MASTER’S DEGREE THESIS

Politecnico di Torino

Master’s Degree in Mechatronic Engineering

Andrea Pietrinferni

Design of an MPC control policy for a non-linear system based on PWA model representation

---

Supervisors:

Massimo CANALE	Politecnico di Torino
Madiha NADRI	Université Lyon 1
Pascal DUFOUR	Université Lyon 1

Abstract

Piecewise affine systems have emerged as focal points of study within the realm of control theory due to their ability to accurately model complex, non-linear dynamical systems. This thesis navigates the motivations behind the extensive investigation of piecewise affine systems and elucidates their practical utility.

The inherent flexibility of piecewise affine models allows for a perfect representation of systems that exhibit distinct behaviors in different regions. This attribute makes them particularly well-suited for applications in robotics, power systems, and process control, where the ability to capture nonlinearity is crucial. Examining the convenience of piecewise affine systems in control scenarios reveals their capacity to address challenging dynamics and facilitate the design of efficient control strategies.

In the domain of optimal control solutions, Model Predictive Control (MPC) emerges as a compelling technique when applied to piecewise affine systems. Toward the conclusion of this exploration, this thesis delves into the MPC's applicability in this context, highlighting the advantages it offers. By leveraging MPC's predictive capabilities, these systems can achieve enhanced performance, increased robustness, and improved adaptability. The synergies between piecewise affine systems and MPC provide a promising avenue for advancing the state-of-the-art in control theory and its practical implementation. This thesis contributes to the ongoing discourse by explaining implications of studying piecewise affine systems, showcasing their versatility, and ultimately underscoring why MPC stands out as an optimal technique for harnessing the full potential of these intriguing dynamical models.

Contents

Contents **2**

1 Introduction **4**

2 State of art **6**

2.1 Piecewise affine dynamical systems 6

2.2 MPC controller 10

2.3 Outline of the thesis 11

3 PWA model **13**

3.1 Mathematical Formulation of PWA Models 13

3.2 Identification and Estimation Techniques for PWA Models 17

3.2.1 Problem statement 20

3.2.2 Estimation of PWA Models from data 23

3.3 Stability and Performance Analysis 24

3.3.1 Dissipativity analysis 24

3.3.2 Lyapunov stability 26

3.4 Uncertainty Models 32

3.5 Example of global model construction through ML 35

3.6 Conclusion 40

4 MPC for PWA systems **42**

4.1 Control of a PWA model 43
4.1.1 Introduction 43
4.1.2 Fundamental Principles of PWA's Control 46
4.2 Introduction of MPC 48
4.2.1 Optimal control: problem formulation 49
4.2.2 Model Predictive Control (MPC) 51
4.2.3 Robustness against uncertainty 52
4.3 MPC for PWA Systems 55
4.4 Robust MPC for PWA systems 58
4.5 Conclusion 62

5 Experimental results 63

5.1 Example I 63
5.2 Example II 71
5.3 Conclusion and Future Developments 75

1 Introduction

The study of physical, biological, and economic systems often involves mathematical modeling using constructs like differential or difference equations. These systems evolve over time or another independent variable, following dynamic relations. To meet specific requirements within a finite period (horizon), external inputs or controls can be applied. However, these requirements must be satisfied while adhering to the inherent limitations and constraints of the systems, such as equipment or safety constraints. When seeking to fulfill these requirements, there may be various controls that achieve the same outcome.

Among these, one control may be considered the "best" in terms of performance. The measure of performance is typically quantified using a cost function. The constraints on inputs and outputs are referred to as constraints, and the optimal control associated with achieving the best performance is known as optimal control. The methodology for solving such problems is termed finite-horizon optimal control.

The design of an infinite-horizon optimal control can be achieved through a receding horizon approach by repeatedly solving finite-horizon optimal control problems. In this methodology, a finite-horizon optimal control problem is solved at each step, yielding an optimal control sequence. However, only the first control sample from this optimal sequence is applied to the system. Subsequently, a new finite-horizon optimal control problem is solved in the next step. This iterative process is known as Model Predictive Control (MPC), or sometimes referred to as model-based predictive control. The advent of MPC as a control strategy has marked a significant leap in the field, providing a framework that offers an efficient and adaptable solution to many control problems, particularly in nonlinear systems.

In this thesis we consider MPC in the context of a complex nonlinear dynamic system, particularly by utilizing a piecewise affine (PWA) model to represent the underlying system dynamics and we will see how both of these aspects are pivotal for addressing the challenges posed by complex nonlinear dynamic systems.

2 State of art

2.1 Piecewise affine dynamical systems

Complex nonlinear dynamic systems abound in various fields, from industrial processes and robotics to environmental monitoring and autonomous vehicles. These systems exhibit intricate behaviors that make their accurate modeling and control a formidable undertaking. Unlike linear systems, nonlinear systems do not adhere to the principles of superposition and proportionality, rendering classical control techniques often ineffective. Their responses are characterized by nonlinearities, discontinuities, and non-convexities, which defy straightforward mathematical representations. A fundamental and historical challenge of control theory is the one of finding systematic design and analysis methods that can apply with a moderate effort to virtually any dynamical system. Methods such as Reinforcement Learning (RL) have the potential of dealing generically, in principle, with very complex dynamical systems, but they suffer from a lack of theoretical guarantees in term for example, of closed-loop performance and stability. These shortcomings prompt the necessity of developing nonlinear control techniques that would exploit as efficiently as possible the deep experience acquired from decades of theory and practice of linear systems. One promising way to achieve such an objective is to work on system models which are structurally simple enough to apprehend and yet able to capture the essential behavior of the nonlinear system. The classes of PieceWise Affine (PWA) models or blended PWA (also called multi-models) seem particularly appealing for establishing a bridge from the rich legacy of linear systems theory to nonlinear systems study. A PWA model represents a dynamic system in which the system's behavior is approximated by a series of linear subsystems, each of which is active in a specific region of the state space. In other words, the overall system is 'partitioned' into different regions, and in each region, the behavior is approximated by a linear model. Piecewise affine models, for this reason, offer a unique advantage in representing complex nonlinear systems. The PWA model can approximate the system dynamics with high fidelity in the presence of nonlinearity and discontinuity. This property makes PWA models an attractive choice when dealing with systems that exhibit abrupt changes or nonlinear phenomena. This class of models is known to have a universal approximation capability. In these years, many reaserchers are focusing on PWA models and their way of being controlled. The research is driven by PWA's demonstrated capability to effectively model and describe a wide range of practical physical systems that exhibit nonlinear characteristics. These nonlinear features include, but are not limited to, dead zones, saturation, hysteresis, and other complex behaviors. PWA systems offer a flexible framework that allows for the representation of intricate dynamics within different regions or modes, making them particularly suitable for capturing the inherent complexities found in real-world systems. As a result, researchers find PWA models to be a powerful and versatile tool for understanding and analyzing systems where traditional linear models may fall short in providing accurate representations. In practical applications, PWA systems belong to the class of hybrid systems and are defined by a polyhedral partition of the system state space. This means that the state space is divided into distinct polyhedral regions, each associated with a specific affine dynamics or model. This partitioning allows for a piecewise representation of the system's behavior, where different modes or regimes govern the dynamics based on the region in which the system currently resides. Such a characterization enables the modeling of complex systems with varying behaviors, making PWA systems valuable in the analysis and control of real-world processes exhibiting nonlinear and hybrid features. During the past few decades, PWA systems have drawn extensively attention and many elegant results on the problems of stability analysis like [44], where a Lyapunov matrix polynomial continous and time-dependent is constructed and used for the e robustness analysis of system, or control context [42] regarding periodic piecewise linear systems. Other optimum results are obtained in estimation [52], where the guaranteed cost control problem with the optimization of two performance indexes is investigated for a class of continuous-time periodic piecewise linear time-delay systems, and identification [13] proposing an online identification framework for the continuous-time PWA models in state-space form, with an arbitrary number of subsystems and the unknown partitions. It has been shown that many general smooth nonlinear systems can be approximated by PWA systems to any degree of accuracy [54]. In tackling the control challenges posed by PWA systems, numerous model-based results have been documented. Notably, some of these results include the use of piecewise Lyapunov functions for system stability analysis [36] and [20]. Most of the model-based control strategies assumed that the models of PWA systems are know in advance. However, in some cases, system models may not be known a priori and must be identified from measured data using identification methods. In practical terms, the control problem for PWA systems is typically divided into two distinct steps: first, the identification of the model, and second, the design of a model-based controller. An alternative approach that circumvents the two-step design procedure of model-based control methods is the data-driven control, explored better in [48], where a data-driven controller is designed to guarantee the asymptotic stability of discrete-time PWA systems in both cases of known and unknown state space partition information.

More and more research endeavors are employing machine learning techniques for the construction of PWA systems [1] and [21], using respectively Clustering and Bayesian approaches to identify Hybrid Systems. This growing trend reflects a recognition of the potential benefits and capabilities that machine learning can bring to the development and understanding of PWA models. In this context, machine learning serves as a valuable tool for system identification and control, offering an alternative or complementary approach to traditional model-based methods. The appeal of machine learning in the construction of PWA systems lies in its ability to adapt to complex and nonlinear dynamics without requiring a detailed understanding of the underlying physics. This adaptability is especially advantageous when dealing with systems characterized by uncertainties, disturbances, or varying operating conditions. The integration of machine learning techniques into PWA modeling not only streamlines the modeling process but also opens up new avenues for addressing control challenges in diverse and dynamic real-world applications. Others use ML to compute a weighting scheme for PWA systems. In [31] the weights are predicted by a deep neural network trained online and combined with multiple linear PID controllers for different operating points.
Resuming, the motivation behind studying piecewise affine dynamical systems stems mainly for two reasons. On one hand, they represent the most straightforward nonlinear extensions of linear systems. On the other hand, these systems can effectively approximate piecewise smooth nonlinear systems with a desired level of accuracy, much like how linear systems are employed to locally approximate smooth nonlinear systems.

2.2 MPC controller

Over the last three decades, there has been a significant surge in the application of control techniques based on dynamic optimization, resulting in enhanced system performance [2]. These applications span various domains, such as maximizing process output or minimizing energy consumption and emissions. The fundamental principle behind these techniques involves employing a mathematical model, typically represented by differential equations describing the process to be controlled. This model is utilized to predict the future behavior of the system and calculate optimized control actions. In some instances, this optimization can be carried out once before the runtime of the process, leading to the development of an offline controller. However, the presence of unknown or unmodeled disturbances often necessitates the use of a feedback controller. Such controllers repeatedly solve optimal control problems in real-time, meaning during the active runtime of the process. This approach is commonly referred to as Model Predictive Control (MPC), emphasizing the real-time, predictive nature of the control strategy in response to dynamic system conditions. Model Predictive Control offers several advantages compared to traditional control approaches. One of its key features is the direct specification of the control objective and desired constraints on the process behavior within an optimal control problem. This eliminates the need for heuristics in controller design and simplifies the tuning process. Mathematical optimization techniques are then employed to derive control actions that represent optimal solutions to these specified control problems.

Furthermore, MPC formulations can incorporate predictive information, enabling the controller to proactively respond to anticipated changes in the system. Additionally, MPC naturally accommodates processes with multiple inputs or outputs, as it can conceptually be applied to dynamic models of any dimension.

However, these advantages come with a trade-off—optimal control problems must be solved in real-time, possibly on embedded controller hardware. This challenge becomes particularly pronounced when the dynamics of the controlled process are fast, necessitating a controller capable of providing feedback at high sampling rates, as is often the case in applications like mechanical or automotive systems. What makes solving MPC problems challenging is the fact that if the model features nonlinear dynamics, the resulting optimisation problem typically becomes nonconvex. Thus, the optimal solution might not be unique and, moreover, many sub-optimal local solutions might exist. This circumstance introduces complications to both the solution procedure, leading to increased computational load, and the theoretical analysis of the closed-loop behavior of the process. Secondly, even when a linear dynamic model results in a convex problem with a unique optimal solution, solving it reliably within short sampling times is still computationally demanding, especially for complex systems. Apart from the computational challenges, considerable effort has been dedicated to exploring the theoretical properties of MPC algorithms.

2.3 Outline of the thesis

The application of MPC to nonlinear systems can present significant computational challenges, especially when real-time processing is essential. The computational burden is further compounded when considering the use of Piecewise Affine (PWA) models, which accurately capture the underlying nonlinear dynamics. While PWAs offer a promising avenue for modeling complex systems, the associated mixed-integer programming problem, a common consequence of employing PWA models in MPC, introduces computational complexities, with the problem being NP-complete. This intricacy not only impacts the efficiency of the MPC algorithm but also poses challenges for real-time applications, particularly when dealing with fast-paced processes, as observed in various mechanical and automotive systems. Addressing these challenges becomes pivotal for unlocking the full potential of MPC in controlling nonlinear and dynamic processes. In this thesis, we delve into the nuanced interplay between MPC, nonlinear systems, and PWA models, exploring strategies to enhance computational efficiency while maintaining the robustness of the control framework.

3 PWA model

The piecewise affine system can be seen as a collection of affine systems each operating within its own designated region. These regions are determined by dividing the combined state space and input space into segments. The behavior of the system is governed by the region in which the state and input vectors reside at a given time. Consequently, the dynamics is controlled by a switch when the state-input vector transitions from one region to another.

The motivation for exploring PWA dynamical systems primarily arises because these systems serve as the most direct nonlinear extensions of linear systems. Additionally, they offer the capability to approximate piecewise smooth nonlinear systems with a high degree of accuracy, such as the way linear systems are employed for locally approximating smooth nonlinear systems.

3.1 Mathematical Formulation of PWA Models

Understanding the foundations and inherent characteristics of PWA models is crucial, for implementing them in control systems. This section aims to provide an exploration of how PWA model are formulated explaining their fundamental principles and the key parameters that influence their behavior. By delving into the complexities of PWA models our goal is to offer an understanding that enables practitioners to utilize these models in various control applications.

PWA models fall under a category of representations that intricately divide the state space into defined regions. Each region corresponds to a model that captures localized insights, into the systems dynamics. Within these regions the systems behavior is represented by a set of linear equations pieced together to create an approximation that captures the nuanced intricacies of the nonlinear system dynamics. This approach allows for an region specific characterization of the systems behavior resulting in a precise and accurate representation.

Mathematically speaking a PWA model can be expressed as a nonlinear autonomous dynamical system $\Sigma : U \to Y$ with a state space representation given by:

$$y = \Sigma(u) = \begin{cases} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= h(x(t), u(t)) \\ x(0) &= x_0 \end{cases} \tag{3.1}$$

where $x(t) \in X \subseteq \mathbb{R}^n$ is the state, $u \in U$ is the input taking values in $U = \mathbb{R}^{n_u}$, $y \in Y$ is the output taking values in $Y = \mathbb{R}^{n_y}$ and x_0 is the initial condition. The functions $f : \mathbb{R}^n \times \mathbb{R}^{n_u} \to \mathbb{R}^n$ and $h : \mathbb{R}^n \times \mathbb{R}^{n_u} \to \mathbb{R}^y$ are called the drift map and output map, respectively.

Piecewise-affine systems represent a class of nonlinear systems wherein the evolution of the state is dictated by a collection of affine equations, each applicable within a distinct region of the state space. This characterization results in a system described by a set of piecewise differential equations. The representation of these systems is as follows:

$$\begin{cases} \dot{x}(t) &= A_i x(t) + B_i u(t) + a_i \\ y(t) &= C_i x(t) + D_i u(t) + c_i \qquad for\ x(t) \in X_i \\ x(0) &= x_0 \end{cases} \tag{3.2}$$

where $A_i \in \mathbb{R}^{n\times n}, a_i \in \mathbb{R}^n, B_i \in \mathbb{R}^{n\times n_u}, C_i \in \mathbb{R}^{n_y\times n}, c_i \in \mathbb{R}^{n_y}$ and $D_i \in \mathbb{R}^{n_y\times n_u}$, for $i \in I := \{1, ..., N\}$. We shall denote $I_0 \subseteq I$ the set containing all i such that $0 \in X_i$. The regions X_i, for $i \in I$, are closed convex polyhedral sets, and may be unbounded. Each face of the polyhedron X_i is in a hyperplane that divides X into two regions. Let

$$G_{i,k} := \{x \in X \mid G_{i,k}x + g_{i,k} \geqslant 0\} \tag{3.3}$$

be a half-plane defined by the k-th face of the polyhedron. Thus, G_i and g_i work as cell identifiers for cell X_i. If x_0 lies in the interior of a cell, this i is unique, and we can recall the appropriate system matrices to evaluate the model (3.1). If x_0 lies on a cell boundary, there are several i that satisfies the vector inequality and the right-hand side may not be uniquely defined. This is the case for non-smooth systems. The region X_i is then characterized by the intersection of all $G_{i,k}$, i.e

$$X_i := \bigcap_{\text{k}} G_{i,k} = \{x \in X \mid G_i x + g_i \geqslant 0\} \tag{3.4}$$

where

$$G_i := \begin{bmatrix} G_{i,1} \\ \vdots \\ G_{i,l_i} \end{bmatrix} \quad g_i := \begin{bmatrix} g_{i,1} \\ \vdots \\ g_{i,l_i} \end{bmatrix} \tag{3.5}$$

and l_i is the number of faces of X_i. The sign $\geqslant$ defines that each value of the vector $G_i x + g_i$ must be positive. The regions X_i have non-empty and pairwise disjoint interiors and are such that $\cup_{i\in I} X_i = X$. Then, $\{X_i\}_{i\in I}$ denotes a finite partition of

X. From the geometry of X_i, the intersection $X_i \cap X_j$ between two different regions is always present in a hyperplane. Let us define by $E_{ij}^T \in \mathbb{R}^n$ and $e_{ij} \in \mathbb{R}$ the vector and the scalar such that

$$X_i \cap X_j \subseteq \{x \in X \mid E_{ij}x + e_{ij} \geqslant 0\}. \tag{3.6}$$

In the next figure is showed the polyhedral partition:

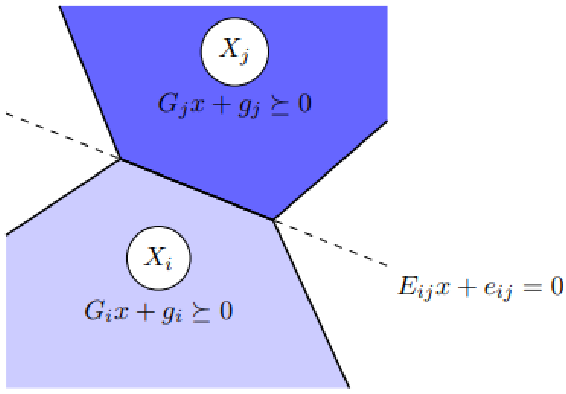

Figure 3.1: Polytopic partition of the state space

Only one model at each sampling time is employed, to reduce computational costs as well as the difficulty of including uncertainty into the optimization problem. Piecewise-affine systems emerge organically when addressing static piecewise-affine nonlinearities like saturations, relays, and dead-zones. Not only do they serve as suitable representations for such nonlinear elements, but they also function as effective approximations for more intricate nonlinear systems, including those featuring smooth separable nonlinearities. Within the control community, these systems have garnered significant attention in recent years. One key factor contributing to this interest is the close resemblance of their description to that of Linear Time-Invariant (LTI) systems. This similarity enables the seamless adaptation of several results from classical control theory, facilitating the application of concepts such as Lyapunov stability and the computation of the L2-gain, of which we will talk later. This inherent connection to well-established control principles makes piecewise-affine systems a valuable and versatile tool for modeling and analyzing complex dynamic processes.

