UNCERTAINTY CHARACTERIZATION IN ROBUST MPC USING AN APPROXIMATE CONVEX HULL

by

Hossein Sartipizadeh

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Electrical Engineering).

Golden, Colorado

Date _____

Signed: _____
Hossein Sartipizadeh

Signed: _____
Dr. Tyrone L. Vincent
Thesis Advisor

Signed: _____
Dr. Robert J. Kee
Thesis Advisor

Golden, Colorado

Date _____

Signed: _____
Dr. Atef Elsherbeni
Professor and Head
Department of Electrical Engineering

ABSTRACT

Model Predictive Control (MPC) of processes when uncertainty is involved is the topic of this thesis. Specifically, a method to characterize parametric uncertainty for robust model predictive control is studied. The goal is to reduce the computational complexity of robust MPC and robust Moving Horizon Estimation (MHE). The main element of this method is the computation of an approximate convex hull that *approximately* covers the system uncertainty in a new output prediction mapping. Given the complete uncertainty set, an approximate convex hull is computed to determine an efficient set of extreme points to represent this set. The calculated set is a subset of the uncertainty set and can be significantly smaller than the original set, which results in decreasing the computational complexity. A measure of the approximation to the original set is used to provide robust output prediction that is guaranteed to hold for all systems in the original set. In other words, the introduced method provides a dynamic guaranteed bound on all possible output trajectories. The control performance of the proposed method will be investigated on the methane reforming process as a key element of fuel cells, and on a DC-DC floating interleaved boost converter.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am most grateful to Dr. Vincent and Dr. Kee for their great direction, amazing support, encouragement, and expertise throughout my PhD. It was a pure honor to work with them and I am thankful to them for believing in me. I would also say a special thanks to my supervisor, Dr. Vincent, whose selfless time and care were sometimes all that kept me going. I would also like to thank the members of my committee, Dr. Moore, Dr. Johnson, and Dr. Zhang for their time, kind help, and expertise.

I would like to thank Dr. Huayang Zhu and Dr. Canan Karakaya for their kind help and direction regarding the methane reforming process. I would also like to thank my great friends Dr. Farshad Harirchi, Farnaz Harirchi, Dr. Mohammad Babakmehr, Dr. Mahmood Ahmadi, Dr. Elham Parsa, Ali Nazem, Soheyl Khademiyan, and Dr. Armin Eftekhari for their support during my PhD.

Non of this would happen without my family support, My beloved mother, brother, and sister.

This thesis was supported by ONR grants N00014-12-1-0201 and N00014-16-1-2780.

This thesis is lovingly dedicated to my mother with all my heart. Her support, encouragement, and endless love have sustained me throughout my life. I hope she rests in peace.

CHAPTER 1

INTRODUCTION

Model Predictive Control (MPC) is widely adopted in industry as an effective means to deal with nonlinear, multivariable, and/or constrained control problems. This type of controller employs an explicit model of the plant to predict the future output behavior of the system. Knowledge of the predicted future is extremely helpful for improving tracking performance in presence of probable disturbances and signal delays.

In MPC, a control law is calculated by solving a constrained, dynamic, optimal control problem with a finite horizon to minimize a cost function that is subject to the system dynamics and physical constraints over a prediction horizon p. Let $u \in \mathbb{R}^m$, $y \in \mathbb{R}^r$, and $x \in \mathbb{R}^n$ be the input, output, and state of the system, respectively, with dynamic equations $x_{k+1} = f(x_k, u_k)$ and $y_k = h(x_k)$. Let $\Phi_k(x_k, u_k)$ be a cost function that measures the cost of state x_k and input u_k at time k representing a dynamic system at the sampling time $t = kT_s$. At each sampling time, the nominal MPC method solves the following optimization problem.

$$\min_{u_0,\cdots,u_{p-1}} \quad \sum_{k=1}^{p} \Phi_k(x_k, u_k) \tag{1.1}$$

$$\text{subject to} \quad \begin{aligned} & x_{k+1} = f(x_k, u_k), x_0 = x(t_0) \\ & y_k = h(x_k) \\ & g(u, y) \leq b \end{aligned} \tag{1.2}$$

where an estimation procedure is involved to find the initial state $x_0 = x(t_0)$. The control action u_0 is applied to the system, and the problem is repeated at the next sampling time.

The finite horizon is utilized to obtain an optimization problem that can be feasibly solved in real time. Early work on MPC focused on the stability of the feedback loop involving the MPC algorithm and physical system. Mayne et al. [1] introduced some stability conditions using Lyapunov theory. According to [1], stability of MPC for linear and nonlinear systems with hard state and constraints may be achieved either by adding a terminal cost and

constraint or by extending the prediction horizon. Studies on the stability and feasibility of MPC were then continued over the last decade (see [2–4]). Moreover, removal of the terminal conditions has been a focal point in recent studies. (for example see [4]).

Since MPC is model based, having an accurate model is vital to achieve an appropriate control result. However, different type of uncertainties, including disturbances and model uncertainties, are inseparable elements of a control application. Uncertainty due to measurement noise and disturbances can often be addressed efficiently under standard assumptions (i.e. i.i.d. Gaussian distribution). Dynamic uncertainty provides a greater computational challenge, but is also important. Commonly, a manufacturer produces multiple units of a product with parameters varying over a range. Consequently, different units may have different behavior. It implies that a single model is not a reliable representative of the real plant. This challenge was a starting point for the development of robust MPC by the end of 1980s. In uncertain cases, the system may be represented by $x_{k+1} = f(x_k, u_k, w_k)$, $y_k = h(x_k, v_k)$ with bounded w and v, or it may be given in a user-defined set called *model set*, i.e., $f \in \mathcal{F} = \{f_1, \cdots\}, h \in \mathcal{H} = \{h_1, \cdots\}$.

A distinguishing feature of robust MPC compared to nominal MPC is that the control objectives and constraints must be satisfied for all realizations of the uncertainties. Let $\mathcal{S}$ be the set of all realizations of the uncertainties associated with any given representation (i.e., $\mathcal{S} = \{\mathcal{F}, \mathcal{H}\}$). A robust MPC (RMPC) approach was initially introduced by Campo in 1987 to deal with parametric uncertainty. This approach solved the min-max optimization problem

$$\begin{aligned} &\min_{u_0,\cdots,u_{p-1}} \ \max_{\mathcal{S}} \ \sum_{k=1}^{p} \Phi_k(x_k, u_k) \\ &\text{subject to (1.2)} \end{aligned} \tag{1.3}$$

Although the min-max problem is conservative due to the fact that the worst case is taken under consideration, it is useful in applications where the violation of a constraint imposes a noticeable cost. For instance, a power producer is responsible for producing the desired

output power regardless of the type and amount of uncertainty. Otherwise, the company is subjected to a significant penalty.

Since $\mathcal{S}$ may be an infinite set, Campo and Morari [5] and, later, Allwright and Papavasiliou [6] replaced all possible realizations with a finite number of realizations under some conditions and limitations. In [5, 6] the authors found a solution for the min-max problem over a finite set of extreme models. They utilized the impulse response description of linear systems to find a model set with finite elements. Using the model set, they presented their algorithms to account for uncertainties in a case where the impulse response is affine in the uncertain parameters along with the constraints and objective function being linear. In this approach, the set of all realizations of uncertainties is replaced with a set of extreme models, called the model set, corresponding to the extreme parameters of the uncertainty range. In this case, q uncertain parameters result in 2^q extreme points and, consequently, 2^q extreme models.

A related robust MPC method uses a polytopic system representation to characterize model uncertainty (e.g. [7–11]). This approach, which is mostly used for Linear Time Variant (LTV) systems, reformulates the min-max problem to an optimization with Linear Matrix Inequality (LMI) constraints. In order to find the robust solution using this method, a polytope of the uncertain plants should be given in state space form [7].

Calafiore and Campi introduced in [12, 13] a probabilistic framework for robust control analysis and design called the scenario approach which could be applied to systems with very general uncertainty distributions. The advantage of this method is that the system can have a general dependence on the uncertain parameters. Instead of solving an optimization with infinite number of constraints, an optimization with limited number of constraints is solved over a set of random samples. In other words, a randomized sampling procedure is utilized to define the model set and a performance guarantee is given based on the number of the taken samples.

Other methods, like uncertainty set construction using Bayesian modeling, have been also studied [14] in robust optimization problems. Also more recently, many robust approaches with robust stability and feasibility guarantees have been presented. [15, 16] are examples of the earlier studies on the stability of uncertain systems with bounded w, v using Lyapunov Theory. Different stability conditions have then been proposed (see [17–20] for example).

For applications where the conservatism is either not necessary or it may result in a poor performance, stochastic MPC would then be a solution. In stochastic MPC, instead of assuming the disturbances to be bounded with constraints satisfied for all possible realizations of uncertainties, disturbances are assumed to be stochastic and not necessarily bounded [21]. The general problem is, therefore, to solve $\min_u E\Phi(x, u)$ where Ea denotes *expectation* of a. In this case, constrains are not to be necessarily satisfied for all realizations. Since the focus of this thesis is on robust MPC using a min-max formulation, readers are referred to useful surveys of stochastic MPC for more details (see [22, 23]).

Although the advent of the introduced robust methods has been a significant achievement in this area, these methods suffer some deficiencies. As it is noted, in [5, 6], as the number of uncertain parameters increases, the number of models in the model set grows exponentially. This may add to the computational complexity of the MPC implementation to the point where it may not be applicable for faster sampling rates. In addition, even for linear systems, the plant is rarely affine in the model parameters. Therefore, the finite parameter set may not cover the true output uncertainty. In polytopic representation, although this type of uncertainty representation are widely used in robust MPC, to the best of the author's knowledge, it is not straightforward to define the appropriate polytopic set for parametric uncertainty situation. In the scenario approach, although the performance is improved by increasing the number of samples from the model set, the computational complexity also increases which could limit the use of this approach in real time. Moreover, a number of these samples may be redundant in that they result in redundant constraints in the robust MPC optimization problem. Furthermore, although for some cases the Bayesian method can

be a useful tool for uncertainty characterization, since the covering element of the Bayesian method is an ellipsoid, the performance of this method depends on the data (uncertaintny) distribution, and, in some cases the computed ellipsoids and the resulted uncertainty set are highly conservative.

In this thesis, we build upon the work of Campo [5] and use the system impulse response as the system representation. In order to remove the limitations of the Campo method for systems where impulse responses are not affine in the parameters, we replace the uncertain parameter set with a finite number of samples similar to the scenario approach. In order to significantly reduce the computational complexity of the scenario approach, we present a new approach to remove the redundant samples. The idea is to map the parameter set to a new prediction mapping that reflects the output trajectory over the prediction horizon. It turns out that extreme points of the convex hull of the finite set of models in the new mapping can be used as the reduced model set, and the rest of the models are redundant and can be removed. This approach significantly decreases the computational complexity while it still guarantees a bound on all output trajectories corresponding to the all uncertainty realizations over the prediction horizon.

Many algorithms have been developed to find the convex hull of a set of points in $\mathbb{R}^n$. Some of the first efficient algorithms were Graham's scan method [24] and the gift wrapping algorithm [25, 26]. For a set of N points these algorithms have $\mathcal{O}(N \log N)$ worst case run time for low dimensions ($n = 2$ or 3) which is the same order as the optimal algorithm. These were followed by many others; a survey can be found in [27]. The focus on low dimensions is well motivated by many applications in computational geometry, image processing and computer vision. However, the proposed RMPC requires finding the convex hull in high dimensions. Specifically, the dimension of interest in this application will be proportional to the size of the prediction window, which can be significantly larger than 3.

For higher dimensions, standard methods are the method of Clarkson and Shor [28] and the Quickhull algorithm [29], which have worst case time complexity in high dimensions of

$\mathcal{O}(N^{\lfloor n/2 \rfloor})$. The time complexity of these algorithms are also closely related to the representational complexity of the convex hull. For a set of points, the convex hull is a polytope, and could be represented by the facets, or by the vertices. For a set of N random data in n dimensions, the expected number of vertices is $\mathcal{O}(\log^{n-1} N)$ [30]. Thus, even in the average case, the representational complexity increases exponentially in the dimension n.

This explosion of complexity in higher dimensions can detrimental not only to finding the convex hull, but in using it, as algorithms that use the convex hull as input may require processing each element of the representation. This gives double motivation for working with an *approximate* convex hull that covers almost the same space as the true convex hull, but with a reduced representational complexity. First, in order to reduce the computational complexity of finding it, and second to reduce the computational burden of using it.

To this end, we first introduce a new method to find the *ϵ-approximate convex hull* of a set in high dimension with a lower time complexity compared to the prior work. The method aims to find minimum number of vertices for a pre-specified violation error ϵ. The developed approximate convex hull computation method is then utilized for characterizing the model uncertainty using the new prediction mapping, and, therefore, introducing a new Robust Moving Horizon Estimation (RMHE). Given the future inputs, the aim of the proposed RMHE is to determine guaranteed bounds on the uncertain output trajectories over the future horizon by taking into account the approximation error ϵ. The introduced estimation approach is then used to develop a novel RMPC. The performance of the proposed controller is examined by implementing the controller on a DC-DC floating interleaved boost converter (FIBC) to handle the input voltage uncertainty, and, a methane reforming to satisfy the physical constraints under uncertain situations.

The main contributions of this work are as follows:

- Introducing a novel method to compute the approximate convex hull of a set in high dimensions.
- Introducing a new mapping to characterize the parametric uncertainty.

- Developing a new RMHE approach using the proposed uncertainty characterization method.
- Developing a new RMPC approach using the proposed uncertainty characterization method.
- Employing the proposed RMPC to control a DC-DC floating interleaved boost converter under uncertain situations.
- Economic control of the methane reforming using a proper model predictive controller.
- Robust control of the methane reforming using the proposed RMPC.

The papers, published or submitted from this work, are listed below:

✓ Hossein Sartipizadeh, Tyrone L. Vincent, and Robert J. Kee. Economic control of methane reforming, in Proc. *IEEE American Control Conference*, 2015.

✓ Hossein Sartipizadeh and Tyrone L. Vincent. Uncertainty Characterization for Robust MPC using an Approximate Convex Hull method, in Proc. *IEEE American Control Conference*, 2016.

✓ Hossein Sartipizadeh and Tyrone L. Vincent. Computationally Tractable Robust Moving Horizon Estimation Using an Approximate Convex Hull, in Proc. *55th IEEE Conference on Decision and Control (CDC)*, 2016.

✓ Hossein Sartipizadeh and Tyrone L. Vincent. A method to calculate the Approximate Convex Hull in High Dimensions, submitted to *Computational Geometry*, 2016.

✓ Hossein Sartipizadeh and Tyrone L. Vincent. A new Robust MPC approach using an Approximate Convex Hull, submitted to *Automatica*, 2016.

✓ Hossein Sartipizadeh and Farnaz Harirchi. Robust Model Predictive Control of DC-DC Floating Interleaved Boost Converter under Uncertainty, submitted to *IEEE GreenTech, 2017*.

This thesis is structured as follows: Some of the preliminaries used in the thesis are introduced in Chapter 2. The focus of Chapter 3 is on developing the approximate convex hull computation method for high dimensions. Using the approximate convex hull method introduced in Chapter 3, the focus of Chapter 4 is on developing a new robust MHE with

a new uncertainty characterization. This estimator is used to develop a novel robust MPC in Chapter 5. Finally, robust control of FIBC and methane reforming using the developed RMPC is investigated in Chapter 6 and Chapter 7, respectively.

CHAPTER 2

BACKGROUND

In this section we briefly review some of the control concepts including MPC and MHE as well as convex hull description widely used in this work in addition to some of the most notable notations.

2.1 Notation

$\mathbb{R}$ is the set of real numbers, with $\mathbb{R}^n$ a length n vector of real numbers. For $x \in \mathbb{R}^n$, x^T denotes the transpose of x and x_i is the ith element of x. For integer s, $1 \leq s < \infty$, $\|x\|_s := (\sum_{i=1}^{n} |x_i|^s)^{\frac{1}{s}}$ and $\|x\|_\infty = \max_i |x_i|$. Given a set Θ with elements θ and function $f(\theta)$, let $f(\Theta)$ denote the set $\{f(\theta)|\theta \in \Theta\}$. For symmetric, invertible matrix $Q \in \mathbb{R}^{n\times n}$, $\|x\|_Q = \sqrt{x^T Q^{-1} x}$. I is the identity matrix, with dimension that may be given as subscripts for clarity. Given a set $\mathcal{S} \subset \mathbb{R}^n$, $|\mathcal{S}|$ is the number of elements in $\mathcal{S}$, $\operatorname{Conv}\mathcal{S}$ is the convex hull of $\mathcal{S}$ and $\mathcal{S}\backslash\mathcal{E}$ is the set obtained by removing the elements in $\mathcal{E}$ from $\mathcal{S}$. For matrix M let ${}^rM$ denote the r^{th} row of M. Also for set $\mathcal{S}$ let ${}^r\mathcal{S}$ be the set of r^{th} row of elements of $\mathcal{S}$.

2.2 Convex Hull

The convex hull of a set of points in $\mathbb{R}^n$ is the smallest convex set which contains all the points and is a fundamental construction for mathematics as computational geometry [29]. This concept is widely used in different fields including data searching and signal processing [31], clustering, image processing, modeling and robust control [11], collision detection [32], and urban planning [33].

Visually, a set is convex if the line between any two arbitrary points inside the set completely lies into the set. In 2 and 3 dimension spaces, the convex hull can be shown as a closed set which the rest of the points lie within. Figure 2.1 demonstrates forming the convex hull in a plane for a given data set $\mathcal{S}$ where the convex hull is the grey polygon. As seen, if

you draw a line between any two points of the set, the line remains in the gray region. The vertices of the grey polygon are called *extreme points* while the rest of the points are *interior points* of $\mathcal{S}$. Note that points on the boundaries are not considered as extreme points.

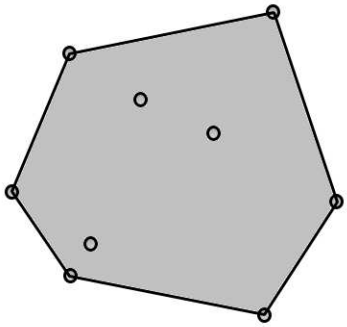

Figure 2.1: Convex hull demonstration in 2D

Similarly in high dimensions, one can find a convex polytope which encompasses all the points. Since visually showing the convex hull in high dimensions is not applicable, we usually use the mathematical description of the convex hull. According to definition, an arbitrary vector $x \in \mathbb{R}^n$ belongs to the convex hull of $\mathcal{S} = \{s_1, \cdots, s_N\}$ ($x \in \text{Conv}\,\mathcal{S}$) iff x can be written as a convex combination of elements of $\mathcal{S}$. In other words, if we can find set $\alpha = \{\alpha_1, \cdots, \alpha_N\}$ so that $x = \sum_{i=1}^{N} \alpha_i s_i$ where α_is are nonnegative and $\sum_{i=1}^{N} \alpha_i = 1$, then $x \in \text{Conv}\,\mathcal{S}$. Using this definition, the following results are concluded:

1. $s_j \in \mathcal{S}$ is an extreme point of $\mathcal{S}$ iff it can not be written as a convex combination of the rest of the points. i.e., there is no α to satisfy $s_j = \sum_{i \neq j} \alpha_i s_i$.

2. $s_j \in \mathcal{S}$ is an interior point of $\mathcal{S}$ iff it can be written as a convex combination of the rest of the points. i.e., there is a vector α so that $s_j = \sum_{i \neq j} \alpha_i s_i$.

2.3 Moving Horizon Estimation

MHE can be employed to determine the current state of the system and therefore predicting the system future behavior. The estimator examines the recent input/output behavior

of the system, and finds the current state for which the system model best matches this behavior. By using a window of data, we get the benefit of noise rejection. With the right model, this state estimate allows us to estimate the current value of the outputs that we cannot directly measure, but that are important to define the control performance.

Assume a system is given in the following variational state space representation[1]

$$\begin{aligned} x_{k+1} &= Ax_k + B\delta u_k + w_k \\ \delta y_k &= Cx_k + D\delta u_k + v_k \end{aligned} \tag{2.1}$$

where $\delta u_k \in \mathbb{R}^m$ is the system input, $\delta y_k \in \mathbb{R}^p$ is the system output, $x_k \in \mathbb{R}^n$ is the system state, and $w_k \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^l$ represent unmeasured signals, denoted process and measurement noises respectively, which account for disturbances as well as model uncertainty. The matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$ define the system dynamics. In addition, the unmeasured signals are given a probabilistic description as independent, identically distributed random sequences with zero mean and covariance

$$cov(w, v) = E\left(\begin{bmatrix} w_k \\ v_k \end{bmatrix}\begin{bmatrix} w_{k+\tau}^T & v_{k+\tau}^T \end{bmatrix}\right) = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}\delta_\tau. \tag{2.2}$$

Since the estimator can only work with system outputs that are actually measured, we define the variable δy^s to be those elements of the output δy_k of (2.1) that are sensed. We also extract the rows of C, D and v_k associated with these elements into variables with superscript s, so that we can write

$$\delta y_k^s = C^s x_k + D^s \delta u_k + v_k^s. \tag{2.3}$$

where R^s is the sub matrix of R associated with the sensed outputs. There are another set of system outputs, both sensed and un-sensed, that are necessary to define the control performance. In a similar manner, we will represent these outputs with variables with subscript r

$$\delta y_k^r = C^r x_k + D^r \delta u_k + v_k^r. \tag{2.4}$$

[1]Because the operating point will be continually changing, it is convent to use a linear system to model the differential behavior.

The MHE solves an optimization problem to find the state sequence that best explains the observed data over a window of p samples into the past, where p is chosen large enough to ensure estimator stability while also ensuring that only data near the current operating point is considered. To determine how well a given sequence "explains the observed data" we measure the energy of the disturbance and noise necessary for the input, output and state sequences to be compatible with the model (2.1). Specifically, given the sequences δu_k, δy_k^s and x_k for $k = -p, \cdots, 0$, one can calculate

$$\begin{aligned} w_k &= x_{k+1} - Ax_k - B\delta u_k \quad k = -p, \ldots, -1 \\ v_k^s &= \delta y_k^s - C^s x_k - D^s \delta u_k \quad k = -p, \ldots, 0. \end{aligned} \tag{2.5}$$

This process is illustrated in Figure 2.2. Clearly, the smaller w_k and v_k are, the better these sequences match the model, and the MHE seeks the smallest such choice. Since the w_k and v_k are modeled as iid random sequences, a well-motivated metric for measuring the size of w_k and v_k is through a weighted Euclidean norm. Without loss of generality, the MHE solves the following optimization problem

$$\begin{aligned} \min_{x_{-p}, \cdots, x_0} \quad & \sum_{k=-p}^{0} \|v_k^s\|_{R^s}^2 + \sum_{k=-p}^{-1} \|w_k\|_Q^2 \\ \text{subject to} \quad & x_{k+1} = Ax_k + B\delta u_k + w_k \\ & \delta y_k^s = C^s x_k + D^s \delta u_k + v_k^s \end{aligned} \tag{2.6}$$

This optimization problem has a quadratic objective function with linear constraints, and thus, the global minimum can be found very efficiently, which is important for real time operation. From the minimizing sequence, $\hat{x}_k$, the estimator extracts $\hat{x}_0$ as the estimate of the current state. In addition, given a input sequence δu_k for $k > 0$, an estimate of the changes of the control relevant system outputs in the future is given by

$$\delta \hat{y}_k^r = C^r \hat{x}_k + D^r \delta u_k. \tag{2.7}$$

Therefore, the absolute value of the output can be calculated as:

$$y_k^r = \hat{y}_0 + \sum_{l=1}^{k} \delta \hat{y}_l^r. \tag{2.8}$$

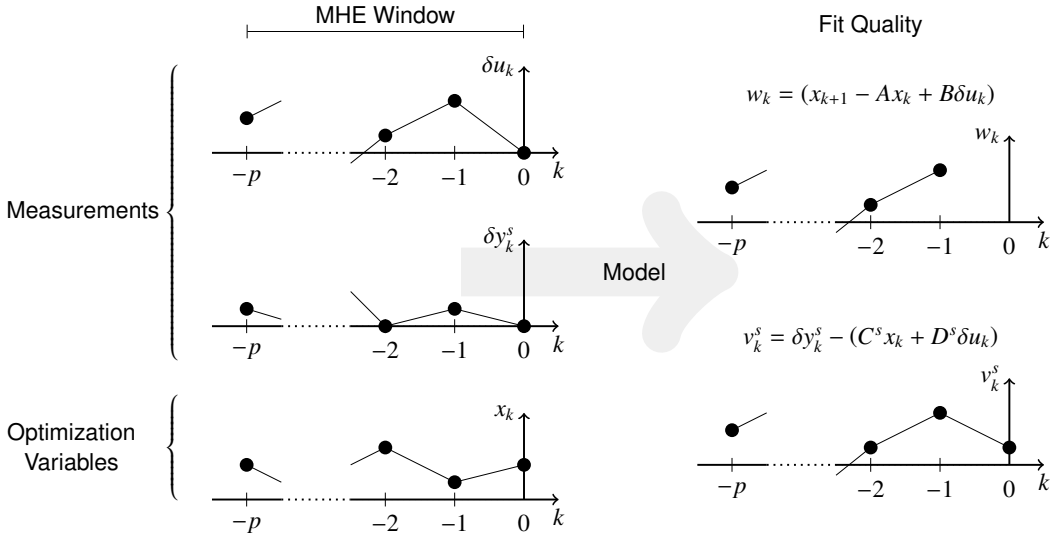

Figure 2.2: State estimation via moving horizon estimation. At each sample time, MHE finds the state trajectory with the best match between model and most recently observed input/output data as measured by the required disturbance signal w_k and measurement noise v_k. The current time is sample $k = 0$.

where y_0 can be directly measured for the sensed output or be estimated using a secondary estimator designed for the large scale.

2.4 Model Predictive Control

MPC is mostly formulated in time domain and the explicit model used by the controller can be simply its state space representation. In general, MPC performs in two steps. First, a prediction of the system signals into the future as a function of the future inputs is obtained and then an optimization problem with future inputs as parameters is solved based on the desired objectives and constraints. The prediction part commonly uses the initial state x_0 obtained by an estimator (as described in Section 2.3) to run the explicit model.

Model predictive control also uses real-time optimization to determine the appropriate control action for the next sample time. As illustrated in Figure 2.3, the optimization variables are the system input sequence δu_k over a window p samples into the future. Using the model, along with the estimate of the current state provided by the MHE, a prediction of

the output sequence δy_k^r is obtained. This output sequence is evaluated based on two types of criteria

- A cost function: The cost for the proposed input and output sequence at time k is represented by the function Φ_k. There are a large variety of forms that this function can take based on the control objectives.
- Constraints: In some cases, there are absolute bounds that must be met by system outputs in order for proper operating, although there is no preference for the system outputs within those bounds. For these outputs, the constraint set

  $$b_k^{min} \leq y_k^r + \zeta \leq b_k^{max}$$

  is added, where b_k^{min} sets the lower bound, b_k^{max} sets the upper bound, and ζ is an auxiliary variable that is non-zero only when the bounds must be violated to achieve a feasible solution. In this case, a second cost function $\Xi(\zeta)$ determines the cost of not meeting the required bounds, and a supervisory controller can initiate shutdown if the value of ζ becomes too large.

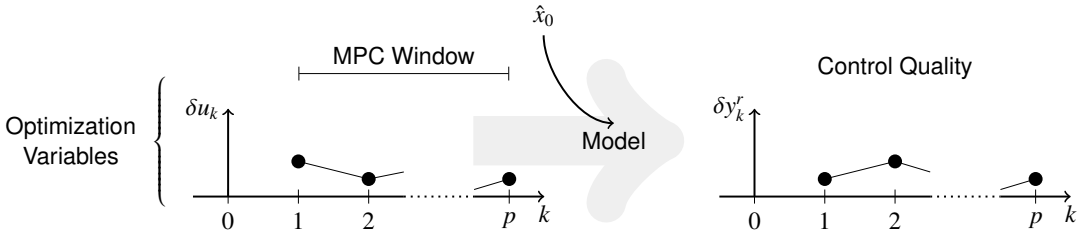

Figure 2.3: Model Predictive Control. At each sample time, MPC finds the input sequence that provides the best output behavior of the system over a window of time into the future. The current time is sample $k = 0$.

After finding the best sequence of future inputs, the controller applies u_0 to the system and repeats the procedure at the next sampling time.

CHAPTER 3

COMPUTING THE APPROXIMATE CONVEX HULL IN HIGH DIMENSIONS

In this chapter, a novel method to find the ϵ-approximate convex hull of a set in high dimensions is presented. This method will be used later as a key element to develop the uncertainty characterization approach introduced in Chapter 4, and consequently, the proposed robust MPC in Chapter 5[2].

3.1 Introduction

In Chapter 2 we described the definition of convex hull. As mentioned, the convex hull of a set $\mathcal{S}$ is the polytope that contains all the points of the set. In practical, in high dimensions the number of extreme points of a set tends to the number of its elements. In this case, an approximation of the convex hull with fewer extreme points may be of interest. An ϵ-approximate convex hull of a set $\mathcal{S}$ is the polytope with fewest extreme points such that all elements of set $\mathcal{S}$ are inside or not farther than ϵ from it. Figure 3.1 illustrates the concept of the approximate convex hull. Let set $\mathcal{S} = \{x_1, \cdots, x_{20}\}$ where $x_i \in \mathbb{R}^2$ be as the points presented in Figure 3.1. The convex hull of $\mathcal{S}$ is shown with solid line and consists of 11 extreme points (shown by black). The ϵ-approximate convex hull of $\mathcal{S}$ is determined with dash lines. As seen, the approximate convex hull *approximately* covers the set $\mathcal{S}$ with only 6 extreme points, and at the worse case, the distance of the elements of $\mathcal{S}$ from the approximated polytope is ϵ.

The search for an approximate convex hull was apparently first considered by Bentley in 1982 [35]. This method used sampling within a grid to achieve a worst case time complexity of $\mathcal{O}(N^{\lfloor n/2 \rfloor} + (1/\epsilon)^{n-1}N)$, where ϵ is the maximum error, N is the number of points, and n is the dimensionality of the space . An approach that uses a grid over angles in polar

[2]This section is based on a submitted paper: Hossein Sartipizadeh and Tyrone L. Vincent. A method to calculate the Approximate Convex Hull in High Dimensions *submitted to Computational Geometry*, 2016 [34].

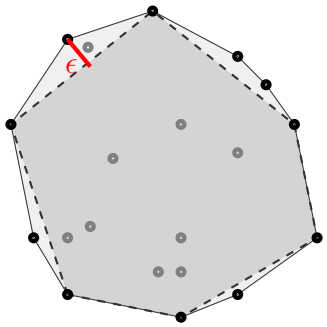

Figure 3.1: ϵ-Approximate convex hull illustration

coordinates was later proposed by [36] with a time complexity of $\mathcal{O}((1/\epsilon)^{n-1}N)$. Although providing a trade-off between run time and accuracy, unfortunately both of these methods also have an exponential dependence on dimension.

In order to tackle convex sets in high dimensions, [37] introduced a useful norm based method to compute the convex hull, with a stated time complexity of $\mathcal{O}(Nn^4)$. This method was further extended in [38]. This method could also be modified to determine an approximate convex hull, although it has not been specifically designed for this purpose. Two drawbacks are that this method does not explicitly consider the quality of the approximation as new vertices are found, and that it uses a fairly large number of vertices for initialization. Based on the initialization approach presented in [37], this method begins with a n-simplex with $n+1$ vertices and therefore is not able to find an optimal approximation with fewer number of vertices. This would be an issue, for example, in cases when the data closely follows a low dimensional hyperplane in a higher dimension. Finally, a sparse modeling method for finding the convex hull is introduced in [39] which can also be utilized for finding an approximate convex hull. The time complexity of the method is experimentally about N^5.

In this chapter, we present a method for finding an approximate convex hull for which the time complexity is $\mathcal{O}(K^{3/2}N^2 \log K)$ where K is the number of iterations of the proposed algorithm and is close to V, the number of vertices of the approximate convex hull, which is usually significantly smaller than N. While much slower than existing methods for low

dimensions, it is quite feasible to apply even in very high dimensions where methods with an exponential dependence on dimension would not even be able to run. The performance of the proposed method compared to the existing methods is investigated through some numerical experiments.

3.2 The Approximate Convex Hull

Let $\mathcal{S}$ be a set of N points in $\mathbb{R}^n$ space, $\mathcal{S} = \{x_1, \ldots, x_N\}$. A key metric that we will utilize not only to define the approximate convex hull but also as an operation in the algorithm is the Euclidean distance of a point $z \in \mathbb{R}^n$ to $\operatorname{Conv} \mathcal{S}$. This function is represented as $d(z, \mathcal{S})$, and its square can be computed via the quadratic program

$$\begin{aligned} d(z,\mathcal{S})^2 = \quad & \min_{\alpha_i} \quad \left\| z - \textstyle\sum_{i=1}^{|\mathcal{S}|} \alpha_i x_i \right\|^2 \\ & \text{s. t.} \quad \alpha_i \geq 0, \quad \textstyle\sum_{i=1}^{|\mathcal{S}|} \alpha_i = 1 \end{aligned} \tag{3.1}$$

Note that $x \in \operatorname{Conv} \mathcal{S}$ if and only if $d(x, \mathcal{S}) = 0$.

This QP problem can be solved using the interior point method [40]. A significant property of the interior point method applied to this problem is that the number of iterations required to reach the optimal solution is independent of n, the dimension of x_i, but rather is dependent on $|\mathcal{S}|$, which is the number of variables. In [41] it is shown that the number of iterations of the interior point method for solving a QP with r independent variables is of order $O(\sqrt{r} \log \frac{r}{\epsilon_0})$ where ϵ_0 is the accuracy precision of the interior point method. The complexity of this problem will also be investigated empirically later.

Since the computational complexity depends on $|\mathcal{S}|$, it is useful for the size of this set to be as small as possible. Fortunately, it is clear that if we have available another set $\mathcal{E}$ (such as the extreme points) such that $\operatorname{Conv} \mathcal{E} = \operatorname{Conv} \mathcal{S}$, then $d(z, \mathcal{E}) = d(z, \mathcal{S})$, and one can use $\mathcal{E}$ for the calculation instead. Finally, we note that the extreme points are the *smallest* subset $\mathcal{E} \subseteq \mathcal{S}$ such that $d(z, \mathcal{E}) = d(z, \mathcal{S})$ for all $z \in \mathbb{R}^n$.

The fact that the extreme points are the smallest subset has a very practical importance in applications that use the convex hull as input, and require computations to be performed

on each element of $\mathcal{E}$. Unfortunately, as discussed in the introduction, the number of extreme points tends to greatly increase in high dimension.

Our objective is to allow for a trade-off between the complexity of the representation of $\operatorname{Conv}\mathcal{S}$ and its accuracy. A subset $\mathcal{E} \subseteq \mathcal{S}$ would be a completely accurate representation if $d(x, \mathcal{E}) = 0$ for all $x \in \operatorname{Conv}\mathcal{S}$. For an approximate representation we will relax the distance to be less than $\epsilon > 0$. Specifically, we have the following definition.

Definition 1. *An* ϵ-approximate convex hull *of* $\mathcal{S}$ *is the convex hull of a minimal subset* $\mathcal{E} \subseteq \mathcal{S}$ *such that for all* $z \in \mathcal{S}$, $d(z, \mathcal{E}) \leq \epsilon$.

Note that we are not guaranteed uniqueness, in that there may be multiple $\mathcal{E}$ with the same number of elements, satisfying the distance requirements. However, as ϵ increases, the size of a minimal representation, $\mathcal{E}$, may be able to decrease.

