018.
[148] S. Yu, C. Böhm, H. Chen, and F. Allgöwer, "Model predictive control of constrained LPV systems," *International Journal of Control*, vol. 85, no. 6, pp. 671-683, 2012.
[149] M. Hadian, A. Ramezani, and W. Zhang, "An interpolation-based model predictive controller for input–output linear parameter varying systems," *International Journal of Dynamics and Control*, pp. 1-14, 2022.
[150] M. Hadian, A. Ramezani, and W. Zhang, "Robust Model Predictive Controller Using Recurrent Neural Networks for Input–Output Linear Parameter Varying Systems," *Electronics*, vol. 10, no. 13, p. 1557, 2021.
[151] M. B. Saltık, L. Özkan, J. H. Ludlage, S. Weiland, and P. M. 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, 2018.
[152] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*: Springer, 1999, pp. 207-226.
[153] I. Alvarado, P. Krupa, D. Limon, and T. Alamo, "Tractable robust MPC design based on nominal predictions," *Journal of Process Control*, vol. 111, pp. 75-85, 2022.
[154] M. Lorenzen, M. Cannon, and F. Allgöwer, "Robust MPC with recursive model update," *Automatica*, vol. 103, pp. 461-471, 2019.
[155] S. Jafari Fesharaki, F. Sheikholeslam, M. Kamali, and A. Talebi, "Tractable robust model predictive control with adaptive sliding mode for uncertain nonlinear systems," *International Journal of Systems Science*, vol. 51, no. 12, pp. 2204-2216, 2020.
[156] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361-1379, 1996.

[157] Y. Lu and Y. Arkun, "A quasi-min-max MPC algorithm for linear parameter varying systems with bounded rate of change of parameters," in *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)*, 2000, vol. 5, pp. 3234-3238: IEEE.

[158] B. Kouvaritakis, J. A. Rossiter, and J. Schuurmans, "Efficient robust predictive control," *IEEE Transactions on automatic control*, vol. 45, no. 8, pp. 1545-1549, 2000.

[159] D. Angeli, A. Casavola, and E. Mosca, "Constrained predictive control of nonlinear plants via polytopic linear system embedding," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal,* vol. 10, no. 13, pp. 1091-1103, 2000.

[160] A. Casavola, D. Famularo, and G. Franze, "A feedback min-max MPC algorithm for LPV systems subject to bounded rates of change of parameters," *IEEE Transactions on Automatic Control*, vol. 47, no. 7, pp. 1147-1153, 2002.

[161] A. Casavola, D. Famularo, G. Franze, and E. Garone, "An improved predictive control strategy for polytopic LPV linear systems," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 5820-5825: IEEE.

[162] A. Casavola, D. Famularo, and G. Franzé, "Robust constrained predictive control of uncertain norm-bounded linear systems," *Automatica,* vol. 40, no. 11, pp. 1865-1876, 2004.

[163] J. Hu and B. Ding, "A periodic approach to dynamic output feedback MPC for quasi-LPV model," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2257-2264, 2020.

[164] X. Ping, Z. Li, and A. Al-Ahmari, "Dynamic output feedback robust MPC for LPV systems subject to input saturation and bounded disturbance," *International Journal of Control, Automation and Systems,* vol. 15, no. 3, pp. 976-985, 2017.

[165] A. Casavola, D. Famularo, and G. Franze, "A predictive control strategy for norm-bounded LPV discrete-time systems with bounded rates of parameter change," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 18, no. 7, pp. 714-740, 2008.

[166] P. Bumroongsri and S. Kheawhom, "An interpolation-based robust mpc algorithm using polyhedral invariant sets," in *2013 European Control Conference (ECC)*, 2013, pp. 3161-3166: IEEE.

[167] D. Rubin, P. Mercader, P.-O. Gutman, H.-N. Nguyen, and A. Bemporad, "Interpolation based predictive control by ellipsoidal invariant sets," *IFAC Journal of Systems and Control*, p. 100084, 2020.

[168] M. Zhao, C. Jiang, X. Tang, and M. She, "Interpolation Model Predictive Control of Nonlinear Systems Described by Quasi-LPV Model," *Automatic Control and Computer Sciences,* vol. 52, no. 5, pp. 354-364, 2018.

[169] W.-J. Mao, "Robust stabilization of uncertain time-varying discrete systems and comments on “an improved approach for constrained robust model predictive control”," *Automatica,* vol. 39, no. 6, pp. 1109-1112, 2003.

[170] E. Garone and A. Casavola, "Receding horizon control strategies for constrained LPV systems based on a class of nonlinearly parameterized Lyapunov functions," *IEEE transactions on automatic control*, vol. 57, no. 9, pp. 2354-2360, 2012.