As a subclass of nonlinear systems, piecewise-affine systems exhibit a diverse range of behaviors that go beyond what is typically observed in Linear Time-Invariant (LTI) models. Notable features include the presence of multiple isolated equilibrium points [19], non-unique steady states [3], and the emergence of limit cycles [22], among others. This underscores the fact that, despite their somewhat "simple" description, piecewise-affine systems are inherently nonlinear. Their dynamic nature encompasses a rich variety of purely nonlinear behaviors, emphasizing their capacity to capture complex and diverse phenomena within the realm of dynamic systems. Due to the connection between the continuous dynamics inside each region of the state space partition and the switching of dynamics when the trajectory crosses a boundary, piecewise-affine systems can be considered as a special class of hybrid systems. One way to represent hybrid systems is a finite collection of $continuos\ dynamics\ f_i \in S$, with $f_i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}, \forall i \in S\ and\ S \in \mathbb{N}$, and a switching function $\sigma : \mathbb{R}^{\times}\mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow S$ that selects which subsystem is active at each step k.

3.2 Identification and Estimation Techniques for PWA Models

As systems become more complex and the demand, for high quality products increases the use of system modeling techniques has become widespread in fields. It is crucial to identify and estimate parameters within Piecewise Affine (PWA) models to implement them in control systems. This section explores the methodologies and techniques used to discern and estimate these parameters within the PWA framework. It provides insights into the challenges involved in this process. Outlines strategic approaches to overcome them. Recognizing the importance of this aspect in enhancing reliability and performance in control applications using PWA models.

There are two approaches to achieve nonlinear system modeling; one based on principle models and another based on data. With advancements in large scale data mining and processing technologies like networks and Gaussian process regression algorithms, a range of data driven identification methods have emerged. These methods are designed to construct models that approximate systems, especially when first principle models are not available. This reflects a shift towards utilizing data based techniques for system modeling due, to complexities involved with systems and strict quality requirements.

The introduction of the multi-model modeling approach is motivated by the recognition that, in contrast to a global model, employing multiple submodels allows for a more accurate prediction in localized regions where process characteristics undergo significant changes. This strategy acknowledges the inherent variability in complex systems and aims to enhance predictive accuracy by tailoring submodels to specific regions of the system where their representations are more adept at capturing nuanced dynamics. We introduce the piecewise affine (PWA) model owing to its capability for describing nonlinear dynamics and the universal approximation properties of the PWA map.

The process of identifying Piecewise Affine (PWA) models mainly involves two components; estimating the models parameters and dividing the state space into regions. Parameter estimation focuses on determining the values of the models parameters whereas region partitioning involves defining regions in the state space and associating each region with model parameters. These two essential steps together contribute to identifying PWA models making them useful, in various domains. To tackle data classification challenges several clustering algorithms, such as K means clustering and its derivative algorithm K means++ are used in system identification. Before using these algorithms users need to specify the number of clusters and initialize clustering centers, which greatly impact the efficiency and accuracy of the clustering process. Moreover some algorithms partition data into categories without incorporating feedback criteria therefore making parameter estimation less precise and potentially affecting classification accuracy. Therefore it is crucial to consider these factors for optimizing the performance of clustering algorithms, in system identification tasks.

It's essential to highlight that region partitioning is a crucial aspect intricately linked with parameter estimation. Given that nonlinear systems generate data that is linearly inseparable, certain classification methods, such as support vector machines, may not entirely delineate the valid regions of the Piecewise Affine (PWA) model[29]. In contrast, Softmax regression presents a viable solution by employing the Softmax function to represent a categorical distribution, effectively addressing the challenge of accurate region partitioning in the context of PWA modeling.

3.2.1 Problem statement

Consider a nonlinear system with input $u \in \mathbb{R}^m$ and output $y \in \mathbb{R}$. It is assumed that input-output data $(u(k), y(k)), k = 1, ..., \bar{N}$, have been collected. The output $y(k)$ can be described by a Nonlinear AutoRegressive eXogenous (NARX) model in the form

$$y(k) = f(x(k)) + \varepsilon(k) \tag{3.7}$$

where $k \in \mathbb{N}$ is the time index, $\epsilon(k) \in \mathbb{R}$ is the error term, Gaussian independent identically distributed random variables with zero mean and variance σ^2. The NARX model is a type of model used in the analysis and prediction of systems. It is a variation of the AutoRegressive eXogenous (ARX) model. With added elements. In a NARX model the current output not depends on values of output and input variables but also on past values of the output variable itself. This feature allows the model to capture patterns in the data effectively. In this section we will explore how a PWA model can naturally emerge from linearization at points. The first example we see is defined on Voronoi-type of polyhedral partition of the regression space [5] which can be obtained either from the equations of the nonlinear system, when available, or directly from input-output data generated by the nonlinear system. A Voronoi-type partition, or Voronoi diagram, is a concept in computational geometry that divides a space into regions based on proximity to a discrete set of points called generators or sites. In a Voronoi diagram, each region is associated with a specific generator and includes all points in space that are closer to that generator than to any other generators.

In equation (3.7) k is the time index and the vector of regressors $x(k)$ is defined as

$$x(k) = [y(k-1)\ y(k-2)\ \cdots y(k-n_a)\ \ u^T(k-1)\ u^T(k-2)\ \cdots u^T(k-n_b)]^T \tag{3.8}$$

where n_a and n_b are the model orders. In this case, the bounded polyhedron $\mathcal{X} \subset R^n, n = na + mn_b$ is referred to as *regressor set*.In case $n_a = 0$, the regressor becomes just

$$x(k) = [u'(k-1)\ u'(k-2)\ \cdots u'(k-n_b)]' \tag{3.9}$$

and the model (3.7) is then called a Nonlinear Finite Impulse Response (NFIR) model. As mentioned in the introduction, our focus lies in deriving a Piecewise Affine (PWA) model for the nonlinear system (3.7) directly from a set $(x(k), y(k))_{k=1}^{N}$ of experimental data generated by the nonlinear system. To achieve this objective, we will explore a specific category of PWA functions characterized by a Voronoi-type partition of $\mathcal{X}$. Indeed if f is continuously differentiable as discussed earlier, then by strategically selecting a finite number s of points c_i in $\mathcal{X}$, we can effectively approximate f by a PWA map $f_{PWA}\ \mathcal{X} \to \mathbb{R}$ defined by

$$f_{PWA}(x) = \begin{cases} a_1^T\ x\ + b_1 & if\ \ x \in \mathcal{X}_1 \\ \vdots & \vdots \\ a_s^T\ x\ + b_s & if\ \ x \in \mathcal{X}_s \end{cases} \tag{3.10}$$

where $(a_i, b_i) \in \mathbb{R}^n \times \mathbb{R},\ i = 1, \cdots, s$, are defined by

$$a_i = \nabla f(c_i), \quad b_i\ =\ f(c_i)\ -\ c_i^T \nabla f(c_i) \tag{3.11}$$

with the notation ∇f referring to the gradient of f with respect to x, and the sets $\mathcal{X}_i$ are given by

$$\mathcal{X}_i = \{x \in \mathcal{X} : \| x - c_i\|_2 \leqslant \| x - c_j\|_2,\ \forall j = 1, \cdots, s\} \tag{3.12}$$

The sets of all the $\mathcal{X}_i$ form the regression space $\mathcal{X}$: $\mathcal{X} = \bigcup_{i=1}^{s} \mathcal{X}_i$ and the interiors are pairwise disjoint $int(\mathcal{X}_i) \bigcap int(\mathcal{X}_j) = \varnothing$ for all $i \neq j$ with $int(\cdot)$ referring to the interior. Such a partition is known as Voronoi partition. The regions $\mathcal{X}_i$ are then called the Voronoi cells while the points c_i are termed the seeds or the generators of the partition. It's not difficult to demonstrate that each set $\mathcal{X}_i$ is a convex polyhedron, which is a set derived from the intersection of a finite number of half-spaces. It should be noted that the proposed method can also be applied to multiple-input multiple-output systems, and relevant expressions are in [1].

More precisely, we can define

$$H_i = [c_1 - c_i \ \cdots \ c_{i-1} - c_i \quad c_{i+1} - c_i \ \cdots \ c_s - c_i]^T \tag{3.13}$$

$$h_i = [\beta_{1,i} \ \cdots \ \beta_{i-1,i} \quad \beta_{i+1,i} \ \cdots \ \beta_{s,i}]^T \tag{3.14}$$

and we can write

$$\mathcal{X}_i = \{x \in \mathcal{X} : H_i x \leqslant h_i\} \tag{3.15}$$

where $\beta_{j,i} = (c_j^T c_j - c_i^T c_i)/2$. The following picture is an example of Voronoi partition in $\mathbb{R}^2$:

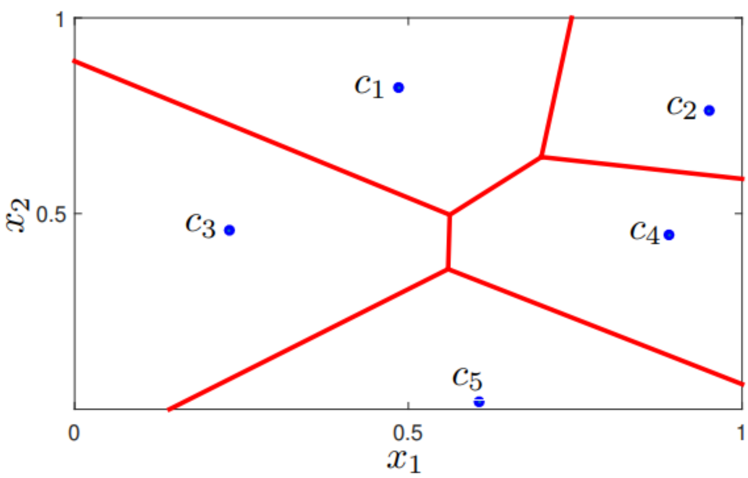

Figure 3.2: Example of Voronoi-type partition of in 5 polyhedral regions

It has a few useful qualities: (a) An understandable interpretation of the seeds c_i as operating points; (b) the affine functions connected to the regions can be viewed as linearization concerning the operational points of the nonlinear systems; (c) the estimate task only requires determining the parameters $(a_i, b_i, c_i)_{i=1}^s$ from observations.

To make things easier, we may write in the sequel

$$f_{pwa}(x) = \tilde{x}^T \theta_i \quad if \;\; x \in \mathcal{X}_i \tag{3.16}$$

with $\tilde{x} = [x^T \;\; 1]^T$ and $\theta_i = [a_i^T \;\; b_i]^T$. This is the compact formulation of the PWA model.

3.2.2 Estimation of PWA Models from data

From now on we view the PWA as a data-generating system. The goal of the estimation task is therefore to infer estimates of the parameters $\{\theta_i\}_{i=1}^s$ and $\{c_i\}_{i=1}^s$ from a finite collection of data $(x(k), y(k))_{k=1}^N$.

A reasonable approach to the estimation problem stated above would be to solve the optimization problem

$$\min_{\substack{\theta_1,\ldots,\theta_s \\ c_1,\ldots,c_s}} \sum_{i=1}^{s} \sum_{x(k)\in X_i(c_1,\ldots,c_s)} \left(y(k) - \theta_i^\top \tilde{x}\right)^2 \tag{3.17}$$

for $\{\theta_i\}_{i=1}^s$ and $\{c_i\}_{i=1}^s$ where $X_i(c_1, \ldots, c_s)$ is defined as (3.12). With this notation it's possible to see the dependence of the sets X_i on the seeds c_i's for more clarity concerning the decision variables. We begin by acknowledging that a straightforward and immediate algorithm for solving equation (3.17) would typically involve alternating between assigning the data points to the regions X_i based on their distances to the c_i points and computing the associated hyperplane parameters $\tilde{\theta}_i$. A basic implementation of this algorithm might entail initializing estimates of (c_i) by random sampling from $\mathbb{R}^n$. However, it is well-known that the performance of such a basic algorithm tends to be poor due to the nonconvex nature of the underlying optimization problem.

3.3 Stability and Performance Analysis

The research domain of piecewise-affine (PWA) systems, as introduced in [43], has garnered significant attention over the last 15 years. Various techniques have been recently explored for the stability analysis of these systems, primarily relying on the computation, facilitated through semi-definite programming, of common quadratic or piecewise-quadratic Lyapunov functions [14], [46], [26]. This focused investigation reflects the growing interest in developing robust stability analysis methods for PWA systems, contributing to the advancement of our understanding and control of complex dynamical systems exhibiting piecewise-affine characteristics. In addition to the approaches using common quadratic or piecewise-quadratic Lyapunov functions, alternative methods have emerged that rely on piecewise-polynomial Lyapunov functions [33], and on PWA Lyapunov functions [18].

In this section we will explore the tools for analyzing systems; dissipativity and Lyapunov stability. These fundamental concepts provide insights. Will help us establish practical conditions, for studying piecewise affine systems. By using dissipativity and Lyapunov stability our goal is to create a framework that enables an understanding and analysis of the complex dynamics present, in piecewise affine systems.

3.3.1 Dissipativity analysis

Input-output properties serve as a cornerstone for characterizing the interaction between the internal behavior of a dynamical system and its environment. This concept lies at the core of the dissipativity theory introduced by Willems [50], [51]. The concept of dissipativity is extremely valuable when studying the performance and resilience of systems. Essentially dissipativity means that a system absorbs energy from its surroundings than it generates. In a sense a dissipative system is characterized by its ability to handle an input power, which represents the rate of supply and a storage function that measures the stored energy. This setup ensures a dissipation of energy highlighting the importance of dissipativity, in comprehending the energy dynamics and behaviors of systems. These principles offer a framework for understanding how energy moves within a system and interacts with its environment providing valuable insights, into the systems dynamics and behavior.

We need to define the $supply\ rate$ as a function $\tilde{w}$ from $U \times Y$ into $\mathbb{R}$. This function is locally absolutely integrable and for all $t_1 \geqslant t_0 \geqslant 0$, it satisfies

$$\int_{t_0}^{t_1} \mid \tilde{w}(u(t), y(t)) \mid \ dt \ < \infty \tag{3.18}$$

The supply rate provides a view of how energy moves, between the system and its surroundings. When energy enters the system it can. Be stored internally. Released. To consider the impact of stored energy we introduce the notion of a storage function. This enables us to describe systems in the manner.

Definition 3.1. ***(Dissipative system)***

A $dynamical\ system$ $\Sigma\ :\ U \to Y$ is said to be dissipative with respect to the supply rate $\tilde{w}\ :\ U \times Y \to \mathbb{R}$ if there exists a nonnegative function $S\ :\ X \to \mathbb{R}_+$, called storage function, such that for all $t1, t0 \in \mathbb{R}_+, t_1 \geqslant t_0$, and $u \in U$,

$$S(x(t_1))\ -\ S(x(t_0))\ \leqslant \int_{t_0}^{t_1} \tilde{w}(u(t), y(t))\ dt \tag{3.19}$$

In the case where S is differentiable, the previous dissipation inequality (3.19) *can be written as*

$$\nabla S(x) \cdot f(x, u) - \tilde{w}(u, y) \leqslant 0 \tag{3.20}$$

for all $u \in U$.

The inequality (3.19) establishes that the generalized energy stored in the system at any future time t_1 cannot exceed the sum of the generalized energy at a specific time t_0

and the energy supplied during the interval between these two time instances. In other words, it implies that there can be no internal creation of "energy" within the system. This constraint encapsulates a fundamental principle, emphasizing the conservation or dissipation of energy in the dynamical behavior of the system.
The connection, between storage functions and Lyapunov functions is quite significant. In some cases the storage function itself can fulfill the requirements of a Lyapunov function, which ensures system stability. Additionally for linear systems with supply rates it has been shown that dissipativity implies the presence of a storage function. This interplay, between dissipativity Lyapunov functions and storage functions offers a framework to analyze and comprehend the stability properties of systems. For more details see [49]. In the next chapter we analyze the Lyapunov stability, explaining why it's fundamental for stability analysis.

3.3.2 Lyapunov stability

Lyapunov-based analysis methods carry out an important role for piecewise linear systems stability. The key component of such an analysis, namely methods for Lyapunov function computations, will be presented in this chapter. In more detail, our exploration will illustrate the process of computing piecewise quadratic and piecewise linear Lyapunov functions using convex optimization techniques. This approach underscores the practicality and efficiency of leveraging convex optimization to derive Lyapunov functions, offering a valuable methodology for stability analysis in piecewise-affine systems.

3.3.2.1 Asymptotical and Exponential Stability

Stability is an aspect when it comes to control systems. It basically means that a system doesn't behave explosively in any way. Initially we focus on stability, which ensures that not does the system avoid explosive behavior but it also settles down after an initial period of change. Specifically we pay attention to stability, which guarantees that the systems state converges, to its equilibrium point within a bound determined by a function of time. This detailed exploration of stability properties is essential for analyzing and designing control systems.