3.3 Finding the Approximate Convex Hull

The quality of an approximate convex hull with vertices $\mathcal{E}$ is determined by both ϵ and $|\mathcal{E}|$, the number of vertices in that we would like both to be small. Thus, there is a Pareto front of optimal solutions. To explore this front, one could either minimize ϵ for a fixed number of vertices or minimize the number of vertices for a fixed error ϵ; we will focus on the former problem. We formulate the following optimization problem to find the approximate convex hull: find the subset of $\mathcal{S}$ of cardinality V or less which minimizes worst case distance of a point in $\mathcal{S}$ to the convex hull of the subset, which is notated as

$$\min_{\mathcal{E} \in \mathcal{P}_V(\mathcal{S})} \max_{z \in \mathcal{S}} d(z, \mathcal{E}).$$

Unfortunately, this is a combinatorial optimization, requiring a search over each element of $\mathcal{P}_V(\mathcal{S})$. In order to achieve an acceptable performance, we will introduce a greedy method that will provide a suboptimal solution. However, this method will still be able to achieve any desired ϵ performance, although with a potentially larger cardinality than optimal.

3.4 Basic Greedy Algorithm

The greedy algorithm finds each element of $\mathcal{E}$ sequentially. That is, suppose at step k, we have identified a set $\mathcal{E}_k$ with $|\mathcal{E}_k| = k$. We then find

$$\hat{x} = \arg \min_{x \in \mathcal{S} \backslash \mathcal{E}_k} \max_{z \in \mathcal{S} \backslash \mathcal{E}_k} d(z, \mathcal{E}_k \cup x), \tag{3.2}$$

and set $\mathcal{E}_{k+1} = \mathcal{E}_k \cup \hat{x}$. This method initiates with the empty set and terminates either after a fixed number of steps k, or after the desired ϵ has been reached. This optimization problem can be solved by searching over all elements of $\mathcal{S} \backslash \mathcal{E}$ for the outer minimization (as well as over all elements of $\mathcal{S} \backslash \mathcal{E}$ in the inner maximization). While this implies a time complexity of $O(VN^2)$ to reach a cardinality of V, this is a vast improvement to searching over $\mathcal{P}_V(\mathcal{S})$. In addition, we will introduce methods that can reduce the search space as the iterations progress.

3.5 Improved Greedy Algorithm

In this section, two strategies are utilized to speed up the basic greedy method introduced in 3.4. First, the solution to (3.2) is explored in more detail and a searching method is introduced to reduce the required time of solving (3.2). Also, we find and remove elements of $\mathcal{S}$ that are interior points of $\mathcal{E}_k$ during processing. The final method to find the approximate convex hull is presented in Algorithm 2.

3.5.1 Directed Search

Let z_i and x_j be enumerations of $\mathcal{S} \backslash \mathcal{E}_k$. Let E be a matrix with i, jth element given by

$$E_{i,j} = d(z_i, \mathcal{E}_k \cup x_j).$$

The solution to (3.2) can then be obtained by solving

$$\hat{j} = \arg \min_j \max_i E_{i,j}$$

and setting $\hat{x} = x_{\hat{j}}$. In other words, the maximum value of each column of E is evaluated, and then minimum among those maximums is found. However, creating E requires N^2

evaluations of the distance function d.

Improvement can be obtained by searching for the optimal element in a more systematic way. To explain this method, an illustrative example is given. Let $N = 4$ and suppose this is the first iteration, and therefore E is a square matrix with 16 elements. Suppose E is populated with the given numbers in Figure 3.2. The maximum of the columns are 6, 7, 4, and 5. The minimum of these is 4 and therefore $\min_j \max_i E = 4$ and $\hat{j} = 3$.

i \ j	1	2	3	4
1	5	1	2	5
2	1	5	1	2
3	0	2	2	5
4	6	7	(4)	0

Figure 3.2: Illustrative example: Matrix E

Instead of computing all elements of E, we can calculate only those that are necessary. The key observation is that the maximum of any subset of elements in a column provides a lower bound on the maximum over the entire column. In the given example, suppose only the elements in grey are evaluated. All of the elements of the third column have been evaluated, thus the maximum over this column is known to be 4. For the remaining columns, since elements in each column have been evaluated that are greater than 4, the maximum for these columns must be greater than 4. Thus, it is clear that $\min_j \max_i E_{i,j}$ must occur for the third column, without having to evaluate the remaining elements. We use this property to develop the following algorithm:

1. Calculate the first row of E.
2. Find the minimum value of the first row and select the corresponding column as the potential candidate for $\hat{j}$.

3. Compute the next element of that column and re-evaluate which column has the smallest maximum. Set that column as the new potential candidate. If two columns have the same maximum, select one at random.
4. Continue until all elements of one of the columns has been found and its maximum is less than or equal to the maximum of the other columns over the computed elements. In this case, $\hat{j}$ is the index of that column and the optimal value $\hat{\epsilon}$ is the maximum value of that column.

In the best case this only requires computing $2N-1$ elements, although it can be N^2 for the worst case.

This method is implemented in Algorithm 1. This implementation stores the current maximum for each column in the vector $\tilde{E} \in \mathbb{R}^n$, along the row index of currently evaluated elements for each column in vector $\tilde{C} \in \mathbb{R}^n$. $\tilde{E}$ is set to the first row of E at the beginning and $\tilde{C}$ to vector 1 which indicates that one element of each column has been computed. Then in each step, the next element of the column with the minimum value is computed. If the new element is greater than the previous value, it is saved as the greatest value in $\tilde{E}$. The procedure is continued until the counter shows that we have calculated all elements of a column and its corresponding element in $\tilde{E}$ is still the minimum of $\tilde{E}$.

Figure 3.3 illustrates the steps of finding $\min_j \max_i E_{i,j}$ for the previous example using Algorithm 1. The elements that are updated in each step are specified by a gray color.

3.5.2 Interior Points

The algorithm can be further improved by finding and eliminating interior points from consideration.

Definition 2. *$z \in \mathcal{S} \backslash \mathcal{E}_{k+1}$ is an* interior point *of* $\operatorname{Conv} \mathcal{E}_{k+1}$ *if* $d(z, \mathcal{E}_{k+1}) = 0$.

It can be shown that (3.2) always has an optimal solution outside of $\operatorname{Conv} \mathcal{E}_k$, and therefore eliminating the interior points will not change the optimal solution.

Algorithm 1 Finding min max of a matrix

1: **Input:** $\mathcal{S}$, $\mathcal{E}$, $N = |\mathcal{S} \backslash \mathcal{E}|$
2: Calculate the first row of E.
3: Set $\tilde{E}$ to the first row of E and $\tilde{C}$ to $\mathbf{1}^{\mathbf{1}\times\mathbf{N}}$.
4: Find the minimum of $\tilde{E}$ and indicate its index with $\hat{j}$.
5: $\tilde{C}_{\hat{j}} = \tilde{C}_{\hat{j}} + 1$.
6: **while** $\max \tilde{C} \leq N$ **do**
7: Calculate $E_{\tilde{C}_{\hat{j}},\hat{j}}$.
8: Set $\tilde{E}_{\hat{j}}$ to $\max(\tilde{E}_{\hat{j}}, E_{\tilde{C}_{\hat{j}},\hat{j}})$.
9: Find the minimum of $\tilde{E}$ and indicate its index with $\hat{j}$.
10: $\tilde{C}_{\hat{j}} = \tilde{C}_{\hat{j}} + 1$.
11: **end while**
12: $\hat{\epsilon} = \tilde{E}_{\hat{j}}$
13: **Output:** $\hat{j}$, $\hat{\epsilon}$

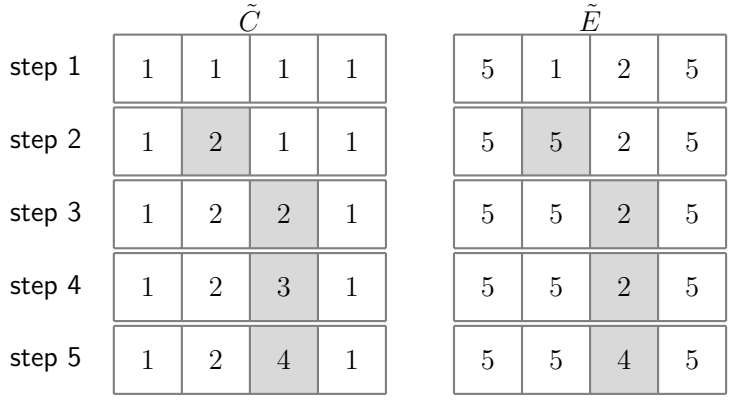

Figure 3.3: Progression of Algorithm 1 when finding $\min_j \max_i E_{i,j}$ for the illustrative example.

Theorem 1. *While* $\operatorname{Conv} \mathcal{E}_k \subset \operatorname{Conv} \mathcal{S}$, (3.2) *has optimal solution* $\hat{x}$ outside *of* $\operatorname{Conv} \mathcal{E}_k$.

Proof. Let us assume that the solution to (3.2) at step k is an interior point of $\operatorname{Conv} \mathcal{E}_k$. In this case, $\operatorname{Conv}(\mathcal{E}_k \cup x_k) = \operatorname{Conv} \mathcal{E}_k$ and it is concluded that $\max_i d(z_i, \mathcal{E}_k \cup x_k) = \max_i d(z_i, \mathcal{E}_k)$. On the hand, for any exterior point x_e, $\max_i d(z_i, \mathcal{E}_k \cup x_e) \leq \max_i d(z_i, \mathcal{E}_k)$ and consequently $\max_i d(z_i, \mathcal{E}_k \cup x_e) \leq \max_i d(z_i, \mathcal{E}_k \cup x_k)$. It is concluded therefore, that solution is not unique and x_e is also an optimal solution of (3.2). Thus, there is always a point at the outside of $\operatorname{Conv} \mathcal{E}_k$ as the optimal solution of (3.2). □

From Theorem 1 the following can be concluded.

- If (3.2) has a unique solution, it will be outside of $\operatorname{Conv}\mathcal{E}_k$.
- If any interior point is the optimal solution of (3.2), the solution is not unique and there is always another optimal solution outside of $\operatorname{Conv}\mathcal{E}_k$.

The second case only happens when some columns of E including some corresponding to the interior points exactly have the same maximum distance. Even in this case, choosing an interior point as x_k is not useful and the algorithm should choose an extreme point of $\operatorname{Conv}\mathcal{E}_k$. In addition, the following is easy to show.

Remark 1. *An interior point of* $\operatorname{Conv}\mathcal{E}_{k_0}$ *will be an interior point of* $\operatorname{Conv}\mathcal{E}_k$ *for* $k \geq k_0$.

These results imply that interior points could be removed from $\mathcal{S}$ as the iterations progress without any changes in the calculated approximate convex hull.

Remark 2. *Since* $E_{i,j} = d(z_i, \mathcal{E}_k \cup x_j)$*, the interior points of* $\mathcal{E}_{k+1}$ *can be found via zeros in the* $\hat{j}$*-th column of* E.

Let $\mathcal{I}_k^{\text{tot}}$ be the set of interior points of $\mathcal{E}_k$, which have been identified at prior steps. At iteration k, the directed search is performed over $(\mathcal{S}\backslash\mathcal{I}_k^{\text{tot}})\backslash\mathcal{E}_k$, and from this the interior points of $\mathcal{E}_k$ from $\mathcal{S}\backslash\mathcal{I}_k^{\text{tot}}$ are identified from zeros in the $\hat{j}$-th column, and placed in set $\mathcal{I}_k$. The set $\mathcal{I}_{k+1}^{tot} = \mathcal{I}_k^{tot} \cup \mathcal{I}_k$ is defined, and the next iteration is performed. This approach significantly decreases the computational time since in each step we remove some of the points for the further processing. It also guarantees that the algorithm acts as preferred and at step $k+1$ will not choose an interior point of $\mathcal{E}_{k+1}$ as the next element of the approximate convex hull.

In addition, it may be that elements of $\mathcal{E}_{k_0}$ become interior points of $\mathcal{E}_k$ for some $k > k_0$. These points can be removed either at the end of each iteration, or at the completion of the algorithm (see Algorithm 2 line 9). This reduces the cardinality of the returned vertex set, which is desirable.

Algorithm 2 Approximate Convex Hull

1: **Input:** $\mathcal{S} = \{x_j\}_{j=1}^{N}$, V, ϵ_{des}
2: **Initialize** $\mathcal{E} \leftarrow$ an extreme point of $\mathcal{S}$, $\mathcal{S}' = \mathcal{S}$.
3: **while** $|\mathcal{E}| < V$ **and** $\epsilon > \epsilon_{des}$ **do** // **Iterate until the desired approximate convex hull is achieved**
4: **for** $x_j \in \mathcal{S}' \backslash \mathcal{E}$ and $z_i \in \mathcal{S}' \backslash \mathcal{E}$ **do**
5: Find $\hat{j}$, ϵ, and $E_{i,\hat{j}}$ by Algorithm 1 where $\hat{j} = \arg\min_j \max_i d(z_i, \mathcal{E} \cup x_j)$.
6: **end for**
7: $\mathcal{I}_k = \{z_i | E_{i,\hat{j}} \leq \epsilon_{des} \quad \forall z_i \in \mathcal{S}' \backslash \mathcal{E}\}$. // **Remove interior points after adding the new extreme point $x_{\hat{j}_k}$ to $\mathcal{E}_{k-1}$ at step k**
8: $\mathcal{S}' \leftarrow \mathcal{S}' \backslash \mathcal{I}_k$, $\mathcal{E} \leftarrow \mathcal{E} \cup x_{\hat{j}}$;
9: **for** $c = 1$ to $|\mathcal{E}|$ **do** // **Check if the elements of $\mathcal{E}_{k-1}$ becomes interior points of $\mathcal{E}_k$**
10: **if** $d(\mathcal{E}_c, \mathcal{E}) = 0$ **then**
11: $\mathcal{E} \leftarrow \mathcal{E} \backslash \mathcal{E}_c$;
12: **end if**
13: **end for**
14: **end while**
15: **Output:** $\mathcal{E}$, ϵ

3.5.3 Initialization

In the first iteration of the greedy method ($k = 1$), (3.2) returns the element closest to the center mass of set $\mathcal{S}$ which is the best approximation when we want to describe set $\mathcal{S}$ only by one element ($V = 1$). However, since we aim to cover the space as much as we can, for $V > 1$ it is better to start from an extreme point of $\operatorname{Conv}\mathcal{S}$. We use the following result to select an extreme point of $\mathcal{S}$ as $\mathcal{E}_1$.

Fact 1. *Let $\mathcal{S}$ to be a set of N points in n dimension. Any element of $\mathcal{S}$ which has a* minimum *or* maximum *in one of the dimensions, is an extreme point of* $\operatorname{Conv}\mathcal{S}$ *[38].*

Remark 3. *In the case that the desired V is significantly bigger than n, one may initialize $\mathcal{E}$ with the best n-complex.*

3.6 Time Complexity Analysis

For Algorithm 2 let K denote the number of iterations of the while loop at line 3. Within this loop, N^2 QP problems are solved to calculate $d(z, \mathcal{E}_k)$, each of which has time complexity $\mathcal{O}(\sqrt{k}\log\frac{k}{\epsilon_0})$ where ϵ_0 is the accuracy precision of the interior point method [41]. As seen,

the time complexity of solving (3.1) is independent of the data dimension n. Therefore, the time complexity of the proposed approximate convex hull is $\mathcal{O}(KN^2\sqrt{K}\log\frac{K}{\epsilon_0})$ or simply $\mathcal{O}(K^{3/2}N^2\log\frac{K}{\epsilon_0})$. Note that $K = V$ unless $\mathcal{E}_{k_0}$ becomes an interior point of $\mathcal{E}_k$ for some $k > k_0$ and is much smaller than N.

Note that by employing the method presented in 3.5.1 the number of QP problems can be much less than N^2. To experimentally show the effect of the directed search method, we ran Algorithm 2 on 390 data sets with 500 random points with $\mathcal{N}(0, I)$. The dimension of the data varied from 2 to 40, with 10 data sets for each dimension.

Figure 3.4 shows the cumulative distribution function (CDF) of the percentage of elements of E which have been computed out of all elements during the experiments. According to the figure, in more than half of the cases we have only computed at most 20% of the elements of E and in 88% of situations we only required to find less than 30% of the elements. Note that the chosen data sets are random. Having a structured data set can even decrease further the number of computed elements.

3.7 Numerical Examples

In this section some numerical examples are given for 2D and higher dimensions for better understanding of the performance of Algorithm 2. In particular, the case where the data clusters around a lower dimensional hyperplane is investigated.

3.7.1 Evaluating the Proposed Algorithm for a 2D Case

Algorithm 2 is implemented on a shown in Figure 3.5. set $\mathcal{S}$ of 20 points in a 2D plane. This set includes 9 vertices and 11 interior points.

Figure 3.5 demonstrates the selected points by the proposed approximate convex hull algorithm for $V = 1, \cdots, 8$ with filled dots. It can be seen that as expected, by choosing a bigger V, the approximate convex hull gets closer to the real convex hull.

Figure 3.6 shows ϵ, the the maximum distance of point in S to the approximate convex hull with V vertices. As seen, by increasing V the error trends down. For $V = 8$, the error

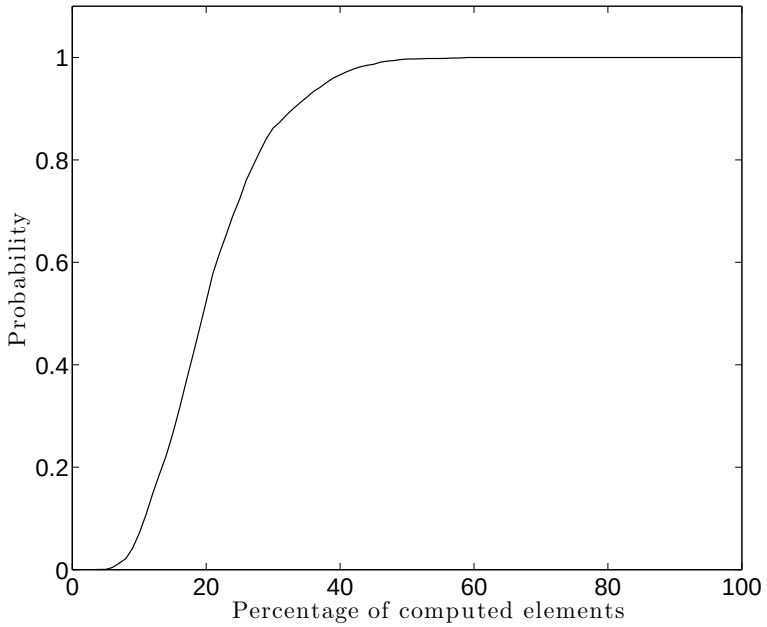

Figure 3.4: Cumulative distribution function of the percentage of computed elements of E to the total elements.

is a small number since the 9th vertex is almost on the one of the edges of the convex hull created by the first eight vertices. The error goes to zero for $V = 9$ since the approximate convex hull algorithm selects all the vertices of $\mathcal{S}$.

This example is indeed a good example of the approximate convex hull application since the convex hull can be precisely approximated by 8 vertices instead of 9 with a negligible error.

3.7.2 Implementation of the Proposed Algorithm for High Dimensions

Here we present a numerical example of finding the approximate convex hull for high dimensional data. Specifically, a data set $\mathcal{S}$ of 4000 points with dimension 20 is selected. The data is the first 20 samples of the step response of the following uncertain mass spring damper system

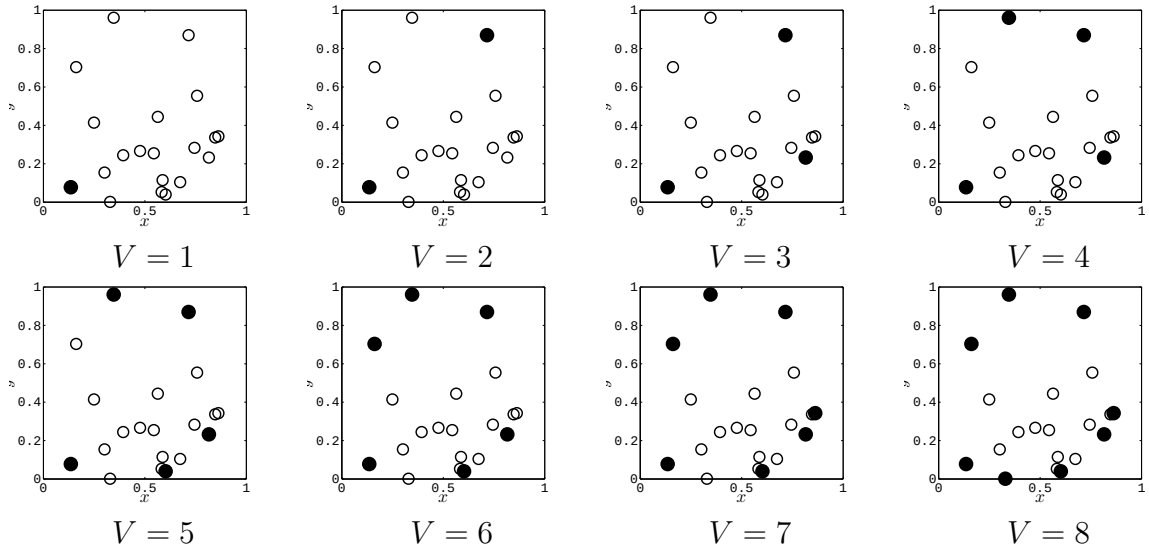

Figure 3.5: The best points found by the approximate convex hull (Algorithm 2) for $V = 1, \cdots, 8$.

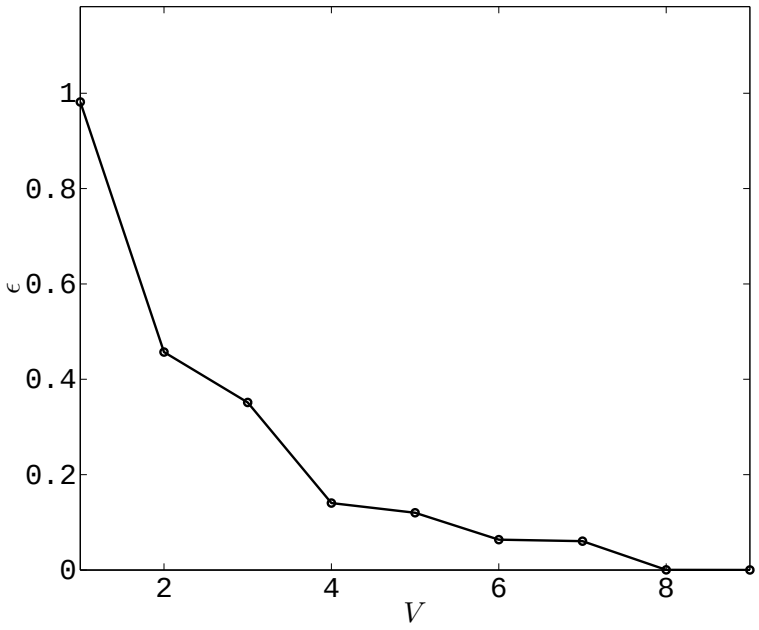

Figure 3.6: The maximum error (ϵ) of the approximate convex hull (Algorithm2) for $V = 1, \cdots, 9$.

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} -c/m & 1 \\ -k/m & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \end{aligned} \tag{3.3}$$

where m is mass, and k and c represent the spring and the damper coefficients with the nominal values 0.72 kg, 0.1 KN/m, 0.3 Kg/s, respectively. For each element of the data set, m, k, and c are randomly generated using a uniform distribution about their nominal values with a $\pm 15\%$ tolerance.

Figure 3.7 presents the result of finding the ϵ-approximate convex hull of $\mathcal{S}$ for $V = 1, \cdots, 20$ where the y-axis is $\epsilon/\kappa_{ss}^{nomi}$ and κ_{ss}^{nomi} indicates the DC gain of the nominal system. As seen, the error decreases to less than 0.093 by taking 7 elements. Choosing 10 elements results in ϵ less than 0.004. By increasing V to 20, the error decreases further. Depending on the application, the appropriate number of elements can be chosen in a trade off with the accuracy.

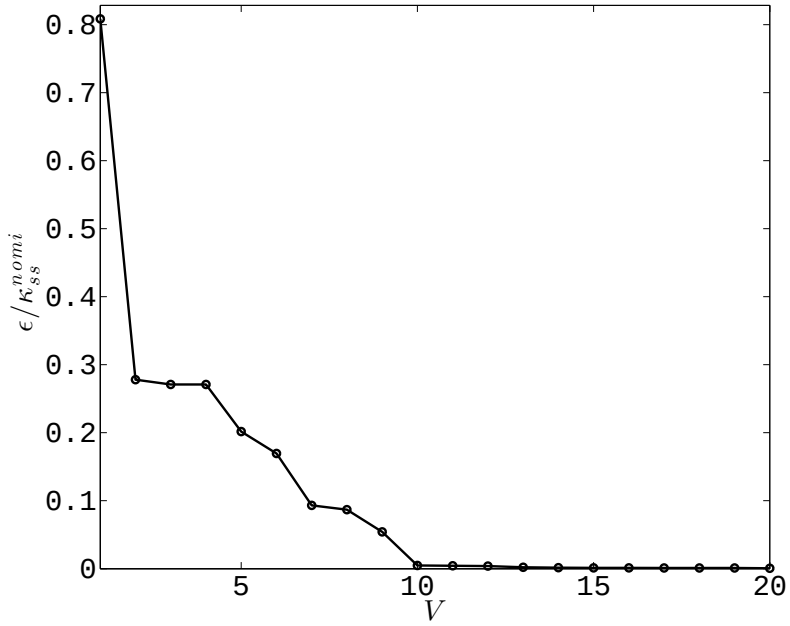

Figure 3.7: Implementation of Algorithm 2 on a set of 4000 samples with $n = 20$.

3.7.3 Low Dimension Data in Higher Dimension

In this example, we focus on the case when the data can be projected to a lower dimension with a small error. In this case, we can define the approximate convex hull with many fewer

vertices than would be required for the full convex hull. To illustrate this, a data set with 200 elements with i.i.d. gaussian distribution $x_i \sim \mathcal{N}(0, I)$ is generated in 5 dimensions, and then projected to a 20 dimension space, after which a small amount of noise is added.

Figure 3.8 indicates the ϵ-approximate convex hull of the data using the proposed method for $V = 1, \cdots, 30$. Note that the error is plotted on a logarithmic axis. It can been seen that by picking 16 points a approximate convex hull can be generated with $\epsilon < .028$. Note that the method presented in [37], would initialize with 21 vertices in a 20 dimensional space.

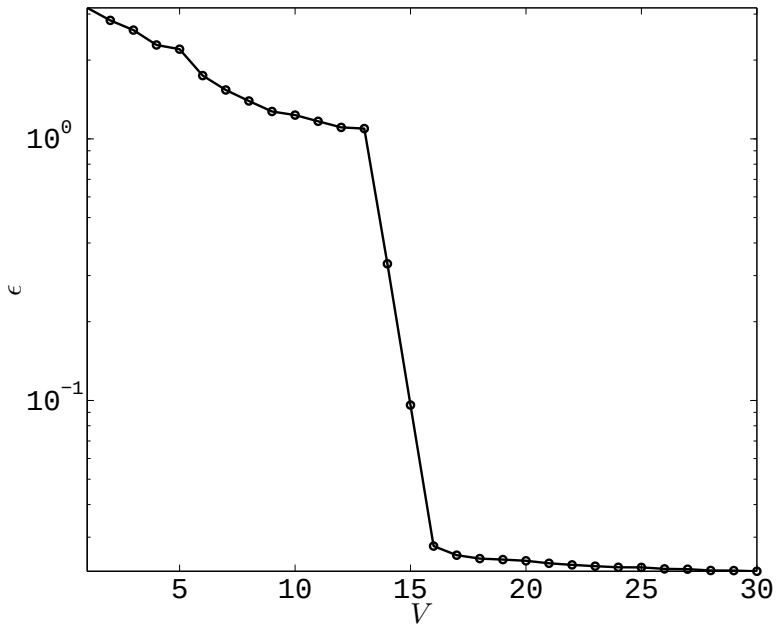

Figure 3.8: Approximate convex hull for a low dimension data set mapped in high dimension.

3.8 Comparison to Other Methods

In order to evaluate the performance of the introduced greedy algorithm, a comparison with commonly used methods is presented in this section. Specifically, we compare the greedy method introduced in this chapter with the sparse method presented in [39] and the convex hull vertices selection method for Support Vector Machine (SVM) presented in [37]. Since the time complexity of Quick-hull method exponentially increases with dimension, this

method could not be run for higher dimensions, and therefore, is not included.

Two different experiments are designed to see the effect of number of data N and data dimension n. Running time and optimality of the computed approximate convex hulls by different methods are chosen as the comparison metrics.

3.8.1 Effect of Number of Points

In this section, we compare the greedy method introduced in this chapter with the sparse method presented in [39] and the convex hull vertices selection method for SVM presented in [37] for fixed dimension. In order to investigate the performance of the three methods with respect to the number of elements N, we vary N from 50 to 1000 and for each case generate 10 random data sets in dimension $n = 10$. The elements of the generated data sets are i.i.d. gaussian with distribution $x_i \sim \mathcal{N}(0, I/10)$. The best ϵ-approximate convex hull then is calculated for $\epsilon = 0.15$. Results are shown by taking average over the ten random data sets.

The *Sedumi* toolbox [42] is used for implementing the second order cone problem used in [39] while the *quadprog* of Matlab has been utilized for the greedy and SVM methods. Both the greedy and vertices selection for SVM methods are initiated from the best n-complex. Figure 3.10 shows the normalized time (by scaling the maximum time to 1) for finding the ϵ-approximate convex hull for the three methods. In addition, Figure 3.10 indicates the number of extreme points of the calculated approximate convex hull for each method.

Although the number of selected extreme points in Sparse method is less than the other methods (Since the solution of this method is close to the global solution), according to Figure 3.9, its time complexity rapidly grows by increasing N. It is worth reporting that because of increasing computational requirements, we were only able to implement this method up to $N = 200$. As seen, the greedy method is slower than the SVM method. However, as it is shown in Figure 3.10, the greedy method selects fewer extreme points than the SVM method. In other words, the presented greedy method is closer to the global solution and results in a better approximation compared to SVM method. In addition,

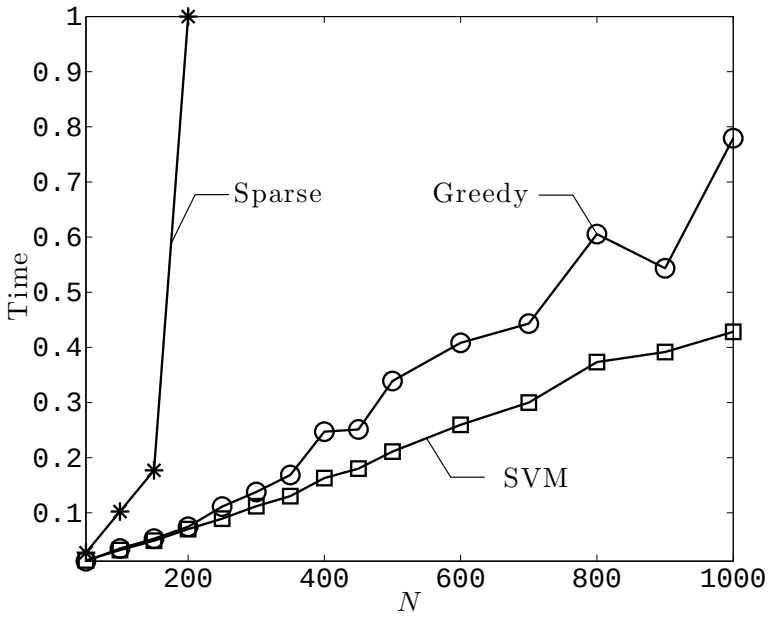

Figure 3.9: Time comparison of the greedy method (introduced in this chapter), vertices selection for SVM (presented in [37]), and Sparse method (presented in [39]). $n = 10$ and N varying from 50 to 1000.

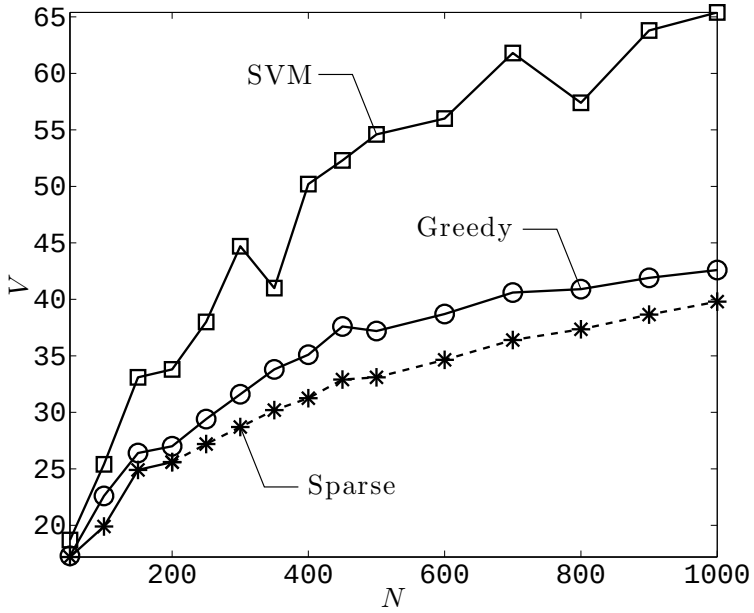

Figure 3.10: Number of extreme points of the calculated ϵ-approximate convex hull by the greed method (introduced in this chapter), vertices selection for SVM (presented in [37]), and Sparse method (presented in [39]). $n = 10$ and N varying from 50 to 1000.

according to Figure 3.9, the time complexity of the greedy method is less than $\mathcal{O}(N^2)$ which is in agreement with the given time complexity in Section 3.6.

3.8.2 Effect of Dimension

The same experiment is performed for the case that N is fixed and the data dimension varies from 2 to 20. N is fixed to 1000 as we prefer to make the analysis on large data sets. Consequently, we have to skip the sparse approach in this analysis. For each dimension, we generate 10 data sets and present their average results. Each data set includes i.i.d. elements with $x_i \sim \mathcal{N}(0, I/n)$ so that at most 25% of the elements are extreme points. The attempt is to find the best ϵ-approximate convex hull where $\epsilon = 0.2$. Both the greedy and SVM methods are initiated from the best n-complex. Figure 3.11 and Figure 3.12 compare the normalized running time and the number of extreme points of the calculated approximate convex hull by the two methods, respectively.

It can be easily seen in Figure 3.12 that although for some cases the greedy method may perform slower than the vertices selection method presented in [37] (Note that the simulation time of the two methods may be different based on the structure of data including number of extreme points and data distribution), the greedy method always picks fewer extreme points. Therefore, in the case that the approximate convex hull is computed off-line and its extreme points are used in an online procedure, the approximate convex hull calculated by greedy method would be a better option due to having fewer extreme points. As seen, the time complexity of the greedy method, unlike the SVM method, does not necessarily increase by increasing dimension. It is a function of K (number of iterations of the greedy method which is close to V), as investigated in Section 3.6. Note that the time complexity of the SVM method increases by dimension since in higher dimensions more partitions are required (see [37] for more detail about its time complexity).