[171] Q. Cheng, M. Cannon, and B. Kouvaritakis, "The design of dynamics in the prediction structure of robust MPC," *International Journal of Control,* vol. 86, no. 11, pp. 2096-2103, 2013.
[172] B. Pluymers, "Robust model based predictive control-an invariant set approach," *status: published,* 2006.
[173] N. Hoai Nam, "Ellipsoidal Set Based Command Governors for Constrained Linear Systems with Bounded Disturbances," 2021.
[174] N. Hoai Nam, "Improved Prediction Dynamics for Robust MPC," 2022.
[175] B. Zhang, Y. Song, and H. Cai, "An efficient model predictive control for Markovian jump systems with all unstable modes," *Nonlinear Analysis: Hybrid Systems,* vol. 44, p. 101117, 2022.
[176] W. Langson, I. Chryssochoos, S. Raković, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica,* vol. 40, no. 1, pp. 125-133, 2004.
[177] J. Fleming, B. Kouvaritakis, and M. Cannon, "Robust tube MPC for linear systems with multiplicative uncertainty," *IEEE Transactions on Automatic Control,* vol. 60, no. 4, pp. 1087-1092, 2014.
[178] D. Muñoz-Carpintero, M. Cannon, and B. Kouvaritakis, "Robust MPC strategy with optimized polytopic dynamics for linear systems with additive and multiplicative uncertainty," *Systems & Control Letters,* vol. 81, pp. 34-41, 2015.
[179] J. Hanema, R. Tóth, and M. Lazar, "Tube-based anticipative model predictive control for linear parameter-varying systems," in *2016 IEEE 55th Conference on Decision and Control (CDC),* 2016, pp. 1458-1463: IEEE.
[180] G. Pannocchia, "Robust model predictive control with guaranteed setpoint tracking," *Journal of Process control,* vol. 14, no. 8, pp. 927-937, 2004.
[181] Y. J. Wang and J. B. Rawlings, "A new robust model predictive control method I: theory and computation," *Journal of Process Control,* vol. 14, no. 3, pp. 231-247, 2004.
[182] D. Limón, I. Alvarado, T. Alamo, and E. F. Camacho, "MPC for tracking piecewise constant references for constrained linear systems," *Automatica,* vol. 44, no. 9, pp. 2382-2387, 2008.
[183] L. Chisci, P. Falugi, and G. Zappa, "Gain-scheduling MPC of nonlinear systems," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal,* vol. 13, no. 3-4, pp. 295-308, 2003.
[184] J. Hanema, M. Lazar, and R. Tóth, "Tube-based LPV constant output reference tracking MPC with error bound," *IFAC-PapersOnLine,* vol. 50, no. 1, pp. 8612-8617, 2017.
[185] H. A. Pipino, M. M. Morato, E. Bernardi, E. J. Adam, and J. E. Normey-Rico, "Nonlinear temperature regulation of solar collectors with a fast adaptive polytopic LPV MPC formulation," *Solar Energy,* vol. 209, pp. 214-225, 2020.
[186] M. Bacic, M. Cannon, Y. I. Lee, and B. Kouvaritakis, "General interpolation in MPC and its advantages," *IEEE Transactions on Automatic Control,* vol. 48, no. 6, pp. 1092-1096, 2003.
[187] Z. Wan, B. Pluymers, M. V. Kothare, and B. De Moor, "Comments on:“Efficient robust constrained model predictive control with a time varying terminal constraint set” by Wan and Kothare," *Systems & control letters,* vol. 55, no. 7, pp. 618-621, 2006.