While there are stable systems where convergence may not be exponential the concept of exponential stability has particular significance in Lyapunov analysis for piecewise linear systems. In the case of linear systems exponential stability go hand in hand making it convenient for analysis. Additionally when dealing with systems an equilibrium point is locally exponentially stable if and only if its linearization around that point is exponentially stable. This connection highlights the importance of stability in analyzing both nonlinear systems providing a valuable tool, for assessing their stability.

In words the linearization provides all the information needed to determine whether an equilibrium is exponentially stable. Additionally in the case of a system global exponential stability is confirmed if and only if there is a Lyapunov function that supports this property.

As first, we have to define what is a Lyapunov function:

Definition 3.2. *V(x) is called a Lyapunov function for the system $\dot{x}(t) = f(x(t))$ if, in a defined space containing the origin, V(x) is positive definite and has continuos derivatives, and if its time derivative along the solutions of the system is negative semi-definite, i.e.* $V(x) = (\partial V/\partial x)f(x) \leqslant 0$.

This linkage between linearization and Lyapunov functions highlights their complementary roles in the analysis of stability properties, providing valuable insights into the behavior of complex dynamical systems. This makes exponential stability the appropriate concept in a piecewise linear approach for smooth nonlinear systems.

Below are described respectively the theorem of asymptotic and exponential stability, proof in [47]

Theorem 3.1. *System* (3.1) *is asymptotically stable if there exist a continuous function* $V : X \to \mathbb{R}_+$, *called a Lyapunov function, and* $\alpha_1, \alpha_2 > 0$, *such that*

$$\alpha_1(| x |) \leqslant V(x) \leqslant \alpha_2(| x |) \tag{3.21}$$

and along any trajectory x, *starting from* x_0, V *satisfies for any* $t \geqslant 0$

$$V(x(t)) - V(x(x_0)) \leqslant -\int_0^T \rho(| x(\tau) |)\, d\tau \tag{3.22}$$

with ρ *a positive definite function.*

Theorem 3.2. *If the conditions in* (3.1) *are satisfied with* $\alpha_i(r) = \sigma_i | r |^2$, *with* $\sigma_i > 0$ *for* $i \in \{1, 2\}$, *and* $\rho(r) = \sigma_3 | r |^2$, *with* $\sigma_3 > 0$, *then the system is exponentially stable.*

3.3.2.2 Quadratic Stability

Quadratic stability refers to the concept of stability that can be proven through the use of a Lyapunov function. The idea of quadratic stability has the origins with Lyapunov, who demonstrated that a quadratic Lyapunov function is both necessary and sufficient for ensuring stability, in linear systems. When we analyze systems quadratic Lyapunov functions are often employed as a tool. Additionally the study of stability heavily relies on the utilization of Lyapunov functions underscoring their significance in examining stability in systems.

In this context, the following result is central: Let us consider the linear time-invariant system possessing a minimal state space representation given by:

$$\begin{cases} \dot{x}(t) = A_i x(t) + B_i u(t) \\ y(t) = C_i x(t) + D_i u(t) \\ x(0) = x_0 \end{cases} \tag{3.23}$$

where $x(t) \in \mathbb{R}^n$ is the state, $u \in U$ is the input taking values in $U = \mathbb{R}^{n_u}$, $y \in Y$ is the output taking values in $Y = \mathbb{R}^{n_y}$ and x_0 is the initial condition. The associated transfer function between u and y is defined as $H(s) \; := \; C(sI \; - \; A)^{-1}B \; + \; D$ where s is the complex Laplace variable.

In the following sections we propose to analyze system (3.23) through the construction of quadratic storage and Lyapunov functions of the form:

$$S(x) \; = \; V(x) \; = \; x^T P x \tag{3.24}$$

Theorem 3.3. *Consider the system* (3.23). *If the convex optimization problem*

$$\begin{cases} P \; = \; P^T \; > \; 0 \\ A^T P \; + \; PA \; < \; 0 \end{cases} \tag{3.25}$$

has a solution, then the origin is globally exponentially stable. It's possible to see the proof in [47].

The primary advantage of the Theorem (3.3) lies in formulating the search for a quadratic Lyapunov function as a convex optimization problem. This approach significantly reduces the computational burden, making the imposition of extra constraints on the form (3.25) a relatively low-cost addition. Solving the multiple Lyapunov inequalities is not significantly more demanding than solving a single Lyapunov equation, as discussed in [11]. This characteristic renders quadratic stability a potent and efficient tool when applicable. The practicality of this approach is evidenced by numerous applications in systems with piecewise linear dynamics, as exemplified in works on fuzzy systems [45].

From a standpoint quadratic stability holds appeal. It allows for the analysis of systems to be framed as optimization problems, which can be efficiently solved using numerical computation techniques. This computational efficiency plays a role, in addressing challenges within control systems and related domains. By approaching system analysis through the lens of optimization quadratic stability methodologies become scalable and applicable to real world scenarios involving dynamical systems.

However there are considerations when applying stability to piecewise linear systems.

While quadratic Lyapunov functions offer attractiveness relying on them may result in overly conservative outcomes or unduly pessimistic assessments of the systems stability. Moreover conventional approaches to stability often do not leverage information about the state space partition during analysis. Additionally these approaches typically do not account for the inclusion of affine terms in the dynamics. This limitation restricts the capacity of stability analysis particularly when dealing with systems that possess affine dynamics or intricate state space structures. Thus it emphasizes the need, for methods that can encompass a wider range of system behaviors.

In the upcoming sections, we will introduce an approach that addresses the shortcomings associated with relying solely on quadratic Lyapunov functions. This method will embrace non-quadratic Lyapunov functions, incorporate information about the state space partition, and permit the inclusion of affine terms in the dynamics. This more flexible approach aims to enhance the accuracy and applicability of stability analysis, accommodating a broader class of systems with varying dynamics and structures.

To facilitate the search for Lyapunov functions, we will introduce a concise matrix parameterization for continuous piecewise quadratic Lyapunov functions. This compact representation aims to streamline the analysis process. Additionally, to mitigate excessive conservatism, we will leverage the observation that each affine dynamics is employed within a limited region of the state space.

We introduce Lyapunov functions that are continuous and piecewise quadratic of the form:

$$V_i(x) = x^T P_i x + 2q_i^T x_i + r_i = \begin{bmatrix} x \\ 1 \end{bmatrix}^T \begin{bmatrix} P_i & q_i \\ q_i^T & r_i \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} := \bar{x}^T \bar{P}_i \bar{x} \tag{3.26}$$

Given that the Lyapunov function and storage function are required to be zero at the origin, we set $r_i = 0$, for all $i \in I_0$. Furthermore, we choose q_i to be zero whenever $i \in I_0$, so that the piecewise-quadratic function behaves as a quadratic function close to the origin. We then obtain the following structure:

$$V(x) = \begin{cases} x^T P_i x & \text{for } x \in X_i, i \in I_0 \\ \begin{bmatrix} x \\ 1 \end{bmatrix}^T \bar{P}_i \begin{bmatrix} x \\ 1 \end{bmatrix} = x^T P_i x + 2q_i^T x + r_i & \text{for } x \in X_i, i \in I_1. \end{cases} \tag{3.27}$$

We now delve into the analysis of the exponential stability of piecewise-affine systems, employing piecewise-quadratic Lyapunov functions. More precisely, we have the following result.

Theorem 3.4. *(PIECEWISE QUADRATIC STABILITY)*

Consider the piecewise-affine system (3.1). *If there exist symmetric matrices $P_i \in \mathbb{S}^n$, vectors $q_i \in \mathbb{R}^n$, scalars $r_i \in \mathbb{R}$, symmetric matrices $U_i, W_i \in \mathbb{S}^{l_i}$ with nonnegative coefficients and zero diagonal and vectors $L_{ijkl} \in \mathbb{R}^{n+1}$ such that*

$$\begin{cases} P_i - E_i^\top U_i E_i \succ 0 \\ A_i^\top P_i + P_i A_i + E_i^\top W_i E_i \prec 0 \end{cases} \qquad for\ i \in I_0$$

$$\begin{cases} \begin{bmatrix} P_i - E_i^\top U_i E_i & q_i - E_i^\top U_i e_i \\ \bullet & r_i - e_i^\top U_i e_i \end{bmatrix} \succ 0 \\ \begin{bmatrix} A_i^\top P_i + P_i A_i + E_i^\top W_i E_i & P_i a_i + A_i^\top q_i + E_i^\top W_i e_i \\ \bullet & 2q_i^\top a_i + e_i^\top W_i e_i \end{bmatrix} \prec 0 \end{cases} \qquad for\ i \in I \setminus I_0 \tag{3.28}$$

$$\begin{bmatrix} P_i & q_i \\ \bullet & r_i \end{bmatrix} = \begin{bmatrix} P_j & q_j \\ \bullet & r_j \end{bmatrix} + L_{ij} \begin{bmatrix} E_{ij} & e_{ij} \end{bmatrix} + \begin{bmatrix} E_{ij} & e_{ij} \end{bmatrix}^\top L_{ij}^\top \quad for\ (i,j) \in I \times I \quad s.t.\ X_i \cap X_j \neq \varnothing \tag{3.29}$$

with $q_i = 0$ *and* $r_i = 0$ *for* $i \in I_0$*, are satisfied, then the piecewise-affine system is exponentially stable. Proof in Appendix B.1 of [47].*

Piecewise quadratic Lyapunov functions were used to analyze stability of continuous-time piecewise linear systems in [20]. Counterpart works for the discrete time case could be found in [15]. The main advantages of the piecewise quadratic Lyapunov function approach include, at first, that the criterion is much less conservative than the existence of a common quadratic Lyapunov function and the searching of piecewise quadratic Lyapunov function is reduced to a set of linear matrix inequalities (LMIs), which admits numerical verification.

3.4 Uncertainty Models

Uncertainty and robustness play pivotal roles in the modeling and analysis of feedback systems. The relationship between a system and its model inherently involves a gap, the significance of which is contingent on the resources allocated for its development. Moreover, a model may serve to represent a class of systems produced using real machinery, introducing variability in their output. Therefore, the concept of uncertainty arises between the model and the actual physical system. This uncertainty encapsulates the variations, discrepancies, and unpredictabilities that inherently exist in the real-world system, emphasizing the importance of accounting for and managing uncertainty in the modeling and analysis processes. In this circumstance it's very important to incorporate feedback into systems to ensure the fulfillment of system specifications even in the presence of variations in system components and external disturbances. Given the inherent mismatch between control design models and the actual system, accounting for this uncertainty becomes crucial to ensure that the conclusions drawn from the model are applicable in real-world scenarios.

To assess robustness, it becomes essential to delineate the sets of admissible uncertainties and disturbances. By specifying these sets, one can systematically evaluate and validate the performance and stability of feedback systems under varying conditions, providing a more comprehensive understanding of their resilience in the face of uncertainties and disturbances.

We can define two classes of uncertainties. The first class is systems

$$\dot{x} = f(x) \tag{3.30}$$

where the function f(x) is uncertain. This situation can arise when $f(x)$ is a piecewise linear approximation of some smooth function. When uncertainty stems from parameters that are either unknown or exhibit time variations, it is typically referred to as $parametric uncertainty$. The second class of uncertainty descriptions deals with systems on the form

$$\begin{aligned} \dot{x} &= f(x, y) \\ \dot{y} &= g(x, y) \end{aligned} \tag{3.31}$$

where $g(x, y)$ is uncertain or lacks a description with an appropriate structure, this form of uncertainty is typically termed $dynamic uncertainty$. This manifests when the variable y represents either an exogenous disturbance or a neglected component.

The conventional approach to address dynamic uncertainties involves incorporating norm-bounded uncertainties within a feedback interconnection as show in Figure 4.1.

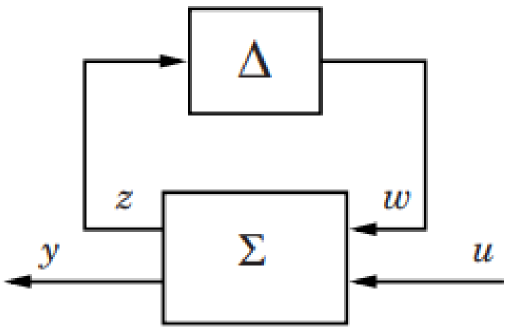

Figure 3.3: Piecewise linear system with uncertainty feedback

In robust control literature, the nominal system Σ is assumed to be linear time invariant, while system nonlinearities and uncertainties are confined to the uncertainty block Δ. In contrast to this, we will allow Σ to be piecewise linear. This flexibility enables us to make choices regarding whether the system nonlinearities should be explicitly modeled within the piecewise linear subsystem or treated as uncertainties in the Δ subsystem. This extra level of flexibility allows us to find a balance, between the complexity of computations and the cautiousness in our analysis. By choosing how to represent nonlinearities we can customize the model according to our requirements. Strike an optimal balance between computational efficiency and the degree of caution, in our analysis. The operator Δ that specify the feedback $u = \Delta y$ may be linear time varying or nonlinear, but is assumed to satisfy the dissipation inequality

$$\int_0^t \begin{bmatrix} y(s) \\ u(s) \end{bmatrix}^T M \begin{bmatrix} y(s) \\ u(s) \end{bmatrix} ds \geqslant 0 \quad \text{for all } t \geqslant 0 \tag{3.32}$$

for some real symmetric matrix M.

It's possible to include uncertainty (Δ) in a Piece-wise affine system, as done in [32] and define it, in the case of discrete time systems, for every submodel X_i, as

$$\begin{bmatrix} x(k+1) \\ z(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} A_i & B_i^1 & B_i^2 & f_i \\ C_i^1 & D_i^{11} & D_i^{21} & g_i^1 \\ C_i^2 & D_i^{12} & D_i^{22} & g_i^2 \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \\ u(k) \\ 1 \end{bmatrix} \tag{3.33}$$

where $w(k) = \Delta(z)(k)$.

Vectors $x(k) \in \mathbb{R}^{n_x}$, $u(k) \in \mathbb{R}^{n_u}$, $y(k) \in \mathbb{R}^{n_y}$ are respectively the state, input and measured output at time k $\in \mathbb{Z}_+$, while w represents the uncertainty and Δ is a causal nonlinear map. The vectors g_i and f_i represent an arbitrary set of constant real valued vectors.

Collection $\{X_i\}$ is the set of (not necessarily closed or bounded) polyhedra, with $\cup X_i = \mathbb{R}^n$, assuming that the regions i are not overlapping. The model i changes with respect to $x(k)$ and $u(k)$, and also with respect to time, k.

3.5 Example of global model construction through ML

The application of Machine Learning (ML), in constructing piecewise affine systems has become quite common. It is a way to handle the complexity of models. This approach proves useful in sectors where phenomena are dynamic and nonlinear requiring a more advanced method than traditional linear models. The decision to use Machine Learning for building piecewise affine systems is justified because it can learn patterns and relationships from data. This ability allows it to model parts of the system adapting accurately to variations without the need for a detailed description of the entire system. Various methods have been developed to implement this approach effectively. Neural networks, feedforward networks have shown their effectiveness, in capturing local nonlinear relationships. This enables the creation of piecewise affine models that can dynamically adjust to regions of the input domain.

We have talk about PWA identification in Section 3.2, but there is indeed a rich literature on PWA system identification. The large diversity of existing methods illustrates a clear surge of interest in these models in the recent years. Examples of ML methods include multiple local estimation and clustering through K-means [1], or recursive bayesian/incremental learning [21] [6], minimum partition into feasible sets (MIN PFS) [8], optimization via mixed integer programming approach [37], sparsity-inducing optimization techniques. A complete overview of systems identification is present in [30] and [16].

However in situations it becomes equally important to view the system as one cohesive global model. In these cases utilizing Machine Learning to create a model of the segmented system becomes crucial. This approach allows for the integration of acquired knowledge, from segmented models ensuring a comprehensive and unified understanding of the system. The implementation of fusion algorithms or recurrent neural networks can facilitate this integration making it easier to transition between regions of the system. It's quite clear that incorporating Machine Learning into constructing segmented linear systems provides an robust solution for capturing the complexity of phenomena. The decision to use either global models will depend on the requirements of the problem, at hand allowing for customization based on system dynamics and prediction or control needs.

In this section, we aim to elaborate on a significant example [31] that elucidates the construction of a global model through linear interpolation of several linear models within a piecewise affine system, facilitated by a weighted system. Consider a system that demonstrates piecewise affine (PWA) behavior, characterized by linear dynamics, in different parts of its input space. In order to create a model that captures the behavior of the system a technique based on linear interpolation is employed. This approach involves blending the outputs of linear models using a system. As we've discussed in this chapter each of these regions is defined as a linear model to accurately depict the dynamics. These local models act as elements capturing the subtleties of the system within their regions. To construct a model we introduce a set of weights that determine how much each local model contributes to the output. These weights are adaptable. Can change based on inputs ensuring flexibility in response to shifting system conditions. By employing interpolation, with these weighted values smooth transitions are achieved between regions resulting in a cohesive and uninterrupted representation of the entire system.

Mathematically, the discrete nonlinear model is of the form

$$\begin{cases} x(k+1) = f\big(x(k), u(k)\big) \\ y(k) = g\big(x(k), u(k)\big) \end{cases} \tag{3.34}$$

with $y \in \mathcal{R}^{n_y}$ the measured outputs, $u \in \mathcal{R}^{n_u}$ the measured inputs and the set of local linear models can be expressed, as already seen, as

$$\begin{cases} x_i(k+1) = f_i\big(x_i(k), u(k)\big) \\ y_i(k) = g_i\big(x_i(k), u(k)\big) \end{cases} \tag{3.35}$$

each tuned to approximate the global system (3.34) around an operating point.
The global output can be expressed as

$$\hat{y}(k) = \sum_{i=1}^{N} w_i(k) y_i(k) \tag{3.36}$$

Here, the weights w_i provide a time-varying adaptation. It is necessary to identify them online so that the outputs of the global model best match those of the plant. The problem then becomes one of finding an online estimate of the weights $w_i(k)$ such that the multi-model (3.35)-(3.36) approximates the initial system (3.34) on a wide range of operating conditions. Many different methods of weight estimation are available in literature.
The determination of the weight w_i in this context often follows a popular approach rooted in the Bayesian interpretation of plant model mismatch [28], [4]. The underlying concept involves employing a recursive Bayesian weighting scheme, which centers around estimating probabilities $p_i(k)$. Each of these probabilities corresponds to the likelihood of the $i - th$ model being valid at the discrete time step t_k. In this Bayesian framework, the weights are dynamically adjusted based on the evolving probabilities of model validity. The recursive nature of the scheme allows for continuous updates, ensuring that the weights accurately reflect the changing confidence in the validity of each model as the system operates over time. Another approach is to use the Distance bades estimator [17].
They decided to compute these weights with a deep neural network trained online and to combine them with multiple linear PID controllers used for different operating points of the system. The focus of this approach, which integrates machine learning with clas-

sical control techniques tailored for linear processes, is to develope a controller for a Waste Heat Recovery System (WHRS) installed on a Heavy-Duty (HD) truck engine. The primary objectives of this approach are to reduce fuel consumption and align with forthcoming pollutant emissions standards.