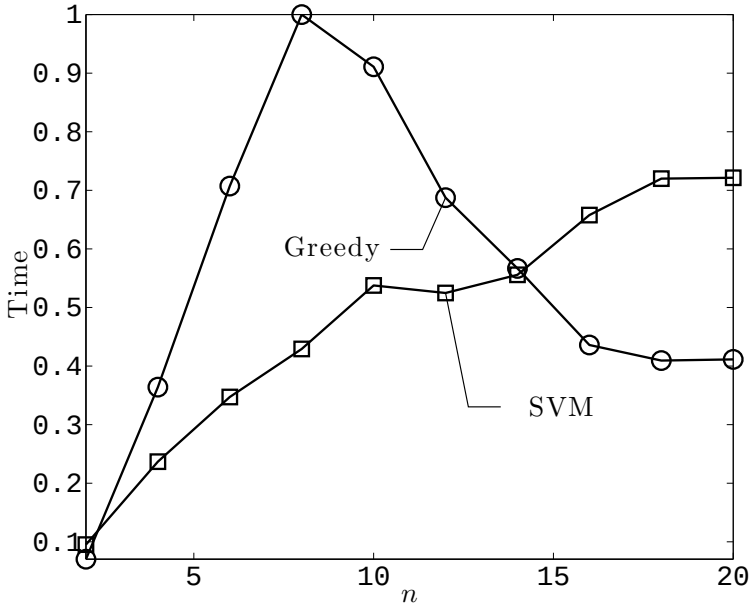

Figure 3.11: Time comparison of the greed method (introduced in this chapter) and vertices selection for SVM (presented in [37]). $N = 1000$ and n varying from 2 to 20.

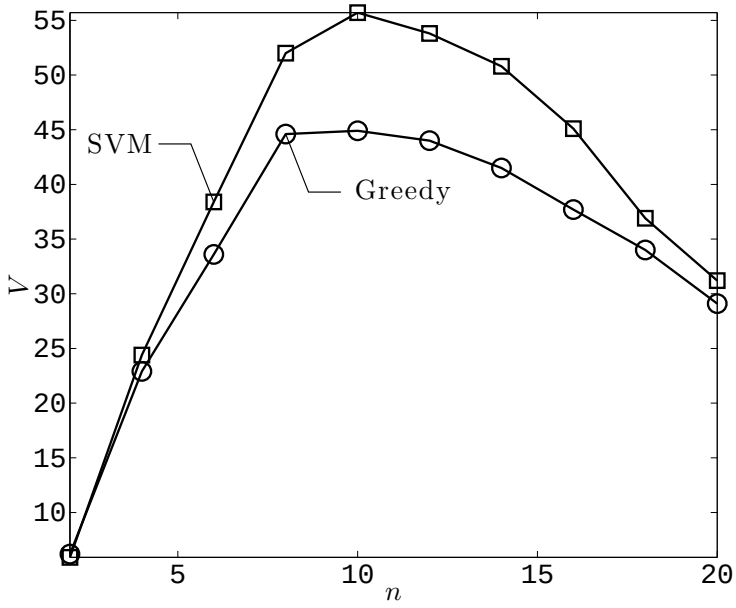

Figure 3.12: Number of extreme points of the calculated ϵ-approximate convex hull by the greed method (introduced in this chapter) and vertices selection for SVM (presented in [37]). $N = 1000$ and n varying from 2 to 16.

3.9 Application in Control: Finding the Polytopic Uncertainty

In this section the usage of the introduced approximate convex hull in computing the polytopic uncertainty description of a system with parametric uncertainty is discussed (see [7] for more details about polytopic uncertainty). Let the state space representation of a system with three masses, two springs, and two dampers be:

$$\begin{aligned}
\dot{x}(t) &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -k_1/m_1 & -b_1/m_1 & k/m_1 & b_1/m_1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ k_1/m_2 & b_1/m_2 & -(k_1+k_2)/m_2 & -(b_1+b_2)/m_2 & k_2/m_2 & b_2/m_2 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & k_2/m_3 & b_2/m_3 & -k_2/m_3 & -b_2/m_3 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ 1/m_1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1/m_3 \end{bmatrix} u(t) \\
y(t) &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} x(t)
\end{aligned} \tag{3.4}$$

where m_*, is mass, and k_*, and b_* indicate the spring and damping coefficients. Let the nominal values of $m_1, m_2, m_3, k_1, k_2, b_1, b_2$ be 1 Kg, 2Kg, 1Kg, 0.5 KN/m, 1KN/m, 1.5 Kg/s, and 1 Kg/s, respectively. Assume the actual parameters vary 5% about their nominal values for masses and spring coefficients and 10 % for damping coefficients.

Let $AB(\theta) = [A(\theta)\ B(\theta)]$ and define set $\mathcal{AB} = \{AB(\theta)|\theta \in \Theta\}$ where Θ is the uncertainty set of parameters. The goal is to determine the uncertainty polytope for the uncertainty range Θ. In other words, find $\mathcal{AB}_{\mathcal{E}} = \{AB_1, \cdots, AB_E\}$ so that any arbitrary set AB_a lies into the convex hull of $\mathcal{AB}_{\mathcal{E}}$, i.e., $AB_a \in \text{Conv}\{AB_1, \cdots, AB_E\}$. We relax the problem to calculate the approximate uncertainty polytope so that any arbitrary AB_a is inside the convex hull of the polytope or not far than ϵ. The distance metric can be chosen based on the application. Let the Frobenius norm be used as the distance metric, i.e. (3.1) has objective of the form

$$\min_{\alpha} \|AB_a - \sum_{i=1}^{E} \alpha_i AB_i\|_F,$$

Note that $AB \in \mathbb{R}^{6\times 8}$ has 16 uncertainty elements, and therefore demands an algorithm suitable for $n = 16$.

Figure 3.13 shows the result of applying the proposed approximate convex hull method to the mass spring damper system where Θ has 500 elements obtained by random sampling of the uncertainty range. According to the figure, by selecting the first 23 calculated extreme points as the uncertainty polytope, we have an approximation with accuracy level of $\epsilon = 0.16$ which the error is about 10% of the error with one element. The accuracy improves by raising the number of extreme points.

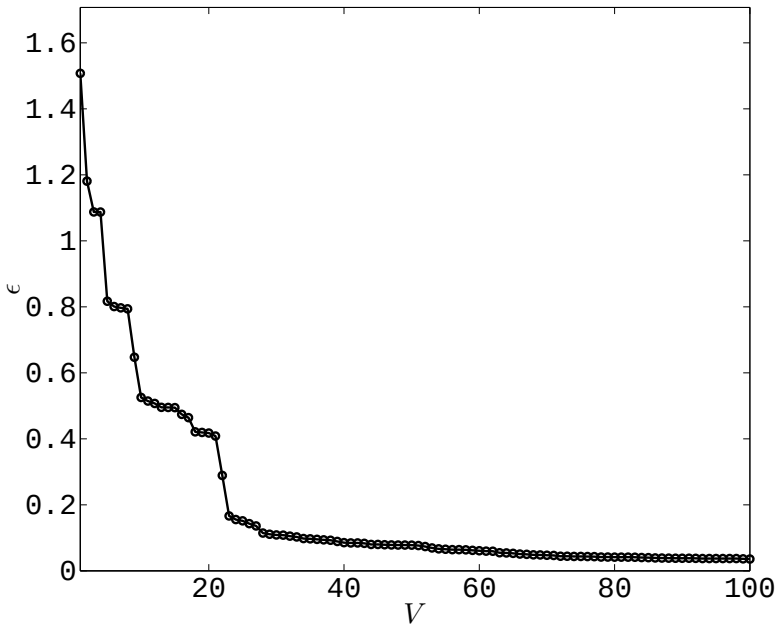

Figure 3.13: Polytopic uncertainty determination using the proposed method.

3.10 Summary

In this chapter, an effective algorithm with time complexity of $\mathcal{O}(K^{3/2}N^2 \log \frac{K}{\epsilon_0})$ was proposed to compute the approximate convex hull of a data set with N points in n dimension. K, the number of iterations of the greedy method, is close to V, the number of vertices of the approximate convex hull. According to the time complexity, this method is highly suitable for the data in high dimensions unlike the family of quick-hull methods. The proposed algorithm uses a greedy method to attempt to find the best approximation to the convex hull for a given number of vertices or a predefined distance violation. A variety of numerical

examples for a wide range of N (up to 4000) and n (up to 50 which can easily go higher) were run to indicate the performance of the introduced greedy method. The comparison results show that the calculated approximate convex hull by the greedy method includes less extreme points than the vertices selection method presented in [37], and significantly faster than the sparse approach [39] and quick-hull family [29].

CHAPTER 4

COMPUTATIONALLY TRACTABLE ROBUST MOVING HORIZON PREDICTOR USING AN APPROXIMATE CONVEX HULL

Output prediction is a vital element of model predictive control. In the case of model uncertainty, a robust prediction of the output is necessary and may improve the MPC efficiency. In this chapter, we propose a new robust estimator/predictor[3] with low computational complexity which will be later used in Chapter 5 to develop a novel robust MPC method for dealing with uncertainties. This predictor benefits a new prediction mapping to characterize the uncertainty. The ϵ-approximate convex hull computation method, introduced in Chapter 3, is used as an important tool for reducing the computational complexity of the predictor[4].

4.1 Introduction

As stated, estimation is a inseparable element of prediction and, therefore, model predictive control. Moving horizon estimation is known as an effective means for state estimation and output prediction for constrained, linear and nonlinear systems. In recent years, MHE has been preferred to the traditional estimation methods such as Kalman filter due to its compatibility with different types of systems and applications[44–47]. This method, first introduced in [48, 49], plays an important role in system monitoring, fault detection, and, especially, model predictive control.

Moving horizon estimation employs an explicit model of the system for estimation. The current state of the system is obtained by solving an optimization problem over a window of

[3]Since an estimator is integrated into the prediction method to find the initial state, we name it estimator/predictor.

[4]This section is based on the paper: Hossein Sartipizadeh and Tyrone L. Vincent. Computationally Tractable Robust Moving Horizon Estimation Using an Approximate Convex Hull *in 55th IEEE Conference on Decision and Control (CDC)*, 2016 [43].

past data (as explained in detail in Section 2.3). Given the explicit model, the current state estimation may be used by a predictor to predict the future output sequence.

In the case of model uncertainty, some robust MHE methods have been proposed (e.g [50–52]) that mainly focus on finding a confidence region for the system states. However, for MPC obtaining confidence intervals for future predictions is of great interest. When the uncertainty set allows for an infinite number of possible system realizations, computing the confidence of region may be challenging. If the states are affine in the uncertain parameters, the region of confidence can be obtained from the parameters that are extreme points of the parameter set [5]. However, if they are not affine, the extreme points may not cover the true output uncertainty. In this case, one may be required to use methods with high computational complexity. In this chapter we focus on reducing the computational complexity of the prediction process when the system states and impulse responses are any arbitrary functions of uncertain parameters.

For clarity, in this study, we only consider parametric uncertainty for Linear Time Invariant (LTI) systems with aim to find guaranteed bounds in which the future system outputs will lie. Let t_0 be the current time. Given the past input-output vectors over a horizon, a robust moving horizon estimator must be able to provide a bound on the all output trajectories compatible with the models in the uncertainty set. Figure 4.1 illustrates this concept. Starting from t_0, the thin solid lines are predicted output trajectories for some specific systems in the uncertainty set while thick solid lines demonstrate the guaranteed bounds on all trajectories provided by the robust predictor.

In the rest of this chapter we introduce a new robust estimator/predictor which can provide a guaranteed bound on the output trajectories with a low computational complexity.

4.2 Methodology

Assume a set of models that represent the system model uncertainty and the past input-output data are given. It is assumed that the set of models characterizes the model uncertainty in the sense that the mapping from past input/output data and future input to

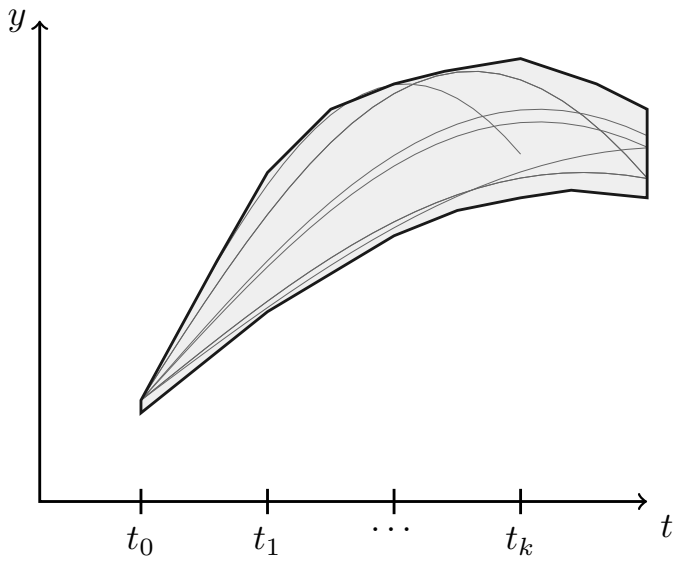

Figure 4.1: Output bounds achieved by a robust predictor

future output for the true system is a convex combination of the mapping elements in the model set. This implies that our desired bounds could be obtained by finding predictions for every element of the model set, and finding the max and min at every time. In general, in the case that there is no restriction on the states, the uncertainty set could be obtained via a sampling of the parameter space. An mentioned in Chapter 1, the motivation for this is provided by the scenario approach for robust control [12, 13, 53]. The scenario approach provides a bound on the number of samples required to attain *a-priori* specified levels of probabilistic guarantee of robustness. Specifically, assume the volume of an uncertainty set build from θ with $|\theta|$ elements is to be covered with maximum violation ϵ^s while the allowed risk of failure is less than a second parameter β. The following number of samples is then required.

$$N_{min} = \frac{2}{\epsilon^s} \ln \frac{1}{\beta} + 2|\theta| + \frac{2|\theta|}{\epsilon^s} \ln \frac{2}{\epsilon^s} \tag{4.1}$$

Let assume the samples taken by the scenario approach give us the bound on the output trajectories (although with a small error corresponding to ϵ^s). Since the number of samples

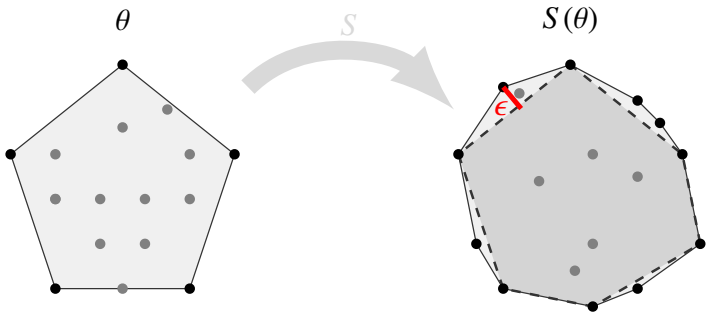

Figure 4.2: Mapping the parameter space to the prediction mapping

is linear in $\frac{1}{\epsilon^s}$, we still need a significant number of samples for a reasonable ϵ^s which could limit the use of this approach in real time. The key idea of the current work is to reduce the number of systems used at run time.

In order to develop the proposed method, we map the finite parameter set, obtained by sampling, to a prediction mapping S. We define this mapping in a way that the extreme points of the convex hull of S give us a bound on the all possible output trajectories and the rest of the points become redundant. Figure 4.2 illustrates the mapping from the parameter set to the prediction mapping. The convex hull boundary is drawn with solid lines. Black dots show the extreme points while gray points are the redundant elements which can be eliminated.

In order to decrease the number of extreme models of the uncertainty set and, consequently, reducing the time complexity, the ϵ-approximate convex hull definition is used, which allows the number of its extreme points to be considerably fewer than for the convex hull. The approximate convex hull is shown with dashed lines in Figure 4.2. This approach gives us the trade off between the number of extreme models of the model set and accuracy.

Choose the set of the approximate convex hull vertices as the model set and denote it with $\mathcal{S}_\mathcal{E}$. $\mathcal{S}_\mathcal{E}$ provides an approximate bound on the output trajectories as illustrated in Figure 4.3 with dashed lines. Because of missing some of the uncertainty realizations, the output may

violate the bounds achieved by approximate convex hull. However, we separately account for the level of approximation (ϵ) by calculating a dynamic upper bound on the output violation, shown by ϵ_k^y at the sampling time k.

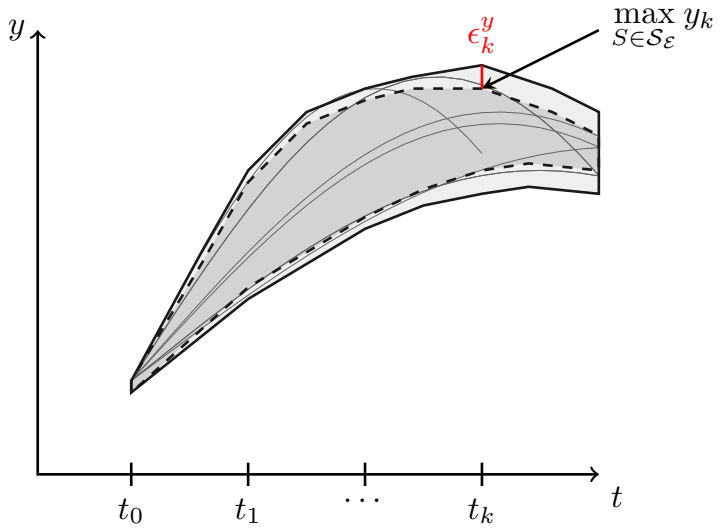

Figure 4.3: Output bounds achieved by the approximate convex hull of the prediction mapping S (dashed lines) and after adding the guaranteed output violation bound ϵ^y (solid lines)

The approximation error of the approximate convex hull can be chosen by the user, and this gives a degree of freedom to designers to make a trade-off between accuracy and computational complexity.

4.3 Introducing the Prediction Mapping

In this section, the new prediction mapping S is defined. This mapping is later used for uncertainty characterization with a low computational complexity.

4.3.1 Modeling Assumptions

Assume the system to be controlled has m inputs and n outputs with the input at sampling time k represented as $u_k \in \mathbb{R}^m$ and the output as $y_k \in \mathbb{R}^r$. A parameterized system model is available of the form

$$\begin{aligned} x_{k+1} &= A(\theta)x_k + B(\theta)\delta u_k \\ \delta y_k &= C(\theta)x_k, \end{aligned} \tag{4.2}$$

where $x \in \mathbb{R}^n$ denotes the vector of system states, $\theta \in \mathbb{R}^q$ is a vector of parameters, and A, B, C are matrices of compatible dimension that are known functions of θ. The input-output feedthrough term is considered zero for convenience. Note that the system model is expressed in terms of input and output variation $\delta u_k := u_k - u_{k-1}$ and $\delta y_k = y_k - y_{k-1}$ (Assume all the outputs are measured, i.e., $y_k^r = y_k^s = y_k$). The system parameters lie in the uncertainty set $\Theta \subset \mathbb{R}^q$.

System (4.2) is assumed to be observable for all $\theta \in \Theta$. The measurements of the system output at sample time k are denoted by z_k, and

$$z_k = y_k + n_k,$$

where $n_k \in \mathbb{R}^r$ represents measurement uncertainty. Because in this study we will be focusing on model uncertainty, our results will be stated for the case when $n_k = 0$. However, the RMHE method presented below can still be applied for non-zero measurement uncertainty, although strict constraint satisfaction would also require accounting for measurement uncertainty due to noise. For an appropriate noise model, i.e., if the noise is assumed to be an i.i.d Gaussian sequence, then this is conceptually straightforward, but the details are eliminated for clarity (it will be discussed in Chapter 5).

4.3.2 Formulation

Without loss of generality, the current sample time is $k = 0$. Given (4.2), the system output variation at time k can be expressed as

$$\delta y_k = C(\theta)A^k(\theta)x_0 + \sum_{\ell=1}^{k} h_\ell(\theta)\delta u_{k-\ell} \tag{4.3}$$

where $h_\ell(\theta) = C(\theta)A^{\ell-1}B(\theta)$ is the impulse response. The first part corresponds to the response of the system due to inputs for $k < 0$, and the second part is the system response due to inputs for $k \geq 0$.

Using this model, the predicted output variation over a length p window into the future can be calculated as [5]

$$Y = Y_0 + \mathcal{I}\left(\mathcal{O}_p(\theta)x_0 + T(\theta)\Delta U_{\mathsf{f}}\right) \tag{4.4}$$

where $Y = \begin{bmatrix} y_1^T & \cdots & y_p^T \end{bmatrix}^T$ is the future output, $Y_0 = \begin{bmatrix} y_0^T & \cdots & y_0^T \end{bmatrix}^T$ is a vector containing repeated elements of the current output, $\Delta U_{\mathsf{f}} = \begin{bmatrix} \delta u_0^T & \cdots & \delta u_{p-1}^T \end{bmatrix}^T$ is the future input variation vector, $\mathcal{O}_p$ is the extended observability matrix defined as

$$\mathcal{O}_p(\theta) = \begin{bmatrix} (CA)^T & (CA^2)^T & \cdots & (CA^p)^T \end{bmatrix}^T, \tag{4.5}$$

$T(\theta)$ is a toeplitz matrix made up of the impulse response,

$$T(\theta) = \begin{bmatrix} h_1(\theta) & 0 & 0 & \cdots & 0 \\ h_2(\theta) & h_1(\theta) & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ h_p(\theta) & h_{p-1}(\theta) & \cdots & h_2(\theta) & h_1(\theta) \end{bmatrix}, \tag{4.6}$$

and $\mathcal{I}$ is the summation operator

$$\mathcal{I} = \begin{bmatrix} I_{r\times r} & 0 & 0 & \cdots & 0 \\ I_{r\times r} & I_{r\times r} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ I_{r\times r} & I_{r\times r} & \cdots & I_{r\times r} & I_{r\times r} \end{bmatrix}. \tag{4.7}$$

For a given θ, the unknowns in (4.4) are the initial state x_0 and current operating point y_0. To determine these, we employ moving horizon estimation. In the absence of measurement noise and disturbances and with a correct model, the MHE state estimate will be correct for a sufficiently long estimation horizon. (This estimator can also reduce the effect of noise and disturbances on the state estimate, but the analysis in this chapter does not explicitly consider noise and disturbances. This will be discussed in the next chapter.) In what follows, we will use p to denote both the estimation horizon (into the past) and the prediction horizon (into the future), although it is quite easy to make these different values.

We assume that the available measurements are the past input sequence u_k, $k = -p, \ldots, -1$, and output measurements z_k, $k = -p, \ldots, 0$. From the input sequence, cal-

[5] Note that $y_k = y_0 + \sum_{i=1}^{k} \delta y_k$

culate δu_k, $k = -p+1, \ldots, -1$. The MHE solves the following optimization problem:

$$\begin{aligned} \min_{y_0, x_{-p+1}, \cdots, x_0} \quad & \sum_{k=-p}^{0} \|v_k\|_R^2 + \sum_{k=-p+1}^{-1} \|w_k\|_Q^2 \\ \text{subject to} \quad & x_{k+1} = A(\theta)x_k + B(\theta)\delta u_k + w_k, \qquad k = -p+1, \cdots, -1 \\ & z_0 - \sum_{\ell=k+1}^{0} \delta z_k = y_0 - \sum_{\ell=k+1}^{0} C(\theta)x_\ell + v_k, \qquad k = -p, \cdots, 0 \end{aligned} \tag{4.8}$$

Since our system is in variational form, the MHE estimates the operating point y_0, along with the state sequence x_k. This is equivalent to the least squares problem

$$\min_X \left\| \Psi(\theta)X - \Gamma(\theta) \begin{bmatrix} \Delta U_{\mathsf{p}} \\ z_0 \\ \Delta Z_{\mathsf{p}} \end{bmatrix} \right\|^2, \tag{4.9}$$

where $X = \begin{bmatrix} y_0^T & x_0^T & \cdots & x_{-p+1}^T \end{bmatrix}^T$, $\Delta U_{\mathsf{p}} = \begin{bmatrix} \delta u_{-1}^T & \cdots & \delta u_{-p+1}^T \end{bmatrix}^T$, $\Delta Z_{\mathsf{p}} = \begin{bmatrix} \delta z_0^T & \cdots & \delta z_{-p+1}^T \end{bmatrix}^T$ and $\Psi(\theta)$ and $\Gamma(\theta)$[6] are given in (4.10).

$$\Psi(\theta) = \begin{bmatrix} 0 & Q^{-\frac{1}{2}} & -Q^{-\frac{1}{2}}A & 0 & \cdots & 0 \\ 0 & 0 & Q^{-\frac{1}{2}} & -Q^{-\frac{1}{2}}A & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots & \\ 0 & 0 & \cdots & 0 & Q^{-\frac{1}{2}} & -Q^{-\frac{1}{2}}A \\ R^{-\frac{1}{2}} & 0 & 0 & \cdots & 0 & 0 \\ R^{-\frac{1}{2}} & -R^{-\frac{1}{2}}C & 0 & \cdots & 0 & 0 \\ \vdots & & & \ddots & \ddots & \\ R^{-\frac{1}{2}} & -R^{-\frac{1}{2}}C & \cdots & -R^{-\frac{1}{2}}C & -R^{-\frac{1}{2}}C & -R^{-\frac{1}{2}}C \end{bmatrix}, \Gamma(\theta) = \begin{bmatrix} Q^{-\frac{1}{2}}B & & & & & \\ & \ddots & & & 0 & \\ & & Q^{-\frac{1}{2}}B & & & \\ & & & R^{-\frac{1}{2}} & & \\ & & & R^{-\frac{1}{2}} & -R^{-\frac{1}{2}} & \\ & 0 & & \vdots & \vdots & \ddots \\ & & & R^{-\frac{1}{2}} & -R^{-\frac{1}{2}} & \cdots & -R^{-\frac{1}{2}} \end{bmatrix}. \tag{4.10}$$

Lemma 1. *If system* (4.2) *is observable for all* θ*,* $p \geq n$ *and* Q *and* R *are invertible, then* (4.9) *has a unique solution.*

Proof. Using matrix row operations and using the property that $\text{rank}(\Psi) = \text{rank}(\Psi Q)$ for any invertible matrix Q, it can be shown that matrix Ψ is full column rank if and only if the matrix $\mathcal{O}(\theta) = \begin{bmatrix} C^T & (CA)^T & \cdots & (CA^{p-1})^T \end{bmatrix}^T$ is full column rank, which is equivalent to the observability of (4.2). Therefore, a unique minimizer X exists. □

[6]The dimensions of the presented matrices are as $\Psi(\theta) \in \mathbb{R}^{r(p+1)+n(p-1) \times pn+r}$, $\Gamma(\theta) \in \mathbb{R}^{r(p+1)+n(p-1) \times r(p+1)+m(p-1)}$.

Let $\hat{X}$ be the minimizer of (4.9). The estimated initial state $\hat{x}_0$ and operating point $\hat{y}_0$ are obtained as

$$\begin{aligned}\hat{x}_0 &= \begin{bmatrix}0 & I_{n\times n} & 0 & \cdots & 0\end{bmatrix}\Psi^{\dagger}(\theta)\Gamma(\theta)\begin{bmatrix}\Delta U_{\mathsf{p}}\\ z_0\\ \Delta Z_{\mathsf{p}}\end{bmatrix},\\ \hat{y}_0 &= \begin{bmatrix}I_{r\times r} & 0 & 0 & \cdots & 0\end{bmatrix}\Psi^{\dagger}(\theta)\Gamma(\theta)\begin{bmatrix}\Delta U_{\mathsf{p}}\\ z_0\\ \Delta Z_{\mathsf{p}}\end{bmatrix}.\end{aligned} \tag{4.11}$$

where $\Psi^{\dagger}(\theta) = (\Psi^T(\theta)\Psi(\theta))^{-1}\Psi^T(\theta)$ is the pseudo inverse of $\Psi(\theta)$. Note that for an uncertain system, both x_0 and y_0 are functions of the uncertain parameters. Replacing x_0 and y_0 with $\hat{x}_0$ and $\hat{y}_0$ respectively in (4.4), the predicted output can be written as

$$Y(\theta) = M(\theta)\begin{bmatrix}\Delta U_{\mathsf{p}}\\ z_0\\ \Delta Z_{\mathsf{p}}\end{bmatrix} + N(\theta)\Delta U_{\mathsf{f}} \tag{4.12}$$

where $N(\theta) = \mathcal{I}T(\theta)$ and

$$M(\theta) = (\begin{bmatrix}I & 0 & \cdots & 0\end{bmatrix} + \mathcal{I}\mathcal{O}_p\begin{bmatrix}0 & I & 0 & \cdots & 0\end{bmatrix})\Psi^{\dagger}(\theta)\Gamma(\theta). \tag{4.13}$$

As (4.12) implies, given known past input-output vectors and future output, M and N determine the output prediction. Therefore, define the prediction mapping as $S = \{M, N\}$.

4.4 Output Bounds-Reduction of the Uncertainty Set

Thus far we have introduced a prediction mapping S. In this section, we explain how to determine guaranteed upper and lower bounds on the prediction error given the true model has parameters in a defined set Θ.

Let the upper and lower bounds of all possible realizations over the prediction horizon be denoted as $\bar{Y} = \begin{bmatrix}\bar{y}_1^T & \cdots & \bar{y}_p^T\end{bmatrix}^T$ and $\underline{Y} = \begin{bmatrix}\underline{y}_1^T & \cdots & \underline{y}_p^T\end{bmatrix}^T$, respectively where $\bar{y}_k = \max_{\theta\in\Theta} y_k$ and $\underline{y}_k = \min_{\theta\in\Theta} y_k$. Computation of these bounds can be accomplished through state estimation and simulation for each element of Θ. However, in many cases, this will be computationally intractable.

In the case that impulse response of the system is affine in the uncertain parameters, some simplification is possible. In this case, one can define the set $\Theta_{\mathcal{C}}$ as the extreme points of

Conv Θ (that is, cannot be represented as a convex combination of the remaining elements). In this case, we can find $\bar{y}_k = \max_{\theta \in \Theta_{\mathcal{C}}} y_k$ and $\underline{y}_k = \min_{\theta \in \Theta_{\mathcal{C}}} y_k$ [5, 6]. However, there are some issues with this. First, if there are many parameters, $\Theta_{\mathcal{C}}$ can still be quite large. Secondly, if the impulse response is not affine in the parameters, the parameters of interest will be those that define the extreme points of Conv $Y(\theta)$, but these will be more difficult to find. A practical method is to create $\Theta_{\mathcal{C}}$ by sampling from Θ similar to scenario approach, but again will require a large number of elements in order to provide good coverage. To solve this problem and reduce the time complexity as illustrated in Figure 4.3, we utilize an approximate convex hull and develop a new approach for finding $\bar{Y}$ and $\underline{Y}$ that accounts for the level of approximation.

We assume that $|\Theta|$ is finite (obtained from a sampling procedure), and establish an arbitrary indexing of the elements of Θ as θ_i, $i = 1, \cdots, |\Theta|$. Define the sets

$$\mathcal{N} = N(\Theta) = \{N(\theta_i) | i = 1, \cdots, |\Theta|\}, \mathcal{M} = M(\Theta) = \{M(\theta) | i = 1, \cdots, |\Theta|\}.$$

Each row of $M(\theta)$ and $N(\theta)$ defines a linear mapping from input/output data to a predicted output at a particular number of samples in the future. Given a set of matrices, $N \in \mathbb{R}^{rp \times mp}$, $M \in \mathbb{R}^{rp \times m(p-1)+r(p+1)}$ and $\alpha \in \mathbb{R}^{|\Theta|}$, $1 \leq s_1 \leq \infty$, set $\mathcal{S} = \{\mathcal{M}, \mathcal{N}\}$ and $S = \{M, N\}$. Let

$$\lambda_k(S, \mathcal{S}, s_1, \alpha) = \|^k M - \sum_{i=1}^{|\Theta|} \alpha_i \, {}^k M(\theta_i)\|_{s_1} \tag{4.14a}$$

$$\lambda_{p+k}(S, \mathcal{S}, s_1, \alpha) = \|^k N - \sum_{i=1}^{|\Theta|} \alpha_i \, {}^k N(\theta_i)\|_{s_1} \tag{4.14b}$$

for $k = 1, \cdots, p$. This gives a measure of the error between M,N, and a specific linear combination of the elements of $\mathcal{M}$, $\mathcal{N}$. Let λ be the vector with elements λ_i. Define the distance of S to the convex hull of set $\mathcal{S}$ as [54]

$$d(S, \mathcal{S}, s_1) = \min_{\alpha \in C} \|\lambda(S, \mathcal{S}, s_1, \alpha)\|_\infty \tag{4.15}$$

where $C = \{\alpha \in \mathbb{R}^l | \sum_{i=1}^{l} \alpha_i = 1, \alpha_i \geq 0\}$.

Theorem 2. *Let* $\operatorname{Conv}\mathcal{S}_\mathcal{E}$ *be the* ϵ*-approximate convex hull of* $\mathcal{S}$*(i.e.,* $d(\mathcal{S}_\mathcal{E}, \mathcal{S}, s_1) < \epsilon$*). Let* $s_2 = \frac{s_1}{1-s_1}$*. Then,*

$$\begin{aligned} \bar{y}_k &\leq \epsilon_k^y + \max_{S\in\mathcal{S}_\mathcal{E}} y_k \\ \underline{y}_k &\geq -\epsilon_k^y + \min_{S\in\mathcal{S}_\mathcal{E}} y_k \end{aligned} \tag{4.16}$$

are two guaranteed upper ($\bar{y}_k$*) and lower(*$\underline{y}_k$*) bounds on output, where*

$$\epsilon_k^y = \epsilon \left\| \begin{bmatrix} \Delta U_\mathsf{p} \\ \Delta Z_\mathsf{p} \end{bmatrix} \right\|_{s_2} + \epsilon_k \|\Delta U_\mathsf{f}\|_{s_2} \tag{4.17}$$

and

$$\epsilon_k = \max_{S\in\mathcal{S}} \min_{\alpha\in C} \lambda_{p+k}(S, \mathcal{S}_\mathcal{E}, s_1, \alpha) \qquad k = 1, \cdots, p \tag{4.18}$$

and y_k *is the prediction using* (4.12) *with matrices contained in* $\mathcal{S}_\mathcal{E}$*.*

Proof. See Appendix A □

Remark 4. ϵ_k *is the bound on the error of the* k*-step ahead prediction of output due to the convex hull approximation, and will naturally rise with increasing* k*. The following relationship holds for* ϵ_k*:* $\epsilon_1 \leq \cdots \leq \epsilon_{p-1} \leq \epsilon_p \leq \epsilon$*.*

Remark 5. *At steady state, without any noise or disturbance, the input and output variations are zero. The calculated bound* ϵ^y*, therefore, converges to zero.*

Remark 6. *When* $\mathcal{S}_\mathcal{E}$ *is the exact convex hull of* $\mathcal{S}$*,* $\epsilon^y = \epsilon = 0$*.*

Remark 7. *Three options* $\{s_1 = 1, s_2 = \infty\}$*,* $\{s_1 = \infty, s_2 = 1\}$*, and* $\{s_1 = 2, s_2 = 2\}$ *can be chosen based on the application.*

Remark 8. *To compute* $\mathcal{S}_\mathcal{E}$*, the approximation algorithm introduced in Chapter 3 is used with the metric* $d(.)$ *given in* (4.15)*.*

In the remainder of the thesis, $\mathcal{S}_\mathcal{E} = \{S_1, \cdots, S_E\} \subseteq \mathcal{S}$ with the corresponding set of parameters $\Theta_\mathcal{E} = \{\theta_1, \cdots, \theta_E\}$ is called the *model set*. This set includes E elements called *extreme models* which denote the extreme points of the convex hull of $\mathcal{S}_\mathcal{E}$ or equivalently

the extreme points of the approximate convex hull of $\mathcal{S}$. Algorithm 3 briefly explains how to implement the proposed robust predictor. Θ indicates the finite set of samples taken by scenario approach. Note that in Algorithm 3 lines 1 to 13 are performed off-line while lines 14 and 15 are processed on-line to find the dynamic violation bounds in future (ϵ_k^y).

Increasing the number of extreme models E results in a smaller ϵ and therefore ϵ_k. However, since the number of constraints linearly increases with the number of extreme models, the computational complexity increases. Therefore, there is a trade off between the uncertainty violation level and the computational complexity.

Algorithm 3 Proposed RMHE

1: **Input:** Θ, E **or** ϵ , Q, R
2: $\mathcal{S} = \emptyset$.
3: **for** $i = 1 : |\Theta|$ **do** // **Map the uncertainty Θ to the prediction mapping S**
4: Form the system matrices $A(\theta_i)$, $B(\theta_i)$, $C(\theta_i)$.
5: Find the impulse response matrix $H(\theta_i)$.
6: Form $\Psi(\theta_i)$, $\Gamma(\theta_i)$, and $N(\theta_i)$.
7: Form $M(\theta_i)$.
8: Form $S(\theta_i)$ using $M(\theta_i)$ and $N(\theta_i)$.
9: $\mathcal{S} \leftarrow \mathcal{S} \cup S(\theta_i)$.
10: **end for**
11: Given $\mathcal{S}$, compute $\mathcal{S}_\mathcal{E}$ for the given E or ϵ. // **Find the model set $\mathcal{S}_\mathcal{E}$ using approximate convex hull**
12: $\Theta_\mathcal{E} = \{\theta | S(\theta) \in \mathcal{S}_\mathcal{E}\}$.
13: Compute $\epsilon_1, \cdots, \epsilon_p$ using (5.10).
14: Given $\epsilon, \epsilon_k, \Delta U_\mathsf{p}, \Delta Z_\mathsf{p}$, compute ϵ_k^y using (5.9). // **Calculate the output violation bound ϵ^y**
15: Find the uncertainty bounds $\bar{y}_k$ and $\underline{y}_k$ from (4.16).
16: **Output:** $\bar{y}_k$, $\underline{y}_k$ for $k = 1, \cdots, p$.

4.5 Illustrative Example

A 2nd order mass-spring-damper system is considered as the case study to show the performance of the introduced RMHE. Let the continuous time state space representation of the mass-spring-damper system be given as

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} -c/m & 1 \\ -k/m & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \end{aligned} \tag{4.19}$$

where m, is mass, and k, and c indicate the spring and damping coefficients with the nominal values of 0.72 Kg, 0.2 KN/m, and 0.4 Kg/s, respectively. Assume the actual parameters vary 15% about their nominal values which implies the following uncertainty ranges,

$$m = \begin{bmatrix} 0.612 & 0.828 \end{bmatrix}, k = \begin{bmatrix} 0.425 & 0.575 \end{bmatrix}, c = \begin{bmatrix} 0.085 & 0.115 \end{bmatrix}.$$

which results in different systems with a wide range of dynamics and DC gains. Figure 4.4 shows the step response of some random systems with parameters in the uncertainty ranges. Dashed lines indicate the upper and lower bounds of the step responses. The uncertainty in the system dynamics and DC gain is noticeable. The goal is to employ a p-step ahead predictor over time with p=10.

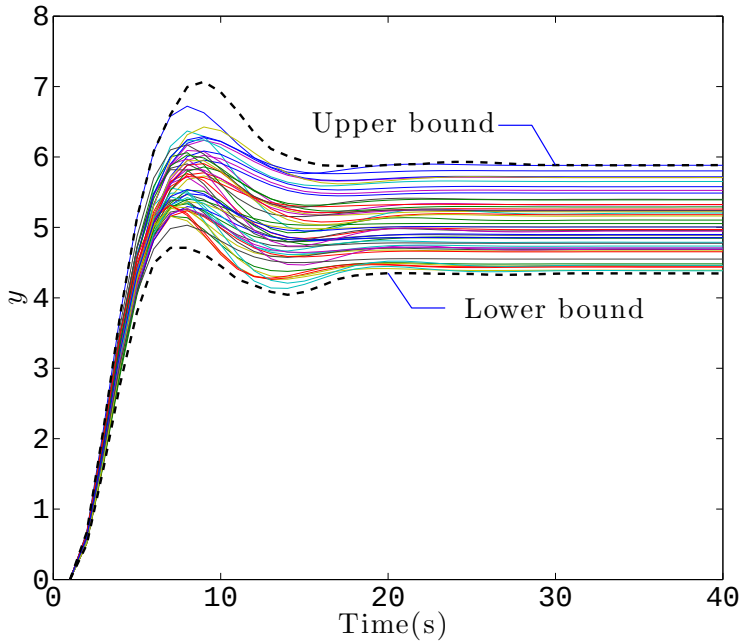

Figure 4.4: Step response range of uncertain systems.

Since the states are not affine in the uncertain parameters, the extreme points of the parameters subspace can not completely cover the uncertainty. In order to characterize the uncertainty, we randomly take 500 samples from the uncertain parameter set and generate $\Theta = \{\theta_1, \cdots, \theta_{500}\}$. Then, using a sampling time $1s$, and weighting matrices $Q = 1$ and $R = 1$, we create set $\mathcal{S}$ and find the approximation of its convex hull, off-line. In this example,

the method has been performed with $s_1 = s_2 = 2$. Let $\operatorname{Conv} \mathcal{S}_{\mathcal{E}}^{MSD}$ be the approximate convex hull of $\mathcal{S}$ computed by the approximate convex hull computation method presented in Chapter 3.

Figure 4.5 shows the approximation error ϵ as a function of E, the number of elements of $\mathcal{S}_{\mathcal{E}}^{MSD}$, for $E = 1, \cdots, 20$. As expected, by selecting more elements, the resulting approximate convex hull is closer to the exact convex hull of $\mathcal{S}^{MSD}$ and consequently ϵ decreases. However, there is also a clear point of diminishing returns.

As seen, the proposed method gives us the freedom to make a trade-off between the number of extreme points and the resulting accuracy. Choose $E = 12$ with corresponding approximation error $\epsilon = 0.065$. In Figure 4.6, ϵ_k is plotted for $k = 1, \cdots, 10$. As expected, the uncertainty grows in time since the uncertainty of the previous steps are accumulated for the next step.

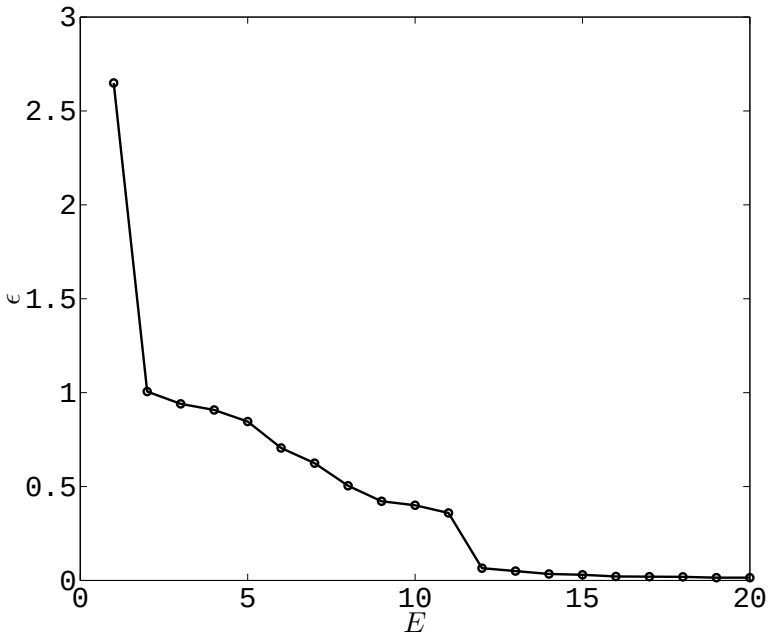

Figure 4.5: Approximate convex hull accuracy using E extreme points.

Given ϵ, ϵ_k, and $\mathcal{S}_{\mathcal{E}}$, we use (4.16) to find the 10-step ahead output prediction bounds for a predefined pulsed input. The plant is run 35 s, and after that we apply the estimator. Figure 4.7 shows the estimation result for a random system. Note that the actual output is

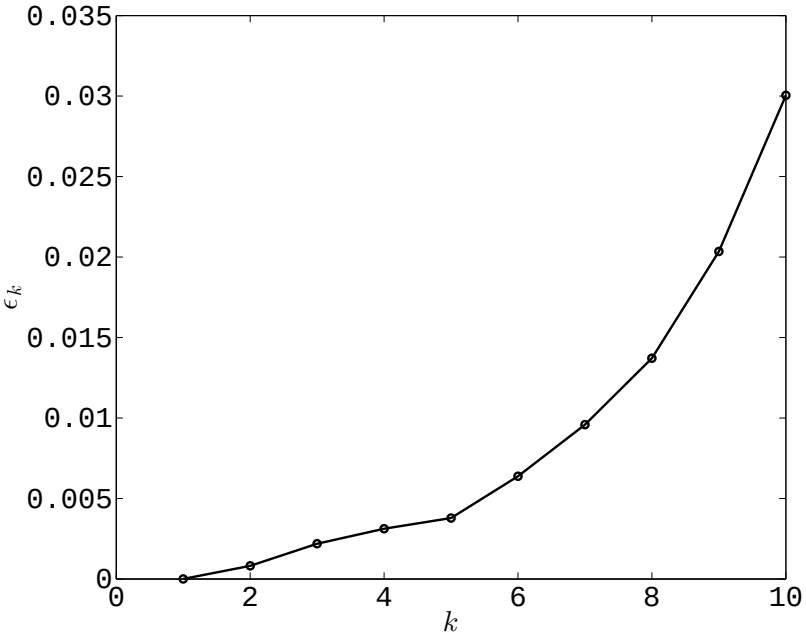

Figure 4.6: ϵ_k for $k = 1, \cdots, 10$ when $E = 12$.

always bounded by the given uncertainty bounds. The output touches both the upper and lower bounds over time but does not violate them.

Figure 4.8 demonstrates the behavior of ϵ^y during estimation. Figure 4.8a shows the output uncertainty level corresponding to the approximation error related to the past data ($\epsilon \left\| [\Delta U_{\mathsf{p}}^T \quad \Delta Z_{\mathsf{p}}^T] \right\|$), and Figure 4.8b indicates the worst uncertainty level caused by the future input variations($\epsilon_p \|\Delta U_{\mathsf{f}}\|$). As seen, the both terms go to zero at the steady state (see Remark 5) which significantly reduces the conservatism of the proposed method.

4.6 Summary

A novel robust moving horizon estimation/prediction approach with a new uncertainty characterization for dealing with the parametric uncertainty was presented in this chapter. A new prediction mapping and the approximate convex hull computation method introduced in Chapter 3 were the main elements of the proposed approach for characterizing the uncertainty. According to Theorem 2 and confirmed by the illustrative example, the output uncertainty level by the approximation process is zero at the steady state. Otherwise, it is

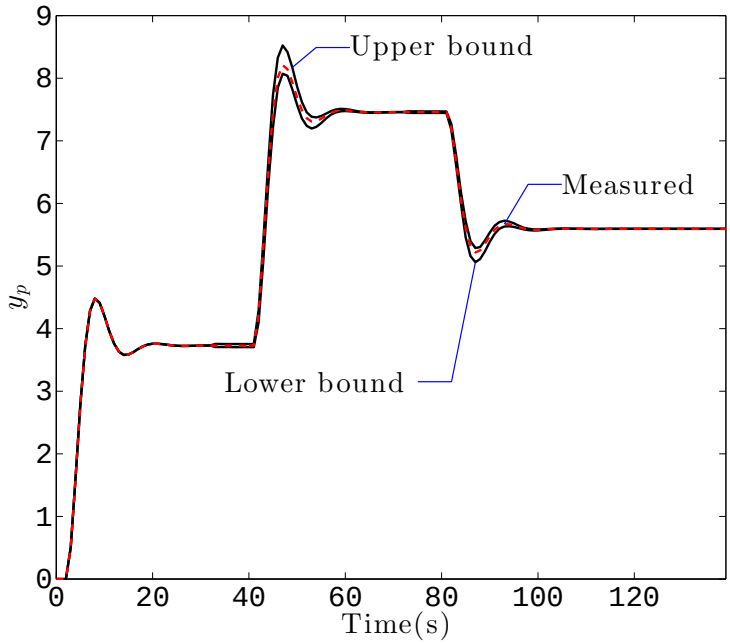

Figure 4.7: 10-step ahead output prediction uncertainty bound. The estimator is applied after 33 s.

a function of ϵ where ϵ can be decreased by increasing the number of extreme models of the computed approximate convex hull. This method allows the designer to make a trade-off between the accuracy and computational complexity by providing a degree of freedom on the number of extreme models. The simulation results confirm that the output is always bounded by the given uncertainty bounds.

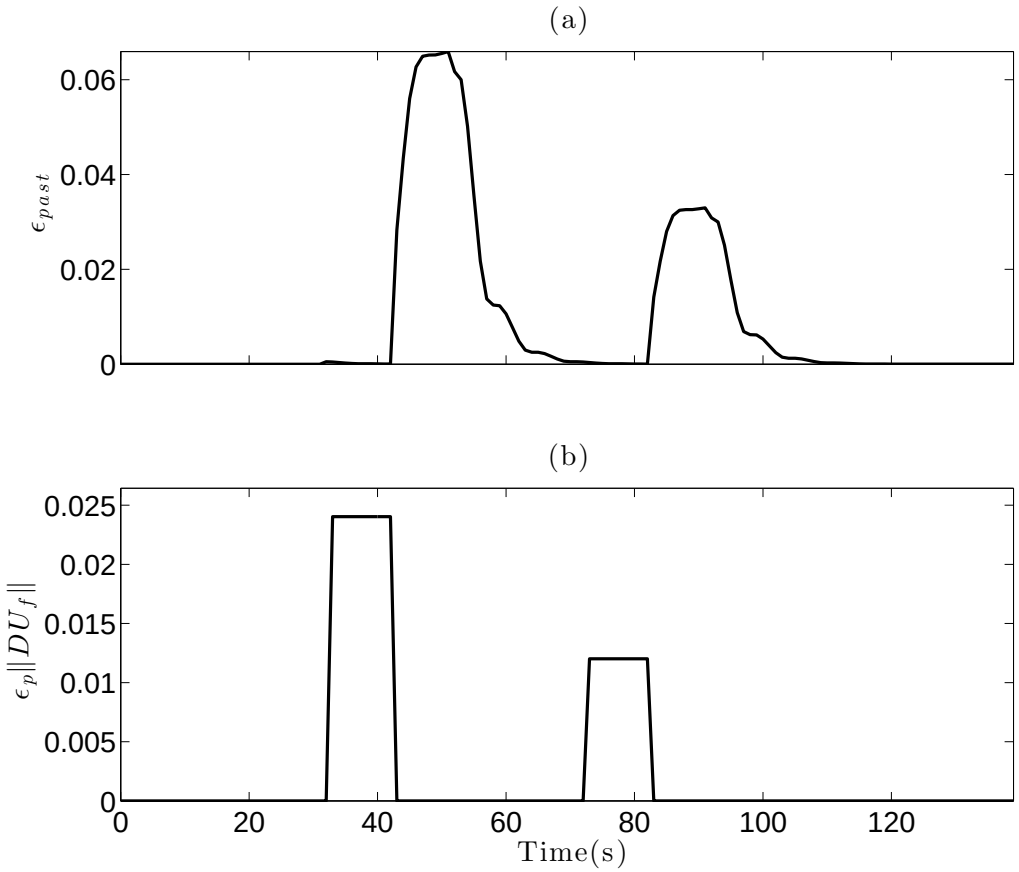

Figure 4.8: Output uncertainty (ϵ_y) corresponding to the convex hull approximation. (a) caused by past data (b) caused by future data

CHAPTER 5

A NEW ROBUST MPC APPROACH USING AN APPROXIMATE CONVEX HULL

This chapter is based on one published paper [7] and one submitted paper [8].

5.1 Introduction

In Chapter 4 we introduced a mapping, S, to help characterize the parametric uncertainty. The main element of the proposed method was the computation of a convex hull that *approximately* covers the system uncertainty in the new prediction mapping. A novel robust moving horizon predictor were introduced to find the guaranteed bounds on the all possible output trajectories. Since the output prediction is a vital part of model predictive control, in this chapter, the proposed robust estimator/predictor is used to design a novel robust MPC. It will be shown through simulation results that the proposed controller has a strong performance in the control of systems with parametric uncertainty and processes with measurement noises.

5.2 RMPC Optimization Problem

In MPC, an objective function and constraints are defined that are functions of the future input and output. In this chapter, we specialize to output constraints that can be expressed as desired minimum and/or maximum values of the output, i.e. $b^{min} \leq Y \leq b^{max}$, where $b^{\min}$ and $b^{\max}$ are vectors the same length as Y. The scalar objective function of the future input and output is designated $J(U_{\mathsf{f}}, Y)$ and is assumed to be convex. The desired input constraints are given by $f(U_{\mathsf{f}}) \leq c$ where f is a linear function of U_{f}. The robust MPC strategy is to minimize the worst case objective function over all $\theta \in \Theta$.

[7] Hossein Sartipizadeh and Tyrone L. Vincent. Uncertainty Characterization for Robust MPC using an Approximate Convex Hull Method. *in American Control Conference, 2016* [54].

[8] Hossein Sartipizadeh and Tyrone L. Vincent. A new Robust MPC approach using an Approximate Convex Hull. *submitted to Automatica, 2016* [55].

$$\begin{aligned} \min_{U_{\mathsf{f}}} \ \max_{\theta\in\Theta} \quad & J(U_{\mathsf{f}}, Y(\theta)) \\ \text{subject to} \quad & Y(\theta) = M(\theta)\begin{bmatrix}\Delta U_{\mathsf{p}} \\ z_0 \\ \Delta Z_{\mathsf{p}}\end{bmatrix} + N(\theta)\Delta U_{\mathsf{f}} \\ & b^{min} \le Y(\theta) \le b^{max} \\ & f(U_{\mathsf{f}}) \le c \end{aligned} \tag{5.1}$$

This problem can be written in epigraph form as follows.

$$\begin{aligned} \min_{U_{\mathsf{f}}, t} \quad & t \\ & Y(\theta) = M(\theta)\begin{bmatrix}\Delta U_{\mathsf{p}} \\ z_0 \\ \Delta Z_{\mathsf{p}}\end{bmatrix} + N(\theta)\Delta U_{\mathsf{f}} \\ \text{subject to} \quad & b^{min} \le Y(\theta) \le b^{max} \\ & J(U_{\mathsf{f}}, Y(\theta)) < t \qquad \forall\theta \in \Theta \\ & f(U_{\mathsf{f}}) \le c \end{aligned} \tag{5.2}$$

The solution to (5.2) is the same as (5.1) and thus the problems are equivalent.

As noted, $|\Theta|$ is finite. However, if $J(.)$ is linear in Y, the solution of (5.2) over Θ is also optimal for the infinite uncertainty set

$$\{\theta \,|M(\theta) \in \operatorname{Conv} M(\Theta), N(\theta) \in \operatorname{Conv} N(\Theta)\}.$$

In the case that $J(.)$ are not linear in Y but is convex, the solution to (5.2) will be feasible for the optimization problem with infinite uncertainty set, but suboptimal. However, feasibility can be more important than optimality in applications where the control system must satisfy constraints to ensure that the system is not damaged.

5.3 Reduction of the Uncertainty Set

When Θ contains many elements, the computational complexity of (5.2) may be too large to meet the sampling time requirements of real-time control. In Chapter 4, we discussed an approach that finds a smaller number of elements that can approximately represent the uncertainty set. To this end, we introduced a prediction mapping $S = \{M(\theta), N(\theta)\}$ where

M and N, defined in Chapter 4, reflect the past and future elements of output. Let $\mathcal{S}$ be the set of all S over the uncertainty range. A distance metric $d(S, \mathcal{S})$ defined in (4.15) was used to find $\mathcal{S}_\mathcal{E}$, the approximate convex hull of $\mathcal{S}$.

In this section, using the same uncertainty characterization method, we develop a new approach for solving (5.2) that accounts for the level of approximation.

Theorem 3. *Let* $\mathrm{Conv}\,\mathcal{S}_\mathcal{E}$ *be the* ϵ*-approximate convex hull of* $\mathcal{S}$ *with* s_1*-norm metric. Let* $s_2 = \frac{s_1}{1-s_1}$*. Let* $J(.)$ *be convex in* Y*. The solution of the following optimization is feasible for* (5.1).

$$\begin{aligned} \min_{U_\mathsf{f}, t} \quad & t \\ & Y = M \begin{bmatrix} \Delta U_\mathsf{p} \\ z_0 \\ \Delta Z_\mathsf{p} \end{bmatrix} + N \Delta U_\mathsf{f} \quad \{M, N\} \in \mathcal{S}_\mathcal{E} \\ \textit{subject to} \quad & Y + \epsilon^y \leq b^{max}, Y - \epsilon^y \geq b^{min} \\ & f(U_\mathsf{f}) \leq c \\ & J(U_\mathsf{f}, Y) < t \end{aligned} \tag{5.3}$$

where

$$\epsilon_k^y = \epsilon \left\| \begin{bmatrix} \Delta U_\mathsf{p} \\ \Delta Z_\mathsf{p} \end{bmatrix} \right\|_{s_2} + \epsilon_k \|\Delta U_\mathsf{f}\|_{s_2} \tag{5.4}$$

and

$$\epsilon_k = \max_{S \in \mathcal{S}} \min_{\alpha \in C} \lambda_{p+k}(S, \mathcal{S}_\mathcal{E}, s_1, \alpha) \qquad k = 1, \cdots, p \tag{5.5}$$

where Y *is the prediction using* (4.12) *with matrices contained in* $\mathcal{S}_\mathcal{E}$*.*

Proof. As proved in Theorem 2, ϵ^y can be used to provide a dynamic guaranteed bound on the possible output trajectories when $\mathcal{S}_\mathcal{E}$ with the approximation error ϵ is used as the model set. Using the same predictor, therefore, the control constraints are satisfied for all the output trajectories.

□

The necessary parameters like ϵ and ϵ_k for $k = 1, \cdots, p$ are calculated off-line (See Algorithm 3).

5.4 Measurement Noise

The presented robust MPC includes a moving horizon estimator. By choosing proper weighting matrices Q and R based on the noise distribution, we may make a trade-off between signal tracking and noise rejection. In order to compensate for the effect of the noise on the prediction of output, an additional term is included on constraints. Let n_k be an i.i.d gaussian noise with $n_k \sim \mathcal{N}(0, \sigma_n^2 I)$. According to (4.12), the output variance is calculated as:

$$\sigma_y^2 = \sigma_n^2 (M_2 M_2^T + M_3 M_3^T) \tag{5.6}$$

where M_2 is the p-th column of M (corresponding to z_0) and M_3 represents its $p+1$ to $2p$ columns (corresponding to ΔZ_{p}). The probability that the prediction is in error greater than $3\sigma_y$ is 0.3%. The output constraints of (5.3) can, therefore, change to

$$\begin{aligned} Y + \epsilon^y + 3\sigma_y &\leq b^{max} \\ Y - \epsilon^y - 3\sigma_y &\geq b^{min} \end{aligned} \tag{5.7}$$

5.5 Implementation for MIMO systems

This approach can be easily used for robust control of MIMO systems. As a reminder, $u \in \mathbb{R}^m$, and $y \in \mathbb{R}^r$ which result in $\Delta U_{\mathsf{f}} \in \mathbb{R}^{mp \times 1}$, and $Y_{\mathsf{f}} \in \mathbb{R}^{rp \times 1}$. One way to cover the output uncertainties is to cover all of them by a single model set $\mathcal{S}_{\mathcal{E}}$ with the corresponding $\{\epsilon, \epsilon_k\}$. In this case, the model set $\mathcal{S}_{\mathcal{E}}$ will be the collection of extreme points that cover each output. However, this is sensitive to the relative scalings between outputs and some unnecessary constraints may be added to the optimization problem which can make the problem more conservative.

Another approach, which may be more efficient, is to find an individual model set for each output. This approach can be carried out by simply taking the corresponding rows of $\mathcal{M}$ and $\mathcal{N}$ to the i-th output. Let $\mathcal{M}_i$, and $\mathcal{N}_i$ to be the corresponding rows of $\mathcal{M}$ and $\mathcal{N}$ to $[y_i(1), \cdots, y_i(p)]^T$ [9]. By setting $\mathcal{S}_i = \{\mathcal{M}_i, \mathcal{N}_i\}$, one can use the ϵ-approximate convex hull method to find the model set $\mathcal{S}_{\mathcal{E}i}$ with corresponding $\{\epsilon_i, \epsilon_{ki}\}$. It can be noted that

[9] which is the $i, i+r, i+2r, \cdots, i+(p-1)r$ th rows of $\mathcal{M}$ and $\mathcal{N}$

$|\mathcal{S}_{\mathcal{E}i}| \leq |\mathcal{S}_{\mathcal{E}}|$ for any $i = 1, \cdots, r$. Therefore, the following control problem is solved for MIMO systems

$$\begin{aligned} \min_{U_{\mathsf{f}}, t} \quad & t \\ & Y_i = M_i \begin{bmatrix} \Delta U_{\mathsf{p}} \\ z_0 \\ \Delta Z_{\mathsf{p}} \end{bmatrix} + N_i \Delta U_{\mathsf{f}} \quad \{M_i, N_i\} \in \mathcal{S}_{\mathcal{E}i} \quad i = 1, \cdots, r \\ \text{subject to} \quad & Y_i + \epsilon^{y_i} \leq b_i^{max}, Y_i - \epsilon^{y_i} \geq b_i^{min} \\ & f(U_{\mathsf{f}}) \leq c \\ & J(U_{\mathsf{f}}, Y) < t \end{aligned} \tag{5.8}$$

where

$$\epsilon_k^{y_i} = \epsilon_i \left\| \begin{bmatrix} \Delta U_{\mathsf{p}} \\ \Delta Z_{\mathsf{p}} \end{bmatrix} \right\|_{s_2} + \epsilon_{ki} \|\Delta U_{\mathsf{f}}\|_{s_2} \tag{5.9}$$

and

$$\epsilon_{ki} = \max_{S \in \mathcal{S}_i} \min_{\alpha \in C} \lambda_{p+k}(S, \mathcal{S}_{\mathcal{E}i}, s_1, \alpha) \qquad k = 1, \cdots, p \tag{5.10}$$

A simulation example of this case will be given later in Chapter 7.

5.6 Discussion on Stability and Feasibility

Different studies on stability and feasibility of min-max robust MPC have been presented in literature[11, 56–61]. The key element of the most stability approaches is to add a terminal cost function and terminal constraints so that insure the objective cost is non-increasing by time while feasibility may be obtained by relaxing the problem or using soft constraints.

5.6.1 Feasibility

The approach presented in this study is a regular min-max problem which employs a new method to reduce the elements of the model set. Therefore, its feasibility may be realized in a similar manner as other min-max approaches. The only difference is the output constraints include dynamic offset ϵ^y. In order to claim that adding ϵ^y does not disrupt feasibility, we show that ϵ^y is bounded. Therefore, using soft constraints on outputs guarantees feasibility.

Lemma 2. *Consider the case that there is no measurement noise. Assume the input variation and system gain K_0 are bounded ($\|\delta u\|_{s_2} \leq \upsilon$, $K_0 = max_{\|\delta u\|_{s_2}=1}\|\delta y\|_{s_2} \leq \kappa$). An upper bound on ϵ_y is:*

$$\epsilon_k^y \leq \epsilon\upsilon \sqrt[s_2]{p}\left(\sqrt[s_2]{1+\kappa^{s_2}}+1\right), \qquad k=1,\cdots,p. \tag{5.11}$$

Proof. From (5.9) and since $\epsilon_k \leq \epsilon$ for any $k \leq p$,

$$\epsilon_k^y \leq \epsilon\left(\left\|\begin{bmatrix}\Delta U_{\mathsf{p}}\\ \Delta Z_{\mathsf{p}}\end{bmatrix}\right\|_{s_2} + \|\Delta U_{\mathsf{f}}\|_{s_2}\right) \tag{5.12}$$

Note that ΔU_{p}, ΔU_{f}, and ΔZ_{p} are p-length vectors. Thus

$$\begin{aligned}\left\|\begin{bmatrix}\Delta U_{\mathsf{p}}\\ \Delta Z_{\mathsf{p}}\end{bmatrix}\right\|_{s_2} &\leq \sqrt[s_2]{p\left(\upsilon^{s_2}+\left(\max_{\|\delta u\|_{s_2}=\upsilon}\|\delta y\|_{s_2}\right)^{s_2}\right)},\\ \|\Delta U_{\mathsf{f}}\|_{s_2} &\leq \sqrt[s_2]{p\upsilon^{s_2}}.\end{aligned} \tag{5.13}$$

By plugging (5.13) in (5.12) and since $\max_{\|\delta u\|_{s_2}=\upsilon}\|\delta y\|_{s_2} = \upsilon\kappa$, (5.11) is easily derived.

□

Remark 9. *If the system is open loop unstable, a local feedback can be used to stabilize the system to obtain a bounded system gain.*

5.6.2 Stability

The min-max RMPC formulation does not provide guaranteed stability by itself unless a terminal function and constraint are used. Different approaches have been used in literature to guarantee the stability (a review on stability of min-max MPC can be found in [21, 62]). One way to guarantee stability, is to add the following terminal constraint to hold the Input-to-State practical Stability (ISpS)

$$\|x_{k+1}\| \leq \beta(\|x_k\|,k) + \gamma(\|d_{k-1}\|) + c \tag{5.14}$$

for a class $\mathcal{KL}$ function of β, class $\mathcal{K}$ function γ, and nonnegative c (see [60] for more detail).These methods could be applied to our formulation using the lifted system in state space form, defined as below,

$$x(k+1) = \begin{bmatrix} 0 \\ I_s\mathcal{O}_p I_y \Psi^+ \Gamma + \begin{bmatrix} 0 & I & 0 \end{bmatrix} \\ I_r \mathcal{O}_p I_y \Psi^+ \Gamma \end{bmatrix} x(k) + \begin{bmatrix} I_r \\ I_s T \\ I_r T \end{bmatrix} \Delta U_{\mathsf{f}}(k) \tag{5.15}$$

where $x(k) = \begin{bmatrix} \Delta U_{\mathsf{p}}(k) \\ \tilde{z}(k) \\ \Delta Z_{\mathsf{p}}(k) \end{bmatrix}$, and

$$\begin{aligned} \Delta U_{\mathsf{p}}(k) &= \begin{bmatrix} \delta u_{kp-1}^T & \delta u_{kp-2}^T & \cdots & \delta u_{kp-p}^T \end{bmatrix}^T, \\ \Delta Z_{\mathsf{p}}(k) &= \begin{bmatrix} \delta z_{kp}^T & \delta z_{kp-1}^T & \cdots & \delta z_{kp-p+1}^T \end{bmatrix}^T, \\ \Delta U_{\mathsf{f}}(k) &= \begin{bmatrix} \delta u_{kp}^T & \delta u_{kp+1}^T & \cdots & \delta u_{kp+p-1}^T \end{bmatrix}^T, \\ \Delta Z_{\mathsf{f}}(k) &= \begin{bmatrix} \delta z_{kp+1}^T & \delta z_{kp+2}^T & \cdots & \delta z_{kp+p}^T \end{bmatrix}^T. \end{aligned} \tag{5.16}$$

$I_y = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \end{bmatrix}$ and $z(k) = z_{kp}$. I_r is the reverse operator matrix, a square matrix with 1 on off-diagonal and zero otherwise. Also, $I_s = [I \ \cdots \ I]$ is the summation operator, and $\tilde{z}(k) = z(k) - z_{ss}$ is the output distance to its steady state value.

5.7 Illustrative Example

The 2nd order mass-spring-damper system (4.19) presented in Chapter 4 with the same uncertainty ranges is considered as the case study to show the performance of the proposed RMPC. Define the control problem as follows, an output constraint (i.e., $y \geq y_{min}$) and non-negative input constraint is to be satisfied while a cost on the input and output variations is to be minimized. After changing the output constraint to a soft constraint in order to guarantee the feasibility, this problem can be written as

$$\begin{aligned} \min_{U_{\mathsf{f}}} \ \max_{\theta \in \Theta} \quad & c_u \|U_{\mathsf{f}}\|_1 + \lambda \|\Delta U_{\mathsf{f}}\| + \gamma \|\zeta\| \\ \text{subject to} \quad & \begin{aligned} & x_{k+1} = A_d(\theta) x_k + B_d(\theta) u_k \\ & y_k = C_d(\theta) x_k && \theta \in \Theta \\ & y_k(\theta) + \zeta_k \geq y_k^{min} && k = 1, \cdots, p \\ & u_k \geq 0 \end{aligned} \end{aligned} \tag{5.17}$$

where A_d, B_d, C_d are the system matrices in discrete time and c_u is the input cost. $\zeta = [\zeta_k]_1^p$ is used to change the output constraint to a soft constraint with aim of making sure that

the problem is always feasible. However, the potential violation is penalized with γ. $\lambda\|\Delta U_{\mathsf{f}}\|$ provides smoother input variations to avoid actuator damage. According to Theorem 3, the following alternative problem is solved instead.

$$\begin{aligned}
\min_{U_{\mathsf{f}}} \quad & c_u\|U_{\mathsf{f}}\| + \lambda\|\Delta U_{\mathsf{f}}\| + t \\
& Y = M \begin{bmatrix} \Delta U_{\mathsf{p}} \\ y_0 \\ \Delta Y_{\mathsf{p}} \end{bmatrix} + N\Delta U_{\mathsf{f}} \qquad \forall\{M, N\} \in \mathcal{S}_{\mathcal{E}}^{MSD} \\
\text{subject to} \quad & -y_k - \zeta_k + \epsilon_k^y \leq -y_k^{min} \qquad k = 1, \cdots, p \\
& \epsilon_k^y = \epsilon \left\| \begin{matrix} \Delta U_{\mathsf{p}} \\ \Delta Y_{\mathsf{p}} \end{matrix} \right\| + \epsilon_k \|\Delta U_f\| \\
& u_k \geq 0 \\
& \gamma\|\zeta\| \leq t
\end{aligned} \tag{5.18}$$

which is a convex optimization that can be reformulated as a second order cone problem. In order to define a finite set of parameters using the scenario approach, choose N so that $\epsilon^s = 0.01$, and $\beta = 0.001$ where the system has three uncertain parameters. According to [53], 4000 samples are required to meet the requirements. By taking 4000 samples uniformly from the uncertain parameter set, we generate $\Theta^{MSD} = \{\theta_1, \cdots, \theta_{4000}\}$. Then, using a sampling time $T_s = 1s$, weighting matrices $Q = 1$, $R = 1$, and choosing $p = 10$, we create set $\mathcal{S}$ and find the approximation of its convex hull, off-line. In this example, the method has been performed with $s_1 = 2, s_2 = 2$. $\mathcal{S}_{\mathcal{E}}^{MSD}$ is the approximate convex hull of $\mathcal{S}$ computed by the approximate convex hull computation method presented in Chapter 3 with corresponding ϵ and ϵ_ks.

Figure 5.1 shows the approximation error ϵ as a function of E, the number of elements of $\mathcal{S}_{\mathcal{E}}^{MSD}$ for $E = 1, \cdots, 20$. As expected, by selecting more elements, the resulting approximate convex hull is closer to the exact convex hull of $\mathcal{S}^{MSD}$ and consequently ϵ decreases.

For implementation, $E = 13$ was chosen, with corresponding approximation error $\epsilon = 0.067$. Given ϵ and set $\mathcal{S}_{\mathcal{E}}$, Figure 5.2 demonstrates ϵ_k for $k = 1, \cdots, 10$ calculated off-line. As expected, the uncertainty increases in time since the uncertainty of the previous steps are accumulated for the next step. It is also observed that $\epsilon_1 \leq \epsilon_2 \leq \cdots \leq \epsilon_{10} \leq \epsilon$.

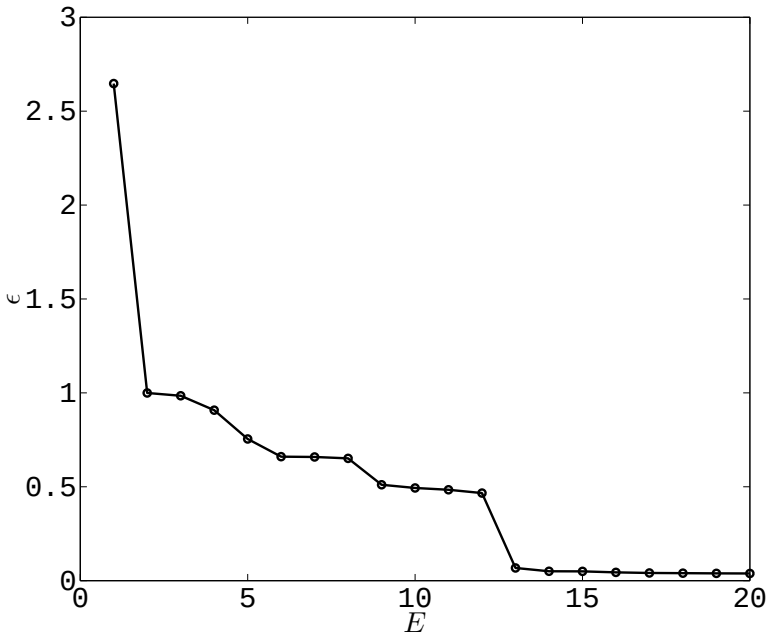

Figure 5.1: Approximate convex hull accuracy using E extreme points.

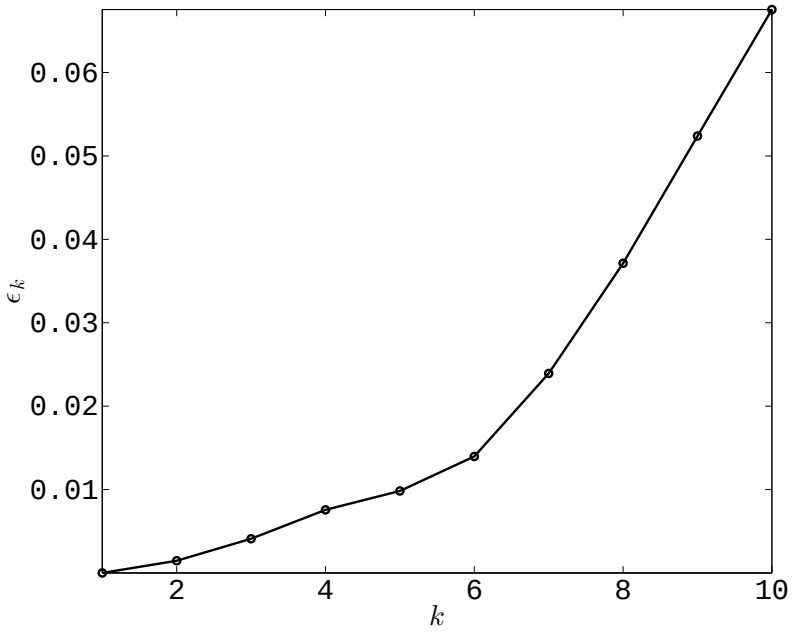

Figure 5.2: ϵ_k for $k = 1, \cdots, 10$ when $E = 13$.

Given ϵ, ϵ_k, and $\mathcal{S}_\mathcal{E}$ and by assuming $c_u = 3$, $\lambda = 50$, and $\gamma = 100$, we use (5.18) to implement the robust MPC for the uncertain mass-spring-damper system. The plant is run 35 s in open loop and after that we apply the controller. Figure 5.3 compares the RMPC performance against the regular MPC for 50 random uncertain systems as the actual plant. The regular MPC uses the same structure as the robust MPC (i.e., it has the same formulation and also it includes the inner MHE) but uses a single nominal model rather than the model set. Note that because of the essence of the problem which aims to minimize the input cost, the system is operated near the constraint. As seen, regular MPC fails to satisfy the output constraint in some transients. On the other hand, the robust controller succeeds to always keep the produced output higher than the required constraint, although with more conservatism. Both the robust and regular MPCs have similar performance at the steady state.

Figure 5.4 demonstrates the behavior of ϵ^y during control. Figure 5.4a shows the output uncertainty level corresponding to the convex hull approximation related to the past data $\left(\epsilon \left\| [\Delta U_\mathsf{p}^T \quad \Delta Y_\mathsf{p}^T] \right\|\right)$, and Figure 5.4b indicates the worst uncertainty level caused by the future input variations ($\epsilon_{10}\|\Delta U_\mathsf{f}\|$). Since $\epsilon_{10} \geq \cdots \geq \epsilon_1$, we only show the uncertainty level for the worst case, i.e., 10-step ahead prediction. Since both the past and future terms go to zero at steady state this reduces the conservatism of the robust approach.

In order to investigate the robust MPC performance in the presence of measurement uncertainty, i.i.d. noise $n_k \sim \mathcal{N}\left(0, 0.004 {y_k^{min}}^2\right)$ is added to the output. Figure 5.5 shows the tracking performance for a randomly generated real plant. Also in order to have a better understanding, ϵ^y is given in Figure 5.6. $3\sigma_y$ has been considered as the noise bound. As seen in Figure 5.5, the output constraint is mostly satisfied by the RMPC while obviously this statement does not hold for the regular MPC. In addition, the estimation procedure of the proposed RMPC produces smoother signals and avoids the input chattering due to noise.

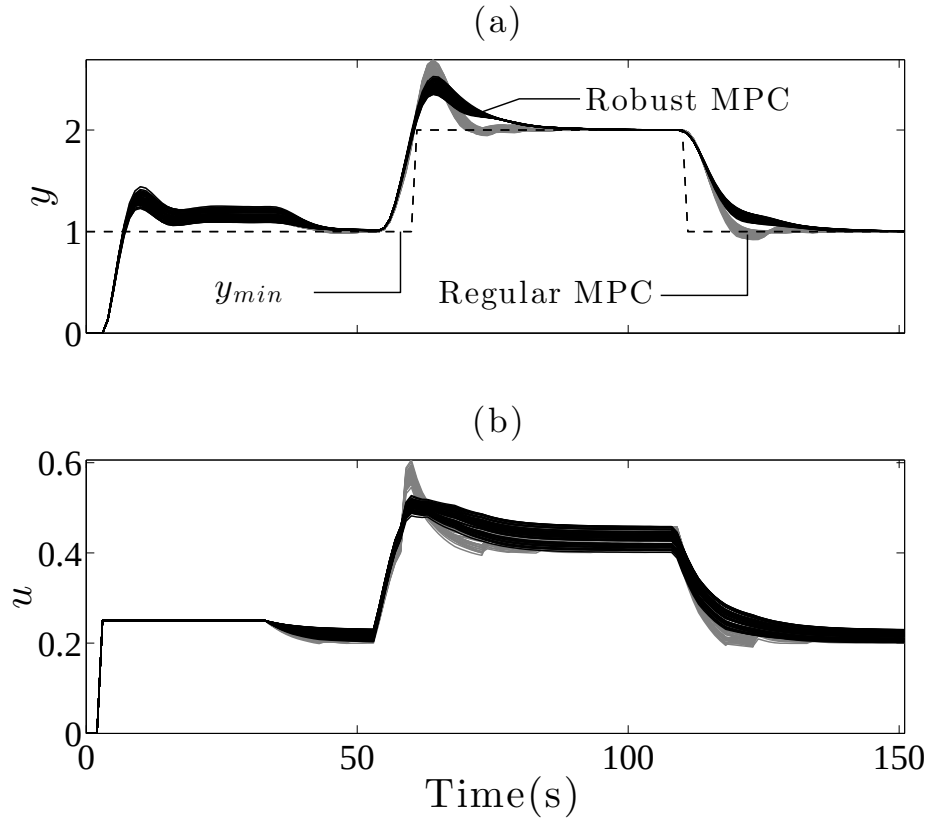

Figure 5.3: Comparison between a regular MPC and the proposed RMPC. The controller is applied after 35 s. (a) output (b) input

5.8 Summary

Employing the new uncertainty characterization approach introduced in Chapter 4, a novel min-max RMPC was presented in this chapter for systems with parametric uncertainty. The approximate convex hull computation method introduced in Chapter 3 was used as the key element of the proposed approach.

The proposed method maps the parameter set to the prediction mapping S (introduced in Chapter 4) with aiming to find a guaranteed bound on the all possible output trajectories. Regardless of the distribution of the parameter set and without requiring the system output to be affine in the parameters, the extreme points of S provide this bound. In order to reduce the number of extreme points for achieving a lower computational complexity, the

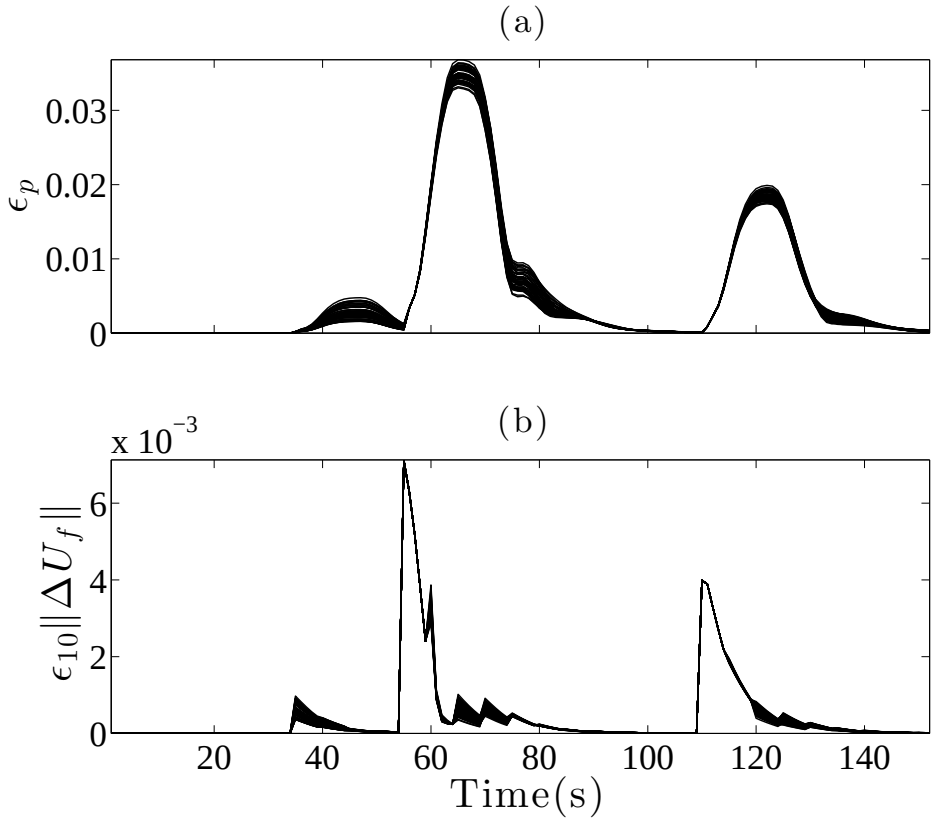

Figure 5.4: Output uncertainty (ϵ^y) corresponding to the convex hull approximation. (a) caused by past data (b) caused by future data

ϵ-approximate convex hull of the new mapping was calculated. It allows the designer to make a trade-off between the accuracy level and computational complexity by providing a degree of freedom on the number of extreme models.

In order to account for the level of approximation on the output bound, a guaranteed bound on output violation was presented in Theorem 3 (The same bound as given in Theorem 2). As noted in Remark 5 and shown by the illustrative example, the output violation bound converges to zero which removes conservatism at steady state.

Integrating an inner estimator, specifically an MHE, in RMPC for finding the initial state of the system and dealing with noise measurement is another feature of the controller that removes the need of using an external estimator.

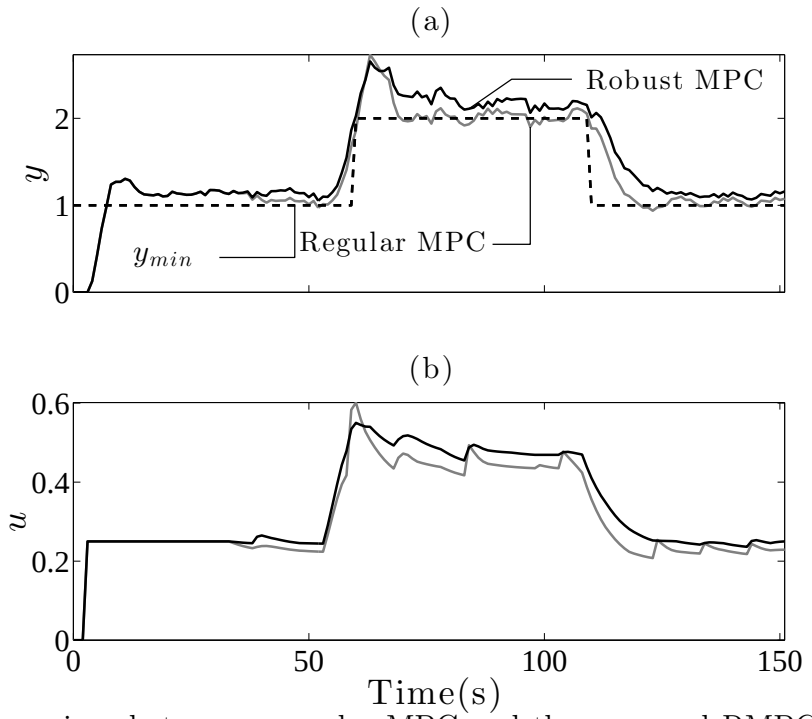

Figure 5.5: Comparison between a regular MPC and the proposed RMPC in the presence of measurement noise. The controller is applied after 33 s. (a) output (b) input

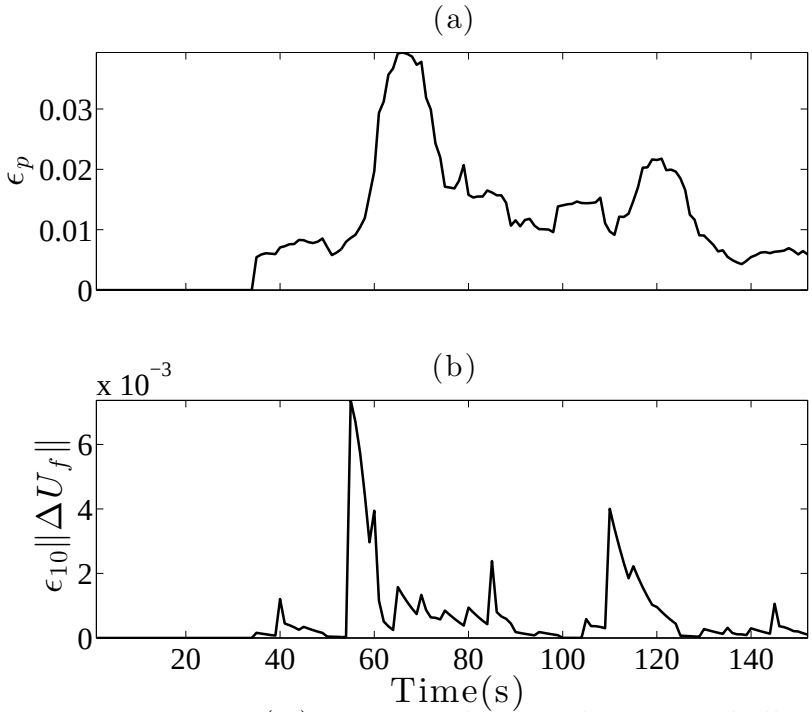

Figure 5.6: Output uncertainty (ϵ_y) corresponding to the convex hull approximation. (a) caused by past data (b) caused by future data

CHAPTER 6

CONTROL OF DC-DC FLOATING INTERLEAVED BOOST CONVERTER UNDER UNCERTAINTY

In this chapter, the proposed robust model predictive control method is applied to an application to evaluate its performance for SISO systems. This case study examines how the uncertainties and disturbances can be handled by the controller[10].

6.1 Introduction

The output voltage of renewable and sustainable energy sources such as photovoltaic (PV) cells is typically an unregulated low level DC voltage. However, a regulated high level DC voltage is required as the input of a three phase inverter used to convert the DC voltage to a proper AC voltage compatible with the grid. Specifically, a DC voltage in the range of 400V is required to feed an AC three-phase 120V or single-phase 220V inverter. One way to increase the output voltage of PV cells is to use a series combination of PV modules. However, since the failure of one module causes the failure of the whole string, using series connected PV modules reduces reliability. To this end, the parallel connection of PV cells with high gain DC-DC boost converters are commonly used to increase the output voltage of PV cells [63, 64].

The Floating Interleaved Boost Converter (FIBC) is a novel power electronic interface which has been proposed for low voltage renewable energy resources such as fuel cells and hybrid electric vehicles [65–67]. The FIBC consists of two conventional boost converters that are connected in series [63]. By using this configuration, we can increase the obtainable voltage gain while reducing the output voltage ripple. Although the FIBC requires a greater number of components than conventional boost converters, it is preferred due to its

[10]This chapter is based on a submitted paper: Hossein Sartipizadeh and Farnaz Harirchi. Robust Model Predictive Control of DC-DC Floating Interleaved Boost Converter under Uncertainty. submitted to *IEEE GreenTech, 2017*

advantages, including higher efficiency, higher voltage gain, less input ripple, lower voltage and current ratings for switches and diodes, lower switching stress, smaller inductors and capacitors, and lower losses and voltage drop through inductive components [64, 68].

In the ideal case that the input voltage of a boost converter is fixed, selecting a proper voltage gain results in the desired output voltage. This gain is adjusted by changing the duty cycle of the converter as the design parameter. Unfortunately, the input voltage of a boost converter varies in practice. For instance in PV cells, although the *Open Circuit Voltage* (VOC) may remain almost constant as long as there is sufficient irradiance light, the output voltage of the PV cell drops when the PV is connected to external loads. This decrement in the voltage of the PV cell is caused by power losses within the cells structure as well as the metallic conductors deposited on the cells surface. Temperature is another factor that also affects the PV cell output voltage. There are different studies on effect of temperature and sun irradiation on PV parameters (for instance see [69]). The output voltage varies about 5% for every 25C changes in the cell temperature. Therefore, the input voltage of the boost converter depends on the sun irradiation condition, temperature, external loads, etc, and consequently it is uncertain. To this end, some control approaches have been employed to reach the desired outputs. For instance, the control of the FIBC converter has been proposed using a Proportion-Integral (PI) controller [63], type III controller [68], nonlinear adaptive controller [70], sliding mode controller [71], and nominal MPC[72]. However, they do not take the model uncertainty of the FIBC into account.

The physical elements of an FIBC, including resistors, capacitors, and inductors, vary about some nominal values. Therefore, the system response may differ from its nominal and will be uncertain. Since individually tuning the duty cycle to tune the voltage gain for each product is not practical, the voltage gain is calculated by the nominal parameters with the price of remaining uncertain.

Thus, keeping the boost converter output voltage at the desired level is challenging and demands an effective control approach. This control approach must be robust against the

input voltage uncertainty as well as physical elements uncertainties. In other words, we prefer a unique controller which can perform on every product under a given uncertainty range with the capability of handling the input voltage unexpected perturbations without need of further tuning. To this end, we employ the proposed RMPC in Chapter 5 to deal with the input and physical element uncertainties. It is worth reporting that prior work has not taken the uncertainty into account (For instance see [63, 68, 70–72]).

6.2 FIBC Model

The configuration of an FIBC is shown in Figure 6.1. In Figure 6.1, when the switch S_i is not conducting, corresponding diode, D_i, is closed and when it is conducting, D_i is open. The relation between the input and output of FIBC can be found as:

$$v_{out} = v_1 + v_2 - v_{in} \tag{6.1}$$

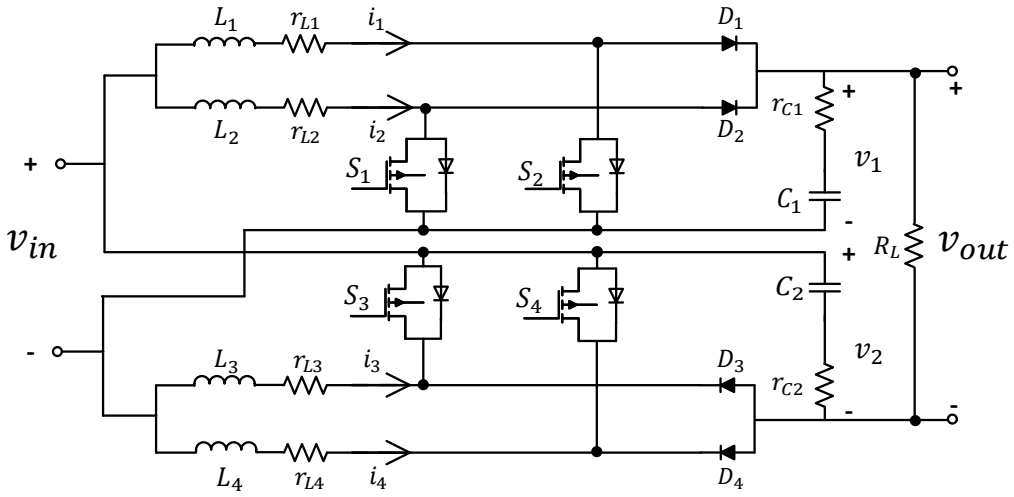

Figure 6.1: DC-DC floating interleaved boost converter schematic

In Figure 6.1, all the capacitors and inductors are the same (i.e., $L_1 = L_2 = L_3 = L_4 = L$, and $C_1 = C_2 = C$). In addition $r_{L_1} = r_{L_2} = r_{L_3} = r_{L_4} = r_L$ and $r_{C_1} = r_{C_2} = r_C$. Therefore, the voltages v_1 and v_2 are calculated like a conventional boost converter as below.

$$v_1 = v_2 = \frac{v_{in}}{1 - D} \tag{6.2}$$

where D is the duty cycle of FIBC. By plugging (6.2) in (6.1), the output voltage is obtained as

$$v_{out} = \frac{1+D}{1-D} v_{in} \tag{6.3}$$

Thus, the gain of the FIBC is $(1+D)$ times greater than a conventional boost converter. Define the duty cycle distance from its nominal value as $\tilde{d} = D - D_0$ where D_0 is a fixed number and it is calculated from (6.3) based on the nominal desired gain. Also let $\tilde{v}_{out} = v_{out} - \bar{v}_{out}$ be the variational voltage. The duty-cycle to output voltage small signal transfer function of an FIBC converter with resistive load R is given as [65]

$$\frac{\tilde{v}_{out}(s)}{\tilde{d}(s)} = K_v \frac{(1-(s/\omega_{zr}))(1+(s/\omega_{zl}))}{(1+(s/\omega_o Q)+(s^2/\omega_o^2))} \tag{6.4}$$

where

$$K_v = \frac{v_{in}\left(2R(1-D_0)^2 - r_L(1+D_0)\right)}{(1-D_0)^2\left(R(1-D_0)^2 + r_L\right)} \tag{6.5a}$$

$$\omega_{zr} = \frac{2R(1-D_0)^2 - r_L(1+D_0)}{L(1+D_0)} \tag{6.5b}$$

$$\omega_{zl} = \frac{1}{r_C C} \tag{6.5c}$$

$$\omega_{zi} = \frac{1}{(RC/(D_0+3)) + r_C C} \tag{6.5d}$$

$$\omega_o = \frac{1}{\sqrt{LC}} \sqrt{\frac{2R(1-D_0)^2 + 2r_L}{R+2r_C}} \tag{6.5e}$$

$$Q = \frac{\omega_o (R+2r_C) LC}{RC\left(r_L + 2r_C(1-D_0)^2\right) + 2(L + r_C r_L C)} \tag{6.5f}$$

The values of the elements of the FIBC converter are given in Table 6.1. v_{in} is uncertain and changes based on the energy source. In order to control the output voltage, $\tilde{d}$ is manipulated to adjust the absolute duty cycle D.

6.3 Implementation

Define the uncertainty vector $\theta = [v_{in}, L, r_L, C, r_C, R_L]^T$. Given the range of each parameter, we take 1000 random samples with uniform distribution to create the uncertainty set

Table 6.1: FIBC model parameters

Parameter	**Nominal value**	**Unit**	**Tolerance**
C	2.2 e-3	F	2%
L	1e-4	H	2%
r_C	4.1 e-2	Ω	2%
r_L	2.0 e-2	Ω	2%
R_L	1e3	Ω	20%

$\Theta = \{\theta_1, \theta_2, \cdots, \theta_{1000}\}$. Then, using the introduced mapping S while assuming $Q = R = I$ and $s_1 = s_2 = 2$, we build $\mathcal{S} = \{S(\theta_1), S(\theta_2), \cdots, S(\theta_{1000})\}$. The following min-max optimization problem can be solved by MPC at each sampling time to achieve the stated control goal

$$\begin{aligned} \min_{U_\mathsf{f}} \max_{\theta \in \Theta} \quad & c_u \|U_\mathsf{f}\|_1 + \lambda \|\Delta U_\mathsf{f}\| + \gamma \|\zeta\| \\ \text{subject to} \quad & x_{k+1} = A_d(\theta) x_k + B_d(\theta) \delta u_k \\ & \delta y_k = C_d(\theta) x_k \qquad\qquad \theta \in \Theta \\ & y_k(\theta) - \zeta_k \leq y_k^{max} \qquad k = 1, \cdots, p \\ & y_k(\theta) + \zeta_k \geq y_k^{min} \end{aligned} \tag{6.6}$$

where A_d, B_d, C_d present the system matrices in discrete time state space representation for variational form(they can be obtained from the FIBC transfer function with a sampling time $T_s = 1ms$). $u_k = \tilde{d}$ is the process input while $y_k = v_{out}$ and $\delta y_k = \tilde{v}_{out}$ are the absolute output and its variation at time k. U_f and Y_f are the future input output vectors defined as Chapter 4. c_u is the input cost, therefore, the first term of the cost function minimizes the control effort cost. We assume $y_{min} = 400V$ to make sure the minimum desired output is generated. Since the cost of the input is minimized, this problem turns to a tracking problem. In addition, we set $y_{max} = 420V$ to avoid large peaks in the output at the transient times. However, sometimes there is no feasible solution in the specified output range. To this end, $\zeta = \{\zeta_1, \cdots, \zeta_p\}$ is used to change the output constraints to soft constraints with the attempt to make sure that the problem is always feasible. However, the potential violation is penalized with γ. $\lambda \|\Delta U_\mathsf{f}\|$ attempts to keep the system in the valid range of linearized

model and avoid it to go far from the operating point.

Given $\mathcal{S}$ we calculate its approximate convex hull $\mathcal{S}_\mathcal{E}$ with corresponding $\{\epsilon, \epsilon_k\}$. According to (5.3), the following alternative problem can be solved.

$$\begin{aligned}
\min_{U_\mathsf{f}} \quad & c_u\|U_\mathsf{f}\|_1 + \lambda\|\Delta U_\mathsf{f}\| + t \\
& Y = M \begin{bmatrix} \Delta U_\mathsf{p} \\ y_0 \\ \Delta Y_\mathsf{p} \end{bmatrix} + N\Delta U_\mathsf{f} \qquad \forall\{M, N\} \in \mathcal{S}_\mathcal{E} \\
\text{subject to} \quad & y_k - \zeta_k + \epsilon_k^y \leq y_k^{max} \qquad k = 1, \cdots, p \\
& -y_k - \zeta_k + \epsilon_k^y \leq -y_k^{min} \\
& \epsilon_k^y = \epsilon \left\| \begin{matrix} \Delta U_\mathsf{p} \\ \Delta Y_\mathsf{p} \end{matrix} \right\| + \epsilon_k \|\Delta U_f\| \\
& \gamma\|\zeta\| \leq t
\end{aligned} \tag{6.7}$$

6.4 Simulation Results

In this section, we investigate the performance of the controller against the parametric uncertainty and the input voltage uncertainty through a simulation experiment. Assume the uncertain parameters of the FIBC can take any value within their uncertainty ranges given in Table 6.1. Also let the input voltage vary over time between 37V to 43V as shown in Figure 6.2. The control goal is to keep the output voltage at $v_{out} = 400$V for any arbitrary system with parameters within the uncertainty range regardless the unexpected changes in v_{in}. In other words, a unique controller is expected to perform under uncertain situations without any needs for further tuning.

Figure 6.3 shows the result of finding $\mathcal{S}_\mathcal{E}$ with E elements. The vertical axis shows the error ϵ and the horizontal axis indicates E. In order to have a better understanding of the approximation improvement by increasing E, the error is shown on a logarithmic scale. As expected, the approximation error decreases as the number of extreme model increases. According to the figure, choosing 25 extreme models results in $\epsilon = 3.246$ while picking 50 models decreases the approximation error to 0.0432 (these points have been highlighted in the figure). Although user can pick any E, these two point may be good candidates according to the error trend. Since 50 models is still considerably fewer than 1000, in this work we

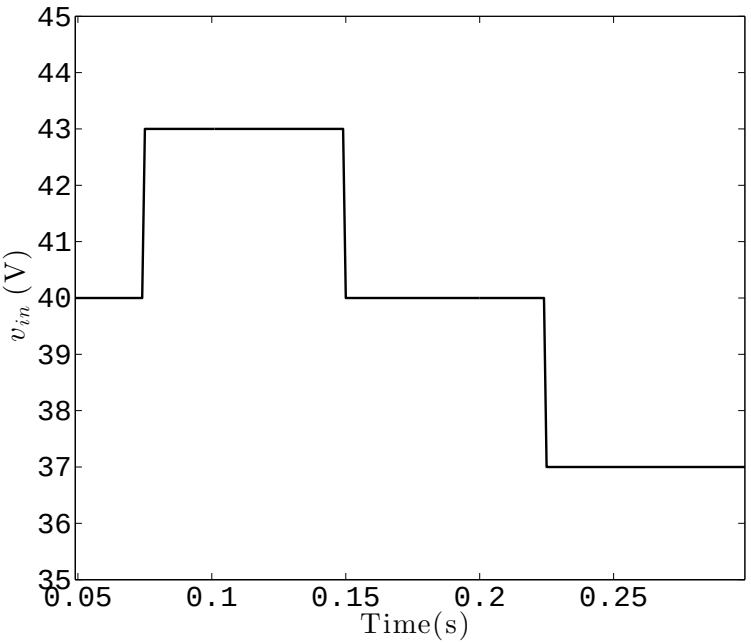

Figure 6.2: The input voltage of the FIBC converter coming from the energy source

prefer to generate $\mathcal{S}_\mathcal{E}$ with 50 models. As seen, adding more extreme models after $E = 50$ results in a negligible decrement in the approximation error while it still increases the time complexity. After defining $\mathcal{S}_\mathcal{E}$ with 50 elements and $\epsilon = 0.0432$, one may use (5.10) to calculate back ϵ_k. Figure 6.4 shows ϵ_k over a 10-length future horizon.

Let choose $c_u = 250$, $\lambda = 2500$, and $\gamma = 0.8$ (These weights can vary based on the importance of their roles stated above). We initiate the system to reach the steady state, and then apply the RMPC controller while v_{in} is perturbed as Figure 6.2. The controller is tested for 40 arbitrary FIBCs with random parameters with uniform distribution in the uncertain ranges. Figure 6.5 shows the tracking results for the random FIBCs. Figure 6.5a demonstrates the output voltage while Figure 6.5b represents the necessary changes in the FIBC duty cycle to compensate the disturbance (input voltage perturbation). The input-output results are plotted in one figure (Figure 6.5) for all the 40 random FIBCs. We individually discuss the performance of the controller against the parametric uncertainty and the input voltage uncertainty.

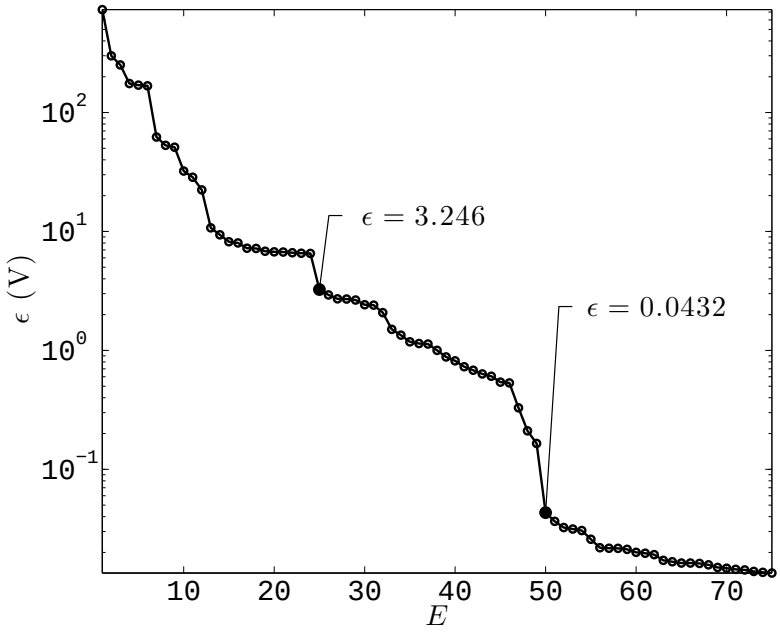

Figure 6.3: FIBC control- The approximation error (ϵ) when the space S is covered with E elements (extreme models).

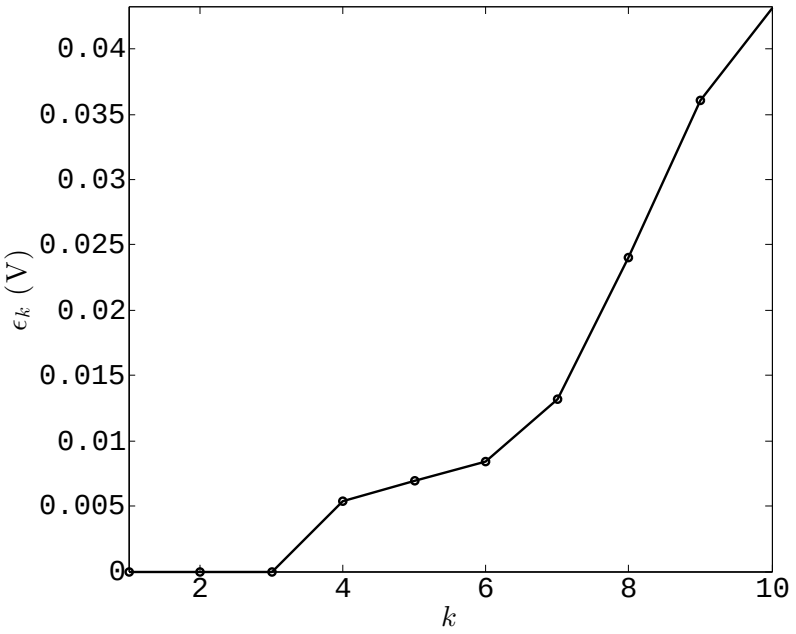

Figure 6.4: FIBC control- ϵ_k for $E = 23$, $k = 1, \cdots, 10$.

- **Performance of the RMPC against the input voltage uncertainty.** As seen, perturbing v_{in} by $3V$ changes the output voltage v_{out} about $30V$ (As noted, the FIBC gain is close to 10). In the case that v_{in} decreases (consequently v_{out} decreases), the controller adds a small deviation to the duty cycle to cancel out the effect of the input voltage changes. When v_{in} suddenly increases, there is a sudden increment in the output voltage. The controller then decreases the duty cycle by adding a negative d. It is worth reporting that in this case since the voltage jumps to 430V, the controller could not find any feasible solution for optimization problem (6.7) if we didn't change the constraints to soft ones.
- **Performance of the RMPC against parametric uncertainty.** As seen in Figure 6.5a, there is a small change in the transient of different systems, seen as thicker lines. However, the controller successfully keeps all the system above the desired reference. At the steady state, the output trajectories of all the systems reach its desired (400V).

Figure 6.6 shows the guaranteed violation bound ϵ^y due to the approximation calculated by the controller. Figure 6.6a and Figure 6.6b represent the violation bounds due to the past input-output and future input (in order, the first and second terms of (5.9)), respectively. As seen and mentioned before, the violation bound goes to zero at steady state, which reduces the conservatism of the min-max problem.

6.5 Summary

In this chapter, robust control of an FIBC in the uncertain situations was studied with an attempt to regulate the FIBC output voltage at the desired level regardless the existing uncertainties. Uncertainty in the FIBC input voltage and uncertainty in the physical elements were taken into account as two common types of uncertainty. Specifically, the proposed robust model predictive control approach introduced in Chapter 5 was used due to its capability in handling the parametric uncertainty as well as physical limitations. Simulation results confirm the solid performance of the robust controller under uncertain situations.

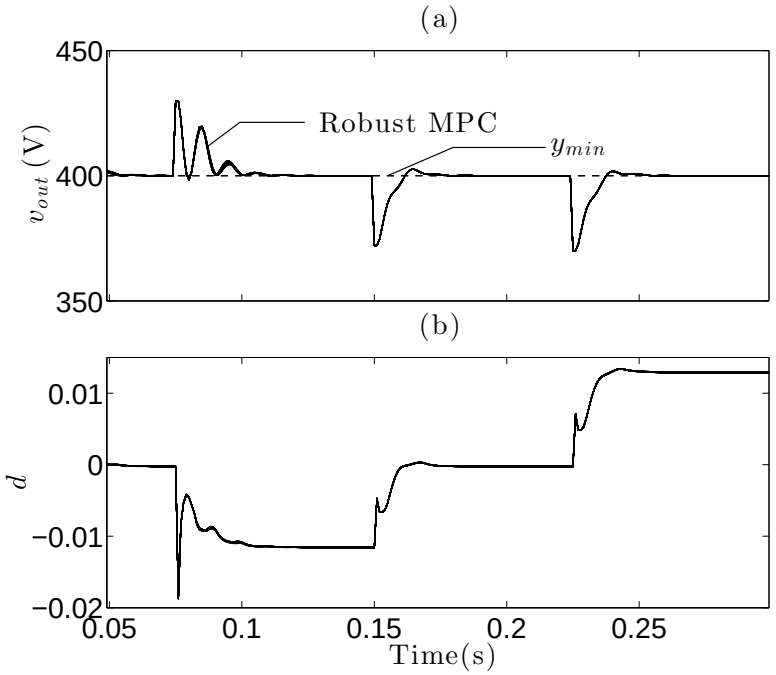

Figure 6.5: FIBC control- Tracking result for 50 random systems after applying the RMPC in the presence of the input voltage changes shown in Figure 6.2. (a) Output voltage (b) Duty cycle deviation from its nominal.

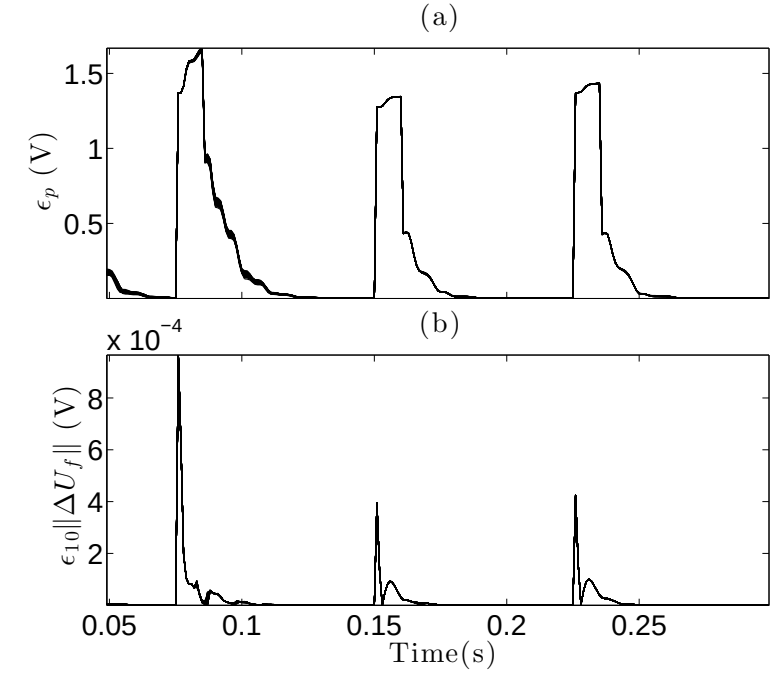

Figure 6.6: FIBC control- Output violation bound (ϵ^y) corresponding to the convex hull approximation error (a) due to past input-output (b) due to future input.

CHAPTER 7

ECONOMIC CONTROL OF METHANE REFORMING

The aim of this chapter is to investigate the application of the proposed robust MPC in control of MIMO systems. Specifically, a methane reforming process that is a key element of a solid oxide fuel cell is selected as the case study and its economic control under uncertainty is discussed[11].

7.1 Introduction

Methane reforming is a process used to produce syngas (H_2 and CO) and can be a key balance of plant component in a fuel cell system. Among three major thermochemical reforming techniques, i.e., steam reforming (SR), partial oxidation (POx), and autothermal reforming (ATR), the latter is more compact and practical for use with mobile fuel cells since it does not need an external heat supply. ATR combines the exothermic partial oxidation with the endothermic steam reforming under thermally neutral conditions [73].

Efficient control of methane reforming results in improved fuel cell performance. While in some applications open loop operation is feasible, real time feedback control is necessary to provide additional system capabilities, such as

- Operating at multiple or varying output conditions. The reformer must provide syngas with a sufficient energy content to meet the demand of the application, for example, as part of the fuel cell system the reformer is required to supply a gas composition to the fuel cell that can meet the current power and heat loads. For some applications these loads can vary, requiring the fuel processor to follow this variation.

[11] Some part of this chapter (Section 7.4) is based on an accepted paper: Hossein Sartipizadeh, Tyrone L. Vincent, and Robert J. Kee. Economic control of methane reforming, in *IEEE American Control Conference*, 2015.

- Meeting operating constraints. The fuel reformer must be operated while avoiding conditions that can lead to damage, such as excessive temperatures or carbon deposition.

- Robustness to disturbances. Although the inputs to the fuel reformer can be tuned for a specific operating point, disturbances such as variations in the environmental temperature, variations in the input fuel mixture, or wear in components such as blowers or injectors can cause the output composition to vary unless corrective action is taken.

- Increased efficiency. There may be a family of inputs which can meet the current operating requirements, and the controller can chose the one that is optimal.

In this chapter, we aim to design an economic controller for methane reforming. To this end, first we design an effective MPC controller for the case that the system parameters are certain but there is measurement uncertainty and operating constraints. However, since many of the methane reforming parameters represent behavior at the micro-scale, their exact values are uncertain. Therefore, we later use the RMPC controller, introduced in Chapter 5, to provide a robust control for methane reforming. In order to have a better understanding of the process, some preliminary concepts are given in Section 7.2 and the methane reforming system is briefly described in Section 7.3.

7.2 Preliminaries

In this section we briefly explain some preliminaries about methane reforming process including coke propensity and calculating the energy content of a gas.

7.2.1 Output Energy Content: Lower Heating Value (LHV)

The heat of combustion is a measure of the energy released as heat when a mixture is completely combusted with oxygen under standard condition (25°C and 1 atm). The heat of combustion for fuels is expressed as the Higher Heating Value (HHV), Lower Heating Value (LHV), and Gross Heating Value (GHV).

The heating value (or energy value) of a substance is the amount of heat released during the combustion of a specified amount of it (usually in mol or kg unit). The higher heating value is determined by bringing all the products of combustion back to the standard temperature, and in particular condensing any produced vapor. HHV assumes all the water components are in liquid state at the end of combustion. LHV is obtained by subtracting the heat of vaporization of the water vapor from HHV. The required energy to vaporize the water is not released as heat.

In this study, to specify the outlet gas energy content, we use LHV. Conceptually, to find the LHV amount, the following steps are performed.

- Pick 1 mole of the fuel mixture.
- Set the gas to the standard condition (25°C and 1 atm).
- Find the amount of oxygen required for complete combustion of fuel, and add this amount of oxygen to the fuel as a new mixture.
- Find the enthalpy of the mixture and call it $H_{reactants}$
- Find the enthalpy of the mixture after equilibrating it (by completely burning the mixture) at the constant temperature and call it $H_{products}$

The enthalpy difference per unit mass of fuel is the heating value, and, the heating value in unites of energy/mass is calculated as

$$HV = \frac{H_{reactants} - H_{products}}{m_{mix}} \tag{7.1}$$

where m_{mix} is the total mass of fuel plus added oxygen. Since the water is completely vaporized, (7.1) is the lower heating value.

A more straightforward method of calculating LHV is simply

$$LHV = m_{H_2} \times LHV_{H_2} + m_{CH_4} \times LHV_{CH_4} + m_{CO} \times LHV_{CO} \tag{7.2}$$

where m_* is the mass of component $*$, LHV_j indicates the LHV of component j in unit of $kJ/mass$. LHV_{H_2}, LHV_{CH_4}, and LHV_{CO} are 119.96, 50.009, and 10.112 MJ/kg, respectively.

The LHV energy can be converted to power by multiplying by the mass flow rate of the gas:

$$\Psi = LHV \times \dot{m} \tag{7.3}$$

where Ψ and $\dot{m}$ represent the power and the gas mass flow rate, respectively.

7.2.2 Estimating Coking Propensity

Coke formation in the reactor is a common and challenging problem in methane reforming. Carbon deposition can decrease the reaction rate and damage the reactor. Overlay, aim is to keep the produced carbon amount in an allowable range. In this study, in order to quantify the carbon deposition, we employ the C-H-O ternary diagram. A mixture can be shown on the ternary diagram by its carbon, hydrogen, and oxygen molar ratio. For instance, CO consists of 1 mole of carbon and 1 mole of oxygen. Then, half of the mixture is carbon and another half is the oxygen while there is no hydrogen. Therefore, CO is shown as the intersection of 0.5 on C and O axes and 0 on H axis. In the C-H-O diagram, Carbon Deposition and No Carbon Deposition regions are distinguished by a boundary line which its location changes based on temperature (Figure 7.1) [74].

To keep system in the coke free region, inputs of the process should be chosen so that the inlet gas composition lies under the carbon deposition line. Assume a methane reforming process where the inlet gas is a mixture of methane, air, and steam. To find the amount of C, H, and O elements of the inlet gas, first we need to calculate the mole fraction of CH_4, O_2, and H_2O based on given input mass flow rates. Using molecular mass we can easily calculate each specie mole fraction as $y_i = \frac{m_i/M_i}{n_{tot}}$ where M_i indicates the molecular mass of specie i and $n_{tot} = \sum m_j/M_j$ is the total number of moles. We only need to calculate CH_4, O_2, and H_2O mole fractions, since H_2, CO, and CO_2, the other species that are produced along the reformer during the process, are zero at inlet. Then,

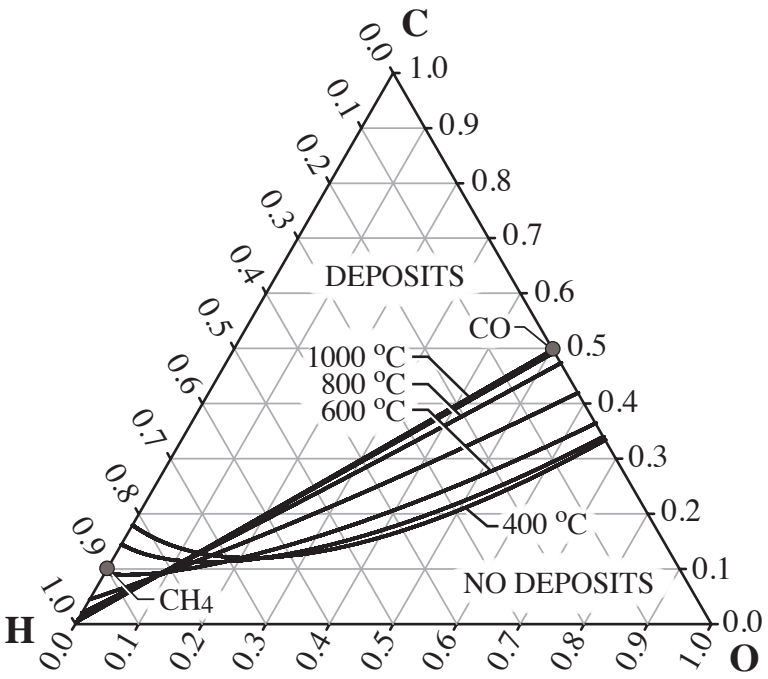

Figure 7.1: C-H-O ternary diagram

$$
\begin{aligned}
C &= \frac{1 \times MF_{CH_4}}{10} \\
H &= \frac{4 \times MF_{CH_4} + 2 \times MF_{H_2O}}{10} \\
O &= \frac{2 \times MF_{O_2} + 1 \times MF_{H_2O}}{10}
\end{aligned}
\tag{7.4}
$$

where MF_j indicates the mole fraction of inlet gas j. Note that they are divided by 10 since CH_4, O_2, and H_2O consist 10 atoms. Finally, the C, H, and O percentages to use in the ternary diagram are calculated as follows,

$$
\begin{aligned}
C_{\%} &= \frac{C}{C + H + O} \\
H_{\%} &= \frac{H}{C + H + O} \\
O_{\%} &= \frac{O}{C + H + O}
\end{aligned}
\tag{7.5}
$$

The system location on the ternary diagram is then specified by $C_{\%}$, $H_{\%}$, and $O_{\%}$. In this work, the distance of this point to the carbon deposition line is denoted with $dist_{CDB}$ and defines a signal constraint that must be met by the control system during operation.

7.3 System description

Figure 7.2 illustrates the main elements of the system. Fuel, air (including 79% N_2, 21% O_2, 1% Ar), and steam are the input streams, which are assumed to be freely actuated to a specific, desired, mass-flow. These input streams are mixed, heated by a heat exchanger (not shown) to 400°C, and enter the reactor. The reactor is made up of multiple tubes of ceramic foam impregnated with the catalyst. The active component of the catalyst is rhodium which is one of the most prominent catalyst for methane reforming in industry. In the case that the process is not running adiabatically (auto-thermal), these tubes are integrated into a second heat exchanger that provides additional flexibility to modulate the temperature of the reactor.

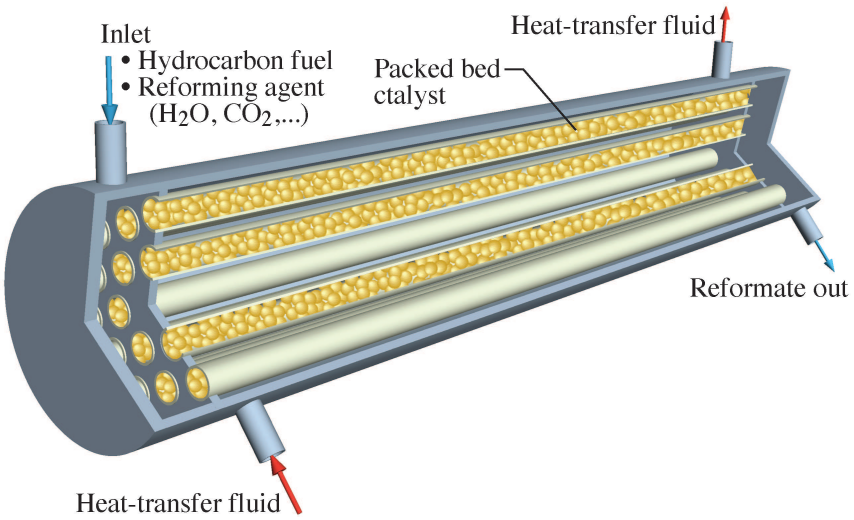

Figure 7.2: Reformer system diagram

There are many physical variables associated with the fuel reformer that are important to ensure proper operation. The most important of these include output composition and temperatures throughout the reformer. In addition, the hydrogen to carbon ration (H_2/CO) may need to remain about a proper value. Although measurement of these variables would be extremely useful as part of a control system, by physical and practical constraints we are limited to a much smaller subset of measurements. In this thesis, we examine the use

of measurements of a subset of the output mixture (specifically, CO and CO_2 concentration using IR absorption measurements) along with a measurement of the thermal conductivity of the output gas, and a temperature measurements from the surface of the tubes enclosing the catalytic foam.

Because the reactor is used as part of a fuel cell system, the operation of the fuel cell also provides independent information that reflects the energy content of the reformer output, namely the current and voltage of the electrical output of the fuel cell, and the temperature of the combustor located downstream of the anode exhaust. Therefore, we also assume that a secondary, slower measurement is available that reflects the energy content of the output stream, although with a delay due to transport to the fuel cell and combustor.

The control will be demonstrated using a detailed numerical model of the fuel reformer. The detailed model, developed in [75], is a one-dimensional (1D) packed-bed reactor code written in MATLAB environment with an interface to Cantera. All the mass transport, energy and momentum continuity equations are solved in axial direction along the catalyst bed. Thermodynamic and transport properties of the gas-phase and catalytic reactions are calculated via Cantera functions.

We split the tubes into a set of 25 mesh points. Since rhodium is highly active and, consequently, the process reactions occur mainly at the beginning of the reactor, the mesh points are chosen so that they are more concentrated near the inlet. The differential equations introduced in [75] hold for each mesh point. These are partial differential equations with respect to displacement and time. A spatial discretization method, specifically the finite-volume discretization method, is used to change the partial differential equations to ordinary differential equations with respect to time. In this method, along the spatial (x) coordinate, the differential value of each unknown variable at point j is evaluated at the faces of the the neighboring cells. Therefore, the differential equations corresponding to x change to algebraic equations. Figure 7.3 illustrates the finite-volume mesh network. The resulting nonlinear ordinary equation sets are solved by MATLAB using the ode15s solver.

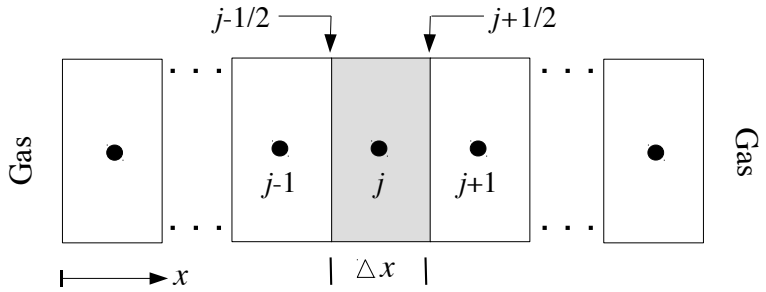

Figure 7.3: Illustration of finite-volume mesh network. The shaded area represents a finite volume for cell j

7.4 Economic Control of Methane Reforming

As an energy source that must compete in the marketplace, optimization of the economic performance of fuel cell systems, including reformers, is of recognized importance [44, 76–79]. [76] presents an economic analysis of a solid oxide fuel cell unit. Also, a low cost fuel cell inverter has been developed and analyzed in [78]. The economic performance of three fuels including methanol, ethanol, and diesel in producing hydrogen in steam reforming has been compared in [79] to conclude which fuel results in a cheaper product. However, this economic point of view not only should be applied in the fuel cell unit design, but also should be taken into account in the process control. Each of the inputs flows that can be actuated by the controller has a cost associated with it. The input fuel must be purchased, blowers use electricity to create the required pressure and velocity of input air, steam must be heated, and the exhaust gases applied to the heat exchangers could be used for other purposes in a combined heat and power system, or will require a certain amount of hydrogen slip in order to supply the combustor. Therefore, the controller not only should reach the physical objectives, but also should be able to reach them by applying the input flows of minimum cost.

Prior work on control of methane reforming and systems including methane reforming, e.g.[80–88], does not consider economic optimization and only focus on system stability, power tracking and physical limitations. However, there exist many sets of inputs that

satisfy the physical tracking constraints. Among them, we can choose the set of inputs that are most economical. The controller developed here is an MPC designed to meet the following control objectives:

- Ensure the outlet energy content of the reformer meets an external load demand.
- Keep system within desired operating limits. The operating limits that are considered here are
  - The temperature inside the reactor is limited to be between lower and higher bounds. An extremely high temperature can damage the catalyst and should be avoided. On the other hand, a low temperature can decrease the amount of the hydrogen produced.
  - The input composition is chosen to avoid carbon deposition in the reactor at the current reactor temperature.
- Reject disturbances. The control objectives should be met despite disturbances such as a change in the fuel composition or inlet gas temperature beside measurement noise.
- Operate efficiently. The controller should apply the input flows of minimum cost.

In this section, we assume a heat exchanger is utilized to heat up the reformer and the system is not running adiabatically. The reactor includes 20 tubes where the nominal values of the parameters associated with a single tube of the reactor are listed in Table 7.1. In addition, we assume that direct measurement of the outlet energy content is slow and with significant delay. This would be the case when the primary mechanism for estimating the energy content of the reformer is via the power produced by a down stream fuel cell.

7.4.1 Control Approach

The control approach is model based. A central feature is a reduced order dynamic model of the reformer which is obtained by an identification process. Based on the described control objectives, five signals are chosen as the model outputs, while four manipulated inputs are selected.

Table 7.1: Nominal parameters of CPOX process

	Value	**Unit**	**Description**
L	2.54×10^{-2}	m	Reactor tube length
D	1.58×10^{-2}	m	Reactor tube diameter
A_s	4×10^4	m^{-1}	Catalyst surface area
τ	2		Tortuosity of foam monolith
ϕ_g	0.75		Porosity of foam monolith
r_p	2.8×10^{-4}	m	Average pore radius
d_p	8.5×10^{-4}	m	Particle diameter of foam
λ_s	1.4	W/m°K	Quartz conductivity
$c_{p,s}$	1225	J/kg °K	Specific heat of quartz
P_{in}	1	atm	Inlet gas pressure
T_{in}	673	°K	Inlet gas Temperature

- Inputs. $u_k = [\dot{m}_{Fuel} \quad \dot{m}_{Air} \quad \dot{m}_{Steam} \quad T_{HEx}]^T$
  - *Fuel mass flow rate* ($\dot{m}_{Fuel}$)
  - *Air mass flow rate* ($\dot{m}_{Air}$)
  - *Steam mass flow rate* ($\dot{m}_{Steam}$)
  - *Heat exchanger temperature* (T_{HEx}): In the current study, we use the heat exchanger temperature as a proxy for actuators that control the fuel cell exhaust.
- Outputs. $y_k = [\Psi \quad T_s \quad \lambda_s \quad X_{CO} \quad X_{CO_2}]^T$
  - *Outlet Power* (Ψ): The fuel cell must produce sufficient energy to meet the external load demand.
  - *Reactor temperature* (T_s): The temperature limitations of the reactor must be met.
  - *Outlet flow thermal conductivity* (λ_s), CO *outlet mole fraction* (X_{CO}) and CO_2 *outlet mole fraction* (X_{CO_2}): These are used to indirectly determine the energy content of the output flow.

Using this model, an MHE, discussed in Section 2.3, is employed to determine the current state of the system as well as the value of important but unmeasured signals. By knowing the

current state of the system, the current and future input trajectories that provide optimal system response while meeting key constraints are obtained by an MPC. The controller cost function will be defined for which the control actions is taken with minimum input cost.

In the remainder of this section, we discuss in detail the foundational elements of the control approach. This includes the application of a system identification method to find a reduced order model that can be used as the plant model by the MHE and MPC, MHE and finally MPC.

7.4.1.1 System Identification

Both the Moving Horizon Estimator and the Model Predictive Controller use a system model as an integral part of their operation. Since this model is used within an optimization routine that runs every sample interval, this model must be relatively simple while the uncertainty of the model is well quantified. Although the reformer system is nonlinear, for sufficiently small deviations, the behavior can be predicted using a linear system. Thus we will identify a linear system representations that are valid for specific operating points. Because the operating point will be continually changing, it is convent to use a linear system to model the differential behavior. Specifically, we consider the state space form (2.1) as the linear model about each operating point.

To find the model parameters A, B, C, D, a system identification method from the class of subspace identification methods will be used. The specific algorithm utilized is described in [89]. The data for system identification is obtained from the detailed model by perturbing the inputs around a predefined operating point, with a sampling rate. Before the identification procedure, input and output data are de-trended and normalized by subtracting from their mean values and dividing by their standard deviations. This normalizes the signals so that they are all approximately the same scale. Since the temperature dynamics are significantly slower than the other outputs, we design two separate experiments for modeling the slow and fast dynamics (low and high frequency) parts. For each output, a 1×4 system is identified (1 output and 4 inputs). These five models are then combined into a single 5×4 minimal

realization.

7.4.1.2 Moving Horizon Estimation

Moving Horizon Estimator is used to find the current state of the small variation model as well as estimate the outlet power which its fast measurement is not available. Since our model is only valid for small signals, we need the *absolute* value of the outputs to rescale them to the real values. The absolute value of the outputs except the outlet power is available from sensor measurements. The absolute value of the outlet power is obtained via a second estimator using the slower and delayed outlet power measurements. As the outlet power measurements contain a delay, the later estimates will contain the same delay. However, this estimate, shown by $\hat{\Psi}_{-d}$,where d denotes the sample number of the delay, can be brought to the current time via the fast estimates of the signal deviation as

$$\hat{\Psi}_0 = \hat{\Psi}_{-d} + \sum_{k=-d+1}^{0} \delta\Psi_k. \tag{7.6}$$

7.4.1.3 Model Predictive Control

For the fuel processing system, the controller must meet the following objectives.

- The outlet power should exceed the external load demand: $\Psi^{min} \leq \Psi$. Since the objective will be to minimize the cost of the inputs, this constraint ensures that the outlet power tracks the external load demand.
- The reactor temperature should be in an acceptable range: $T_s^{min} \leq T_s \leq T_s^{max}$.
- Carbon deposition along the reactor during the process should be avoided.
- Input variation should be smooth, $|\delta u| \leq \delta u^{max}$.
- The controller should be robust against noise.
- All the objectives should be achieved with minimum input cost.

The following assumptions are assumed to hold:

- Inlet gases can be perfectly mixed before entering to the reactor.

- The boundary between the reactor and the surrounding heat exchanger is at a uniform temperature.
- Thermal conductivity, solid temperature, CO and CO_2 all can be measured at the outlet with a $t_s = 0.05$ s sampling time.
- Power conveyed to the external load can be measured 4 times per second. However, each measurement will be available only after a delay of $t_d = 0.5$ seconds after the corresponding output flow has left the reactor. This delay accounts for transport delays, as well as the computational time for the estimator.

The designed model predictive controller determines the appropriate control action for the next sample time by solving the following optimization problem in each sample time:

$$\begin{aligned}
\min_{\delta u_0, \cdots, \delta u_P} \quad & \sum_{k=1}^{P} \Phi_k(y_k^r, u_k) + \Xi(\zeta) \\
\text{subject to.} \quad & x_{k+1} = Ax_k + B\delta u_k \\
& \delta y_k^r = C^r x_k + D^r \delta u_k \\
& y_k^r = y_0^r + \sum_{\ell \le k} \delta y_\ell^r \\
& u_k = u_0 + \sum_{\ell \le k} \delta u_\ell \\
& -\delta u^{max} \le \delta u \le \delta u^{max} \\
& T_s^{min} \le T_s + \zeta_{T_s} \le T_s^{max} \\
& \Psi^{min} \le \Psi + \zeta_\Psi \\
& dist_{CDB}^{min} \le dist_{CDB}
\end{aligned} \tag{7.7}$$

where $*^{min}$ and $*^{max}$ set the lower and upper bounds of *, respectively. ζ is an auxiliary variable that is non-zero only when the bounds must be violated to achieve a feasible solution. The optimization variables are the system input sequence δu_k over the prediction horizon p into the future. $\Phi_k(.)$, the cost function of the controller, is defined for which all objectives are satisfied with minimum input costs:

$$\Phi_k(y_k^r, u_k) = \gamma \|\delta u_k\|_2^2 + t_s(\lambda_{Fuel}|\dot{m}_{Fuel}| + \lambda_{Air}|\dot{m}_{Air}| + \lambda_{Steam}|\dot{m}_{Steam}|) + \lambda_{HEx}|T_{HEx}| \tag{7.8}$$

Since the cost of each input is obtained by multiplying its price by the total amount of the input consumed in future (note that $m = t_s\dot{m}$), we minimize a weighted sum of the absolute value of the inputs. For the rest of the terms of the cost function, we use the Euclidean norm as the metric. The term $\gamma\|\delta u_k\|_2^2$ prevents the input from excessive changes or chattering in the presence of noise. It is also useful when two different sets of inputs give us the same or a close minimum value. In this case the preference is to keep the inputs at the previous state. The second cost function $\Xi(\zeta)$ determines the cost of not meeting the required bounds, and defines as follow:

$$\Xi(\zeta) = \gamma_\Psi\|\zeta_\Psi\|_2 + \gamma_{T_s}\|\zeta_{T_s}\|_2. \tag{7.9}$$

a supervisory controller can initiate shutdown if the value of ζ becomes too large.

The variable $dist_{CDB}$ represents the CDB distance. Since for a given reactor condition the safety margin increases with temperature, it is only necessary to check the coking propensity at the inlet, which has the lowest temperature. Then, by considering a safety margin and keeping the system in a predefined distance from the barrier it can be ensured that the system remains in the coke free region. The safety margin is given by $dist_{CDB}^{min}$. Since producing coke into the reactor is to be avoided, this constraint is hard constraint. If there is case that there is no solution in free coke region, then reactor shutdown would be initiated. In this study the barrier is considered to be constant since the inlet gas temperature is constant.

7.4.2 Simulation Results and Discussion

The simulation results of the identification process and the results of the designed economic controller are presented in this section.

7.4.2.1 Identification Results

The 25-mesh detailed model was run at the operating point 6.25e-6 kg/s fuel flow, 4e-5 kg/s air flow, 5e-7 kg/s steam flow, 973.15 K heat exchanger temperature. The operating point has been chosen in which each reactor produces 250W. Two experiments were run with

perturbations around this operating point defined by two PRBSs with different frequency ranges corresponding to the fast and slow parts of the model with 2% deviation for all inputs except heat exchanger temperature, for which it is a 10% deviation.

The detailed simulation was run with simulated time of 2 minutes and sampling rate of $t_s = .05$s for the faster input perturbations (2400 samples), and 4 minutes for the slower perturbations (4800 samples). 1500 samples were used for identification and rest used for validation. The state dimension of the identified 5×4 model is 11.

Since the identified model is to be used by the MPC for predicting the future behavior of the system over a specific horizon, we validate it by a p-step ahead predictor where p is the prediction horizon of the MPC. In this study the MSE will be reported based on this signal. The validation results represent a quite accurate model such that the validation MSE of the reactor temperature, power, thermal conductivity, CO and CO_2 mole fractions validation are 0.49067, 4.67e-10, 1.68e-8, 2.89e-9, and 3.55e-9, respectively.

A representative part of the validation results are shown in Figure 7.4. This shows the results for 15 step ahead prediction, or 0.75 seconds. Note that the model is quite accurate in predicting the behavior of all the outputs over this range.

We have presented result only at a single operation point, but the same process applies to obtain models valid at other operating points. However, it will be demonstrated in later sections that a single linear model can be valid over a relatively wide operating range.

7.4.2.2 Controller Results

The values of the parameters chosen for the MHE and MPC are shown in Table 7.2. $*_1$ and $*_2$ indicate parameter * of the faster and slower estimators, respectively. The costs of inputs have been chosen according to the real cost of input gases showed in Table 7.3. In Table 7.3, it is assumed that the price of methane is the same as natural gas price and is equal to 4 USD/1million Btu or 0.2077 USD/kg. The necessary energy for heating up 1kg of fuel, air, and steam from 25° C to 400° C is 821, 375, and 3240 kJ, respectively. The cost of releasing 1 kJ energy is 8e-6 USD since 1 kWh energy can be gained from 7.86 ft^3

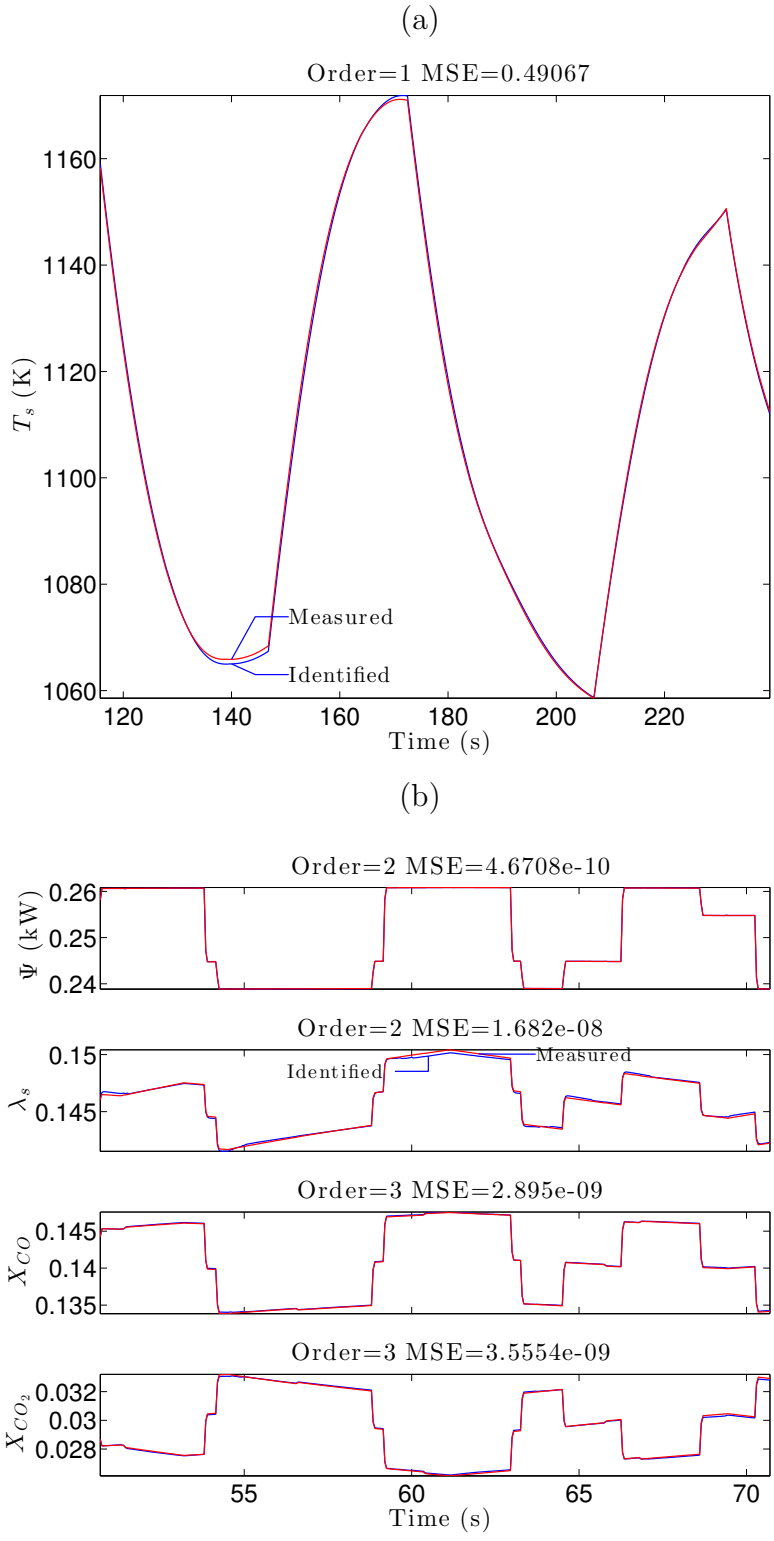

Figure 7.4: Validation results of system identification. Dashed lines show the detailed model while the solid line indicates the output of the identified model. (a) Reactor Temperature (b) All other outputs.

Table 7.2: MHE and MPC parameters used in economic control of methane reforming

Parameter	**Value**	**Parameter**	**Value**
P	15	R_2	0.0334
Q_1	I	Q_2	$0.2I$
γ	0.05	γ_{T_s}	10
γ_Ψ	20	λ_{Fuel}	213
λ_{Steam}	26	λ_{Air}	3
λ_{HEx}	2.94e-5	$dist_{CDB}^{min}$	0.03
T_s^{min}	1000 K	T_s^{max}	1350 K
δu^{max}	$\begin{bmatrix}1.25e-6 & 8e-6 & 1e-7 & 1.946\end{bmatrix}'$		
R_1	$\text{diag}\begin{bmatrix}0.0179 & 0.0144 & 0.0294 & 0.0660\end{bmatrix}$		

natural gas. Therefore the cost of heating up 1kg of each input to 400°C is as shown in the second column of Table 7.3. Furthermore, it is assumed that the volume of the circulating air through the heat exchanger is 4 times of the total volume of the tubes and therefore the price of heating up the air is 2.94e-8 USD/K [12].

Table 7.3: Input costs of methane reforming

	Price (USD/kg)	**Cost of heating up to 400°C (USD/kg)**	**Total(USD/kg)**
Fuel	0.207	0.0065	0.2135
H_2O	0	0.026	0.026
Air	0	3e-3	0.003

The MPC controller has been applied to the system after running it open loop for 3.5 s to make sure both estimators reach their steady state. The closed loop system has been simulated for around 12 seconds with varying external load demand between 4.6 and 5.6 kW. A 1% white gaussian measurement noise has been added to the simulated measurements. Figure 7.5a presents the results for the outlet power. As seen, the controller has a strong performance in power estimation and noise rejection. Although we do not have fast measurement of the outlet power, results show a perfect estimation of the power and a successful switch between different level of demands. One of the key factors of the controller robustness against noise is that the controller uses the estimated signals which are much smoother than the real noisy signals. This robustness against noise prevents inputs to continuously change

[12]The prices are reported based on the gas cost in 2015

because of output variations caused by noise.

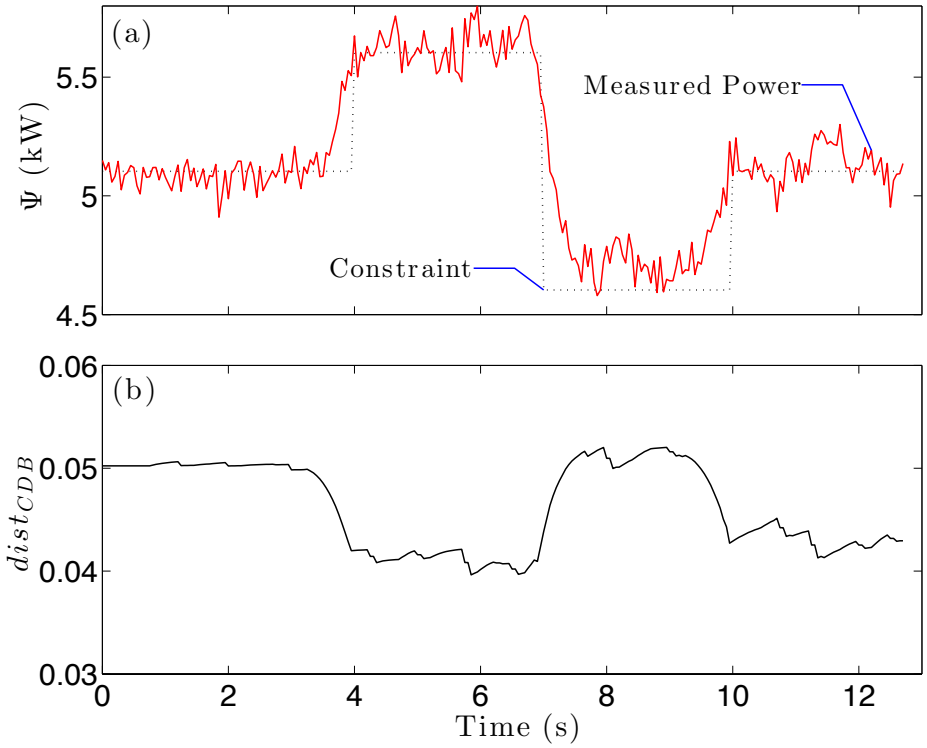

Figure 7.5: Controller performance. (a) Outlet power (b) Distance from the carbon region barrier.

Although we have not defined a tracking objective for the controller, we can see that the power tends to the minimum acceptable power. This naturally occurs because the controller minimizes the cost of inputs.

Figure 7.5b confirms that the controller succeed to prevent carbon deposition by keeping the system in the desired distance from the barrier of the ternary diagram. As seen, the distance is always more than 0.03 during the run.

The optimal inputs applied by the controller is shown in Figure 7.6a. Note that the air and heat exchanger temperature trend down during the run. This is a result of the input cost optimization, which recognizes the cost of maintaining the temperature of the heat exchanger. This optimization does not occur more quickly because of the constraints placed on the controller action at each time step.

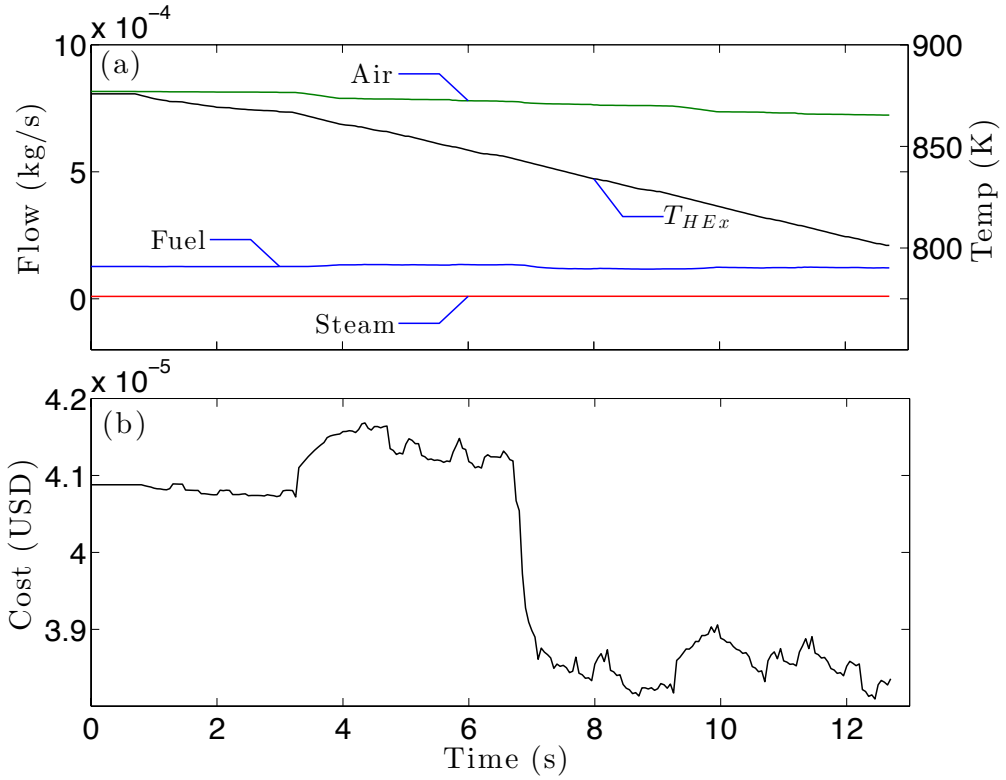

Figure 7.6: Economic control of methane reforming (a) Inputs. (b) Total cost of inputs.

The total cost of inputs selected by the controller in each step on the prediction horizon p is shown in Figure 7.6b. As seen, the input cost trends down during the time even though the power demand at the beginning and the end of simulation is the same. In other words, the controller makes the process more profitable without decreasing the control performance. The cost keeps going down until one of the outputs is bounded by its physical constraints.

7.5 Robust Model Predictive Control of Methane Reforming under Parametric Uncertainty

In a methane reformer, some of the parameters such as the reactor length and diameter, and also foam properties including tortuosity, porosity, specific catalyst area, etc. are uncertain and may vary from a reactor to another one. Consequently, different reformers may have different specification and do not share the same behavior. This is explored in [90] where the author discusses the effect of parametric uncertainty on the outlet temperature

and gas composition. On the other hand, we can not identify a separate model and design a controller for each single reformer. Therefore, the designed controller must be robust against the parametric uncertainty somehow that it guarantees the control performance for any values of the uncertain parameters within the given ranges.

The focus of this section is on designing a robust model predictive control for ATR (i.e., no external source is used to heat up the reactor) with the capability of satisfying the control objectives in presence of parametric uncertainty. The employed method should still guarantee the stability and be applicable on-line. We assume that the nominal value of the parameters and their tolerance around the nominal value are given. In other words, the exact value of the parameters are not available and we only know they lie in given ranges.

7.5.1 Uncertain Parameters

Manufacturing variations will give rise to associated variations in the physical parameters of the reformer. [90] presents a practical uncertainty range for some of these parameters for a CPOX process on rhodium foam. A maximum 2.5% tolerance is also taken into account as the uncertainty of the length and diameter of the reactor. Table 7.4 indicates the nominal values of the uncertain parameters and their tolerances around nominal values, used in this study.

Table 7.4: Uncertainty ranges of methane reforming parameters

Param.	**Nominal value**	**Unit**	**Tolerance**
ϕ_g	0.25		50%
d_p	2.8e-4	m	60%
A_s	4e4	m^{-1}	30%
$c_{p,s}$	1225	J/kg°K	20%
λ_s	1.4	W/mK	24%
ρ_s	2.2×10^3	kg/m^3	20 %
τ	3		50%
L	5e-2	m	2.5%
D	1e-2	m	2.5%

7.5.2 Uncertainty Analysis

Parametric uncertainty can cause uncertainty in both the steady state and transient responses. The steady state uncertainty results in an output bias. Although the controller can compensate for the bias, it will cause problem when a fast measure of the output is not available. In addition, the amount of bias could vary after switching between different operating points. Note that the process is nonlinear and we linearize it around different operating points. On the other hand, the presence of uncertainty in the model dynamics can cause tracking failure during the change in the tracking reference.

In this section, the sensitivity of three control outputs, the carbon to oxygen ratio, the outlet solid temperature, and the outlet LHV to the uncertain parameters is evaluated in both steady state and transient time scales. Mass flow rates of fuel, oxygen and steam are the three input of the system. Figure 7.7a, Figure 7.7b, and Figure 7.7c represent the change in the steady state value of the hydrogen to carbon ratio, solid temperature, and the outlet LHV power where the parameters are chosen at the boundaries of the uncertainty range. As seen, the impact of the uncertainty in porosity, catalyst surface area, solid conductivity, tortuosity, and, the reactor diameter on the outputs steady state is dominant and more significant than the other parameters. Figure 7.8a and Figure 7.8b plot the impulse response of the outlet solid temperature where parameters are chosen at the lower and upper bounds of the uncertainty range. According to the Figure 7.8, the uncertainty in the porosity and specific catalyst area can make a huge change in the system dynamics. Figure 7.9 specifically shows the impulse response for the given uncertainty range of porosity and specific catalyst area.

According to the steady state and dynamics uncertainty analysis, we design the controller based on the six dominant uncertain parameters which have significant impacts on the system behavior, i.e., porosity, specific catalyst area, catalyst surface area, solid conductivity, tortuosity, and reactor diameter.

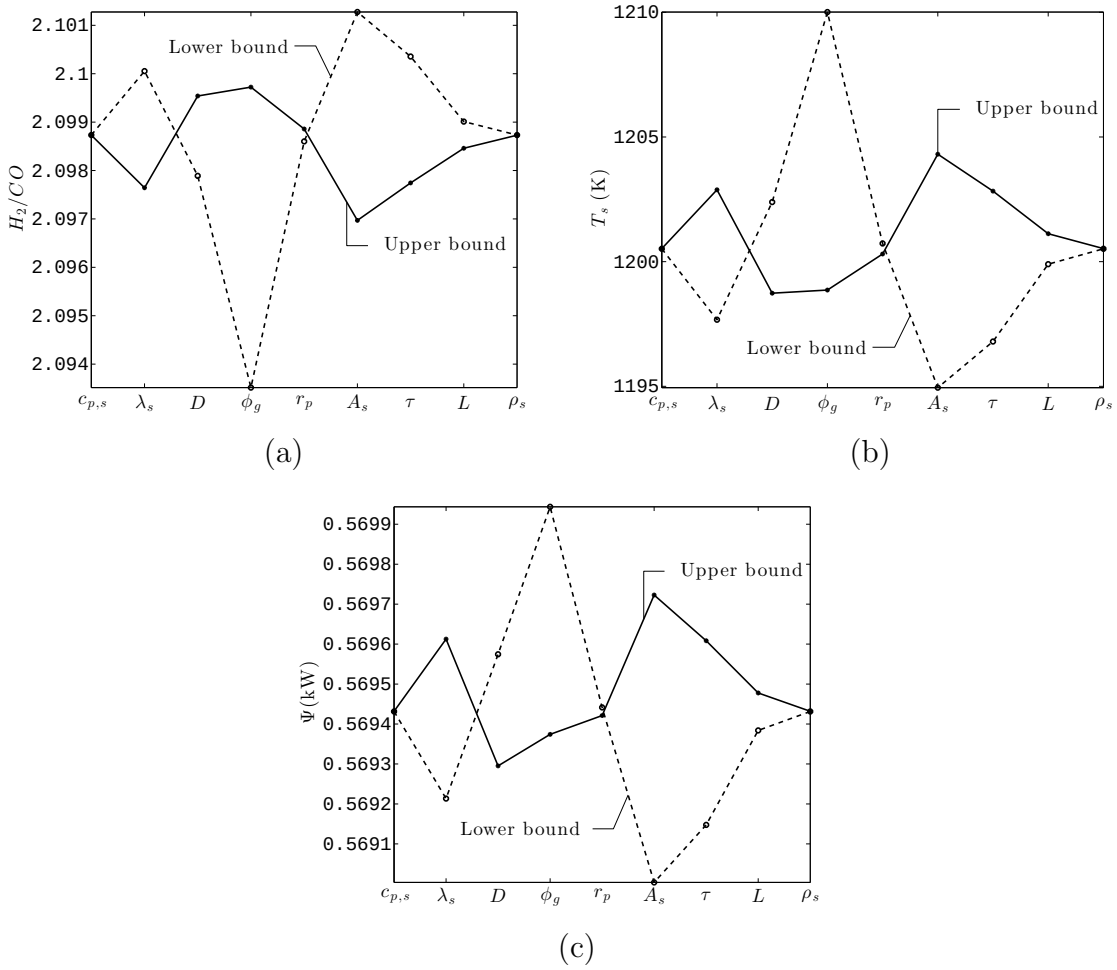

Figure 7.7: Uncertainty Analysis: uncertainty in the steady state of (a) hydrogen to carbon ratio (b) outlet solid temperature (c) outlet power, when the uncertain parameters are set to their possible minimum or maximum values

7.5.3 Control Objectives and Physical Considerations

For this scenario, the process is running AT and the proper sampling time for this case will be $5s$. The other physical assumption will be the same as 7.4. In addition, we add one control objective to what considered in Section 7.4. We aim to keep the hydrogen to carbon ration (H_2/CO) about 2 based on Fischer-Tropsch synthesis. In addition, the controller must be robust against uncertainties. The control objectives and assumptions are given below for consistency.

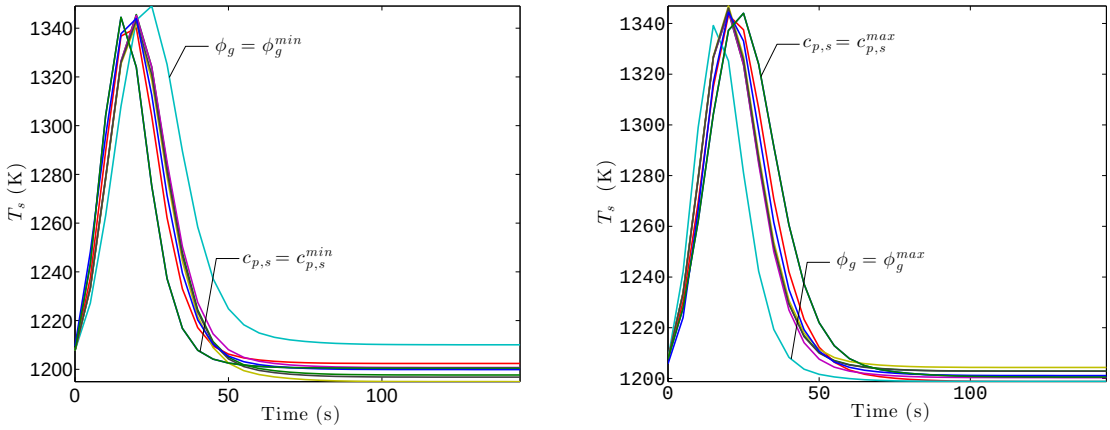

Figure 7.8: Uncertainty Analysis: uncertainty in the outlet temperature dynamics (a) lower bound of the uncertainty ranges (b) upper bound of the uncertainty ranges

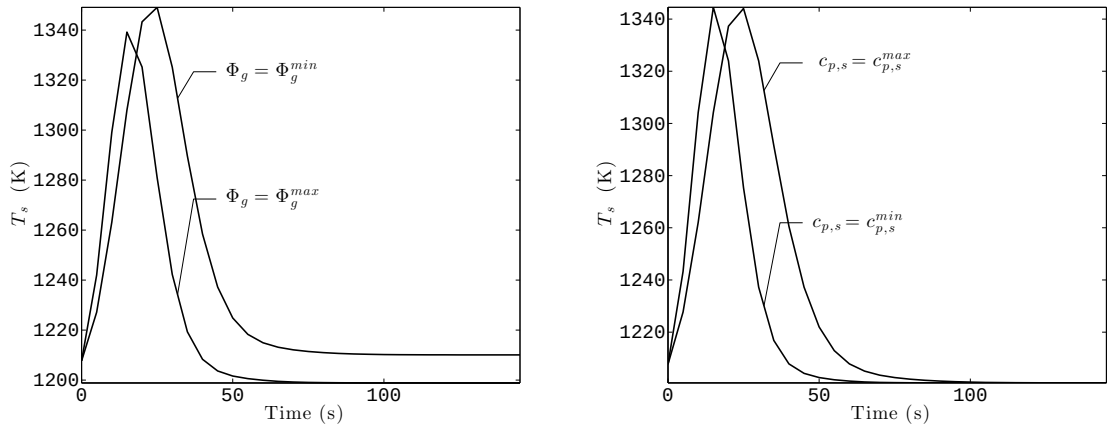

Figure 7.9: Sensitivity analysis on the outlet temperature: (a) Porosity impact (b) Specific catalyst area impact

- The outlet power should exceed the external load demand: $\Psi^{min} \leq \Psi$.
- The reactor temperature should be in an acceptable range: $T_s^{min} \leq T_s \leq T_s^{max}$.
- Hydrogen to carbon ratio should remain close to 2.
- Carbon deposition along the reactor during the process should be avoided.
- Input variation should be smooth, $|\delta u| \leq \delta u^{max}$.

- All the objectives should be achieved with minimum input cost.
- The controller should be robust against measurement uncertainty.
- The stated objectives must be satisfied for any system realization in the given uncertainty range.

The following assumptions are assumed to hold:

- Pure oxygen is injected at the inlet instead of air.
- Inlet gases can be perfectly mixed before entering to the reactor.
- System is running auto-thermal.
- Solid temperature, CO and H_2 (and therefore H_2/CO ratio), and outlet power, all can be measured at the outlet with a $t_s = 5$ s sampling time.

7.5.4 Methodology

Based on the control objectives given in Section 7.5.3, three signals are chosen as the model outputs, while three manipulated inputs are selected.

- Inputs. $u_k = [\dot{m}_{Fuel} \quad \dot{m}_{Oxygen} \quad \dot{m}_{Steam}]^T$
  - *Fuel mass flow rate* ($\dot{m}_{Fuel}$)
  - *Oxygen mass flow rate* ($\dot{m}_{Oxygen}$)
  - *Steam mass flow rate* ($\dot{m}_{Steam}$)
- Outputs. $y_k = [H_2/CO \quad T_s \quad \Psi]^T$
  - *Hydrogen to carbon ratio*(H_2/CO): Must remain close to 2.
  - *Reactor temperature* (T_s): The temperature limitations of the reactor must be met.
  - *Outlet power* (Ψ): The fuel cell must produce sufficient energy to meet the external load demand.

Define the parameter set to be the dominant parameters investigated in Section 7.5.2, $\theta^{SOFC} = [\phi_g, c_{p,s}, \lambda_s, A_s, \tau, D]^T$. In order to design the RMPC controller, first we sample

θ^{SOFC} uniformly and add the extreme points of the parameter space of the dominant parameters to the taken samples to create Θ^{SOFC}. Specifically, we take 500 random samples and add 64 samples corresponding to the parameter ball extreme points. Then, we run the numerical model for every elements of Θ^{SOFC} (by setting the parameters to θ_i) and identify a linear variational model for each point. The model identification process is done using a subspace method. Using these models, one can find the impulse responses corresponding to the Θ^{SOFC} elements and form $\mathcal{S}^{SOFC} = \{S_1, \cdots, S_{564}\}$.

After scaling the inputs-outputs of the system, using the proposed approximate convex hull method, we calculate three model sets $\mathcal{S}_{\mathcal{E}1}$, $\mathcal{S}_{\mathcal{E}2}$, and, $\mathcal{S}_{\mathcal{E}3}$ with $\{\epsilon_1, \epsilon_{k1}\}$, $\{\epsilon_2, \epsilon_{k2}\}$, $\{\epsilon_3, \epsilon_{k3}\}$ corresponding to the three outputs, including, hydrogen to carbon ratio, solid temperature, and outlet power. These model sets have a fewer elements than the original model set $\mathcal{S}^{SOFC}$. Therefore, the control problem is to solve

$$\begin{aligned}
\min_{U_{\mathsf{f}},t} \max_{\mathcal{S}_{\mathcal{E}1},\mathcal{S}_{\mathcal{E}2},\mathcal{S}_{\mathcal{E}3}} \quad & \Phi(Y_{\mathsf{f}}, U_{\mathsf{f}}) + \Xi(\zeta^{min}, \zeta^{max}) \\
& Y_i = M_i \begin{bmatrix} \Delta U_{\mathsf{p}} \\ z_0 \\ \Delta Z_{\mathsf{p}} \end{bmatrix} + N_i \Delta U_{\mathsf{f}} \quad \{M_i, N_i\} \in \mathcal{S}_{\mathcal{E}i} \quad i = 1,2,3 \\
\text{subject to} \quad & Y_i + \zeta_i^{max} + \epsilon^{y_i} \leq b_i^{max}, Y_i + \zeta_i^{min} - \epsilon^{y_i} \geq b_i^{min} \quad i = 1,2,3 \\
& dist_{CDB} \geq dist_{CDB}^{min} \\
& U_{\mathsf{f}} = U_{-1} + \mathcal{I}\Delta U_{\mathsf{f}}, U_{\mathsf{f}} \geq 0
\end{aligned} \tag{7.10}$$

where

$$\begin{aligned}
\Phi(Y_{\mathsf{f}}, U_{\mathsf{f}}) &= \gamma_{du} \|\Delta U_{\mathsf{f}}\|_2^2 + \sum_{j=1,2,3} t_s \lambda_j \|U_{j\mathsf{f}}\|_1 \\
\Xi(\zeta^{min}, \zeta^{max}) &= \sum_{i=1,2,3} \gamma_i (\|\zeta_i^{min} + \zeta_i^{max}\|_2
\end{aligned} \tag{7.11}$$

where j and i reflect the input and output indices, respectively. A supervisory controller can initiate shutdown if the value of ζ^{min} or ζ^{max} becomes too large. λ_j is the cost of the j-th input as listed in Table 7.3.

Table 7.5: Physical constraints

Parameter	Value	Parameter	Value
T_s^{min}	1000 K	T_s^{max}	1350 K
H_2/CO^{min}	1.98	H_2/CO^{max}	2.03
$dist_{CDB}^{min}$	0.03	Ψ^{min}	0.45 kW

7.5.5 Simulation Results

We choose $u = [1.1e-5, 1.2e-5, 1.87e-6]$ kg/s as the operating point to make sure the carbon to oxygen ratio is close to 1, The system is autothermal, and, it lies in the free carbon region on ternary diagram. The numerical model was run for 16 cases for 2000 s with sampling time 5 s. Then 75% of the data was used to identify a 3×3 model for each case. the obtained models are from the order of 12. Then, after scaling inputs by $[1e-5, 1e-5, 2e-6]$ and outputs by $[2, 1200, 0.5]$, the 16 models were mapped to S domain (using $Q = R = I_{3\times3}$) to create $\mathcal{S}^{SOFC}$.

Figure 7.10 shows the results of finding $\mathcal{S}_{\mathcal{E}1}$, $\mathcal{S}_{\mathcal{E}2}$, and $\mathcal{S}_{\mathcal{E}3}$ using the approximate convex hull method. These model sets are to cover the uncertainty on y_1 (hydrogen to carbon ration), y_2 (solid temperature), and y_3 (outlet power). As seen, selecting 11, 5, and 6 extreme models for $\mathcal{S}_{\mathcal{E}1}$, $\mathcal{S}_{\mathcal{E}2}$, and $\mathcal{S}_{\mathcal{E}3}$ results in a good decrement in the error (about 100 times).

After finding the approximate model sets with corresponding ϵ_is, ϵ_{ki}s are calculated back as presented in Figure 7.11.

The controller has been applied to some random systems to satisfy the physical limitations, presented in Table 7.5. Figure 7.13 shows the controller performance when a small measurement noise is added. The controller is applied after 65 s. As seen, the robust MPC is able to convey and keep all the output trajectories within their limits. The 10-step ahead prediction of the output violation bound, calculated by the controller for the scaled system, is given in Figure 7.14.

Figure 7.12 shows the distance of the operating point from the carbon region of the ternary diagram, over the control process. As seen, the carbon barrier distance is always

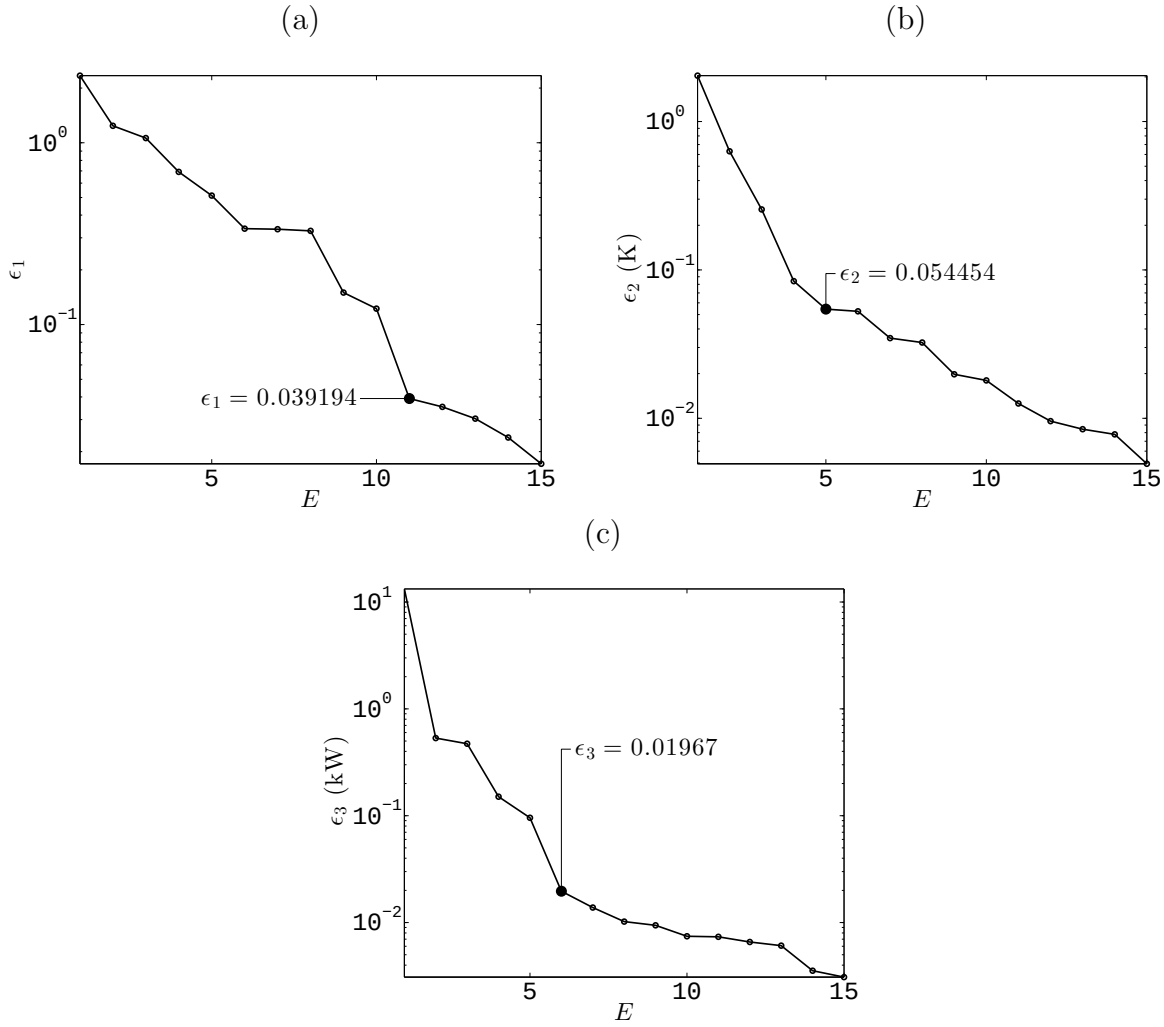

Figure 7.10: Model set determination for RMPC of methane reforming. The model sets are related to (a) hydrogen to carbon ratio (b) outlet solid temperature (c) outlet power.

kept above the expected limit (0.03).

7.6 Summary

In this chapter, the control of a methane reforming process as a key element of a solid oxide fuel cell in two different scenarios was studied. In the first scenario in Section 7.4, the system parameters were assumed to be known. The controller was designed in a way that the physical constraints are satisfied with the optimum inputs from the minimum cost. The second scenario, presented in Section 7.5, accounted for the model uncertainty to achieve a robust control for the case that the system parameters are uncertain. In detail, three outputs,

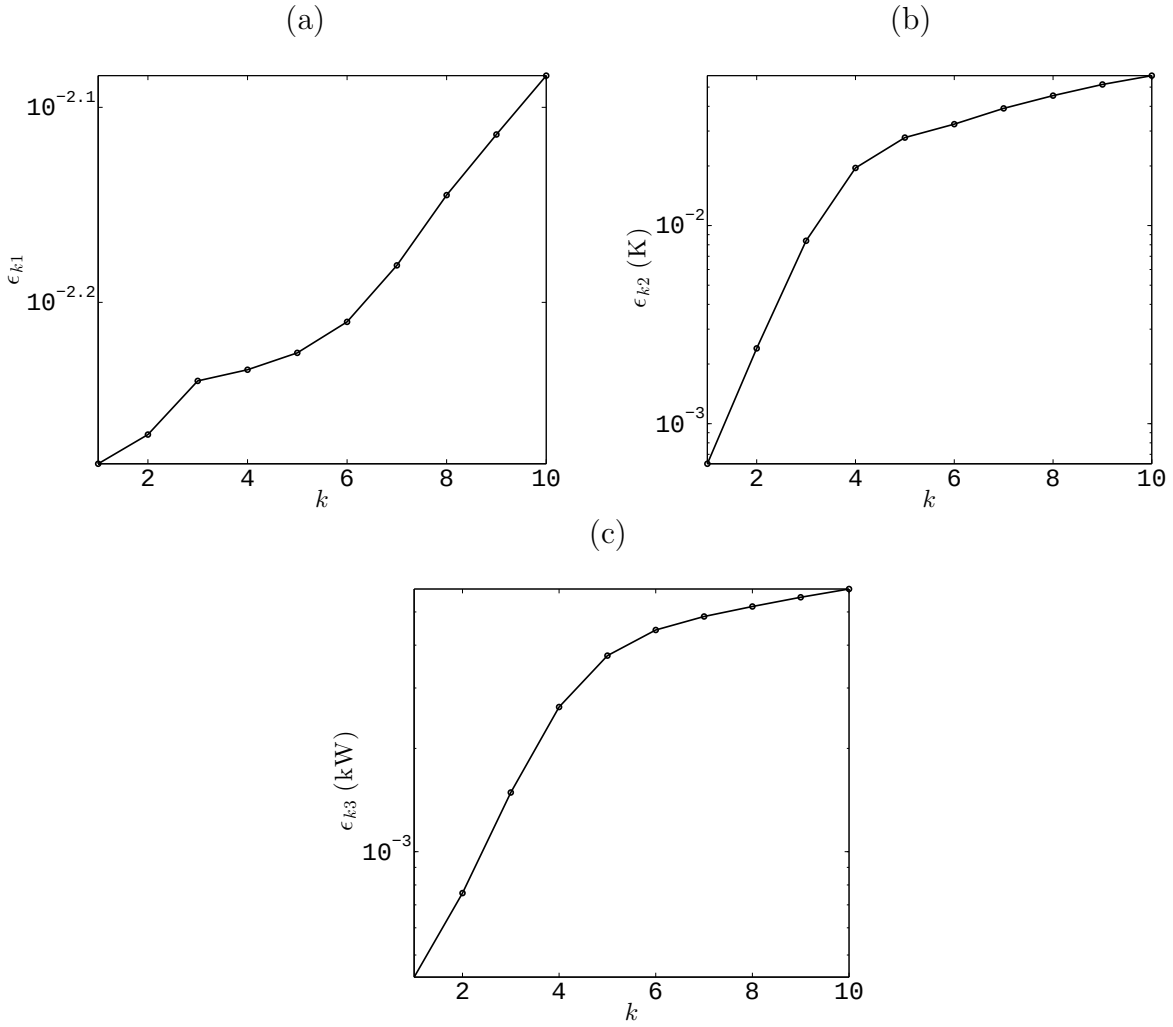

Figure 7.11: Model set determination for RMPC of methane reforming. calculating back ϵ_k for (a) hydrogen to carbon ratio (b) outlet solid temperature (c) outlet power.

including hydrogen to carbon ratio, solid temperature, and outlet power were expected to be in a certain range. The introduced RMPC in Chapter 5 was used to control of the system after some sensitivity analysis. Simulation results show the solid performance of the proposed controller in dealing with MIMO systems.

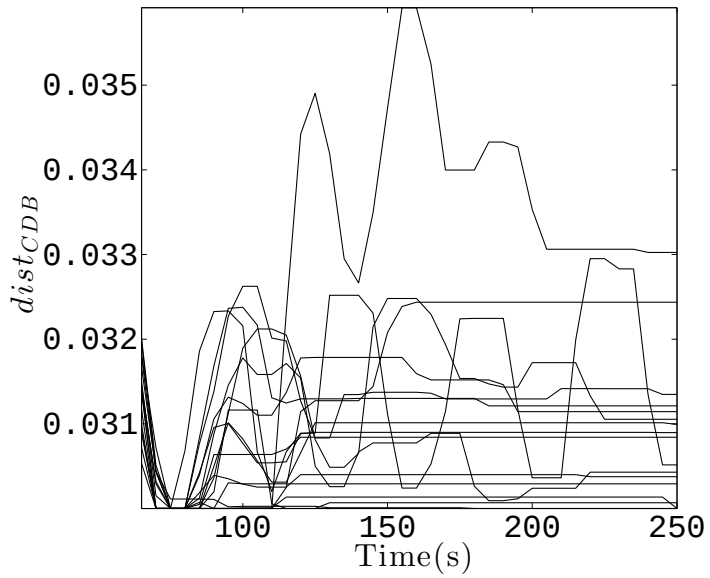

Figure 7.12: Distance of the methane reforming operating point from the carbon region of the ternary diagram, resulted by robust MPC.