[188] P. Bumroongsri, "Tube-based robust MPC for linear time-varying systems with bounded disturbances," *International Journal of Control, Automation and Systems*, vol. 13, no. 3, pp. 620-625, 2015.
[189] R. Findeisen and F. Allgöwer, "An introduction to nonlinear model predictive control," in *21st Benelux meeting on systems and control*, 2002, vol. 11, pp. 119-141: Technische Universiteit Eindhoven Veldhoven Eindhoven, The Netherlands.
[190] N. Blet, D. Megias, J. Serrano, and C. De Prada, "Nonlinear MPC versus MPC using on-line linearisation—a comparative study," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 147-152, 2002.
[191] W. F. Lages and J. A. V. Alves, "Real-time control of a mobile robot using linearized model predictive control," *IFAC Proceedings Volumes*, vol. 39, no. 16, pp. 968-973, 2006.
[192] J. H. Lee, "Model predictive control: Review of the three decades of development," *International Journal of Control, Automation and Systems*, vol. 9, no. 3, p. 415, 2011.
[193] F. Bertoncelli, F. Ruggiero, and L. Sabattini, "Linear time-varying MPC for nonprehensile object manipulation with a nonholonomic mobile robot," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 11032-11038: IEEE.
[194] D. Li, X. Tao, N. Li, and S. Li, "Switched offline multiple model predictive control with polyhedral invariant sets," *Industrial & Engineering Chemistry Research*, vol. 56, no. 34, pp. 9629-9637, 2017.
[195] K. Worthmann, M. W. Mehrez, M. Zanon, G. K. Mann, R. G. Gosine, and M. Diehl, "Model predictive control of nonholonomic mobile robots without stabilizing constraints and costs," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1394-1406, 2015.
[196] M. Kogel and R. Findeisen, "Robust output feedback MPC with reduced conservatism for linear uncertain systems using time varying tubes," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 2571-2577: IEEE.
[197] S. G. Vougioukas, "Reactive trajectory tracking for mobile robots based on non linear model predictive control," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 3074-3079: IEEE.
[198] D. Gu and H. Hu, "Receding horizon tracking control of wheeled mobile robots," *IEEE Transactions on control systems technology*, vol. 14, no. 4, pp. 743-749, 2006.
[199] M. W. Mehrez, G. K. Mann, and R. G. Gosine, "Stabilizing nmpc of wheeled mobile robots using open-source real-time software," in *2013 16th International Conference on Advanced Robotics (ICAR)*, 2013, pp. 1-6: IEEE.
[200] J. L. Piovesan and H. G. Tanner, "Randomized model predictive control for robot navigation," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 94-99: IEEE.
[201] Y. Liu, S. Yu, B. Gao, and H. Chen, "Receding horizon following control of wheeled mobile robots: A case study," in *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2015, pp. 2571-2576: IEEE.

[202] M.-M. Ma, S. Li, and X.-J. Liu, "Tracking control and stabilization of wheeled mobile robots by nonlinear model predictive control," in *Proceedings of the 31st Chinese Control Conference*, 2012, pp. 4056-4061: IEEE.
[203] S. Yu, X. Li, H. Chen, and F. Allgöwer, "Nonlinear model predictive control for path following problems," *International Journal of Robust and Nonlinear Control,* vol. 25, no. 8, pp. 1168-1182, 2015.
[204] H. X. Araújo, A. G. Conceição, G. H. Oliveira, and J. Pitanga, "Model predictive control based on LMIs applied to an omni-directional mobile robot," *IFAC Proceedings Volumes,* vol. 44, no. 1, pp. 8171-8176, 2011.
[205] F. Rußwurm, W. Esterhuizen, K. Worthmann, and S. Streif, "On MPC without terminal conditions for dynamic non-holonomic robots," *IFAC-PapersOnLine,* vol. 54, no. 6, pp. 133-138, 2021.
[206] Y. Tian-Tian, L. Zhi-Yuan, C. Hong, and P. Run, "Formation control and obstacle avoidance for multiple mobile robots," *Acta Automatica Sinica,* vol. 34, no. 5, pp. 588-593, 2008.
[207] D. Henriksson and J. Åkesson, *Flexible implementation of model predictive control using sub-optimal solutions*. Univ., 2004.
[208] K. Benbouabdallah and Q. Zhu, "Improved genetic algorithm lyapunov-based controller for mobile robot tracking a moving target," *Research Journal of Applied Sciences, Engineering and Technology,* vol. 5, no. 15, pp. 4023-4028, 2013.
[209] L. T. Biegler, "Efficient solution of dynamic optimization and NMPC problems," in *Nonlinear model predictive control*: Springer, 2000, pp. 219-243.
[210] X. Wang, D. Han, Y. Lin, and W. Du, "Recent progress and challenges in process optimization: Review of recent work at ECUST," *The Canadian Journal of Chemical Engineering,* vol. 96, no. 10, pp. 2115-2123, 2018.
[211] K. M. Abughalieh and S. G. Alawneh, "A survey of parallel implementations for model predictive control," *IEEE Access,* vol. 7, pp. 34348-34360, 2019.
[212] I. J. Wolf and W. Marquardt, "Fast NMPC schemes for regulatory and economic NMPC–A review," *Journal of Process Control,* vol. 44, pp. 162-183, 2016.
[213] A. S. Conceicao, A. P. Moreira, and P. J. Costa, "A nonlinear model predictive control strategy for trajectory tracking of a four-wheeled omnidirectional mobile robot," *Optimal Control Applications and Methods,* vol. 29, no. 5, pp. 335-352, 2008.
[214] H. Lim, Y. Kang, C. Kim, J. Kim, and B.-J. You, "Nonlinear model predictive controller design with obstacle avoidance for a mobile robot," in *2008 IEEE/ASME International Conference on Mechtronic and Embedded Systems and Applications*, 2008, pp. 494-499: IEEE.
[215] T. A. Teatro, J. M. Eklund, and R. Milman, "Nonlinear model predictive control for omnidirectional robot motion planning and tracking with avoidance of moving obstacles," *Canadian Journal of Electrical and Computer Engineering,* vol. 37, no. 3, pp. 151-156, 2014.
[216] A. Bouzerdoum and T. R. Pattison, "Neural network for quadratic optimization with bound constraints," *IEEE transactions on neural networks,* vol. 4, no. 2, pp. 293-304, 1993.