In the next rows we explain briefly how it works: At each time step t_k, the network takes as input not only the latest model errors $\epsilon(k)$ but also the sequences of errors $\big(\epsilon_1(k), ..., \epsilon_1(k-d), ..., \epsilon_N(k), ..., \epsilon_N(k-d)\big)$ along with $\big(u(k), ..., u(k-d)\big)$, providing in this way information about the local evolution of the error. It's possible to divide the algorithm in two section: Neural Network estimator and Online training. In the Neural Network estimator context, let $z(k) \in \mathcal{R}^{(n_y \times N + n_u)\times d}$ be the flattened representation of the local modeling errors on the previous d sampling times concatenated with the process inputs:

$$\begin{aligned} z(k) =& [\epsilon_1(k), ..., \epsilon_1(k-d), ..., \epsilon_N(k), ..., \epsilon_N(k-d), \\ & u(k), ..., u(k-d)]. \end{aligned} \tag{3.37}$$

In order to construct the network architecture from the input $z(k)$ to the output $C(z(k))$, they use a sequence of linear layers each followed by a local non-linearity of type rectified linear unit (ReLU):

$$\widetilde{C}(z(k)) = [\tilde{c}_1(z(k)), ..., \tilde{c}_N(z(k))] \tag{3.38}$$

$$= L^p\phi(L^{p-1}\phi(L^{p-2}...\phi(L^1 z(k))...)) \tag{3.39}$$

$$C(z(k)) = \Phi(\widetilde{C}(z(k))), \tag{3.40}$$

where the activation function $\phi(.) = \max(., 0)$ is called the element-wise ReLU function, and Φ is the softmax function:

$$\Phi(c_i(k)) = \frac{\exp(\tilde{c}_i(k))}{\sum_{j=1}^{N} \exp(\tilde{c}_j(k))}, \tag{3.41}$$

which is a mathematical function that converts a vector of numbers into a probability distribution. It guarantees that the predicted outputs W_i are assigned values between 0

and 1 with a sum equal to 1.

The outputs of the neural network is so computed and characterized by the predicted weights W_i for the different linear models:

$$W(k+1) = [w_1(k+1), ..., w_N(k+1)] = C(z(k)). \tag{3.42}$$

The next figure explain the representative scheme of the multi-model controller: It's

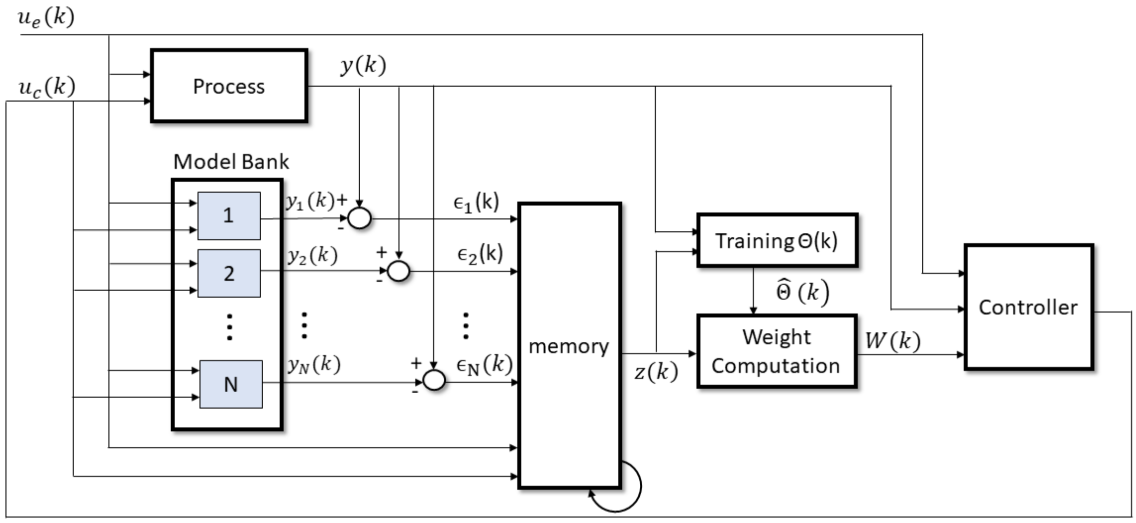

Figure 3.4: Proposed multi-model controller

possible to see that the weights W_i predicted by the network are used by the controller to generate the control signal towards the process and local-models. In return the neural network estimator receives process and models outputs that are used to improve the estimator parameters online.

A deep neural network (DNN) is a non-linear function defined by a parameter vector Θ, which can approximate any continuous function on compact subsets $\mathbb{R}^N$ given a sufficient number of parameters. This parameter is update to minimize a loss between measurements y and predicted $\hat{y}$, making the dependency on network parameters Θ explicit, is given by

$$\hat{y}(k) = \sum_{i=1}^{N} c_i(z_k, \Theta) y_i(k) \tag{3.43}$$

The loss function to minimize is the $\mathcal{L}_2$ loss given as

$$\mathcal{L}_2 = (y - \hat{y})^2. \tag{3.44}$$

and the algorithm used to iteratively update Θ is the stochastic gradient descent

$$\Theta \leftarrow \Theta - \alpha \nabla_\Theta \sum_{(y,\hat{y}) \in B} \mathcal{L}_2(y, \hat{y}), \tag{3.45}$$

where α is the learning rate and B is the current batch of a replay buffer, procedure that should reduce correlations in the data and it should smooth changes over data distribution when the controller changes operating points.

In the successive sections of [31] is possible to go further into details in the connection of this system of weights with the PID controller and in the application to an Organic Rankine Cycle simulator.

3.6 Conclusion

In this chapter we have explored the complexities of piecewise affine systems, which may seem simple but actually showcase a range of behaviors. We have given an explanation of the methodology used to analyze these systems. This analytical approach provides a foundation, for understanding the dynamics and stability of piecewise affine systems giving us insights, into their dynamic properties.

Expanding upon this discourse, we have extensively discussed the mathematical formulation of piecewise-affine systems, describing the aspects of identification and estimation, conducted thorough stability and performance analyses, and addressed uncertainty modeling. Additionally, we have presented a concrete example illustrating the construction of a global model, using a system of weights generated by neural networks, employed for multimodal control, showcasing the practical applicability and efficacy of the methods discussed throughout this chapter.

By exploring these various aspects of piecewise-affine systems and their associated methodologies, we have laid a solid foundation for understanding and leveraging their potential across a wide range of applications, further advancing the field of control theory and system dynamics.

4 MPC for PWA systems

The primary objective of this thesis is to establish a connection, between the effectiveness of MPC and the adaptability of PWA models in order to create a control policy that can adjust to systems. It will delve into the complexities of combining these two elements and highlight their potential in addressing the control challenges posed by systems.

Model Predictive Control (MPC) is a control strategy that incorporates analysis to calculate control actions in a manner. At each time step MPC plans a trajectory for a period. Then it applies the control action to the system. This process is repeated iteratively at subsequent time steps based on new measurements. By utilizing prediction MPC can account for uncertainties and changes, in system behavior enabling it to make informed decisions at each iteration.

Linear MPC has been widely used in a number of industries [35] due to its relative simplicity and robustness. Nonlinear MPC is more appropriate for handling complex processes with underlying nonlinear dynamics. Nevertheless, computations for nonlinear MPC may become prohibitively slow, making it difficult to handle the process model in real time. Piecewise affine (PWA) models [9]can accurately represent the underlying nonlinear dynamic system.

Several results about stability of PWA systems and MPC schemes for such systems can be found in the literature. In [9], one of the early results in ensuring closed-loop stability for Model Predictive Control applied to Piecewise Affine systems is achieved using a terminal equality constraint approach. This method, however, has limitations, as it requires a long prediction horizon to ensure feasibility of the optimal problem, leading to computationally demanding challenges. Another approach, presented in [27], employs a terminal set and a terminal cost to ensure stability of the MPC scheme for continuous PWA systems, particularly when the state equilibrium (the origin) is located within one of the polyhedra of the system's partition. In [24] this approach is extended to PWA systems where the origin lies at the intersection of some polyhedra of the partition.
In the first section of this chapter we will introduce the most important principles of PWA's control, passing then to an introduction of the MPC theory, defining the meaning of the optimal control problem and analyzing its behaviour with and without uncertainty. In the third section, we shift our focus to MPC for PWA systems, before moving to the last one, where we define the robust version of MPC for PWA.

4.1 Control of a PWA model

4.1.1 Introduction

In the field of control systems accurately representing nonlinear dynamics is incredibly important. Piecewise Affine (PWA) models are a tool, in this regard. They naturally come into play in applications especially when dealing with phenomena like zones, saturations, relays or hysteresis. Moreover piecewise affine systems are approximations for types of nonlinear systems. As a result they provide a framework for analyzing and designing systems. In years significant progress has been made in understanding the properties of piecewise affine systems. This includes advancements, in solution existence and uniqueness stability analysis and controller design.

These models provide a versatile and tractable approach to capturing the intricacies of system behavior within distinct regions of the state space. This introduction seeks to illuminate the significance of Piecewise Affine models in control applications, outlining their unique advantages and contributions in tackling the challenges posed by nonlinear systems.

In the realm of contemporary control systems, the pervasive presence of inherent nonlinearities in real-world processes poses a significant challenge. Conventional linear models, designed for simplicity, often prove inadequate in faithfully representing the intricate nonlinear behaviors inherent in these systems. The complexities introduced by nonlinear dynamics, including discontinuities and sudden changes in behavior, demand an innovative and adaptive approach to model formulation. Traditional control strategies, built upon linear foundations, are confronted with limitations when addressing the nuanced intricacies of nonlinear systems.

Although the field of nonlinear control techniques is making progress it remains challenging to develop an universally applicable theory and methodology. This challenge arises from the complexity and difficulty of controlling systems, which exceeds the challenges encountered in linear systems. Firstly accurately deriving a model, which's often crucial, for control systems is not an easy task. Secondly designing controllers for systems presents a problem even with an accurate model at hand. These complexities highlight the necessity, for approaches and methodologies to effectively address the intricacies of controlling systems.

As a result, there is a pressing need for novel methodologies that can effectively capture and navigate the complexities inherent in these dynamic processes. Piecewise Affine models present a compelling solution to this issue. By partitioning the state space into distinct polyhedral regions, each governed by a linear model, PWA models provide a flexible and adaptive representation of system dynamics. This inherent adaptability al-

lows them to capture the diverse nonlinear behaviors within different regions, making them particularly suited for systems characterized by varying modes of operation.
A key strength of PWA models lies in their ability to approximate complex nonlinearities through the judicious use of piecewise linear segments. This enables the representation of intricate system responses within each region, offering a computationally efficient means to navigate the challenges associated with nonlinear dynamics.
Among advantages already said, it should be noted that the use of PWA systems permits introducing thresholds and other discontinuities in a natural way that is not available in other algebraic approaches to nonlinear system theory.
Resuming, the piecewise linear approximation facilitates the translation of nonlinear complexities into a more manageable and analytically tractable framework and they constitute a robust modeling framework that can effectively capture the behavior of various nonlinear systems, including those exhibiting chaotic dynamics. As systems traverse different regions, characterized by varying linear models, PWA models adeptly capture the abrupt changes in behavior. This feature is particularly advantageous in control applications where the ability to navigate through disparate system modes is essential. The control strategies applied to these systems inherently involve switch control mechanisms. In this context, the control action transitions between different affine or linear subsystems based on specific switch conditions determined by the system states.
Recent advancements in methodologies for analyzing both piecewise linear and piecewise affine systems have garnered significant interest, particularly in addressing control challenges associated with such systems.

4.1.2 Fundamental Principles of PWA's Control

When multi-linear models are used to represent a nonlinear system, the mature linear control theory can be easily utilized to design linear controllers. Broadly speaking, methods employing multi-linear models can be categorized into two main types: soft switching and hard switching. In soft switching methods a global model is constructed by combining each local linear model with a different weight. The subsequent design involves creating a global controller for the overall model. Alternatively, in the soft switching approach, a linear controller is individually designed for each local model, and the aggregate of these controllers, each with different weights, is considered as the global controller. On the other hand, in hard switching methods [14,16,18], the approach involves designing a distinct local linear controller for each corresponding local linear model. Subsequently, these controllers are scheduled based on predefined rules to formulate a global controller.

While various linear control techniques can be applied in multi-linear model methods, both general soft switching and hard switching approaches may encounter challenges such as oscillations and slow adaptation when the system undergoes switches. The dynamic nature of switching between different local models poses difficulties in maintaining stability and responsiveness, leading to potential oscillations and delays in adaptation within the control system. However, general hard-switching and soft-switching multilinear model-based approaches may encounter difficulty in scheduling linear subsystems and thus may incur slow adaptation and/or oscillation when the system transits between different subsystems. To avoid slow adaptation and/or oscillation during subsystems controller scheduling and get an effective global control policy for the entire operating regions, the scheduling of multiple controllers should be considered in a unified framework. For instance, [25] proposed a value function-based approach for scheduling multiple controllers, aiming to derive an effective global control policy that covers the entire operating range.

Indeed, in the implementation of a multilinear model-based approach, the supervisor controller introduces multiple modes to the system. These modes are linked to different linear subsystems and are typically represented by discrete states, contributing to the hybrid character of the system. Given this hybrid nature, recent advancements in optimal control methods tailored for hybrid systems are well-suited for comprehensively capturing the dynamics of the subsystems, weakening slow adaptation and/or oscillation, and guaranteeing the overall control performance of nonlinear systems for the entire operating regions. The evolution of continuous states in response to discrete states and the switching of discrete states based on certain conditions contribute to the hybrid nature of the system. This interaction between continuous and discrete states allows the formulation of a hybrid model that amalgamates both aspects into a unified framework [12]. Consequently, designing a global optimal controller under this hybrid model enhances the overall control performance of the nonlinear system across its entire operating region [41].

The control imposed on such piecewise affine systems is naturally switch control, where the control action switches from an affine subsystem to another according to the switch conditions depending on the system states. A unified method for such switch control design has been established by employing piecewise continuous Lyapunov functions [20], [34], as seen in the previous chapter. However, LMI based approaches have the drawback of being conservative, the more conservative the larger the number of regions in the polyhedral partition of the state-space. In this thesis we formulate as a verification problem the issue of characterizing the stability of a feedback system composed of a PWA system and a constrained MPC controller, whose explicit solution can be found in PWA form [10].

4.2 Introduction of MPC

Model predictive control (MPC) is an advanced method of process control that is used to control a process while satisfying a set of constraints. It has emerged as a powerful paradigm in the control theory landscape, capable of addressing the main issues of complex nonlinear systems. MPC operates by iteratively optimizing control inputs over a finite prediction horizon based on a dynamic system model. It provides the flexibility to accommodate nonlinearities, constraints, and uncertainties, making it a compelling choice for tackling the complexities inherent in nonlinear systems. Model predictive controllers rely on dynamic models of the process, most often linear empirical models obtained by system identification. The main advantage of MPC is the fact that it allows the current timeslot to be optimized, while keeping future timeslots in account. This is achieved by optimizing a finite time-horizon, but only implementing the current timeslot and then optimizing again, repeatedly, thus differing from a linear–quadratic regulator (LQR). MPC, by applying at state x the first control in a finite sequence of control actions obtained by solving online a constrained, discrete-time, optimal control problem, traded arduous off-line computation of a control law $u = \kappa(x)$ for repeated on-line solution of a constrained dynamic optimal control problem; this trade-off was perfectly acceptable for control applications for which the optimal control problem could be solved within one sampling interval. The optimal control problem had, for computational reasons, to have a finite horizon so that the resultant controller did not guarantee stability but stability was achieved in most process applications since the horizon of the optimal control problem was normally sufficiently long.

4.2.1 Optimal control: problem formulation

In the optimal control literature the plant to be controlled is usually described in terms of state-space methods. The reason is that a lot of analysis will be done within a Lyapunov framework, which is most naturally performed in the state-space. The discrete-time systems with state x and control u described by:

$$x(k+1) = f(x(k), u(k)) \tag{4.1}$$

$$y(k) = h(x(k)) \tag{4.2}$$

where $x(k) \in \mathbb{R}^n, u(k) \in \mathbb{R}^m, y(k) \in \mathbb{R}^p$ denote the state, control input and measured output respectively. It is a standing assumption that the system is both controllable and observable. The controlled system is required to satisfy the state and control constraints $x \in \mathbb{X}$ and $u \in \mathbb{U}$ respectively where, usually $\mathbb{X}$ is a closed subset of $\mathbb{R}^n$ and $\mathbb{U}$ is a compact subset of $\mathbb{R}^m$; more generally, the state and control are required to satisfy $(x, u) \in \mathbb{Z} \subset \mathbb{R}^{n \times m}$.

We use **u** to define a control sequence and $\phi(k; x, \mathbf{u})$ to denote the state solution of (4.1) at step k when the initial state is x at step 0 and the control sequence **u** is applied. By definition $\phi(0; x, \mathbf{u}) := x$.

The control objective is to steer the state of the system in a finite number of steps N to a "safe" region X_f, that for instance might be the origin or any other set point, in a "best" way. In control systems, performance is typically quantified through a performance measure, often referred to as a cost function. The primary goal is to control the plant in a manner that minimizes this cost function. Beyond straightforward performance measures, objectives like tracking can be reformulated as the task of guiding the system to a designated safe set. This can be achieved through various means, such as extending the model appropriately, selecting suitable coordinates, or defining the cost function in a specific manner.