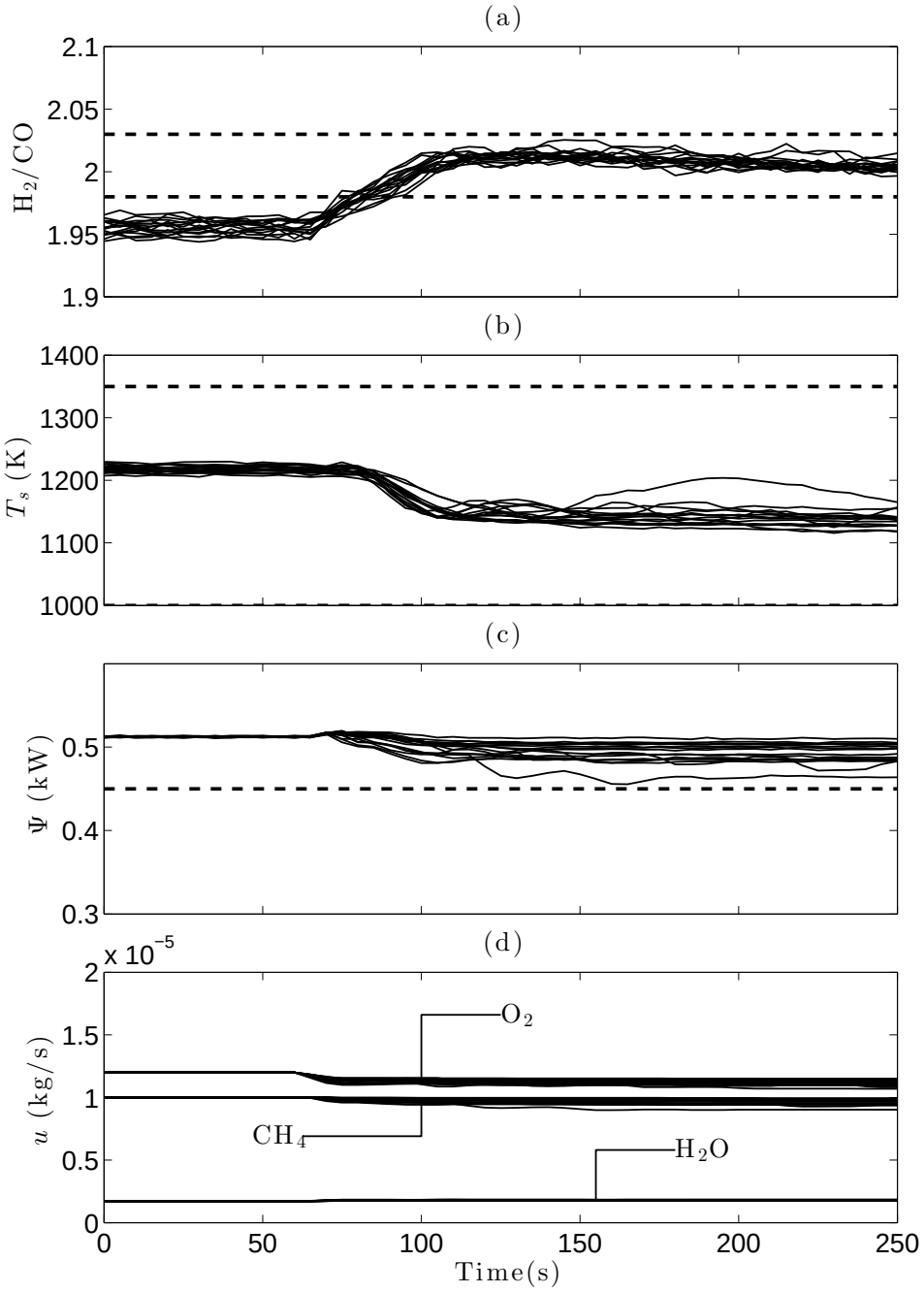

Figure 7.13: Control performance of robust control of methane reformer (a,b,c) outputs. (d) inputs. Controller is applied after 65 s.

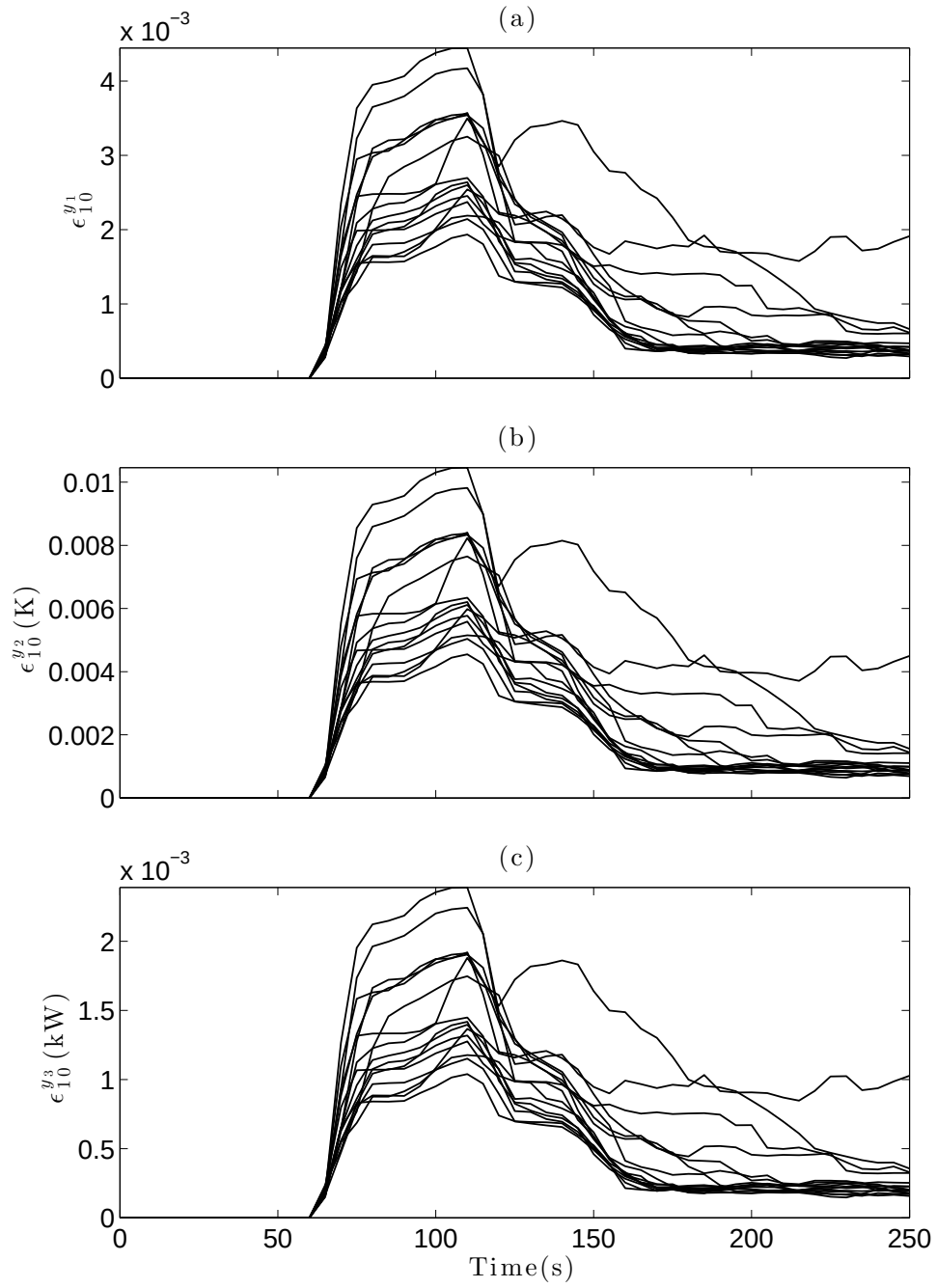

Figure 7.14: Output violation bounds provided by RMPC controller (a) ϵ_{10}^{y1} (b) ϵ_{10}^{y2} (c) ϵ_{10}^{y3}.

CHAPTER 8

CONCLUSION AND DISCUSSION

In this thesis we developed a new open loop robust MPC with the capability of handling measurement and model uncertainties. The proposed controller is able to provide a guaranteed dynamic bound on the predicted system output trajectories with a low computational complexity. At each step the optimum input is calculated for the worst case situation by maximizing the objective function over the uncertainty set. The computational complexity, therefore, grows by increasing the number of uncertainty set elements. The true uncertainty set may contain infinite elements. However, a finite set Θ is obtained using a sampling procedure similarly to scenario approach. The finite set obtain by the scenario approach still includes a number of samples which imposes a high computational complexity.

In order to reduce the complexity and establish the proposed robust MPC, we used the concept of convex hull to reduce the number of uncertainty set elements. In Chapter 4, we introduced a new prediction mapping (denoted with S) which represents the output predictions. The invaluable property of this mapping is that its convex hull vertices provide a bound on the output trajectories. In other words, the output upper and lower bounds predicted using the extreme points of the uncertainty set in S domain, is also an upper and lower bounds for any arbitrary system within the uncertainty range. This implies that the rest of the points are redundant and can be removed from the uncertainty set which highly decreases the computational complexity of the robust MPC.

Since the introduced prediction mapping S lives in high dimensions and the number of extreme points could be comparable to the number of original set elements, we brought the idea of using an ϵ-approximate convex hull which results in a significantly fewer extreme points. This approach gives the user a degree of freedom to make a trade off between the time complexity and accuracy. For this purpose, a novel greedy ϵ-approximate convex hull

was introduced in Chapter 3. The method effectiveness for high dimensions was proved by numerical examples and comparison to prior work. It is worth reporting that this method can be used for uncertainty characterization in other areas as well.

Using the proposed ϵ-approximate convex hull method, we shrink the uncertainty set to the set $\mathcal{S}_{\mathcal{E}}$, called model set. In this case, the output trajectories may violate the bounds given by the model set due to the approximation error. In order to deal with this issue, we provided a dynamic bound on the output violation (denoted by ϵ_k^y at time k) which accounts for the approximation error ϵ.

The performance of the stablished robust MPC was evaluated by implementing on two case studies. The robust control of a DC-DC floating interleaved boost converter where the input voltage and the model are uncertain was studied in Chapter 6.The simulation results prove the capability of the proposed controller to deal with disturbances and uncertainties for SISO system. In Chapter 7, we investigated the implementation of the robust controller on the methane reforming as a MIMO system. According to the simulation results, the designed controller has a solid performance in satisfying all the control objectives and meeting the physical constraints for any experimented arbitrary system within the uncertainty range.

APPENDIX - PROOF OF THEOREM 2

Let $y_k^{\theta_{act}}$ be the actual system output at time k where the actual system is an arbitrary system with uncertain parameters $\theta_{act} \in \Theta$. Also, let $y_k^{\theta_i}$ be the k-step ahead prediction of the output using the i^{th} element of $\Theta_{\mathcal{E}}$ (θ_i).

If $\operatorname{Conv}\Theta_{\mathcal{E}} = \operatorname{Conv}\Theta$, it can be easily shown that $y^{\theta_{act}} \in \operatorname{Conv} y(\Theta_{\mathcal{E}})$ and consequently the actual output is always bounded by the predicted output set corresponded to $\Theta_{\mathcal{E}}$. In the case of approximate convex hull though, the actual output could violate from the boundaries of the predicted outputs. This violation is obtained from $|y_k^{\theta_{act}} - \sum_{i=1}^{E} \alpha_i y_k^{\theta_i}|$ where $y_k^{\theta_*}$ is the k-th row of $Y(\theta_*)$, and according to (4.12) it is calculated as

$$y_k^{\theta_*} =^k M(\theta_*) \begin{bmatrix} \Delta U_{\mathsf{p}} \\ z_0 \\ \Delta Z_{\mathsf{p}} \end{bmatrix} +^k N(\theta_*)\Delta U_{\mathsf{f}}.$$

By adding and subtracting z_0, the violation bound can be calculated as $|(y_k^{\theta_{act}} - z_0) - \sum_{i=1}^{E} \alpha_i (y_k^{\theta_i} - z_0)|$.

By setting $\Delta U_{\mathsf{p}}, \Delta Y_{\mathsf{p}}$ to zero in (4.8) and according to the structure of Ψ and Γ matrices defined in (4.10), it can be shown that $\hat{X} = \begin{bmatrix} z_0 & 0 & \cdots & 0 \end{bmatrix}$ is the exact solution of the least square problem (4.8) (i.e., the system is at the steady state and its steady state output value and, therefore, its k-step ahead prediction are z_0). Therefore, z_0 can be written as

$$z_0 = y_k =^k M \begin{bmatrix} 0 \\ z_0 \\ 0 \end{bmatrix}$$

for any k. Then,

$$y_k^{\theta_*} - z_0 =^k M(\theta_*) \begin{bmatrix} \Delta U_{\mathsf{p}} \\ 0 \\ \Delta Z_{\mathsf{p}} \end{bmatrix} +^k N(\theta_*)\Delta U_{\mathsf{f}}.$$

A guaranteed bound regarding the output violation due to the approximation process, therefore, can be obtained as equation (A.1). (A.1c) is achieved from the Holder's inequality. According to (4.14), it can be written as (A.1d). Let $\hat{\alpha} = \arg\min \lambda(S, \mathcal{S}_{\mathcal{E},\alpha})$

$$|y_k^{\theta_{act}} - \sum_{i=1}^{E} \alpha_i y_k^{\theta_i}| = \left| \left({}^kM(\theta_{act}) - \sum_{i=1}^{E} \alpha_i \, {}^kM(\theta_i) \right) \begin{bmatrix} \Delta U_\mathsf{p} \\ 0 \\ \Delta Z_\mathsf{p} \end{bmatrix} + \left({}^kN(\theta_{act}) - \sum_{i=1}^{E} \alpha_i \, {}^kN(\theta_i) \right) \Delta U_\mathsf{f} \right| \tag{A.1a}$$

$$\leq \left| \left({}^kM(\theta_{act}) - \sum_{i=1}^{E} \alpha_i \, {}^kM(\theta_i) \right) \begin{bmatrix} \Delta U_\mathsf{p} \\ 0 \\ \Delta Z_\mathsf{p} \end{bmatrix} \right| + \left| \left({}^kN(\theta_{act}) - \sum_{i=1}^{E} \alpha_i \, {}^kN(\theta_i) \right) \Delta U_\mathsf{f} \right| \tag{A.1b}$$

$$\leq \left\| {}^kM(\theta_{act}) - \sum_{i=1}^{E} \alpha_i \, {}^kM(\theta_i) \right\|_{s_1} \left\| \begin{bmatrix} \Delta U_\mathsf{p} \\ 0 \\ \Delta Z_\mathsf{p} \end{bmatrix} \right\|_{s_2} + \left\| {}^kN(\theta_{act}) - \sum_{i=1}^{E} \alpha_i \, {}^kN(\theta_i) \right\|_{s_1} \left\| \Delta U_\mathsf{f} \right\|_{s_2} \tag{A.1c}$$

$$= \lambda_k(S^{\theta_{act}}, \mathcal{S}_\mathcal{E}, \hat{\alpha}) \left\| \begin{bmatrix} \Delta U_\mathsf{p} \\ \Delta Z_\mathsf{p} \end{bmatrix} \right\|_{s_2} + \lambda_{k+p}(S^{\theta_{act}}, \mathcal{S}_\mathcal{E}, \hat{\alpha}) \left\| \Delta U_\mathsf{f} \right\|_{s_2} \tag{A.1d}$$

$$\leq \epsilon \left\| \begin{bmatrix} \Delta U_\mathsf{p} \\ \Delta Z_\mathsf{p} \end{bmatrix} \right\|_{s_2} + \epsilon_k \left\| \Delta U_\mathsf{f} \right\|_{s_2} \tag{A.1e}$$

for a fixed S. Since for any $\theta \in \Theta$ including θ_{act} the approximate convex hull method guarantees that $\lambda_k(S(\theta), \mathcal{S}_\mathcal{E}, \hat{\alpha}) \leq \epsilon$ for $1 \leq k \leq p$, and because $\lambda_{p+k}(S(\theta_{act}), \mathcal{S}_\mathcal{E}, \hat{\alpha}) \leq \max_{\theta \in \Theta} \lambda_{p+k}(S(\theta), \mathcal{S}_\mathcal{E}, \hat{\alpha}) = \epsilon_k$, for $1 \leq k \leq p$, (A.1e) is a guaranteed violation bound on the model set output.

Note that for a fixed set of $\Theta_\mathcal{E}$ and, therefore, $\mathcal{S}_\mathcal{E}$, according to the structure of T, and consequently N, it is easily implied that $\lambda_{p+k}(.) \leq \lambda_{p+k+1}(.)$ for $1 \leq k < p$ (since $\left\| h_1 \quad h_2 \right\|_{s_1} \geq \left\| h_1 \quad 0 \right\|_{s_1}$). Thus, according to the ϵ_k formula presented in (5.10) it is proved that $\epsilon_1 \leq \cdots \leq \epsilon_P \leq \epsilon$.

(A.1e) implies that the following statement gives a guaranteed bound on the actual output:

$$\begin{aligned} y_k^{\theta_{act}} &\leq +\epsilon_k^y + \max_{i=1,\cdots,E} y_k^{\theta(i)}, \\ y_k^{\theta_{act}} &\geq -\epsilon_k^y + \min_{i=1,\cdots,E} y_k^{\theta(i)}. \end{aligned} \tag{A.2}$$

where $y^{\theta(i)}$ is the output corresponded to the i-th model of the computed model set $\mathcal{S}_\mathcal{E}$.

REFERENCES CITED

[1] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789 – 814, 2000. ISSN 0005-1098.

[2] Lars Grüne and Jürgen Pannek. *Nonlinear model predictive control*. Springer, 2011.

[3] Lars Grüne. Nmpc without terminal constraints. In *Proceedings of the IFAC Conference on Nonlinear Model Predictive Control*, volume 2012, pages 1–13, 2012.

[4] Lars Grüne. Economic receding horizon control without terminal constraints. *Automatica*, 49(3):725–734, 2013.

[5] Peter J. Campo and M. Morari. Robust model predictive control. In *American Control Conference, 1987*, pages 1021–1026, June 1987.

[6] J.C. Allwright and G.C. Papavasiliou. On linear programming and robust model-predictive control using impulse-responses. *Systems & Control Letters*, 18(2):159 – 164, 1992. ISSN 0167-6911.

[7] Mayuresh V. Kothare, Venkataramanan Balakrishnan, and Manfred Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32 (10):1361 – 1379, 1996. ISSN 0005-1098.

[8] A. Bemporad, F. Borrelli, and M. Morari. Min-max control of constrained uncertain discrete-time linear systems. *IEEE Transactions on Automatic Control*, 48(9):1600–1606, Sept 2003. ISSN 0018-9286.

[9] M. Cannon and B. Kouvaritakis. Optimizing prediction dynamics for robust mpc. *IEEE Transactions on Automatic Control*, 50(11):1892–1897, Nov 2005. ISSN 0018-9286.

[10] B. Pluymers, J. A. Rossiter, J. Suykens, and B. De Moor. A simple algorithm for robust MPC. In *Proceedings of the IFAC World Congress*, Prague, Czech Republic, March 2005.

[11] D. Munoz-Carpintero, M. Cannon, and B. Kouvaritakis. Recursively feasible robust mpc for linear systems with additive and multiplicative uncertainty using optimized polytopic dynamics. In *52nd Annual Conference on Decision and Control (CDC), 2013 IEEE*, pages 1101–1106, Dec 2013.

[12] G.C. Calafiore and M.C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, May 2006. ISSN 0018-9286.

[13] Marco C. Campi, Simone Garatti, and Maria Prandini. The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2):149 – 157, 2009. ISSN 1367-5788.

[14] T. Campbell and J.P. How. Bayesian nonparametric set construction for robust optimization. In *American Control Conference (ACC), 2015*, pages 4216–4221, July 2015.

[15] Zhong-Ping Jiang and Yuan Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37(6):857 – 869, 2001. ISSN 0005-1098.

[16] Eduardo D. Sontag and Yuan Wang. On characterizations of the input-to-state stability property. *Systems & Control Letters*, 24(5):351 – 359, 1995. ISSN 0167-6911.

[17] Mircea Lazar, WP Maurice H Heemels, and Andy R Teel. Lyapunov functions, stability and input-to-state stability subtleties for discrete-time discontinuous systems. *IEEE Transactions on Automatic Control*, 54(10):2421–2425, 2009.

[18] Davide Martino Raimondo, Daniel Limon, Mircea Lazar, Lalo Magni, and Eduardo Fernández Camacho. Min-max model predictive control of nonlinear systems: A unifying overview on stability. *European Journal of Control*, 15(1):5–21, 2009.

[19] Mircea Lazar, WP Maurice H Heemels, and Andrew R Teel. Further input-to-state stability subtleties for discrete-time systems. 2010.

[20] Florian Bayer, Mathias Burger, and Frank Allgower. Discrete-time incremental ISS: A framework for robust NMPC. In *2013 European Control Conference (ECC)*, pages 2068–2073. IEEE, 2013.

[21] David Q Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

[22] B. Kouvaritakis, M. Cannon, and V. Tsachouridis. Recent developments in stochastic {MPC} and sustainable development. *Annual Reviews in Control*, 28(1):23 – 35, 2004. ISSN 1367-5788.

[23] Basil Kouvaritakis and Mark Cannon. Developments in robust and stochastic predictive control in the presence of uncertainty. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 1(2):021003, 2015.

[24] R.L. Graham. An efficient algorith for determining the convex hull of a finite planar set. *Information Processing Letters*, 1(4):132 – 133, 1972. ISSN 0020-0190.

[25] Donald R Chand and Sham S Kapur. An algorithm for convex polytopes. *Journal of the ACM (JACM)*, 17(1):78–86, 1970.

[26] R.A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*, 2(1):18 – 21, 1973. ISSN 0020-0190.

[27] T. Ebert, J. Belz, and O. Nelles. Interpolation and extrapolation: Comparison of definitions and survey of algorithms for convex and concave hulls. In *2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 310–314, Dec 2014.

[28] Kenneth L. Clarkson and Peter W. Shor. Applications of random sampling in computational geometry, ii. *Discrete & Computational Geometry*, 4(1):387–421, 1989. ISSN 0179-5376.

[29] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22(4):469–483, December 1996. ISSN 0098-3500.

[30] Rex Allen Dwyer. *Average-case Analysis of Algorithms for Convex Hulls and Voronoi Diagrams*. PhD thesis, Pittsburgh, PA, USA, 1988. Order No. GAX88-17713.

[31] David Donoho and Jared Tanner. Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1906):4273–4293, 2009. ISSN 1364-503X.

[32] M.K. Ponamgi, D. Manocha, and M.C. Lin. Incremental algorithms for collision detection between polygonal models. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):51–64, Jan 1997. ISSN 1077-2626.

[33] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, September 1991. ISSN 0360-0300.

[34] Hossein Sartipizadeh and Tyrone L. Vincent. A method to calculate the approximate convex hull in high dimensions. *submitted to Computational Geometry*, 2016.

[35] Jon Louis Bentley, Franco P. Preparata, and Mark G. Faust. Approximation algorithms for convex hulls. *Commun. ACM*, 25(1):64–68, January 1982. ISSN 0001-0782.

[36] Zong-Ben Xu, Jiang-She Zhang, and Yiu-Wing Leung. An approximate algorithm for computing multidimensional convex hulls. *Applied Mathematics and Computation*, 94 (2):193 – 226, 1998. ISSN 0096-3003.

[37] Di Wang, Hong Qiao, Bo Zhang, and Min Wang. Online support vector machine based on convex hull vertices selection. *IEEE Transactions on Neural Networks and Learning Systems*, 24(4):593–609, April 2013. ISSN 2162-237X.

[38] H.R. Khosravani, A.E. Ruano, and P.M. Ferreira. A simple algorithm for convex hull determination in high dimensions. In *2013 IEEE 8th International Symposium on Intelligent Signal Processing (WISP)*, pages 109–114, Sept 2013.

[39] Ehsan Elhamifar, Guillermo Sapiro, and Rene Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1600–1607. IEEE, 2012.

[40] Stephen Boyd. *Convex Optimization*. Cambridge University Press, 2004.

[41] Xinzhong Cai, Guoqiang Wang, and Zihou Zhang. Complexity analysis and numerical implementation of primal-dual interior-point methods for convex quadratic optimization based on a finite barrier. *Numerical Algorithms*, 62(2):289–306, 2013. ISSN 1017-1398.

[42] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[43] Hossein Sartipizadeh and Tyrone L. Vincent. Computationally tractable robust moving horizon estimation using an approximate convex hull. In *55th IEEE Conference on Decision and Control (CDC)*, 2016.

[44] Hossein Sartipizadeh, Tyrone L. Vincent, and Robert J. Kee. Economic control of methane reforming. In *American Control Conference*, 2015.

[45] Jinfeng Liu. Moving horizon state estimation for nonlinear systems with bounded uncertainties. *Chemical Engineering Science*, 93:376–386, 2013.

[46] C. Rao, J. Rawlings, and D. Mayne. Constrained state estimation for nonlinear discrete-time systems: stability and moving horizon approximations. *IEEE Transactions on Automatic Control*, 48(2):246–258, Feb 2003. ISSN 0018-9286.

[47] Christopher V. Rao and James B. Rawlings. *Nonlinear Model Predictive Control*, chapter Nonlinear Moving Horizon State Estimation, pages 45–69. Birkhäuser Basel, Basel, 2000. ISBN 978-3-0348-8407-5.

[48] Hannah Michalska and David Q Mayne. Moving horizon observers and observer-based control. *IEEE Transactions on Automatic Control*, 40(6):995–1006, 1995.

[49] PE Moraal and JW Grizzle. Observer design for nonlinear systems with discrete-time measurements. *IEEE Transactions on Automatic Control*, 40(3):395–404, 1995.

[50] Sihem Tebbani, Laurent Le Brusquet, Emil Petre, and Dan Selisteanu. Robust moving horizon state estimation: Application to bioprocesses. In *17th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 539–544, 2013.

[51] Matthew Ellis, Jing Zhang, Jinfeng Liu, and Panagiotis D Christofides. Robust moving horizon estimation based output feedback economic model predictive control. *Systems & Control Letters*, 68:101–109, 2014.

[52] Zhao Haiyan, Chen Hong, and Ma Yan. Robust moving horizon estimation for system with uncertain measurement output. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 673–677, 2009.

[53] Giuseppe C Calafiore and Lorenzo Fagiano. Robust model predictive control via scenario optimization. *IEEE Transactions on Automatic Control*, 58(1):219–224, 2013.

[54] Hossein Sartipizadeh and Tyrone L. Vincent. Uncertainty characterization for robust MPC using an approximate convex hull method. In *American Control Conference*, 2016.

[55] Hossein Sartipizadeh and Tyrone L. Vincent. A new robust MPC approach using an approximate convex hull method. *submitted to Automatica*, 2016.

[56] Zhi Qiang Alex Zheng. Robust control of systems subject to contraints. 1995.

[57] Zhi Q. Zheng and M. Morari. Robust stability of constrained model predictive control. In *American Control Conference, 1993*, pages 379–383, June 1993.

[58] Alberto Bemporad and Manfred Morari. Robust model predictive control: A survey. In A. Garulli and A. Tesi, editors, *Robustness in identification and control*, volume 245 of *Lecture Notes in Control and Information Sciences*, pages 207–226. Springer London, 1999. ISBN 978-1-85233-179-5.

[59] D. Peaucelle, D. Arzelier, O. Bachelier, and J. Bernussou. A new robust d-stability condition for real convex polytopic uncertainty. *Systems & Control Letters*, 40(1):21 – 30, 2000. ISSN 0167-6911.

[60] Daniel Limón, T Alamo, Francisco Salas, and Eduardo F Camacho. Input to state stability of min–max mpc controllers for nonlinear systems with bounded uncertainties. *Automatica*, 42(5):797–803, 2006.

[61] Florian Bayer, Mathias Bürger, and Frank Allgöwer. Discrete-time incremental ISS: A framework for robust NMPC. In *Proc. European Control Conf.(ECC)*, pages 2068–2073, 2013.

[62] Davide Martino Raimondo, Daniel Limon, Mircea Lazar, Lalo Magni, and Eduardo Fernández Camacho. Min-max model predictive control of nonlinear systems: A unifying overview on stability. *European Journal of Control*, 15(1):5–21, 2009.

[63] Christopher D Lute, Marcelo Godoy Simões, Danilo Iglesias Brandão, Ahmed Al Durra, and SM Muyeen. Development of a four phase floating interleaved boost converter for photovoltaic systems. In *2014 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 1895–1902. IEEE, 2014.

[64] Christopher D Lute, Marcelo Godoy Simões, Danilo Iglesias Brandão, Ahmed Al Durra, and SM Muyeen. Experimental evaluation of an interleaved boost topology optimized for peak power tracking control. In *IECON 2014-40th Annual Conference of the IEEE Industrial Electronics Society*, pages 2096–2102. IEEE, 2014.

[65] Mohammad Kabalo, Damien Paire, Benjamin Blunier, David Bouquain, Marcelo Godoy Simões, and Abdellatif Miraoui. Experimental evaluation of four-phase floating interleaved boost converter design and control for fuel cell applications. *IET power electronics*, 6(2):215–226, 2013.

[66] Mohammad Kabalo, Benjamin Blunier, David Bouquain, Marcelo Godoy Simões, and Abdellatif Miraoui. Advanced hybrid dual loop control for multi-phases interleaved floating DC-DC converter for fuel cell applications. In *Industry Applications Society Annual Meeting (IAS), 2012 IEEE*, pages 1–8. IEEE, 2012.

[67] S. Choi, V.G. Agelidis, J. Yang, D. Coutellier, and P. Marabeas. Analysis, design and experimental results of a floating-output interleaved-input boost-derived DC-DC high-gain transformer-less converter. *IET power electronics*, 4(1):168–180, 2011.

[68] Farnaz Harirchi, M Godoy Simoes, Ahmed Al Durra, and SM Muyeen. Short transient recovery of low voltage-grid-tied DC distributed generation. In *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 1149–1155. IEEE, 2015.

[69] Moein Choobineh, Paulo C Tabares-Velasco, and Salman Mohagheghi. Optimal energy management of a distribution network during the course of a heat wave. *Electric Power Systems Research*, 130:230–240, 2016.

[70] H El Fadil, F Giri, JM Guerrero, and B Salhi. Adaptive control of interleaved boost converter for fuel cell energy. In *Proceedings of the 2011 American Control Conference*, pages 3905–3910. IEEE, 2011.

[71] Angel Cid-Pastor, Roberto Giral, Javier Calvente, VI Utkin, and Luis Martinez-Salamero. Interleaved converters based on sliding-mode control in a ring configuration. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 58(10):2566–2577, 2011.

[72] Matías Aguirre, Samir Kouro, Jose Rodriguez, and Haitham Abu-Rub. Model predictive control of interleaved boost converters for synchronous generator wind energy conversion systems. In *2015 IEEE International Conference on Industrial Technology (ICIT)*, pages 2295–2301. IEEE, 2015.

[73] D.L. Hoang and S.H. Chan. Modeling of a catalytic autothermal methane reformer for fuel cell applications. *Applied Catalysis A: General*, 268(1-2):207 – 216, 2004. ISSN 0926-860X.

[74] K Sasaki and Y Teraoka. Equilibria in fuel cell gases ii. the cho ternary diagrams. *Journal of The Electrochemical Society*, 150(7):A885–A888, 2003.

[75] Huayang Zhu, Robert J. Kee, Jeffrey R. Engel, and David T. Wickham. Catalytic partial oxidation of methane using rhsr- and ni-substituted hexaaluminates. *Proceedings of the Combustion Institute*, 31(2):1965 – 1972, 2007. ISSN 1540-7489.

[76] AA Trendewicz and RJ Braun. Techno-economic analysis of solid oxide fuel cell-based combined heat and power systems for biogas utilization at wastewater treatment facilities. *Journal of Power Sources*, 233:380–393, 2013.

[77] Andrea Mazzucco and Masoud Rokni. Thermo-economic analysis of a solid oxide fuel cell and steam injected gas turbine plant integrated with woodchips gasification. *Energy*, pages 1–16, 2014. ISSN 0360-5442.

[78] R. Gopinath, Sangsun Kim, Jae-Hong Hahn, P.N. Enjeti, M.B. Yeary, and J. W. Howze. Development of a low cost fuel cell inverter system with dsp control. *IEEE Transactions on Power Electronics*, 19(5):1256–1262, Sept 2004. ISSN 0885-8993.

[79] Seyyed Mohsen Mousavi Ehteshami and Siew Hwa Chan. Techno-economic study of hydrogen production via steam reforming of methanol, ethanol, and diesel. *Energy Technology & Policy*, 1(1):15–22, 2014.

[80] J.T. Pukrushpan, AG. Stefanopoulou, S. Varigonda, L.M. Pedersen, S. Ghosh, and Huei Peng. Control of natural gas catalytic partial oxidation for hydrogen generation in fuel cell applications. *IEEE Transactions on Control Systems Technology*, 13(1):3–14, Jan 2005. ISSN 1063-6536.

[81] Jay Pukrushpan, Anna Stefanopoulou, Subbarao Varigonda, Jonas Eborn, and Christoph Haugstetter. Control-oriented model of fuel processor for hydrogen generation in fuel cell applications. *Control Engineering Practice*, 14(3):277 – 293, 2006. ISSN 0967-0661.

[82] So-Ryeok Oh, Jing Sun, H. Dobbs, and J. King. Model predictive control for power and thermal management of an integrated solid oxide fuel cell and turbocharger system. *IEEE Transactions on Control Systems Technology*, 22(3):911–920, May 2014. ISSN 1063-6536.

[83] T. Das and S. Snyder. Adaptive control of a solid oxide fuel cell ultra-capacitor hybrid system. *IEEE Transactions on Control Systems Technology*, 21(2):372–383, March 2013. ISSN 1063-6536.

[84] Vanesa M. GarcA, Maria Serra, Jordi Llorca, and Jordi Riera. Design of linear controllers applied to an ethanol steam reformer for {PEM} fuel cell applications. *International Journal of Hydrogen Energy*, 38(18):7640 – 7646, 2013. ISSN 0360-3199.

[85] D. Recio-Garrido, C. Ocampo-Martinez, and M. Serra-Prat. Design of optimization-based controllers applied to an ethanol steam reformer for hydrogen production. *International Journal of Hydrogen Energy*, 37(15):11141 – 11156, 2012. ISSN 0360-3199.

[86] S. Bauer. Fuel reformer control system and method, March 18 2004. US Patent App. 10/245,268.

[87] Matthew J. Kupilik and Tyrone L. Vincent. Control of a solid oxide fuel cell system with sensitivity to carbon formation. *Journal of Power Sources*, 222(0):267 – 276, 2013. ISSN 0378-7753.

[88] Ali Al-Matouq and Tyrone Vincent. On-line estimation of inlet and outlet composition in catalytic partial oxidation. *International Journal of Hydrogen Energy*, 39(13):6953 – 6966, 2014. ISSN 0360-3199.

[89] Katrien De Cock and Bart De Moor. Subspace identification methods. *contribution to section 5.5, control systems robotics and automation of EOLSS, UNESCO Encyclopedia of Life Support Systems*, 1 of 3:933 – 979, 2003.

[90] Jorge E.P. Navalho, Jose M.C. Pereira, Ana R. Ervilha, and Jose C.F. Pereira. Uncertainty quantification in the catalytic partial oxidation of methane. *Combustion Theory and Modelling*, 17(6):1067–1095, 2013.