[217] K. C. Tan, H. Tang, and Z. Yi, "Global exponential stability of discrete-time neural networks for constrained quadratic optimization," *Neurocomputing,* vol. 56, pp. 399-406, 2004.
[218] Y. Xia and J. Wang, "A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 51, no. 7, pp. 1385-1394, 2004.
[219] Y. Xia, G. Feng, and J. Wang, "A novel recurrent neural network for solving nonlinear optimization problems with inequality constraints," *IEEE Transactions on neural networks,* vol. 19, no. 8, pp. 1340-1353, 2008.
[220] Y. Xia and J. Wang, "A general projection neural network for solving monotone variational inequalities and related optimization problems," *IEEE Transactions on Neural Networks,* vol. 15, no. 2, pp. 318-328, 2004.
[221] L. V. Nguyen and X. Qin, "Some results on strongly pseudomonotone quasi-variational inequalities," *Set-Valued and Variational Analysis,* vol. 28, no. 2, pp. 239-257, 2020.
[222] Y. Pan and J. Wang, "Model predictive control for nonlinear affine systems based on the simplified dual neural network," in *2009 IEEE Control Applications,(CCA) & Intelligent Control,(ISIC)*, 2009, pp. 683-688: IEEE.
[223] Z. Wu and P. D. Christofides, "Optimizing process economics and operational safety via economic MPC using barrier functions and recurrent neural network models," *Chemical Engineering Research and Design,* vol. 152, pp. 455-465, 2019.
[224] N. Lanzetti, Y. Z. Lian, A. Cortinovis, L. Dominguez, M. Mercangöz, and C. Jones, "Recurrent neural network based MPC for process industries," in *2019 18th European Control Conference (ECC)*, 2019, pp. 1005-1010: IEEE.
[225] J. Gomez-Ortega and E. F. Camacho, "Neural network MBPC for mobile robot path tracking," *Robotics and computer-integrated manufacturing,* vol. 11, no. 4, pp. 271-278, 1994.
[226] D. Gu and H. Hu, "Wavelet neural network based predictive control for mobile robots," in *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0*, 2000, vol. 5, pp. 3544-3549: IEEE.
[227] J.-Q. Huang and F. L. Lewis, "Neural-network predictive control for nonlinear dynamic systems with time-delay," *IEEE Transactions on Neural Networks,* vol. 14, no. 2, pp. 377-389, 2003.
[228] Y. Pan and J. Wang, "A neurodynamic optimization approach to nonlinear model predictive control," in *2010 IEEE International Conference on Systems, Man and Cybernetics*, 2010, pp. 1597-1602: IEEE.
[229] J. Deng, Z. Li, and C.-Y. Su, "Trajectory tracking of mobile robots based on model predictive control using primal dual neural network," in *Proceedings of the 33rd Chinese Control Conference*, 2014, pp. 8353-8358: IEEE.
[230] Z. Li, J. Deng, R. Lu, Y. Xu, J. Bai, and C.-Y. Su, "Trajectory-tracking control of mobile robot systems incorporating neural-dynamic optimized model predictive approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 46, no. 6, pp. 740-749, 2015.