The cost criterion is written as a sum of *stage costs* $\ell(x, u)$ satisfying $\ell(0,0) = 0$. Performance can also be expressed with respect to the safe region X_f where we may have a cost criterion (terminal cost) V_f. By combining these two measures of performance, we can formulate a comprehensive cost function:

$$V_N(x, \mathbf{u}) = \sum_{i=0}^{N-1} \ell\left(x_i, u_i\right) + V_{\mathrm{f}}\left(x_N\right), \tag{4.3}$$

where $\mathbf{u} := [u_0^\top \ u_1^\top \dots u_{N-1}^\top]^\top$ and $x_i := \phi(i; x, \mathbf{u})$. For a given initial condition x, the set of feasible input sequences is defined by:

$$\Pi_N(x) := \left\{\mathbf{u} \in \mathbb{R}^{Nm} : x_i \in X, u_i \in U \quad \forall i \in \mathbb{N}_{[0,N-1]}, x_N \in X_{\mathrm{f}}\right\} \tag{4.4}$$

We denote with X_N the set of initial states for which a feasible input sequence exists, i.e.

$$X_N := \{x \in \mathbb{R}^n : \Pi_N(x) \neq \emptyset\} \tag{4.5}$$

Then, the *finite horizon optimal control problem* is formulated as follows:

$$\mathbb{P}_N(x): \quad V_N^0(x) := \inf_{\mathbf{u} \in \Pi_N(x)} V_N(x, \mathbf{u}) \tag{4.6}$$

The optimal control problem $\mathbb{P}_N(x)$ yields an optimal control sequence
$\mathbf{u}_N^0(x) \in arg\ min_{\mathbf{u} \in \Pi_N(x)} V_N(x, \mathbf{u})$:

$$\mathbf{u}_N^0(x) = \left[\left(u_0^0(x)\right)^T \left(u_1^0(x)\right)^T \cdots \left(u_{N-1}^0(x)\right)^T\right]^T. \tag{4.7}$$

It is called *optimizer* and it is designed to guide the system from any initial condition $x \in X_N$ to the safe region X_f in N steps. Crucially, this is achieved without violating the specified constraints and represents the most effective open-loop control strategy for ensuring system safety and performance. The function $V_N^0 : X_N \to \bar{\mathbb{R}}$ assigns to each state $x \in X_N$ the minimum value of the performance index and is named as *optimal value function*.

4.2.2 Model Predictive Control (MPC)

In the previous section we have seen the main ingredients of a constrained finite-horizon optimal control problem for a general nonlinear system (4.1). We can obtain an infinite-horizon controller by repeatedly solving the finite-horizon optimal control problem (4.6) where the current state of the plant is used as an initial state for the optimization. The procedure involves implementing only the first control sample from the computed optimal control sequence, and this entire process is repeated at the next step when new measurements of the state become available. This is referred to as the $receding\ horizon$ implementation of the controller and the resulting design method is called $model predictive control$ (MPC).

In the following figure is well explained the MPC operation and its evolution during the time:

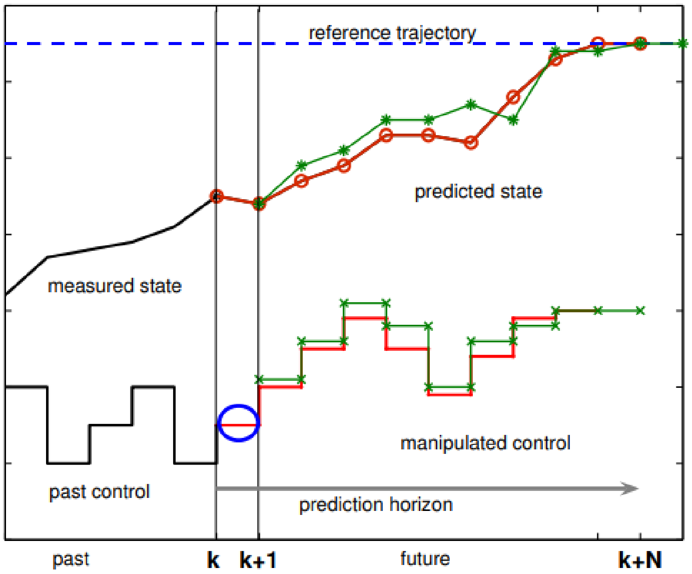

Figure 4.1: The model predictive control setup.

It's possible to visualize graphically the mathematical behaviour: given the event pair (k, x), i.e. $x(k) = x$, the optimization problem (4.6) is solved yielding the optimal

control sequence $\mathbf{u}_N^0(x)$. Only the first control $u_0^0(x)$ is applied to the system at step k. At the next step $k+1$ a new optimization problem is solved over a shifted horizon.
We can explicit the MPC law

$$\kappa_N(x) = u_0^0(x) \tag{4.8}$$

An intrinsic characteristic of MPC is that the optimization problem (4.6) is executed in open-loop, even if the MPC law (4.8) is a feedback law. Consequently, an open-loop control is employed in the prediction, while the actual controller of the plant (i.e., the MPC) operates in closed-loop form:

$$x(k+1) = f\left(x(k), \kappa_N(x(k))\right), y(k) = h(x(k)). \tag{4.9}$$

The main issues in MPC involve the feasibility of on-line optimization and closed-loop stability. These two issues are interconnected, and we will delve into more details on them in the following sections.

4.2.3 Robustness against uncertainty

Incorporating uncertainty into the mathematical representation of a system introduces the critical aspect of robustness. A controlled system is deemed robust when it maintains stability and meets performance specifications within a certain range of model variations and in the presence of specific disturbances. The exploration of stability and performance robustness is crucial to ensure the favorable behavior and safety of controlled systems. Various approaches are available in the literature for studying robustness. In this section we explain robustness using a min-max game framework, where the controller acts as the minimizing player, and the plant model along with the disturbance act as the maximizing players. Two main categories of robust optimal control strategies are explored: open-loop min-max control and feedback min-max control. In open-loop min-max control, a single control sequence is utilized to minimize the worst-case cost. On the other hand, in feedback min-max control, the worst-case cost is minimized over a sequence of feedback control laws. Each of these approaches will be briefly discussed. The robust control problem under consideration involves guiding an uncertain system, constrained by hard state and input limits, towards a safe or target set. Simultaneously, the objective is to minimize a worst-case performance function, addressing the challenges posed by uncertainties in the system.

Mathematically, the problem can be formulated based on the assumption that the plant is represented by a difference equation, taking the form:

$$x(k+1) = f(x(k), u(k), w(k)) \tag{4.10}$$

$$y(k) = h(x(k)) \tag{4.11}$$

where $w(k) \in \mathbb{R}^n$ is an additive disturbance that is assumed to lie in a compact subset $\mathbb{W}\ of\ \mathbb{R}^n$. Boundedness of w is required to permit satisfaction of the constraints in the optimal control problem for all possible realizations of the sequence $\mathbf{w} := [w_0^\top\ w_1^\top \dots w_{N-1}^\top]^\top$, which denote a realization of the disturbance over the prediction horizon N. Also, let $\phi(k; x, \mathbf{u}, \mathbf{w})$ denote the solution of (4.10) at step k when the initial state is x at step 0, the control sequence $\mathbf{u}$ and the disturbance sequence is $\mathbf{w}$. By definition $\phi(0; x, \mathbf{u}, \mathbf{w}) := x$. For a given initial state x, control sequence $\mathbf{u}$ and disturbance realization $\mathbf{w}$, the cost function $V_N(x, \mathbf{u}, \mathbf{w})$ is:

$$V_N(x, \mathbf{u}, \mathbf{w}) := \sum_{i=0}^{N-1} \ell\left(x_i, u_i\right) + V_{\mathrm{f}}\left(x_N\right) \tag{4.12}$$

where $x_i := \phi(i; x, \mathbf{u}, \mathbf{w})$ and thus $x_0 = x$. For each initial condition x we define the set of feasible open-loop input sequences $\mathbf{u}$:

$$\Pi_N^{\mathrm{ol}}(x) := \left\{\mathbf{u} : x_i \in X, u_i \in U \quad \forall i \in \mathbb{N}_{[0,N-1]}, x_N \in X_{\mathrm{f}}, \forall \mathbf{w} \in \mathcal{W}\right\} \tag{4.13}$$

where $\mathcal{W} := W^N$. Also, let X_N^{ol} denote the set of initial states for which a feasible input sequence exists:

$$X_N^{\mathrm{ol}} := \left\{x : \Pi_N^{\mathrm{ol}}(x) \neq \emptyset\right\}. \tag{4.14}$$

The $finite\ horizon\ open\ loop\ min-max$ control problem is defined

$$\mathbb{P}_N^{\text{ol}}(x): \quad V_N^{0,\text{ol}}(x) := \inf_{\mathbf{u}\in\Pi_N^{\text{ol}}(x)} \max_{\mathbf{w}\in\mathcal{W}} V_N(x,\mathbf{u},\mathbf{w}) \tag{4.15}$$

Typically V_N is a continuous function and since W is a compact set, it follows that the maximum is reached in and it is finite.

When an open-loop min-max control is applied in a receding horizon fashion we refer to this design method as open-loop min-max MPC. The open-loop formulation, while computationally advantageous, tends to be conservative as the set of feasible trajectories may significantly diverge from the origin [39]. Effective control in the presence of disturbances often necessitates optimization over feedback policies rather than open-loop input sequences. Feedback control helps prevent trajectories from diverging excessively and generally leads to improved performance compared to the open-loop case. This advantage arises from the increased degree of freedom in the optimal control problem associated with feedback strategies.

We now present the feedback min-max optimal control formulation. In this case we define the decision variable in the optimal control problem, for a given initial condition x as a control $policy$

$$\pi := (\mu_0(\cdot), \mu_1(\cdot), \ldots, \mu_{N-1}(\cdot)) \tag{4.16}$$

where each $\mu_i(\dot{})$ is a feedback law. Also, let $x_k = \phi(k; x, \pi, \mathbf{w})$ denote the solution of (4.10) at step k when the initial state is x at step 0, the control is determined by the policy π and the disturbance sequence is **w**. For each initial condition x we define the set of feasible policies π:

$$\Pi_N^{\text{fb}}(x) := \left\{\pi : \mu_i \in U, x_i \in X \forall i \in \mathbb{N}_{[0,N-1]}, x_N \in X_{\text{f}}, \forall \mathbf{w} \in \mathcal{W}\right\} \tag{4.17}$$

Also, let X_N^{fb} denote the set of initial states for which a feasible policy exists

$$X_N^{\text{fb}} := \left\{x : \Pi_N^{\text{fb}}(x) \neq \emptyset\right\}. \tag{4.18}$$

The $finite\ horizon\ feedback\ min - max$ control problem is defined as:

$$\mathbb{P}_N^{\text{fb}}(x): \quad V_N^{0,\text{fb}}(x) := \inf_{\pi \in \Pi_N^{\text{fb}}(x)} \max_{\mathbf{w} \in \mathcal{W}} V_N(x, \pi, \mathbf{w}) \tag{4.19}$$

The receding horizon implementation of a feedback min-max control is referred to as feedback min-max MPC. Given the assumption that the disturbance is bounded, the optimal controller's achievement is constrained to steering the state to a neighborhood of the origin, denoted as X_f. Subsequently, a local controller, represented as κ_f, is employed to sustain the state within X_f for any conceivable realizations of disturbances. As an alternative, in [7] it is proposed that a dynamic programming approach be used to obtain an explicit expression for the feedback MPC law.

The fundamental elements of finite-horizon optimal control and its receding horizon implementation, known as Model Predictive Control (MPC), have been outlined for general nonlinear systems. Additionally, various solutions to key challenges in MPC, including feasibility, robustness, and closed-loop stability, were discussed, relying on the utilization of a terminal set and a terminal cost approach.

4.3 MPC for PWA Systems

PWA models are incredibly useful when it comes to explaining systems, with behaviors that don't follow a straight line but rather change in specific ranges. To effectively implement MPC with these models it's essential to develop a model that accurately captures how the system moves between connected subsets and its dynamics. By using optimization techniques based on linear or quadratic programming algorithms we can smoothly handle any changes. Ensure a seamless transition between the various operating modes of the system. When adapting MPC to PWA models it becomes crucial to create a model that precisely represents the transitions, between interconnected subsets and their dynamics. A common approach involves utilizing a set of linear or affine equations that represent the system within each region of the state space defined by the PWA model.

The main challenge is determining which region the system currently belongs to, as the matrices A and B can vary discontinuously between regions. This requires a switching strategy that identifies the current region based on the system state and activates the corresponding equations. Practically implementing this approach might involve defining a set of constraints that capture transitions between regions, enabling the MPC controller to handle the discontinuities in the PWA system. The accuracy of the modeling and the robustness of the switching strategy will be crucial to ensure reliable and effective control of the PWA system.

Another important fact to consider is the discrete nature of PWA models. Since PWA models describe systems with nonlinear behaviors and discrete transitions between different regions, it's crucial to adapt optimization algorithms to consider these characteristics. Firstly, the presence of affine regions and discrete transitions requires special attention to formulating optimization constraints. Optimization algorithms should be able to handle the switching between different dynamics of the system, ensuring that control is consistent with transitions between regions.

Additionally, the selection of prediction and control horizons within MPC must be carefully considered, taking into account the discrete nature of the PWA model. Proper horizon choices can help capture transitions between regions and ensure optimal control performance. Incorporating a robust switching strategy that accurately identifies the current system region is also crucial. This involves a careful analysis of the activation conditions of different affine subsets, enabling the controller to promptly respond to changes in the system's behavior.

Let's assume we have a PWA model with N affine regions. The associated predictive model could be expressed as:

$$\begin{aligned} x(k+1) &= A_i x(k) + B_i u(k) \quad \text{if } x(k) \in X_i \\ y(k) &= C_i x(k) \end{aligned} \tag{4.20}$$

where i represents the corrisponding region, $x(k)$ is the state, $u(k)$ is the control, $y(k)$ is the output. The matrices $A_i \in \mathbb{R}^{n\times n}, B_i \in \mathbb{R}^{n\times m}, C_i \in \mathbb{R}^{p\times n}$ and $a_i \in \mathbb{R}^n$ are different for each region. Here, X_i is a polyhedral partition of the state space $\mathbb{R}^n$.

The system (4.20) is subject to hard input and output constraints:

$$\begin{aligned} X &= \left\{x \in \mathbb{R}^n : |y_j| \leqslant y_{j,\max}, \forall j \in \mathbb{N}_{[1,p]}\right\} \\ U &= \left\{u \in \mathbb{R}^m : |u_j| \leqslant u_{j,\max}, \forall j \in \mathbb{N}_{[1,m]}\right\}, \end{aligned} \tag{4.21}$$

The objective is to drive the system (4.20) from an arbitrary initial state x to the origin while adhering to the constraints on inputs and outputs (4.21), limiting also the finite-horizon quadratic cost function defined as

$$V_N(x, \mathbf{u}) = \sum_{i=0}^{N-1} \ell(x_i, u_i) + V_\mathrm{f}(x_N) \tag{4.22}$$

where the stage cost is given by the quadratic expression

$$\ell(x, u) = x^T Q x + u^T R u, \tag{4.23}$$

such that $Q = Q^T \succ 0, R = R^T \succ 0$ (positive definite matrices), $\mathbf{u} := [u_0^\top \; u_1^\top \ldots u_{N-1}^\top]^\top$ and $x_i := \phi(i; x, \mathbf{u})$. We assume that at each step k the state $x(k)$ is available (can be measured or estimated).

For each initial condition x we define the set of feasible control sequences $\mathbf{u}$:

$$\Pi_N(x) = \left\{\mathbf{u} : x_i \in X, u_i \in U \quad \forall i \in \mathbb{N}_{[0,N-1]}, x_N \in X_\mathrm{f}\right\} \tag{4.24}$$

where also a *terminal constraint* $x_N \in X_f$ is added. Let X_N denote the set of initial states for which a feasible input sequence exists:

$$X_N = \{x : \Pi_N(x) \neq \emptyset\} \tag{4.25}$$

The MPC law is then obtained as follows. At event (x, k) (i.e. the state of the PWA system at step k is x) the following optimal control problem is solved:

$$V_N^0(x) = \inf_{\mathbf{u}\in\Pi_N(x)} V_N(x, \mathbf{u}). \tag{4.26}$$

Let $\mathbf{u}_N^0(x) = \left[(u_0^0(x))^T \; (u_1^0(x))^T \cdots \left(u_{N-1}^0(x)\right)^T\right]^T$ define a minimizer of the optimization problem (4.26)

$$\mathbf{u}_N^0(x) \in \arg \min_{\mathbf{u} \in \Pi_N(x)} V_N(x, \mathbf{u}) \tag{4.27}$$

and let $\mathbf{x}^0 = \left[x^T \; (x_1^0)^T \cdots (x_N^0)^T\right]^T$ denote the optimal state trajectory (i.e. $x_i^0 = \phi(i; x, \mathbf{u}_N^0(x))$). We obtain an implicit MPC law:

$$\kappa_N(x) = u_0^0(x). \tag{4.28}$$

referred, in this case, only to PWA systems. This process is repeated at each sampling step, considering the new information about the state and continuously adjusting the control to the complex and discrete dynamics of the PWA model.

4.4 Robust MPC for PWA systems

There is always a difference between the PWA models used in the mathematical analysis and the actual physical system, so it is important to consider this uncertainty to ensure that the results will hold in reality. In such a scenario, the uncertainty description captures the variations in the parameters within each polyhedral partition, providing a more realistic representation of the actual system behavior. Robust control techniques, such as Robust Model Predictive Control (RMPC), can be employed to ensure stability and performance in the presence of these uncertainties. The challenge lies in designing controllers that can handle the variability in model parameters across different partitions and maintain robustness in the face of uncertainties. Various forms of model uncertainty descriptions have been proposed for PWA systems, including: (1) PWA system with additive disturbance [29], (2) local affine parameter-dependent PWA models [40], and (3) PWA systems with normbounded uncertainty [38].

Here, to define robust stability of an MPC controller for a Piecewise Affine system, we will consider a polytopic uncertain linear system description, presenting an uncertain structure of PWA systems that is a set of polytopic parameter varying models. To explain it in a better way, we say that in each partition of the PWA systems, the PWA subsystem is represented by a polytopic uncertainty. Indeed, the described uncertainty formulations provide a framework to embed more general nonlinear systems into the PWA framework.

Indeed, implementing robust Model Predictive Control (MPC) for Piecewise Affine (PWA) systems involves solving an online optimization problem that depends on the current state of the system. The computational complexity can increase with the size of the problem, and the presence of state switching in PWA systems further adds to the computational burden.

The model considered here is the (4.20)

$$x(k+1) = A_i x(k) + B_i u(k) \quad \text{if } x(k) \in X_i \tag{4.29}$$

where $i \in S := \{1, 2, ..., s\}$ represents the corrisponding region and s denotes the number of discrete modes. $x(k)$ is the state, $u(k)$ is the control. $A_i \in \mathbb{R}^{n\times n}, B_i \in \mathbb{R}^{n\times m}$. Here, X_i is a convex polyhedron containing the origin for all $i \in S$.

The state and input gains are uncertain and time-varying within each polyhedral partition of a Piecewise Affine system, it introduces additional complexity to the control problem. The uncertainty in these matrices implies that the system's dynamics can change over time, and the exact variations are not known in advance:

$$\begin{bmatrix} A_i(k) & B_i(k) \end{bmatrix} \in \Omega_i, \quad \forall k \geqslant 0, \tag{4.30}$$

where $\Omega_i := \mathrm{Co}\left\{\begin{bmatrix} A_i^1 & B_i^1 \end{bmatrix}, \begin{bmatrix} A_i^2 & B_i^2 \end{bmatrix}, \ldots, \begin{bmatrix} A_i^L & B_i^L \end{bmatrix}\right\}$, and $\begin{bmatrix} A_{il} & B_{il} \end{bmatrix} := \begin{bmatrix} A_i^l & B_i^l \end{bmatrix}$ represents a vertex of the polytope Ω_i, i.e., $\forall k$ there exist L_i nonnegative coefficients λ_i^l (with $l = 1, 2, \ldots, L_i$) such that

$$\sum_{l=1}^{L} \lambda_i^l = 1, \qquad [A_i(k) \quad B_i(k)] = \sum_{l=1}^{L} \lambda_i^l(k) \begin{bmatrix} A_{il} & B_{il} \end{bmatrix}. \tag{4.31}$$

MPC, as already explained, is a step-by-step optimization technique where new measurements are calculated at each step and a model of the process is used to predict future outputs of the systems. A cost function like (4.23) is minimized to compute the future control action.

In this section we want to transform the predictive control problem for the uncertain PWA system (4.20) to a set of LMI (Linear Matrix Inequalities) problems. The proposed approach suggests using a set of PWA Lyapunov functions, each corresponding to different sampling times or instances where the state may switch from one partition to another. This is an interesting strategy for dealing with closed-loop PWA systems. Consider a PWA quadratic function

$$V(x) = x^T P_i x \qquad \begin{pmatrix} x \\ u \end{pmatrix} \in \mathcal{X}_i \tag{4.32}$$

where $i \in S$. At sampling time k, $x(k|k)$ denotes the true measured state $x(k)$. In ordeer to obtain an upper bound on the robust performance objective, we let $V_\infty(x(\infty \mid k)) = 0$ because of $x(\infty \mid k) = 0$ and the following requirement is required (as explained in the Theorem (3.1) in section 3.3.2

$$\begin{aligned} & V(x(k+j+1 \mid k)) - V(x(k+j \mid k)) \\ & \quad < -\left[x(k+j \mid k)^T Q x(k+j \mid k) + u(k+j \mid k)^T R u(k+j \mid k)\right] \\ & \quad \forall \left[A_i(k+j) \quad B_i(k+j)\right] \in \Omega_i. \end{aligned} \tag{4.33}$$

Summing (4.33) for $j = 0 \to \infty$ we get the following inequality:

$$\max_{[A_i(k+j)\ B_i(k+j)] \in \Omega} J_\infty(k) < V(x(k \mid k)). \tag{4.34}$$

Therefore, the robust MPC algorithm, with the state-feedback controller of the form $u(k+j|k) = F_p x(k+j|k)$, with $F_p \in \mathbb{R}^{m \times n}$, can be denoted as follows. At each time step k, the state-feedback controller can be synthesized by minimizing the following problem:

$$\begin{aligned} &\min_{\gamma, P_i} \gamma \\ &\text{s.t. } x(k \mid k)^T P_i x(k \mid k) \leqslant \gamma \\ &\text{and (4.33),} \end{aligned} \tag{4.35}$$

where $\left[x^T(k \mid k) u^T(k \mid k)\right]^T \in \mathcal{X}_i$, and γ is a suitable nonnegative coefficient to be minimized. In the end, supposing that the switching sequence of the PWA system from one partition to another is known, the optimization problem (4.35) can be solved by the LMI already described in the Theorem (3.4) in section 3.3.2.

We have seen that the computational complexity can increase with the size of the problem, and the presence of state switching in PWA systems further adds to the computational burden. In [55] a new method to reduce the computational complexity is presented. They proposed an improved robust MPC for PWA systems, consisting in an online computation of the robust MPC when the state is outside of the region where the origin is included and in a sequence of asymptotically stable attraction domains constructed off-line once the state enters in the region including the origin. They proposed also an example to proof the off-line stabilizing state-feedback laws, demonstrating the good behaviour of the controller. In our opinion, in the example provided the number of models was only two and maybe increasing the number of models to 10 or more, the control system presented may not be optimal. A future challenge would be to test it on a more complex nonlinear system and assess its effectiveness. At that point, additional modifications may be necessary.

4.5 Conclusion

In this chapter, we have delved into the control of Piecewise Affine (PWA) models using Model Predictive Control (MPC), a powerful technique in control engineering. We began by introducing the fundamental principles of PWA control, highlighting the unique characteristics associated with controlling nonlinear and hybrid systems. Subsequently, we provided an overview of MPC, discussing its formulation as an optimal control problem and its application in real-time control of dynamic systems

Through our exploration of MPC for PWA systems, we have demonstrated the potential of this approach in achieving robust and adaptive control in complex and uncertain environments. By leveraging predictive models and optimizing control actions over a finite time horizon, MPC offers a systematic framework for addressing constraints and optimizing performance, even in the presence of disturbances and uncertainties.

Moreover, our discussion on robust MPC for PWA systems underscores the importance of considering uncertainty and disturbances in control design. By incorporating robustness-enhancing techniques such as constraint tightening and disturbance rejection, robust MPC ensures stable and reliable control performance in the face of varying operating conditions and disturbances.

Overall, this chapter has provided valuable insights into the theory and application of MPC for PWA systems, highlighting its effectiveness in addressing control challenges and achieving desired performance objectives. Moving forward, further research and development in this area could lead to advancements in control strategies for a wide range of nonlinear and hybrid systems, with potential applications across various industrial sectors and domains.

5 Experimental results

This chapter presents two cases of application of the Model Predictive Control (MPC) to a Piecewise Affine (PWA) system, as described in the previous chapters. The experiments conducted aim to assess the performance, stability, and feasibility of employing MPC in controlling such a dynamic system. The implementation of MPC on a PWA system holds significance in various engineering applications, particularly in fields where complex nonlinear dynamics need to be effectively managed and controlled.

5.1 Example I

The experiment focuses on a fluid level control process, representing a popular example for nonlinear application. The system consists of two interconnected cylindrical tanks R_1 and R_2 [53], as described in the following figure

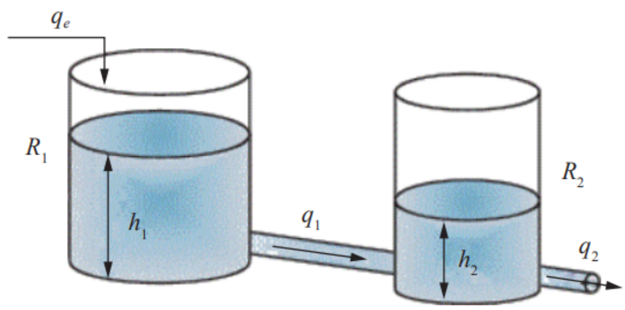

Figure 5.1: Level control process

The objective is to regulate the water level h_2 in tank R_2 by adjusting the input flow q_e. The dynamics of this process are governed by principles of Bernoulli's equation and the conservation of mass, resulting in the following system of differential equations:

$$\begin{aligned} \frac{\mathrm{d}h_1}{\mathrm{d}t} &= \frac{q_e}{S_1} - \frac{\alpha_1}{S_1}\sqrt{2g\left(h_1 - h_2\right)} = f_1\left(h_1, h_2, q_e\right) \\ \frac{\mathrm{d}h_2}{\mathrm{d}t} &= \frac{\alpha_1}{S_2}\sqrt{2g\left(h_1 - h_2\right)} - \frac{\alpha_2}{S_2}\sqrt{2gh_2} = f_2\left(h_1, h_2, q_\epsilon\right) \end{aligned} \tag{5.1}$$

where h_1 and h_2 are the water levels in R_1 and R_2, respectively, S_1 and S_2 are the areas of R_1 and R_2, α_1 is the area of the pipes linking R_1 to R_2, α_2 is the area of the outlet pipes of R_2, q_e is the incoming flow, q_1 is the flow between R_1 and R_2, q_2 is the outlet flow and g is the gravity.

The parameter's values of the level control process are chosen as: $\alpha_1 = 0.002\ \mathrm{m}^2, \alpha_2 = 0.002\ \mathrm{m}^2, S_1 = 0.25\ \mathrm{m}^2, S_2 = 0.1\ \mathrm{m}^2, l = 1.2\ \mathrm{m}, g = 9.81\ \mathrm{m/s}^2$.

The objective of the experiment is to construct a piecewise affine system and then control it using MPC. The focus of this thesis is on control, but first, I'll briefly explain the construction of the PWA system. The PWA system can be expressed, as seen in Section 3.2, as

$$y(k) = \begin{cases} \theta_1^T \bar{\varphi}(k) + e(k) & \text{if} \quad \varphi(k) \in H_1 \\ & \vdots \\ \theta_s^T \bar{\varphi}(k) + e(k) & \text{if} \quad \varphi(k) \in H_s \end{cases} \tag{5.2}$$

where

$$\varphi(k) = \begin{bmatrix} y(k-1) \ \dots \ y(k-n_a) & u(k-1) \ \dots \ u(k-n_b) \end{bmatrix}^\top \tag{5.3}$$

$$\theta_i = \begin{bmatrix} a_{i,1} \ \dots \ a_{i,n_a} & b_{i,1} \ \dots \ b_{i,n_b} & g_i \end{bmatrix}^\top \tag{5.4}$$

$$\bar{\varphi} = \begin{bmatrix} \varphi^T & 1 \end{bmatrix}^\top . \tag{5.5}$$

- $y(k) \in \mathbb{R}, u(k) \in \mathbb{R}, e(k) \in \mathbb{R}, s \in \mathbb{N}$ are respectively the output, the input, the additive noise and the number of sub-models.

- $\theta_i \in \mathbb{R}^{n_a+n_b+1}$ is the parameter vector of the i^{th} submodel having n_a and n_b as orders.

- $a_{i,j}$ and $b_{i,j}$ represent the coefficients of the i^{th} submodel while g_i represents an independent affine parameter of the i^{th} sub-model.

- $\varphi(k) \in \mathbb{R}^{n_a+n_b}$ is the regressor vector.

- $H_i \in \mathbb{R}^{n_a+n_b}$ is the polyhedral partition of the i^{th} submodel.

In the paper [53] the identification of the process is done throw the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm, which is a popular clustering algorithm used in machine learning. This computational technique enables the categorization of data into separate groups, relying on specific density criteria. In essence, it identifies dense clusters as distinct classes, with gaps of low density separating them. Furthermore, this approach effectively filters out anomalies during the classification process and has the capability to ascertain the quantity of classes present. In [23] the DBSCAN algorithm is more detailed.

They obtained parameter vectors $\theta_i = \{a_{i,1}, b_{i,1}, b_{i,2}\}_{i=1,2,3}$

$$\begin{aligned} \theta_1 &= \{0.7915, 2.2125, 5.2863\} \\ \theta_2 &= \{0.8877, 0.1246, 7.5378\} \\ \theta_3 &= \{0.9105, -0.7323, 8.4514\} \end{aligned} \tag{5.6}$$

and the regions $\{H_i\}_{i=1,2,3}$, computed throw support-vector machine (SVM) approach,are as follows:

$$
\begin{aligned}
H_1 &= \{\varphi \in \mathbb{R}^3; \quad [-4.989 - 0.105 - 0.103]^T \varphi(k) - 0.2528 \leqslant 0\} \\
H_2 &= \{\varphi \in \mathbb{R}^3; \quad [-4.5735 - 0.0207 - 0.0348]^T \varphi(k) + 0.1977 \leqslant 0 \\
&\qquad \text{and } [-4.989 - 0.105 - 0.103]^T \varphi(k) - 0.2528 > 0\} \\
H_3 &= \{\varphi \in \mathbb{R}^3; \quad [-4.5735 - 0.0207 - 0.0348]^T \varphi(k) + 0.1977 > 0\}
\end{aligned}
\tag{5.7}
$$

The result of their simulation is shown in the following figure:

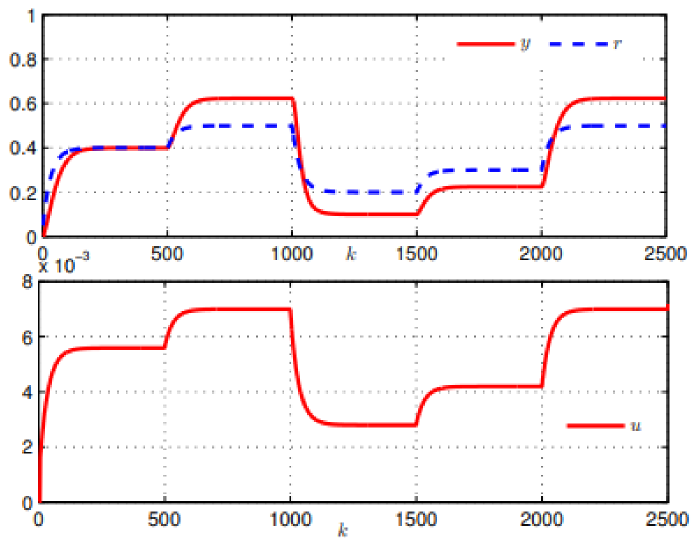

Figure 5.2: Paper [53] control results using Hybrid MPC

It's possible to see that the output and the reference trajectory don't have the same dynamics. In order to implement it and try to obtain a better result, we decided to use Matlab and Simulink.

The Simulink block scheme is

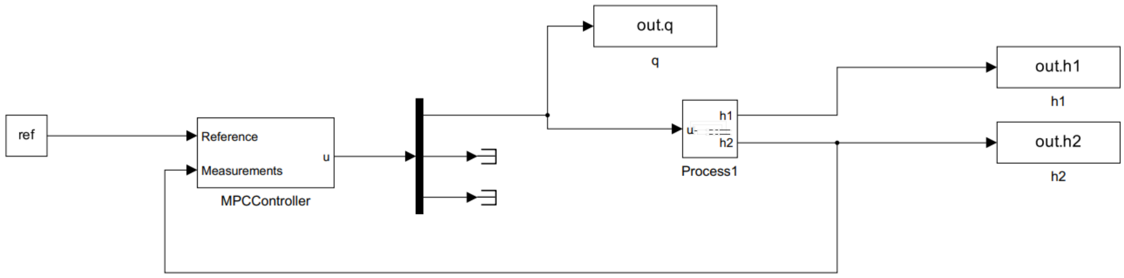

Figure 5.3: Simulink Model

where the process is defined by the two differential equations, each followed by the integrator and the saturation blocks, as shown in the following scheme

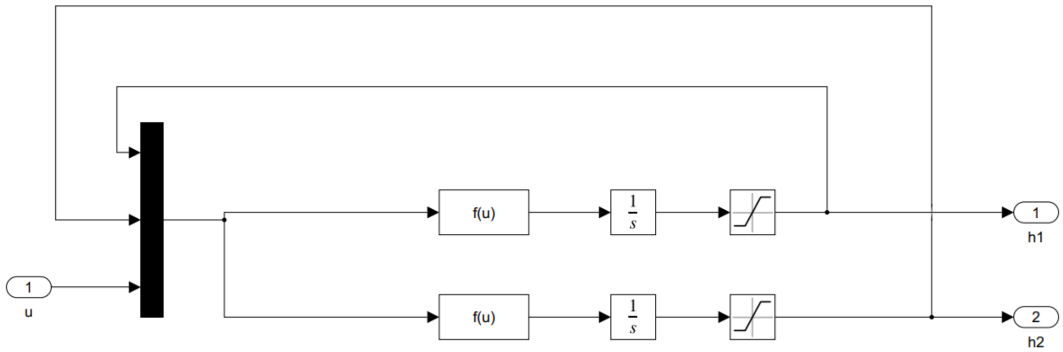

Figure 5.4: Process blocks scheme

The simulink model is iteratively launched and stopped N times, with updates to the controller conditions based on the linear submodel of the selected PWA system, determined by the switching conditions. The total number of controllers is thus 3. Once the submodel is defined, the respective θ is used to determine the system output $y(k)$ and it will directly influence the value of the cost function. The control signal outputted by the MPC then enters the process, from which the two variables h1 and h2 are measured, to be passed to MATLAB for the subsequent step. At each cycle, the controller's actions are adjusted based on the current operating conditions, ensuring adaptive and efficient control of the system dynamics.