[231] C. Lian, X. Xu, H. Chen, and H. He, "Near-optimal tracking control of mobile robots via receding-horizon dual heuristic programming," *IEEE transactions on cybernetics*, vol. 46, no. 11, pp. 2484-2496, 2015.
[232] Y.-C. Huang and H.-Y. Li, "Receding Horizon Optimal controller for reference trajectory tracking in Mars entry guidance," in *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, 2016, pp. 2442-2449: IEEE.
[233] U. Rosolia, S. De Bruyne, and A. G. Alleyne, "Autonomous vehicle control: A nonconvex approach for obstacle avoidance," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 2, pp. 469-484, 2016.
[234] M. Bujarbaruah, X. Zhang, M. Tanaskovic, and F. Borrelli, "Adaptive MPC under time varying uncertainty: Robust and Stochastic," *arXiv preprint arXiv:1909.13473*, 2019.
[235] J. Tang, "Motion planning for mobile robot with uncertainty: a model predictive control approach," 2021.
[236] J. Hu and B. Ding, "Output feedback robust MPC for linear systems with norm-bounded model uncertainty and disturbance," *Automatica*, vol. 108, p. 108489, 2019.
[237] L. Pacheco, N. Luo, and J. Ferrer, "Local model predictive control experiences with differential driven wheeled mobile robots," in *2008 IEEE International Conference on Automation, Quality and Testing, Robotics*, 2008, vol. 2, pp. 377-382: IEEE.
[238] Y. Yoon, T. Choe, Y. Park, and H. J. Kim, "Obstacle avoidance for wheeled robots in unknown environments using model predictive control," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 6792-6797, 2008.
[239] K. Kanjanawanishkul and A. Zell, "Path following for an omnidirectional mobile robot based on model predictive control," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3341-3346: IEEE.
[240] J. R. Pitanga, H. X. Araújo, A. G. Conceição, and G. H. Oliveira, "Stable model-based predictive control for wheeled mobile robots using linear matrix inequalities," *IFAC-PapersOnLine*, vol. 48, no. 19, pp. 33-38, 2015.
[241] Z. Zeng, H. Lu, and Z. Zheng, "High-speed trajectory tracking based on model predictive control for omni-directional mobile robots," in *2013 25th Chinese Control and Decision Conference (CCDC)*, 2013, pp. 3179-3184: IEEE.
[242] D. Navarro, *Learning Statistics with R: A Tutorial for Psychology Students and Other Beginners: Version 0.5*. Citeseer, 2013.
[243] C. J. Ostafew, A. P. Schoellig, T. D. Barfoot, and J. Collier, "Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking," *Journal of Field Robotics*, vol. 33, no. 1, pp. 133-152, 2016.
[244] G. Williams *et al.*, "Information theoretic MPC for model-based reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1714-1721: IEEE.
[245] A. Al-Araji, "Design of a cognitive neural predictive controller for mobile robot," Brunel University School of Engineering and Design PhD Theses, 2012.
[246] S. J. Yoo, Y. H. Choi, and J. B. Park, "Generalized predictive control based on self-recurrent wavelet neural network for stable path tracking of mobile robots: adaptive learning rates approach," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 6, pp. 1381-1394, 2006.

[247] H. Hu and D. Gu, "Generalised predictive control of an industrial mobile robot," in *IASTED International Conference on Intelligent Systems and Control, Santa Barbara, CA, USA*, 1999, pp. 28-30: Citeseer.
[248] J. Li *et al.*, "Neural approximation-based model predictive tracking control of non-holonomic wheel-legged robots," *International Journal of Control, Automation and Systems*, vol. 19, no. 1, pp. 372-381, 2021.
[249] Y. Kebbati, N. A. Oufroukh, V. Vigneron, and D. Ichalal, "Neural Network and ANFIS based auto-adaptive MPC for path tracking in autonomous vehicles," in *18ème IEEE International Conference on Networking, Sensing and Control (ICNSC 2021)*, 2021.
[250] E. Alcalá, V. Puig, and J. Quevedo, "LPV-MPC control for autonomous vehicles," *IFAC-PapersOnLine*, vol. 52, no. 28, pp. 106-113, 2019.
[251] C. P. Fernández, "Stabilizing model predictive control for wheeled mobile robots with linear parameter-varying and different time-scales," in *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, 2018, pp. 314-319: IEEE.
[252] A. Casavola, W. Lucia, and F. Tedesco, "A networked-based MPC architecture for constrained LPV systems," *IFAC-PapersOnLine*, vol. 48, no. 26, pp. 158-163, 2015.
[253] S. Liu, K. Liu, Z. Zhong, J. Yi, and H. Aliev, "A novel wheeled mobile robots control based on robust hybrid controller: Mixed H2/H∞ and predictive algorithm approach," *Journal of King Saud University-Computer and Information Sciences*, 2021.
[254] G. Zhang, W. Qin, Q. Qin, B. He, and G. Liu, "Varying gain mpc for consensus tracking with application to formation control of omnidirectional mobile robots," in *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, 2016, pp. 2957-2962: IEEE.
[255] S. Han and S. Lee, "Sampled-Data MPC for Leader-Following of Multi-Mobile Robot System," *The Transactions of the Korean Institute of Electrical Engineers*, vol. 67, no. 2, pp. 308-313, 2018.
[256] A. Subiantoro, M. S. A. Hadi, and A. Muis, "Distributed linear parameter varying model predictive controller with event-triggered mechanism for nonholonomic mobile robot," in *2019 International Conference on Advances in the Emerging Computing Technologies (AECT)*, 2020, pp. 1-6: IEEE.
[257] E. Fernandez-Camacho and C. Bordons-Alba, *Model predictive control in the process industry*. Springer, 1995.
[258] L. Grüne and J. Pannek, "Nonlinear model predictive control," in *Nonlinear Model Predictive Control*: Springer, 2017, pp. 45-69.
[259] Q. Chi, Z. Fei, K. Liu, and J. Liang, "Latent-variable nonlinear model predictive control strategy for a pH neutralization process," *Asian Journal of Control*, vol. 17, no. 6, pp. 2427-2434, 2015.
[260] H. Fukushima, T.-H. Kim, and T. Sugie, "Adaptive model predictive control for a class of constrained linear systems based on the comparison model," *Automatica*, vol. 43, no. 2, pp. 301-308, 2007.
[261] B. Zhu and X. Xia, "Adaptive model predictive control for unconstrained discrete-time linear systems with parametric uncertainties," *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 3171-3176, 2015.

[262] B. Tian, W. Fan, R. Su, and Q. Zong, "Nonlinear robust adaptive deterministic control for flexible hypersonic vehicles in the presence of input constraint," *Asian Journal of Control,* vol. 17, no. 6, pp. 2303-2316, 2015.
[263] H. Mo and G. Farid, "Nonlinear and adaptive intelligent control techniques for quadrotor uav–a survey," *Asian Journal of Control,* vol. 21, no. 2, pp. 989-1008, 2019.
[264] D. Q. Mayne, M. M. Seron, and S. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica,* vol. 41, no. 2, pp. 219-224, 2005.
[265] Y. Zhao, H. Gao, and T. Chen, "Fuzzy constrained predictive control of non-linear systems with packet dropouts," *IET Control Theory & Applications,* vol. 4, no. 9, pp. 1665-1677, 2010.
[266] M. Hadian, N. AliAkbari, and M. Karami, "Using artificial neural network predictive controller optimized with Cuckoo Algorithm for pressure tracking in gas distribution network," *Journal of Natural Gas Science and Engineering,* vol. 27, pp. 1446-1454, 2015.
[267] F. J. Lin, S. Y. Lee, and P. H. Chou, "Intelligent integral backstepping sliding-mode control using recurrent neural network for piezo-flexural nanopositioning stage," *Asian Journal of Control,* vol. 18, no. 2, pp. 456-472, 2016.
[268] M. M. Morato, J. E. Normey-Rico, and O. Sename, "Model predictive control design for linear parameter varying systems: A survey," *Annual Reviews in Control,* 2020.
[269] J. Mohammadpour and C. W. Scherer, *Control of linear parameter varying systems with applications*. Springer Science & Business Media, 2012.
[270] J. S. Shamma, "An overview of LPV systems," in *Control of linear parameter varying systems with applications*: Springer, 2012, pp. 3-26.
[271] J. Hu and B. Ding, "An efficient offline implementation for output feedback min-max MPC," *International Journal of Robust and Nonlinear Control,* vol. 29, no. 2, pp. 492-506, 2019.
[272] K. Liu, "Identification of linear time-varying systems," *Journal of Sound and Vibration,* vol. 206, no. 4, pp. 487-505, 1997.
[273] Y. Yang and B. Ding, "Model predictive control for LPV models with maximal stabilizable model range," *Asian Journal of Control,* vol. 22, no. 5, pp. 1940-1950, 2020.
[274] F. Zhou, M. Gan, and C. P. Chen, "Robust model predictive control algorithm with variable feedback gains for output tracking," *IEEE Transactions on Industrial Electronics,* 2020.
[275] H. M. Calderón, P. S. Cisneros, and H. Werner, "qLPV predictive control-a benchmark study on state space vs input-output approach," *IFAC-PapersOnLine,* vol. 52, no. 28, pp. 146-151, 2019.
[276] H. Abbas, J. Hanema, R. Tóth, J. Mohammadpour, and N. Meskin, "An improved robust model predictive control for linear parameter-varying input-output models," *International Journal of Robust and Nonlinear Control,* vol. 28, no. 3, pp. 859-880, 2018.
[277] P. S. Cisneros and H. Werner, "Stabilizing model predictive control for nonlinear systems in input-output quasi-LPV form," in *2019 American Control Conference (ACC)*, 2019, pp. 1002-1007: IEEE.