We aimed to verify that it's possible to achieve a better outcome respect to Figure 5.2. Indeed, through our experimentation, we succeeded in aligning the output closely with the reference:

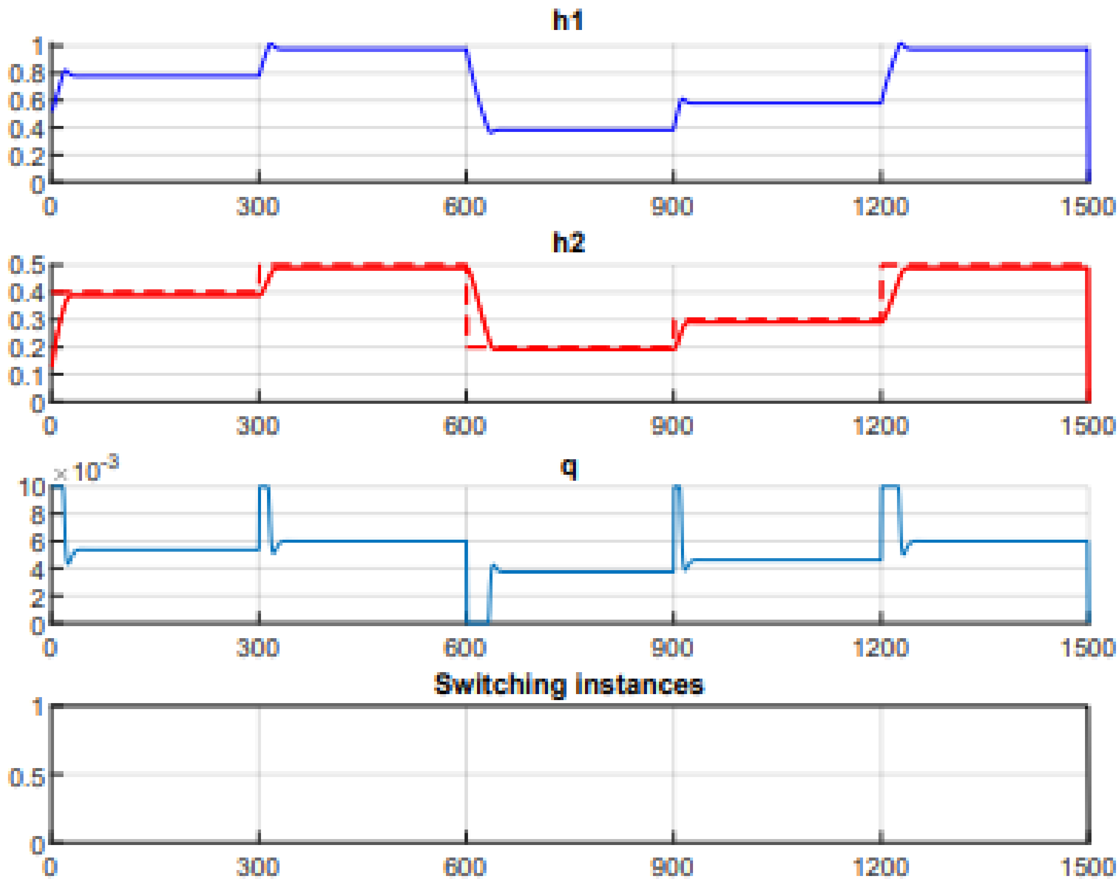

Figure 5.5: Plot with same reference of the paper

Figure (5.5) illustrates that utilizing MPC for such processes can be effective, as evidenced by the near perfect tracking of the reference by the output and by the absence of significant overshoots, while the rise time is optimal.

The only issue identified in the simulation (5.5) is the absence of switching between models. Initially, this was considered as a coding error, but, through further experimentation, it was revealed that the switching mechanism operates correctly for significantly smaller reference values as illustrated below:

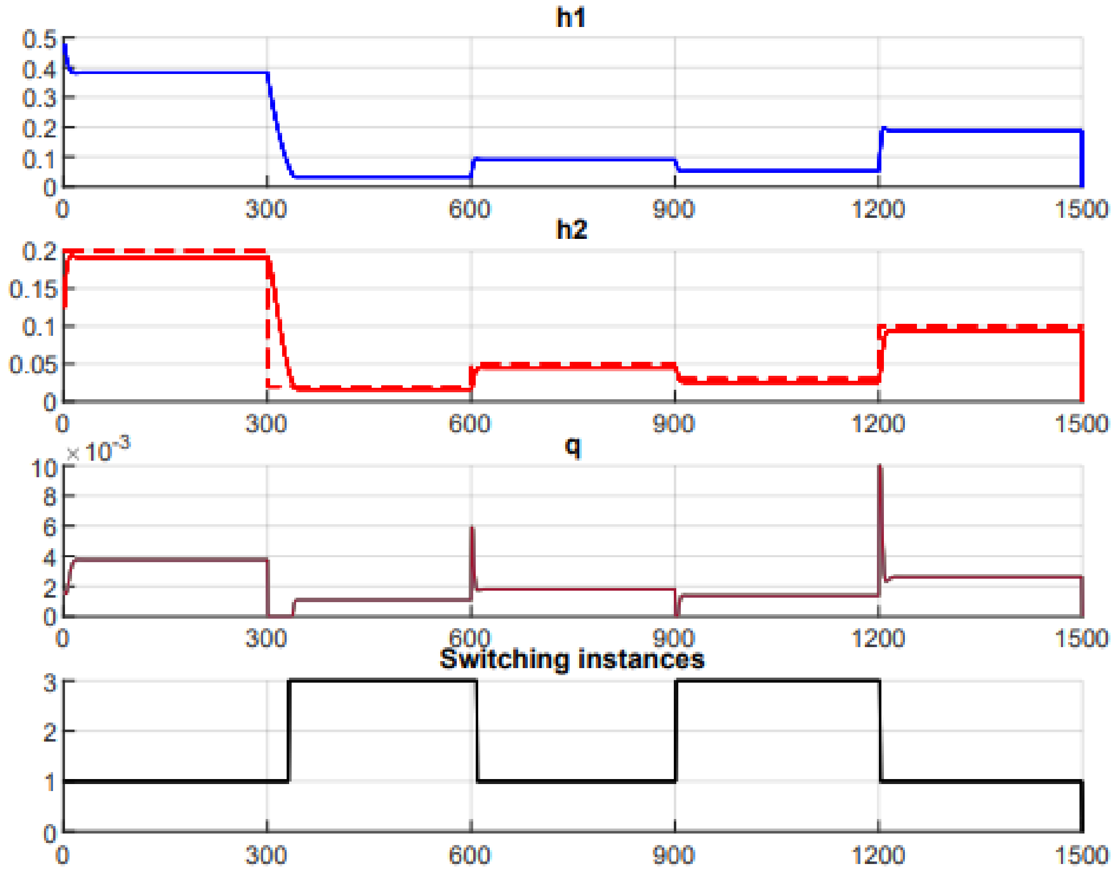

Figure 5.6: Plot with smaller reference

Here, it's possible to see the reference values, each for 300 steps, of 0.2, 0.02, 0.05, 0.03, and 0.1, respectively. It's observable that in this scenario, the switching between submodels works correctly, but only between models 1 and 3. The switching to model 2 never activates because the inequality condition H_2 in (5.7) is never met with these reference values, despite repeated checks in the code. It's worth noting that the paper does not specify the switching instants, so we cannot conclusively state whether, in their experiment, with our same reference values, switching occurs between all three submodels. This suggests that the switching logic of the paper may require adjustments to accommodate the reference values effectively. In order to do that, we modified the treshold value defined in (5.7) from $0.25\ to\ -2.1$ and from $-0.19\ to\ -1.2$. The plot obtained is the following:

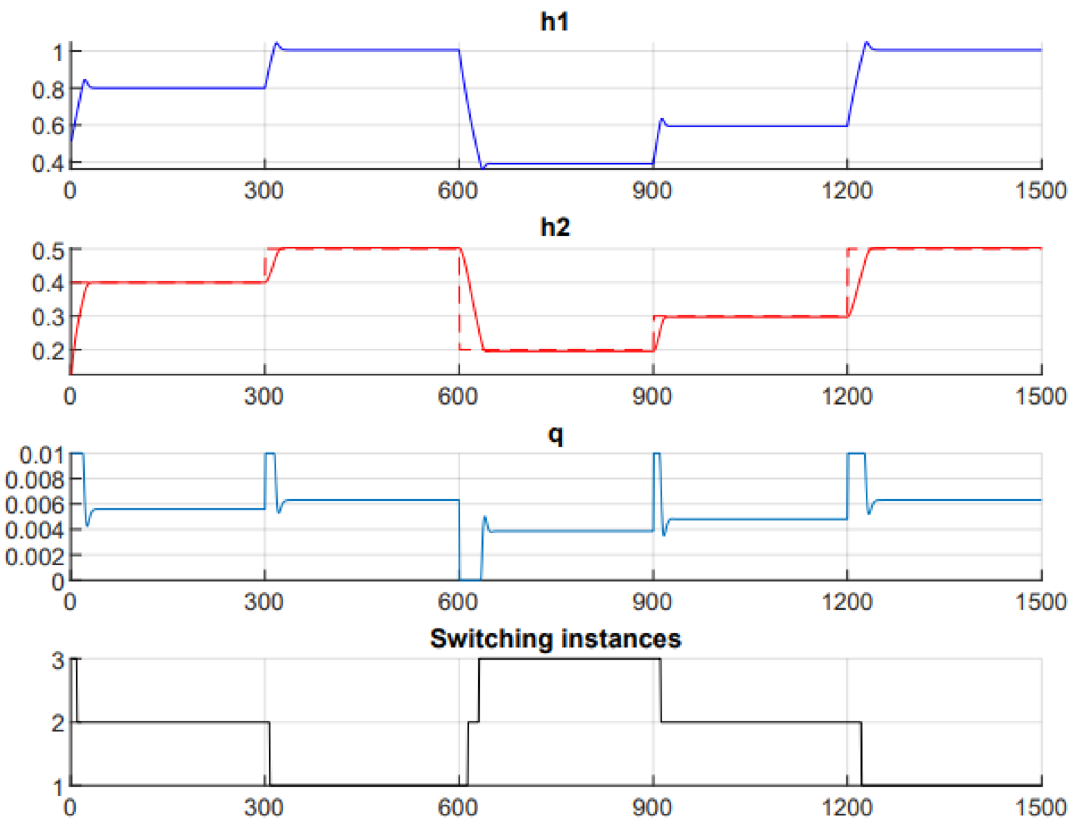

Figure 5.7: Plot with different tresholds

This graph highlights consistency in switching between different models, each covering a specific range of output values, thus demonstrating the correctness of the code.

In contrast to the paper, through our experiments, we successfully demonstrated the effectiveness of MPC in controlling the level of interconnected tanks, showcasing its potential applicability in similar processes with nonlinear dynamics. This allowed us to effectively regulate the liquid levels in the tanks while considering constraints in the sys-

tem. This highlights the versatility of predictive control techniques, particularly when applied to complex systems like the level control system considered in our study.
We believe that the difference between our results and those reported in the paper stems from the fact that they employ one-step predictive control, where the horizon is set to one, and they utilize an explicit solution of u to compute the state at the next time step. In contrast, we choose to use a prediction horizon $H_p = 15$ and a control horizon $H_u = 10$, ensuring a better prediction of future states and thus improved tracking of the reference. As demonstrated by the plots, our approach proves to be more effective.

5.2 Example II

Subsequently, we implemented another experimental application of MPC for PWA systems, based on the same two-tanks model Figure 5.1, so the same Simulink block diagram and mathematical model were used. The difference lies in the construction of the models and the method of switching between them. The saturation level was augmented, in order to have a larger range of operation points. Each model is constructed around an equilibrium point, denoted as $\bar{q}, \bar{h_1}\ and\ \bar{h_2}$, identified by fixing a value of h_2 and setting the differential equations (5.1) equal to zero, as the system under such conditions remains time-invariant. The values of $\bar{h_2}$ chosen are in order for the models: $0.1,\ 0.4, 0.8, 1.2$ metres. Around each of these equilibrium points, as explained earlier, we can approximate the system behavior as linear. Four MPC controllers were then developed, each tailored to its respective model.
The switching, however, is based on selecting the model that predicts the minimum error between the current system measurement and the estimation of each individual model. At each time step, the liquid level in the second tank (output y) was measured and stored. Additionally, a future estimation $\hat{y}$ was computed for each model. The es-

timation error for each model was calculated as the l_2 norm of the difference between the last three measured output values and the last three estimated values.

$$\begin{aligned} Y &= \begin{bmatrix} y(k) & y(k-1) & y(k-2) \end{bmatrix} \\ \hat{Y} &= \begin{bmatrix} \hat{y}(k) & \hat{y}(k-1) & \hat{y}(k-2) \end{bmatrix} \\ \varepsilon &= \|Y - \hat{Y}\|_2^2 \end{aligned} \tag{5.8}$$

This approach of considering multiple past measurements aimed to enhance the robustness of the switching system. The model with the smallest error was selected for control action at each time step. However, to address the issue of excessive switching observed in the initial simulation runs, a threshold was introduced. This threshold, when exceeded, triggers the system to switch from one model to another.

The threshold was determined based on errors dynamic and desired control performance. If the estimation error surpasses this threshold, indicating a significant deviation from the current model's predictive capabilities, the system initiates a switch to an alternative model better suited to the prevailing conditions.

The next figure shows the obtained results:

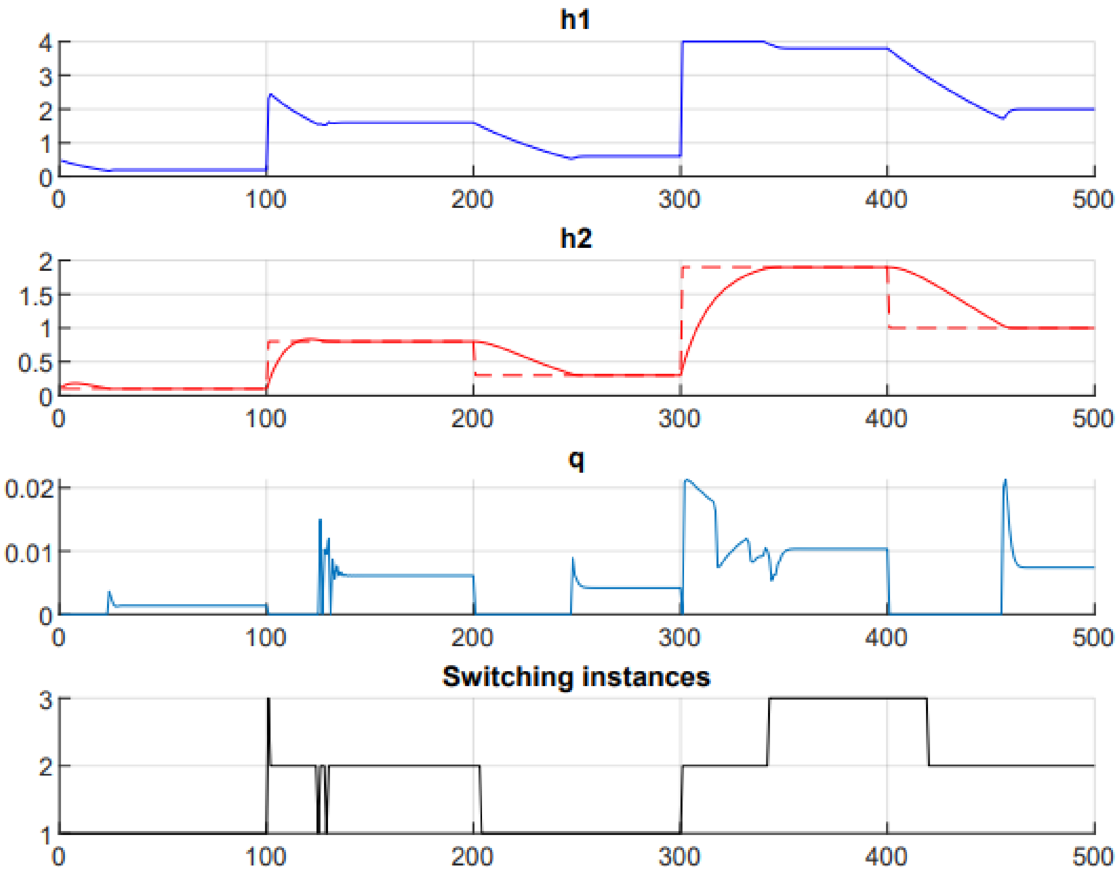

Figure 5.8: Plot of 4 Models PWA system

By implementing the threshold-based switching mechanism, the frequency of unnecessary switches was significantly reduced, leading to smoother operation and enhanced stability. The MPC system effectively tracked desired trajectories while minimizing oscillations and overshoots. The optimal result was found with prediction horizon $H_p = 40$, control horizon $H_u = 15$, $Q = 20, R = 1$, and $\tau = 1.5 \times 10^{-3}$. It is possible to notice that only three out of the four submodels are primarily used. Indeed, model 4, which has the linearization point with the highest value of $\bar{h_2}$, is not used. Model 1, characterized by the lowest value of $\bar{h_2}$, is engaged when h_2 exhibits very small magnitudes. As the output h_2 approaches approximately 1 meter, model 2 takes over. Subsequently, for higher peaks in h2, the control shifts to model 3.

As a final validation for the correct functioning of the system, we decided to introduce a random disturbance to the output of approximately 5% of its value. At first, the result was not satisfactory, as a significant overshoot occurred on the output h_2 when its value transitioned from $h_2 = 0.1$ to $h_2 = 0.8$. To mitigate this overshoot and enhance controller performance, we extended the horizons to $H_p = 50$ and $H_u = 20$, thereby achieving excellent results even in the presence of a disturbance, as depicted in the following figure.

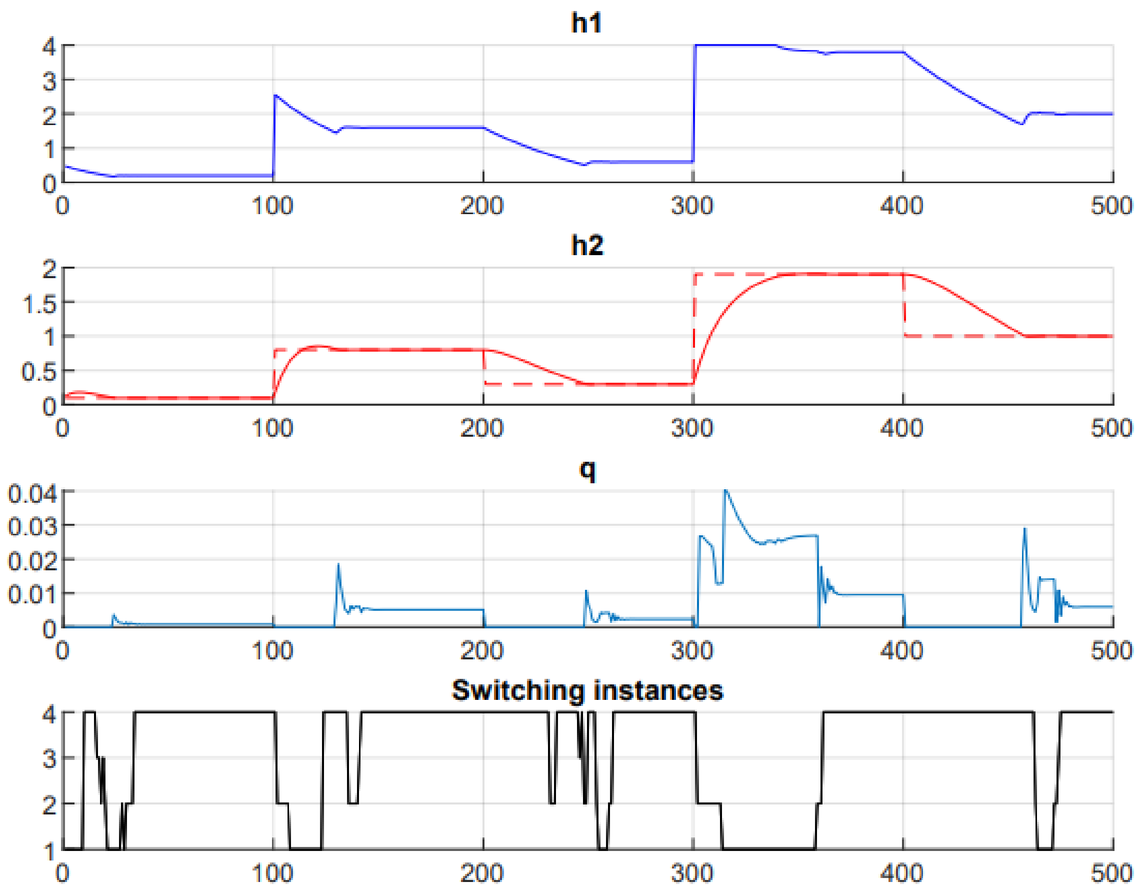

Figure 5.9: Plot of 4 Models PWA system with output disturbance

In this case, it is evident that even submodel 4 is utilized in the switching, demonstrating the system's correctness.

5.3 Conclusion and Future Developments

Through the exploration and experimentation with Model Predictive Control (MPC) applied to Piecewise Affine (PWA) systems, this chapter has provided valuable insights into the efficacy and adaptability of MPC techniques for controlling complex nonlinear processes. By implementing and fine-tuning MPC controllers on a two-tanks model, we have demonstrated the versatility and robustness of this control strategy in achieving precise tracking of reference trajectories while minimizing overshoots and oscillations. Our approach involved constructing multiple linear submodels around equilibrium points and dynamically switching between them based on prediction errors, ensuring adaptive and efficient control in varying operating conditions. Additionally, the introduction of a threshold-based switching mechanism effectively reduced unnecessary switches, further enhancing system stability and performance. Furthermore, our investigation into the effects of horizon selection and disturbance handling underscored the importance of parameter tuning and system robustness in real-world applications. Overall, the successful implementation and validation of MPC techniques in this study highlight their potential for addressing control challenges in diverse industrial processes, paving the way for future advancements in control system design and optimization.

Looking ahead, several avenues for future research and development in MPC for PWA systems present themselves. Firstly, exploring advanced optimization techniques such as model predictive control with state and input constraints could further enhance control performance and robustness, particularly in scenarios with stringent operational constraints. Additionally, investigating the integration of machine learning methods, such as reinforcement learning or data-driven modeling, could offer insights into optimizing MPC controllers in real-time based on historical process data, leading to improved adaptability and performance in dynamic environments. Lastly, exploring MPC applications in emerging fields such as renewable energy systems, smart grids, and autonomous vehicles could offer exciting opportunities for addressing complex control challenges and advancing sustainability and efficiency goals. Overall, continued research and innovation in MPC for PWA systems hold the potential to revolutionize control engineering practices and drive progress across various industrial sectors.

Bibliography

[1] "A clustering technique for the identification of piecewise affine systems". In: *Automatica* 39.2 (2003), pp. 205–217. ISSN: 0005-1098. DOI: https://doi.org/10.1016/S0005-1098(02)00224-8.

[2] "A survey of industrial model predictive control technology". In: *Control Engineering Practice* 11.7 (2003), pp. 733–764. ISSN: 0967-0661. DOI: https://doi.org/10.1016/S0967-0661(02)00186-7.

[3] N. Van De Wouw A. Pavlov A. Pogromsky and H. Nijmeijer. "On convergence properties of piecewise affine systems". In: *International Journal of Control* 80.8 (2007), pp. 1233–1247. DOI: 10.1080/00207170701261978. eprint: https://doi.org/10.1080/00207170701261978. URL: https://doi.org/10.1080/00207170701261978.

[4] Brian Aufderheide and B Wayne Bequette. "Extension of dynamic matrix control to multiple models". In: *Computers & Chemical Engineering* 27.8-9 (2003), pp. 1079–1096.

[5] Laurent Bako and Olfa Yahya. "Piecewise Affine System identification: A least harmonic mean approach". In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. 2019, pp. 2660–2665. DOI: 10.1109/CDC40024.2019.9030163.

[6] Laurent Bako et al. "A recursive identification algorithm for switched linear/affine models". In: *Nonlinear Analysis: Hybrid Systems* 5 (May 2011), pp. 242–253. DOI: 10.1016/j.nahs.2010.05.003.

[7] A. Bemporad, F. Borrelli, and M. Morari. “Min-max control of constrained uncertain discrete-time linear systems”. In: *IEEE Transactions on Automatic Control* 48.9 (2003), pp. 1600–1606. DOI: 10.1109/TAC.2003.816984.

[8] A. Bemporad et al. “A bounded-error approach to piecewise affine system identification”. In: *IEEE Transactions on Automatic Control* 50.10 (2005), pp. 1567–1580. DOI: 10.1109/TAC.2005.856667.

[9] Alberto Bemporad and Manfred Morari. “Control of systems integrating logic, dynamics, and constraints”. In: *Automatica* 35.3 (1999), pp. 407–427. ISSN: 0005-1098. DOI: https://doi.org/10.1016/S0005-1098(98)00178-2. URL: https://www.sciencedirect.com/science/article/pii/S0005109898001782.

[10] Alberto Bemporad et al. “The explicit linear quadratic regulator for constrained systems”. In: *Automatica* 38.1 (2002), pp. 3–20. ISSN: 0005-1098. DOI: https://doi.org/10.1016/S0005-1098(01)00174-1. URL: https://www.sciencedirect.com/science/article/pii/S0005109801001741.

[11] Stephen Boyd et al. *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, 1994. DOI: 10.1137/1.9781611970777. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611970777. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611970777.

[12] M.S. Branicky, V.S. Borkar, and S.K. Mitter. “A unified framework for hybrid control: model and optimal control theory”. In: *IEEE Transactions on Automatic Control* 43.1 (1998), pp. 31–45. DOI: 10.1109/9.654885.

[13] Yingwei Du et al. “Online Identification of Piecewise Affine Systems Using Integral Concurrent Learning”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 68.10 (2021), pp. 4324–4336. DOI: 10.1109/TCSI.2021.3099828.

[14] Najmeh Eghbal, Naser Pariz, and Ali Karimpour. “Discontinuous piecewise quadratic Lyapunov functions for planar piecewise affine systems”. In: *Journal of Mathematical Analysis and Applications* 399.2 (2013), pp. 586–593. ISSN: 0022-247X.

DOI: https://doi.org/10.1016/j.jmaa.2012.09.054. URL: https://www.sciencedirect.com/science/article/pii/S0022247X12007950.

[15] Giancarlo Ferrari-Trecate et al. "Analysis of discrete-time piecewise affine and hybrid systems". In: *Automatica* 38.12 (2002), pp. 2139–2146. ISSN: 0005-1098. DOI: https://doi.org/10.1016/S0005-1098(02)00142-5. URL: https://www.sciencedirect.com/science/article/pii/S0005109802001425.

[16] Andrea Garulli. "A Survey on Switched and Piecewise Affine System Identification". In: vol. 45. July 2012, pp. 344–355. ISBN: 9783902823069. DOI: 10.3182/20120711-3-BE-2027.00332.

[17] Vincent Grelet et al. "Explicit multi-model predictive control of a waste heat Rankine based system for heavy duty trucks". In: *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE. 2015, pp. 179–184.

[18] Pascal Grieder et al. "Stabilizing low complexity feedback control of constrained piecewise affine systems". In: *Automatica* 41.10 (2005), pp. 1683–1694. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2005.04.016. URL: https://www.sciencedirect.com/science/article/pii/S0005109805001482.

[19] A. Hassibi and S. Boyd. "Quadratic stabilization and control of piecewise-linear systems". In: *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No.98CH36207)*. Vol. 6. 1998, 3659–3664 vol.6. DOI: 10.1109/ACC.1998.703296.

[20] M. Johansson and A. Rantzer. "Computation of piecewise quadratic Lyapunov functions for hybrid systems". In: *IEEE Transactions on Automatic Control* 43.4 (1998), pp. 555–559. DOI: 10.1109/9.664157.

[21] A.L. Juloski, S. Weiland, and W.P.M.H. Heemels. "A Bayesian approach to identification of hybrid systems". In: *IEEE Transactions on Automatic Control* 50.10 (2005), pp. 1520–1533. DOI: 10.1109/TAC.2005.856649.

[22] Tatsuya Kai and Ryo Masuda. “Limit cycle synthesis of multi-modal and 2-dimensional piecewise affine systems”. In: *Mathematical and Computer Modelling* 55.3 (2012), pp. 505–516. ISSN: 0895-7177. DOI: https://doi.org/10.1016/j.mcm.2011.08.028. URL: https://www.sciencedirect.com/science/article/pii/S0895717711005097.

[23] Zeineb Lassoued and Kamel Abderrahim. “Identification and control of nonlinear systems using PieceWise Auto-Regressive eXogenous models”. In: *Transactions of the Institute of Measurement and Control* 41.14 (2019), pp. 4050–4062. DOI: 10.1177/0142331219846239. eprint: https://doi.org/10.1177/0142331219846239. URL: https://doi.org/10.1177/0142331219846239.

[24] Mircea Lazar et al. “Stabilization conditions for model predictive control of constrained PWA systems”. In: vol. 5. Jan. 2005, 4595–4600 Vol.5. ISBN: 0-7803-8682-5. DOI: 10.1109/CDC.2004.1429508.

[25] Jong Min Lee and Jay H. Lee. “Value function-based approach to the scheduling of multiple controllers”. In: *Journal of Process Control* 18.6 (2008), pp. 533–542. ISSN: 0959-1524. DOI: https://doi.org/10.1016/j.jprocont.2007.10.016. URL: https://www.sciencedirect.com/science/article/pii/S0959152407001606.

[26] Paolo Mason, Mario Sigalotti, and Jamal Daafouz. “Comparison between classes of state-quadratic Lyapunov functions for discrete-time linear polytopic and switched systems”. In: *Systems & Control Letters* 61.11 (2012), pp. 1062–1068. ISSN: 0167-6911. DOI: https://doi.org/10.1016/j.sysconle.2012.07.010. URL: https://www.sciencedirect.com/science/article/pii/S0167691112001430.

[27] D.Q. Mayne and Saša V. Raković. “Model predictive control of constrained piecewise affine discrete-time systems”. In: *International Journal of Robust and Nonlinear Control* 13 (Feb. 2003), pp. 261–279. DOI: 10.1002/rnc.817.

[28] Naresh N Nandola and Sharad Bhartiya. "A multiple model approach for predictive control of nonlinear hybrid systems". In: *Journal of Process Control* 18.2 (2008), pp. 131–148.

[29] I. Necoara et al. "Robustly stabilizing MPC for perturbed PWL systems". In: *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005 (CDC-ECC'05)*. Seville, Spain, Dec. 2005, pp. 3759–3764.

[30] Simone Paoletti et al. "Identification of Hybrid Systems A Tutorial". In: *European Journal of Control* 13.2 (2007), pp. 242–260. ISSN: 0947-3580. DOI: https://doi.org/10.3166/ejc.13.242-260. URL: https://www.sciencedirect.com/science/article/pii/S0947358007708221.

[31] Johan Peralez et al. "Data-driven multi-model control for a waste heat recovery system". In: *2020 59th IEEE Conference on Decision and Control (CDC)*. 2020, pp. 5501–5506. DOI: 10.1109/CDC42340.2020.9304418.

[32] Panagiotis Petsagkourakis, William Paul Heath, and Constantinos Theodoropoulos. "Stability analysis of piecewise affine systems with multi-model predictive control". In: *Automatica* 111 (2020), p. 108539. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2019.108539. URL: https://www.sciencedirect.com/science/article/pii/S0005109819304005.

[33] Stephen Prajna and Antonis Papachristodoulou. "Analysis of Switched and Hybrid Systems - Beyond Piecewise Quadratic Methods". In: vol. 4. July 2003, 2779–2784 vol.4. ISBN: 0-7803-7896-2. DOI: 10.1109/ACC.2003.1243743.

[34] A. Rantzer and M. Johansson. "Piecewise linear quadratic optimal control". In: *IEEE Transactions on Automatic Control* 45.4 (2000), pp. 629–637. DOI: 10.1109/9.847100.

[35] James Blake Rawlings, David Q Mayne, and Moritz Diehl. *Model predictive control: theory, computation, and design*. Vol. 2. Nob Hill Publishing Madison, WI, 2017.

[36] Luis Rodrigues and Stephen Boyd. "Piecewise-affine state feedback for piecewise-affine slab systems using convex optimization". In: *Systems & Control Letters* 54.9 (2005), pp. 835–853. ISSN: 0167-6911. DOI: https://doi.org/10.1016/j.sysconle.2005.01.002. URL: https://www.sciencedirect.com/science/article/pii/S0167691105000125.

[37] Jacob Roll, Alberto Bemporad, and Lennart Ljung. "Identification of piecewise affine systems via mixed-integer programming". In: *Automatica* 40.1 (2004), pp. 37–50. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2003.08.006. URL: https://www.sciencedirect.com/science/article/pii/S000510980300270X.

[38] C. Rowe and J. Maciejowski. "Robust constrained receding horizon control of PWA systems with norm-bounded input uncertainty". In: *Proceedings of the 2003 American Control Conference, 2003.* Vol. 5. 2003, 3949–3954 vol.5. DOI: 10.1109/ACC.2003.1240453.

[39] P.O.M. Scokaert and D.Q. Mayne. "Min-max feedback model predictive control for constrained linear systems". In: *IEEE Transactions on Automatic Control* 43.8 (1998), pp. 1136–1142. DOI: 10.1109/9.704989.

[40] Olav Slupphaug, Lars Imsland, and Bjarne Foss. "Uncertainty Modeling and Robust Output Feedback Control of Nonlinear Discrete Systems: A Mathematical Programming Approach". In: *Modeling, Identification and Control* 22 (Jan. 2001). DOI: 10.4173/mic.2001.1.3.

[41] Chunyue Song, Bing Fang Wu, and Ping Li. "A hybrid model-based optimal control method for nonlinear systems using simultaneous dynamic optimization strate-

gies". In: *Journal of Process Control* 22 (2012), pp. 852–860. URL: https://api.semanticscholar.org/CorpusID:120849314.

[42] Xiaoqi Song and James Lam. "Bumpless H^∞ for periodic piecewise linear systems". In: *Automatica* 135 (2022), p. 109941. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2021.109941. URL: https://www.sciencedirect.com/science/article/pii/S0005109821004660.

[43] E. Sontag. "Nonlinear regulation: The piecewise linear approach". In: *IEEE Transactions on Automatic Control* 26.2 (1981), pp. 346–358. DOI: 10.1109/TAC.1981.1102596.

[44] "Stability and stabilization of periodic piecewise linear systems: A matrix polynomial approach". In: *Automatica* 94 (2018), pp. 1–8. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2018.02.015.

[45] K. Tanaka, T. Ikeda, and H. O. Wang. "Robust stabilization of a class of uncertain nonlinear systems via fuzzy control: quadratic stabilizability, H^∞ control theory, and linear matrix inequalities". In: *Trans. Fuz Sys.* 4.1 (Feb. 1996), pp. 1–13. ISSN: 1063-6706. DOI: 10.1109/91.481840. URL: https://doi.org/10.1109/91.481840.

[46] Giancarlo ferrari trecate et al. "Analysis of discrete-time piecewise affine and hybrid systems". In: *Automatica* 38 (Dec. 2002), pp. 2139–2146.

[47] Sergio Waitman. "A piecewise-affine approach to nonlinear performance". Theses. Université de Lyon, July 2018. URL: https://theses.hal.science/tel-01917511.

[48] Meng Wang et al. "Data-driven control for discrete-time piecewise affine systems". In: *Automatica* 155 (2023), p. 111168. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2023.111168. URL: https://www.sciencedirect.com/science/article/pii/S0005109823003291.

[49] Jan Willems. "Dissipative Dynamical Systems". In: *European Journal of Control* 13 (Dec. 2007), pp. 134–151. DOI: 10.3166/ejc.13.134-151.

[50] Jan C. Willems. "Dissipative dynamical systems part I: General theory". In: *Archive for Rational Mechanics and Analysis* 45 (1972), pp. 321–351. URL: https://api.semanticscholar.org/CorpusID:123076101.

[51] Jan C. Willems. "Dissipative dynamical systems Part II: Linear systems with quadratic supply rates". In: *Archive for Rational Mechanics and Analysis* 45 (1972), pp. 352–393. URL: https://api.semanticscholar.org/CorpusID:122845740.

[52] Xiaochen Xie and James Lam. "Guaranteed cost control of periodic piecewise linear time-delay systems". In: *Automatica* 94 (2018), pp. 274–282. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2018.04.047. URL: https://www.sciencedirect.com/science/article/pii/S0005109818302322.

[53] Lassoued Zeineb and Abderrahim Kamel. "Predictive Control of nonlinear systems based on PWA model". In: *2022 IEEE 21st international Ccnference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*. 2022, pp. 448–453. DOI: 10.1109/STA56120.2022.10019110.

[54] Yanzheng Zhu et al. "A Descriptor System Approach to Stability and Stabilization of Discrete-Time Switched PWA Systems". In: *IEEE Transactions on Automatic Control* 63.10 (2018), pp. 3456–3463. DOI: 10.1109/TAC.2018.2797173.

[55] Yuanyuan Zou and Shaoyuan Li. "Robust Model Predictive Control for Piecewise Affine Systems". In: *Circuits, Systems and Signal Processing* 26 (June 2007), pp. 393–406. DOI: 10.1007/s00034-006-0519-2.