[278] M. M. Morato, J. E. Normey-Rico, and O. Sename, "Novel qLPV MPC design with least-squares scheduling prediction," *IFAC-PapersOnLine,* vol. 52, no. 28, pp. 158-163, 2019.
[279] H. S. Abbas, R. Toth, N. Meskin, J. Mohammadpour, and J. Hanema, "A robust MPC for input-output LPV models," *IEEE Transactions on Automatic Control,* vol. 61, no. 12, pp. 4183-4188, 2016.
[280] J. A. Rossiter and Y. Ding, "Interpolation methods in model predictive control: an overview," *International Journal of Control,* vol. 83, no. 2, pp. 297-312, 2010.
[281] S. Wollnack and H. Werner, "LPV-IO controller design: An LMI approach," in *2016 American Control Conference (ACC)*, 2016, pp. 4617-4622: IEEE.
[282] J. Liu, X. Chen, D. Muñoz de la Peña, and P. D. Christofides, "Sequential and iterative architectures for distributed model predictive control of nonlinear process systems," *AIChE Journal,* vol. 56, no. 8, pp. 2137-2149, 2010.
[283] E. Kayacan, E. Kayacan, H. Ramon, and W. Saeys, "Distributed nonlinear model predictive control of an autonomous tractor–trailer system," *Mechatronics,* vol. 24, no. 8, pp. 926-933, 2014.
[284] V. Yaramasu and B. Wu, *Model predictive control of wind energy conversion systems*. John Wiley & Sons, 2016.
[285] S. V. Raković and W. S. Levine, *Handbook of model predictive control*. Springer, 2018.
[286] C. Hu, X. Wei, and Y. Ren, "Passive fault-tolerant control based on weighted LPV tube-MPC for air-breathing hypersonic vehicles," *International Journal of Control, Automation and Systems,* vol. 17, no. 8, pp. 1957-1970, 2019.
[287] X. Wang, X. Zhang, and X. Yang, "Delay-dependent robust dissipative control for singular LPV systems with multiple input delays," *International Journal of Control, Automation and Systems,* vol. 17, no. 2, pp. 327-335, 2019.
[288] J. Hu and B. Ding, "One-step ahead robust MPC for LPV model with bounded disturbance," *European Journal of Control,* vol. 52, pp. 59-66, 2020.
[289] E. Alcalá, V. Puig, J. Quevedo, and U. Rosolia, "Autonomous racing using Linear Parameter Varying-Model Predictive Control (LPV-MPC)," *Control Engineering Practice,* vol. 95, p. 104270, 2020.
[290] Z. Xu, J. Zhao, J. Qian, and Y. Zhu, "Nonlinear MPC using an identified LPV model," *Industrial & Engineering Chemistry Research,* vol. 48, no. 6, pp. 3043-3051, 2009.
[291] S. Zhang, L. Dai, and Y. Xia, "Adaptive MPC for constrained systems with parameter uncertainty and additive disturbance," *IET Control Theory & Applications,* vol. 13, no. 15, pp. 2500-2506, 2019.
[292] E. Alcala, V. Puig, J. Quevedo, T. Escobet, and R. Comasolivas, "Autonomous vehicle control using a kinematic Lyapunov-based technique with LQR-LMI tuning," *Control engineering practice,* vol. 73, pp. 1-12, 2018.
[293] E. Alcalá, V. Puig, J. Quevedo, and O. Sename, "Fast zonotope-tube-based LPV-MPC for autonomous vehicles," *IET Control Theory & Applications,* vol. 14, no. 20, pp. 3676-3685, 2020.
[294] K. Mondal, A. A. Rodriguez, S. S. Manne, N. Das, and B. Wallace, "Comparison of kinematic and dynamic model based linear model predictive control of non-

holonomic robot for trajectory tracking: Critical trade-offs addressed," in *IASTED International Conference on Mechatronics and Control*, 2019.

[295] F. Ke, Z. Li, and C. Yang, "Robust tube-based predictive control for visual servoing of constrained differential-drive mobile robots," *IEEE Transactions on Industrial Electronics,* vol. 65, no. 4, pp. 3437-3446, 2017.

[296] K. M. M. Rathai, J. Amirthalingam, and B. Jayaraman, "Robust tube-MPC based lane keeping system for autonomous driving vehicles," in *Proceedings of the Advances in Robotics*, 2017, pp. 1-6.

[297] E. Alcala, V. Puig, J. Quevedo, and T. Escobet, "Gain-scheduling LPV control for autonomous vehicles including friction force estimation and compensation mechanism," *IET Control Theory & Applications,* vol. 12, no. 12, pp. 1683-1693, 2018.

[298] M. Sabouri and M. H. Asemani, "LPV Controller Design for Trajectory Tracking of Non-holonomic Wheeled Mobile Robots in the Presence of Slip," in *2021 29th Iranian Conference on Electrical Engineering (ICEE)*, 2021, pp. 715-720: IEEE.

[299] W. Kwon and S. Lee, "Sampled-Data Model Predictive Tracking Control for Mobile Robot," *International Journal of Electrical and Information Engineering*, vol. 11, no. 4, pp. 468-471, 2017.

[300] P. Coelho and U. Nunes, "Lie algebra application to mobile robot control: a tutorial," *Robotica,* vol. 21, no. 5, pp. 483-493, 2003.

APPENDIX A

Copyright permission letters from Co-Authors

To Whom It May Concern:

I, Chris Zhang, hereby grant permission to Mr. Mohsen Hadian to include the following papers in his thesis titled "Robust Model Predictive Control for Linear Parameter varying Systems and its Application in a Medcial Robots".

1. *Hadian, M., & **Zhang, W.** (2022). A Review of Control Techniques for Medical Wheeled Mobile Robot: Towards Application of Resilient Model Predictive Control with Linear Parameter Varying Systems. Annual Reviews in Control. Under review.*
2. *Hadian, M., & **Zhang, W.**, Ramezani, A. (2021). Interpolation-based Model predictive controller for input-output linear parameter varying systems. International Journal of Dynamics and Control. Published.*
3. *Hadian, M., Ramezani, A., & **Zhang, W.** (2021). Robust Model Predictive Controller Using Recurrent Neural Networks for Input–Output Linear Parameter Varying Systems. Electronics, 10(13), 1557. Published.*
4. *Hadian, M., **Zhang, W.**, & Ghanevati, A. (2022). Application of Nonlinear Model Predictive Control for an Anaerobic Digestion Process Using an Input-output LPV System Identified by LS-SVM. International Journal of dynamics and control, revised.*
5. *Hadian, M., **Zhang, W.**, & Etesami, D. (2022). An Improved Interpolated Model Predictive Control based on Recurrent Neural Networks for a nonholonomic differential-drive mobile robot with Quasi-LPV Representation: computational complexity and conservatism. International Journal of Robust and Nonlinear Control, under review.*

I am aware that all University of Saskatchewan theses are also posted in the digital USask eCommons thesis repository, making the thesis openly available on the internet.

Date:

Signature:

To Whom It May Concern:

I, Danial Etesami, hereby grant permission to Mr. Mohsen Hadian to include the following papers in his thesis titled "Robust Model Predictive Control for Linear Parameter varying Systems and its Application in a Medcial Robots".

1. *Hadian, M., Zhang, W., &* ***Etesami, D.*** *(2022). An Improved Interpolated Model Predictive Control based on Recurrent Neural Networks for a nonholonomic differential-drive mobile robot with Quasi-LPV Representation: computational complexity and conservatism. International Journal of Robust and Nonlinear Control, under review.*

I am aware that all University of Saskatchewan theses are also posted in the digital USask eCommons thesis repository, making the thesis openly available on the internet.

Date:

Signature:

To Whom It May Concern:

I, Amin Ramezani, hereby grant permission to Mr. Mohsen Hadian to include the following papers in his thesis titled "Robust Model Predictive Control for Linear Parameter varying Systems and its Application in a Medcial Robots".

1. *Hadian, M., & Zhang, W., **Ramezani, A.** (2021). Interpolation-based Model predictive controller for input-output linear parameter varying systems. International Journal of Dynamics and Control. Published.*
2. *Hadian, M., **Ramezani, A.**, & Zhang, W. (2021). Robust Model Predictive Controller Using Recurrent Neural Networks for Input–Output Linear Parameter Varying Systems. Electronics, 10(13), 1557. Published.*

I am aware that all University of Saskatchewan theses are also posted in the digital USask eCommons thesis repository, making the thesis openly available on the internet.

Date:

Signature:

To Whom It May Concern:

I, Amin Ghanevati, hereby grant permission to Mr. Mohsen Hadian to include the following papers in his thesis titled "Robust Model Predictive Control for Linear Parameter varying Systems and its Application in a Medcial Robots".

1. *Hadian, M., Zhang, W., & **Ghanevati, A.** (2022). Application of Nonlinear Model Predictive Control for an Anaerobic Digestion Process Using an Input-output LPV System Identified by LS-SVM. International Journal of dynamics and control, revised.*

I am aware that all University of Saskatchewan theses are also posted in the digital USask eCommons thesis repository, making the thesis openly available on the internet.

Date:

Signature: