Imperial College London

Approximate robust model predictive control design using off-line computational methods

Yishen Xiao
CID:01886384

Supervisor: Dr Imad M. Jaimoukha

A Thesis submitted in fulfilment of requirements for the degree of
Master of Science
Control Systems
of Imperial College London

Department of Electrical and Electronic Engineering
Imperial College London
September 15, 2021

Abstract

Model predictive control(MPC) is an advanced control scheme that has now been widely used, but its performance will be ruined by the long computation time online if it is designed to handle systems with uncertainty.

This project is aimed to develop 3 computational efficient methods to relieve online computational burden for discrete linear time-invariant(DLTI) systems with bounded disturbance imposed by linear constraints on state and input.

The project first introduces the scheme of nominal MPC(NMPC) without uncertainty and relating mathematical knowledge background. Then online robust MPC scheme based on linear matrix inequality(LMI) with causal feedback control law is investigated and formulated. Later on, two computational efficient methods are introduced, proposed, and illustrated in details. One is called single LMI method and the other is a mehod of using offline lookup table. Also, their performances on reducing online computation time are analyzed and discussed with simulation results on a numerical case.

In the rest parts of the project, a novel MPC control scheme based on robust control invariant(RCI) set called 'Tube-based MPC' is introduced and simulated. Last but not least, a comprehensive comparison of three methods mentioned before is carried out in terms of online computation time and optimality.

Index Terms: Robust Model Predictive Control, Causal State-feedback Control, LMI-based MPC, Robust Control Invariant Set, Tube-based MPC

Acknowledgment

Here, I would like to thank some people who have played essential roles in the period of my project within these three months.

Firstly, I would like to thank Dr. Imad M. Jaimoukha, who is my supervisor. He respected my interest and passion for the choice of the method to be applied in my project. Besides, his patience and encouragement have helped me a lot, and he always gave me valuable suggestions on my project and guided me to the right path when my project was in trouble.

Secondly, I would like to thank Anastasis Georgiou, a Ph.D. student of Dr. Jaimoukha, who helped me introduce his projects and methods in detail.

Thirdly, I would like to thank my friends, Weijie Wang, Xiaofeng Mao, Fei Teng, and Kai Tang, classmates of mine at Imperial College London. We always studied at the central library and had meals together during this challenging year with the epidemic. Thanks for their companion and support in both daily life and spirits. Additionally, I also would like to express my thankfulness to Jing Li, and we had a lot of pleasure discussions on our projects, and her optimistic spirit encouraged me a lot.

At last, I would like to thank my parents, who always give me material and spiritual support unconditionally. It is their selfless care that makes it possible for me to complete my MSc project successfully.

Contents

List of Figures

List of Tables

Abbreviations

MPC: Model Predictive Control

NMPC: Nominal Model Predictive Control

RMPC: Robust Model Predictive Control

CRMPC: RMPC with Causal State-feedback Control

DLTI: Discrete Linear Time-invariant

LMI: Linear Matrix Inequality

SDP: Semi-definite Programming

SDPR: Semi-definite Programming Relaxation

CVX: Convex Programming Toolbox

QP: Quadratic Programming

RCI: Robust Control Invariant

LC-RCI: Low Complecity Robust Control Invariant

Chapter 1

Introduction

1.1 Model predictive control

Model predictive control(MPC) is a robust control algorithm widely applied in vehicles, energy, aeronautics, medical industries [1] [2], based on its ability to handle constraints and update control law according to read time conditions. However, MPC has its defects; one is that MPC has to go through vast amounts of computational efforts to solve the optimization problem in each prediction, which requires the high computational performance of the hardware. Otherwise, the long computational time might ruin the performance of control. This project is aimed to investigate such a problem.

1.2 LMI-based Robust model predictive control

LMI-based RMPC scheme is designed to deal with a system under the influence of disturbance and uncertainty, which is closer to the actual system. This algorithm aims to take advantage of using semidefinite programming to compute an optimal control sequence by solving an LMI-based optimization problem online [3]. The advantage of this scheme is that it can show explicit incorporation of uncertain items within the optimization and the polynomial-time that the optimization problem requires for its solution [4], which can give people a better understanding of the structure of the optimization problem.

1.3 Tube-based Model Predictive Control

An alternative Tube-based RMPC algorithm for handling the uncertainty of the system is proposed based on the RMPC algorithm [5–7]. The earlier Tube-based algorithm is very conservative, like the Min-max algorithm. In order to reduce the conservative nature of considering the uncertainty, a novel tube-based RMPC algorithm is developed by using a separation control strategy, which means the nominal system (uncertainty-free system) is separated from the whole system. The control of the existing system is transformed into the control of the nominal system, which controls the system state in a subset of state constraints [8, 9]. This subset of state constraints is called(looks like a) tube. The tube's existence ensures that the existing system satisfies the constraints, which makes it possible to design more straightforward and faster algorithms.

1.4 Outline of the Thesis

The thesis structure is shown in Fig 1.1, where we can observe that the first two chapters mainly introduce the ideas of the topics briefly and give a description of some valuable and important mathematical tools that will be used frequently later.

Chapter 3 introduces the mathematical formulation of the nominal MPC scheme, which is the basis of the following contents.

Chapter 4 and chapter 5 mainly investigate the LMI-based CRMPC scheme, which is a traditional way of solving RMPC problems online. One online and one offline computational efficient method are proposed to reduce the online computation time, and the performance is discussed and analyzed based on simulation results on a simple numerical case.

Chapter 6 and Chapter 7 mainly introduce the Tube-based MPC scheme. The idea of RCI set is described, and a minimal low complexity RCI set is designed and calculated as a basis of Tube-based MPC.

In the last chapter, comprehensive comparisons of the performance of the three methods proposed and introduced are conducted on the same numerical case in terms of online

computation time reduction and optimality.

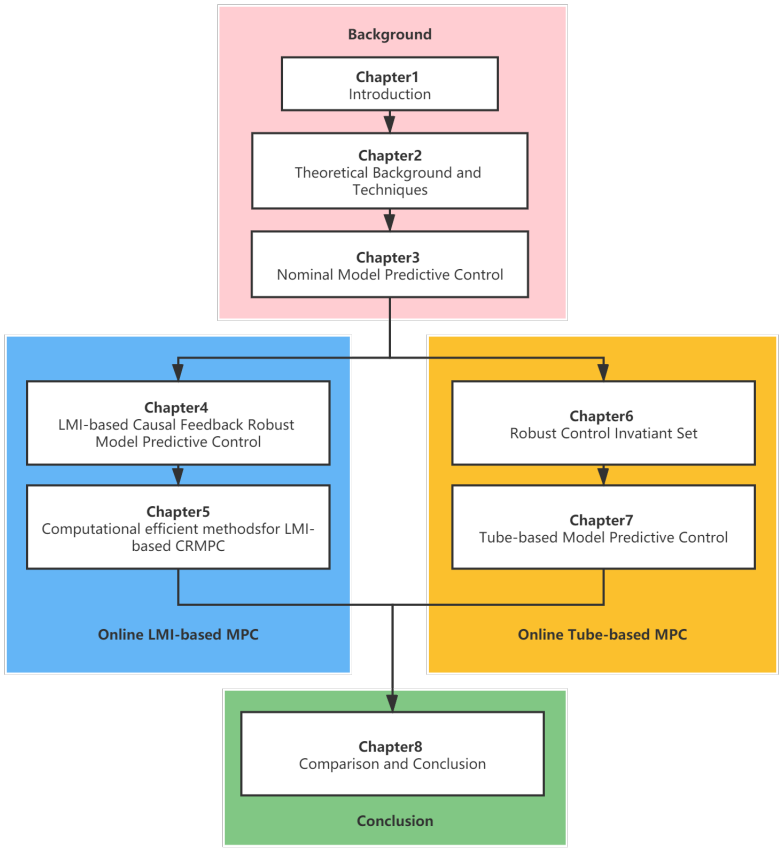


Figure 1.1: outline of the thesis

1.5 Contributions of the Project

The contribution of this project is to introduce, propose and illustrate one online method and two offline methods to reduce the online computation time of RMPC in detail about their ideas and algebraic formulations. Also, the performance of those methods is compared and analyzed based on simulation results on a simple numerical case.

Chapter 2

Theoretical Background and Techniques

THIS chapter will briefly introduce those critical mathematical techniques that will be frequently used in the algebraic formulation of the following problems.

2.1 Quadratic Programming(QP)

Quadratic Programming(QP) is a classic optimization programming used to minimize the quadratic objective function(cost function) with linear constraints. The standard form of QP is described as following. (2.1):

$$\begin{aligned} &\min_{x} \frac{1}{2} x^T H x + f^T x \\ &\text{s.t.} \quad \begin{aligned} A_{eq} x &= b_{eq} \\ A_{in} x &\leq b_{in} \end{aligned} \end{aligned} \tag{2.1}$$

Where H is usually a Hessian matrix, and f^T is composed of the Jacobi matrix of the gradient. And the variable x is subject to a set of both equality and inequality constraints. QP is a helpful tool in linear controller designing in that it is easy to build and compute and is efficient in its performance.

2.2 Min-Max Problem

The min-max problem is originated from game theory. It is described as finding the minimum cost of the objective function under the condition of the maximum value of several input variables, that is to say, finding the optimal solution in the worst case.
It is sometimes applied to minimize the possible loss for the worst case (maximum loss scenario). In control theory, it is often used to handle the system with norm-bounded uncertainty and disturbance. To manage such a nonlinear and non-convex scheme, we usually use semi-definite program relaxation (SDPR) to introduce a slack variable as an upper(lower) of the original cost function to transfer it into a convex problem then apply an online solver to calculate the optimal solutions. However, this strategy is very conservative because it has to consider the worst-case when obtaining the optimization solutions to ensure feasibility.

2.3 Linear Matrix Inequality

LMI is short for linear matrix inequality, which represents linear inequalities with a set of matrix variables. The standard form of LMI is shown as follows.

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i \geq 0.$$

where $x \in \mathbb{R}^m$ is a variable and F_0 and F_i are symmetric matrices which are given as coefficients.

2.4 Semi-definite Programming Relaxation

Semi-infinite programming is widely used in industry and research because many practical problems can be transformed or approximated as a combination of SDP problems. SDP is a subject of convex optimization, and it aims at solving problems with linear cost functions. Therefore, it is appropriate to solve RMPC problems with LMIs constraints. The meaning of "relaxation" is an approximation, usually shown as a slack variable for the upper/lower

bound of the given cost function. SDPR can be formulated in the following scheme:

$$\min_x c^T x$$
$$\text{s.t. } F(x) = F_0 + \sum_i^n x_i F_i \geq 0,$$

where $x = [x_1, x_2, \cdot, x_n]^T \in \mathbb{R}^n, F_i = F_i' \in \mathbb{R}^{m \times m}$.

2.5 Schur Complement

Schur complement is a handy tool in this project. It is usually applied in transforming nonlinear terms into a higher-order component. It is formed as follows:
Given that

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix}.$$

Then,

$$X > 0 \Leftrightarrow (X_{22} > 0 \text{ and } \overbrace{X_{11} - X_{12}^T X_{22}^{-1} X_{12}}^{s} > 0)$$

Where S is called the Schur complement of x [10]. It can be proved by:

$$\begin{bmatrix} I & -X_{12}X_{22}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \begin{bmatrix} I & 0 \\ -X_{22}X_{12}^T - 1 & I \end{bmatrix} = \begin{bmatrix} S & 0 \\ 0 & X_{22} \end{bmatrix}.$$

Therefore, the nonlinear part $X_{12}^T X_{22}^{-1} X_{12}$ can be represented as the linear matrix inequality X by using Schur complement.
An extended lemma of Schur complement is proposed and proved in [11], which is introduced as follows: If there exists $X = X^T$:

$$\mathcal{L} := \begin{bmatrix} R & AB \\ \star & Z \end{bmatrix} \succ 0 \Leftrightarrow \mathcal{L}_1 := \begin{bmatrix} R & A \\ \star & X^{-1} \end{bmatrix} \geq 0, \mathcal{L}_2 := \begin{bmatrix} X & B \\ \star & Z \end{bmatrix} \geq 0.$$

In most of the cases, it is frequent to solve the nonlinearity caused by a couple of two variables in the form of $XY + Y^T X^T$, the extended lemma called 'Bilinear Matrix Inequality'

(BMI) is used to take two variables apart and linearize the LMI, which says that matrix inequality $L := Z + XY + Y^T X^T \succeq 0$ is equivalent to matrix inequality

$$\begin{bmatrix} Z + XX^T + Y^T Y & X - Y^T \\ \star & I \end{bmatrix} \geq 0$$

An extended BMI proposed in [11] is given as:

$$L := Z + XY + Y^T X^T \geq 0 \tag{2.2}$$

is satisfied if and only if there exist matrix variables, of appropriate dimensions, $Q = Q^T \geq 0, E = E^T \geq 0, G_1$, G_2, F, and H such that $M := \begin{bmatrix} E & Y \\ \star & Q \end{bmatrix} \geq 0$, and

$$\begin{bmatrix} Z + Q + XEX^T & F - XG_1 & H - XG_2 \\ \star & G_1 + G_1^T - E & F^T + G_2 - Y \\ \star & \star & H^T + H - Q \end{bmatrix} \geq 0. \tag{2.3}$$

It is proved as following:

It shows that

$$XY + Y^T X^T = Q + XPX^T - V^T MV$$

where $V^T := \begin{bmatrix} -X & I \end{bmatrix}$. Replace the above expression in (2.2), take a Schur complement and perform a congruence transformation with diag $\left(I, M_o^T\right)$ with $M_o := \begin{bmatrix} G_1 & G_2 \\ F & H \end{bmatrix}$, it has:

$$\begin{bmatrix} Z + Q + XEX^T & V^T M_o \\ \star & M_o^T M^{-1} M_o \end{bmatrix} \geq 0 \tag{2.4}$$

To deal with nonlinear term $M_o^T M^{-1} M_o$, the following slack-variable identity is used:

$$M_o^T M^{-1} M_o = M_o + M_o^T - M + (M_o - M)^T M^{-1} (M_o - M) \tag{2.5}$$

Replace the $(2,2)$ entry of (2.4) by the first three terms on the RHS of (2.5), then it gives (2.3).

2.6 S-Procedure

An S-procedure or S-theorem is a mathematical result, which gives the conditions for a particular quadratic inequality to be transformed into a sequence of another quadratic inequalities combination.

Let M_1 and M_2 be symmetric matrices, f_1 and f_2 be vectors and c_1 and c_2 be real numbers. Assume that there is some x_0 such that the strict inequality $x_0^T M_1 x_0 + 2f_1^T x_0 + c_1 < 0$ holds. Then the implication

$$x^T M_1 x + 2f_1^T x + c_1 \leq 0 \Longrightarrow x^T M_2 x + 2f_2^T x + c_2 \leq 0$$

holds if and only if there exists some non-negative number λ such that

$$\lambda \begin{bmatrix} M_1 & f_1 \\ f_1^T & c_1 \end{bmatrix} - \begin{bmatrix} M_2 & f_2 \\ f_2^T & c_2 \end{bmatrix} \geq 0.$$

This lemma is used in this paper to transform a sequence of linear matrix inequalities to one inequality, reducing the computational effort.

2.7 Matlab CVX Toolbox

CVX is a modeling system to construct and solve regular convex programs (DCPs). CVX supports a wide range of standard problem types, including linear and quadratic programs (LP/QP), semi-infinite plans (SDP), and second-order conic programs (SOCP). One of its advantages can be implemented in Matlab and then turn Matlab into an optimization modeling language. Model specifications are constructed using joint Matlab operations and functions. Additionally, the standard Matlab code can also be mixed with these specifications in flexible ways, making it simple to perform the calculations for optimization

problems or formulate the results from their solutions.

Chapter 3

Nominal Model Predictive Control

IN this chapter, the nominal MPC scheme is presented, which means the system is not influenced by any disturbance and uncertainty. The formulation of nominal MPC will be introduced by presenting the dynamical formulation of the system, cost function and the constraints imposed on the state and input of the system. Then the design of the controller for the nominal system will be shown later on and simulation result on a numerical example demonstrates the performance of the nominal MPC controller.

3.1 System Description

Given a discrete time system as:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned} \tag{3.1}$$

Where $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$ and $y_k \in \mathbb{R}^{n_y}$ are the state, input and output at kth step, and $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_u \times n_u}$, $C \in \mathbb{R}^{n_y \times n_x}$ and $D \in \mathbb{R}^{n_y \times n_u}$ are the state, input, output and feedforward matrix of the system. The prediction step k belongs to the time set $T_N = \{0, 1, \cdots, N-1\}$ where $N > 0$ is the prediction horizon, thus item labelled with N meaning the terminal one in each prediction iteration.

Quadratic cost function is often used in MPC for its simplicity and efficiency, which is

given in the following form:

$$J = x_N^T P x_N + \sum_{i=0}^{N-1} (x_i^T Q_i x_i + u_i^T R_i u_i)$$

Where $P_N \geq 0, Q_i \geq 0, R_i \geq 0$ are symmetric positive-definite penalty matrices for terminal states, states and input respectively. By adjusting the parameters in penalty matrices, we can put weight on the specific state or input to satisfy our requirements.
Meanwhile, the constraints of state and input are usually set as their own boundaries, which can be described as the following:

$$\begin{aligned} \underline{x} \leq x_k \leq \overline{x}, \quad \forall k \in \{0, 1, 2 \ldots, N\} \\ \underline{u} \leq u_i \leq \overline{u}, \quad \forall i \in \{0, 1, 2 \ldots, N-1\} \end{aligned}$$

Where $\underline{x}_k, \overline{x}_k, \underline{u}_k, \overline{u}_k$ denote the upper bound and lower bound for state and input at prediction step k, respectively. Hence, the optimal input control signal is obtained by solving the following optimal problem:

$$\begin{aligned} \min_{u_k} \quad & J = x_N^T P x_N + \sum_{i=0}^{N-1} (x_i^T Q_i x_i + u_i^T R_i u_i) \\ \text{s.t.} \quad & \underline{x}_k \leq x_k \leq \overline{x}_k, \quad \forall k \in \{0, 1, 2 \ldots, N\} \\ & \underline{u} \leq u_i \leq \overline{u}, \quad \forall i \in \{0, 1, 2 \ldots, N-1\} \end{aligned} \tag{3.2}$$

3.2 Algebraic Formulation

The algebraic formulation of the state according to each input can be expressed as the following:

$$
\begin{aligned}
x_1 &= Ax_0 + B_u u_0, \\
x_2 &= A^2 x_0 + AB_u u_0 + B_u u_1, \\
x_3 &= A^3 x_0 + A^2 B_u u_0 + AB_u u_1, + B_u u_2 \\
&\vdots \\
x_N &= A^N x_0 + A^{N-1} B_u u_0 + A^{N-2} B_u u_1 + \cdots + B_u u_{N-1}.
\end{aligned} \tag{3.3}
$$

The (3.3) can be written in stack form with states vector and input vector as

$$
x = \tilde{A} x_0 + \tilde{B}_u u, \tag{3.4}
$$

where the stack form of states and inputs are written as:

$$
x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^{Nn_x \times 1}, u = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix} \in \mathbb{R}^{Nn_u \times 1},
$$

and $\tilde{A}$ and $\tilde{B}_u$ are defined as:

$$
\tilde{A} = \begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^N \end{bmatrix} \in \mathbb{R}^{Nn_x \times n_x}, \tilde{B} = \begin{bmatrix} B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ A^2 B & AB & B & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{N-1} B & A^{N-2} B & A^{N-3} B & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{Nn_x \times Nn_u}.
$$

3.3 Cost Function

For simplicity, let $Q_k = Q$, $R_k = R$ and the cost function J can be rewritten as the matrix form:

$$\tilde{Q} = \left[\begin{array}{cccc|c} Q & 0 & \cdots & 0 & 0 \\ 0 & Q & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & Q & 0 \\ \hline 0 & 0 & \cdots & 0 & P \end{array}\right], \tilde{R} = \begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R \end{bmatrix},$$

Where $Q, P \in \mathbb{R}^{n_x \times n_x}$ and $R \in \mathbb{R}^{n_u \times n_u}$, then it has $\tilde{Q}, \tilde{P} \in \mathbb{R}^{Nn_x \times Nn_x}$ and $\tilde{R} \in \mathbb{R}^{Nn_u \times Nn_u}$. Moreover, the cost function for all state in each prediction iteration can be rewritten in a dense matrix from, substituting (3.4) into it gives:

$$J(x_0, u) = \frac{1}{2} u^T H u + G^T u + \tilde{c} \tag{3.5}$$

Where

$$H = \tilde{R} + \tilde{B}^T \tilde{Q} \tilde{B}$$

$$G = \tilde{B}^T \tilde{Q} \tilde{A} x_0$$

$$\hat{c} = x_0^T \left(Q_0 + \tilde{A}^T \tilde{Q} \tilde{A} \right) x_0$$

Then the cost function (3.5) is a linear quadratic programming of variable u and the global optimal solution u^* can be easily solved by using derivative. During which process the value of $\hat{c}$ will not affect the optimal solution u^* in that it is a constant and will be eliminated when using derivative.

3.4 Constraints

For the constraints, here only introduce the general linear stage constraints on state and input, which can be expressed as:

$$x_k \in \mathcal{X} = \{x \in \mathbb{R}^2 : \underline{x} \leq x \leq \overline{x}\}$$

and

$$u_k \in \mathcal{U} = \{u \in \mathbb{R} : \underline{u} \leq u \leq \overline{u}\}$$

For the convenience of coding, it can be written as one side matrix form as:

$$\begin{bmatrix} x \\ -x \end{bmatrix} \leq \begin{bmatrix} \overline{x} \\ -\underline{x} \end{bmatrix}$$

and

$$\begin{bmatrix} u \\ -u \end{bmatrix} \leq \begin{bmatrix} \overline{u} \\ -\underline{u} \end{bmatrix}$$

Where, $\overline{x}, \underline{x} \in \mathbb{R}^{Nn_x}$ and $\overline{u}, \underline{u} \in \mathbb{R}^{Nn_u}$ are the upper and lower bound of the state and input respectively. Substituting the equation (3.4) into the above linear matrix inequality, the constraints on state can be expressed in terms of control variable u, and combine the constraints on input, a linear inequality in terms of u is shown as:

$$\begin{bmatrix} I_{N\times n_u} \\ -I_{N\times n_u} \\ \tilde{B} \\ -\tilde{B} \end{bmatrix} \leq \begin{bmatrix} \overline{u} \\ -\underline{u} \\ \overline{x} - \tilde{A}x_0 \\ -\overline{x} + \tilde{A}x_0 \end{bmatrix} \tag{3.6}$$

Therfore, the nominal MPC problem can be formulated as a QP in terms of control variable u :

$$\begin{aligned} \min_{u} \quad & J(u) = u^T \hat{A} u + 2\hat{b}^T u \\ \text{subject to} \quad & (3.6) \end{aligned} \tag{3.7}$$

At each sampling instant, the QP(3.7) is solved to get an optimal control input to update the state, and then the whole process is repeated with the new updated state as x_0 in the next iteration.

3.5 Numerical Case Study

In this section, the performance of the designed nominal MPC will be shown by a numerical example in MATLAB R2021a with function `quadprog()`. The DLTI system matrix A and input matrix B is given by:

$$A = \begin{bmatrix} 1 & 0.8 \\ 0.5 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \tag{3.8}$$

Then $n_x = 2$, $n_u = 1$. Prediction horizon is set as $N = 5$.

The system is unstable with the eigenvalues of $\lambda_1 = 1.6325$ and $\lambda_2 = 0.3675$, but the pair (A, B) is controllable, then the case can be stabilized by applying suitable control signal u. The cost penalty matrices Q and P for state and R for input is given by:

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 1.$$

The constraints imposed on state and input of the nominal system are stage constraints, which are

$$x_k \in \mathcal{X} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -7 \\ -7 \end{bmatrix} \leq x \leq \begin{bmatrix} 7 \\ 7 \end{bmatrix} \right\}, \quad k = \{1, 2, \cdots, N\}.$$

and

$$u_i \in \mathcal{U} = \{u \in \mathbb{R} : -7 \leq u \leq 7\}, \quad i \in \{1, 2, \cdots, N-1\}.$$

Then the state trajectory of the controller is shown in Fig.3.1, where red line with star marks is the state of NMPC and the blue box shows feasible region for state. It is observed that with the designed controller, the state can be controlled to the origin without breaking state constraints. The code for the below can be found by the URL in Appendix, named as `Nominal_MPC_final.m`.

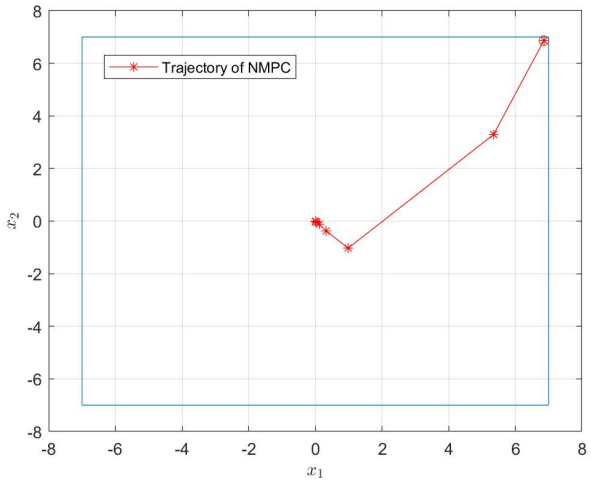


Figure 3.1: Trajectory of NMPC

The input signal of NMPC is shown in Fig.3.2, where red line with star marks is the input signal U of the designed controller and the cyan/light blue line and black line represent the upper and lower bounds of input signal, respectively. Is is observed that the input signal decreases with the process of NMPC and becomes state at zero, which means the system is stable.

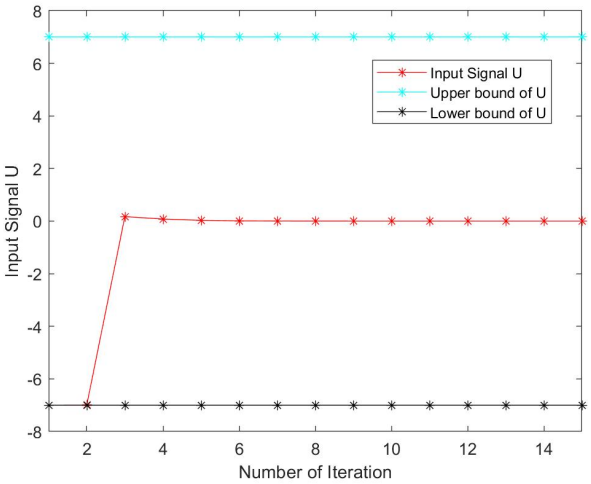


Figure 3.2: Input Signal of NMPC

Given that the system is affected by bounded disturbance as: $w_i \in \mathcal{W}_i := \{w \in \mathbb{R} : -1 \leq$

$w \leq 1\}, \quad i \in \{1, 2, \cdots, N-1\}$. Then the state trajectory and input signal with the same controller can be found in Fig.3.3 and Fig.3.4, it is apparent that under the influence of bounded disturbance, the states are very unstable around the origin and input signal U is oscillating around 0. To conclude, the performance for the controller designed for NMPC is terrible for system with bounded disturbance, and a specific scheme for such system is discussed in the next chapter.

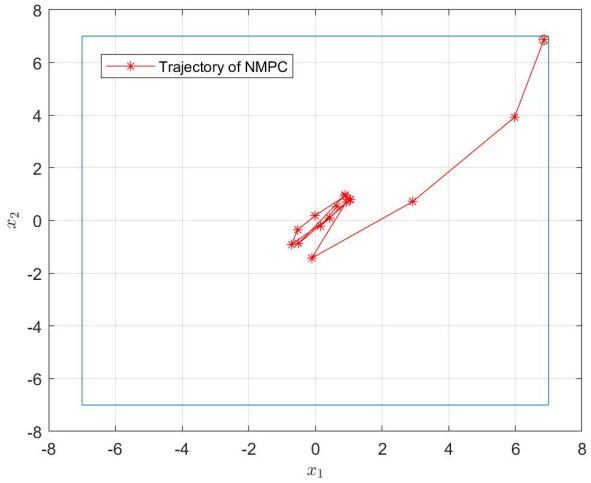


Figure 3.3: Trajectory of NMPC with Disturbance

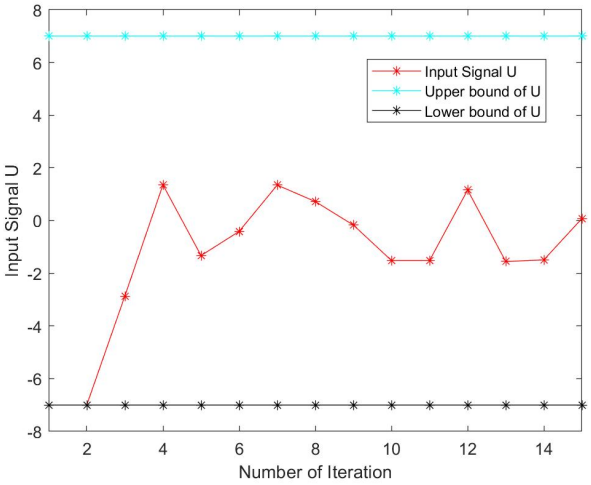


Figure 3.4: Input Signal of NMPC with Disturbance

Chapter 4

LMI-based Causal Feedback Robust Model Predictive Control

THE aim of this chapter is to investigate the robust model predictive control for dealing with norm-bounded disturbance included in the model. Besides, a causal feedback control strategy is applied in the system structure and the performance of the controller is simulated on a numerical case.

4.1 System Description

The DTLI model of RMPC scheme can be described as following:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + B_d w_k \\ f_k &= C_f x_k + D_f u_k + D_{fw} w_k \\ f_N &= \hat{C}_f x_N \\ z_k &= C_z x_k + D_z u_k + D_{zw} w_k \\ z_N &= \hat{C}_z x_N \end{aligned} \quad , \tag{4.1}$$

where $x_k \in \mathbb{R}^n$,$u_k \in \mathbb{R}^{n_u}$,$w_k \in \mathbb{R}^{n_w}$,$f_k \in \mathbb{R}^{n_f}$,$z_k \in \mathbb{R}^{n_z}$are the state, input, disturbance, constraints signal and costs signal vectors at prediction step k respectively. The prediction step k belongs to the time set $T_N = \{0, 1, \cdots, N-1\}$ where $N > 0$ is the prediction

horizon, thus item labelled with N meaning the terminal one in each prediction iteration. The disturbance is considered as the form

$$w_k \in \mathcal{W}_k := \{w \in \mathbb{R}^{n_w} : \underline{w}_k \leq w \leq \overline{w}_k\} \tag{4.2}$$

where $\underline{w}_k < \overline{w}_k$, $k \in T_N$ are given lower and upper boundaries of the disturbance at each prediction step k. Then $\underline{w} \in \mathbb{R}^{Nn_w}$ and $\overline{w} \in \mathbb{R}^{Nn_w}$ are the lower and upper stage boundary of disturbance w respectively.

For unity, (4.1) can be written in a matrix form as (4.3).

$$\begin{bmatrix} x_{k+1} \\ f_k \\ z_k \end{bmatrix} = \begin{matrix} n_x \\ n_f \\ n_z \end{matrix} \overset{\begin{matrix} n_x & n_u & n_w \end{matrix}}{\begin{bmatrix} A & B_u & B_p \\ C_f & D_{fu} & D_{fw} \\ C_z & D_{zu} & D_{zw} \end{bmatrix}} \begin{bmatrix} x_k \\ u_k \\ w_k \end{bmatrix}$$
$$\begin{bmatrix} f_N \\ z_N \end{bmatrix} = \begin{bmatrix} \hat{C}_f \\ \hat{C}_z \end{bmatrix} x_N \tag{4.3}$$

For the robust model predictive control(RMPC), it is required that for all $k \in T_N$, to find u_k such that the future constrained outputs satisfy $f_k \leq \overline{f}_k$ and $f_N \leq \overline{f}_N$ for all $w_k \in \mathcal{W}_k$, and the cost function

$$J = \max_{w_k \in \mathcal{W}_k} \sum_{k=0}^{N} (z_k - \bar{z}_k)^T (z_k - \bar{z}_k) \tag{4.4}$$

is minimized, where $\bar{z}_k$, $k \in T_N$ is a given reference trajectory. Note that f_k,$k \in T_N$ and f_N may be chosen to represent polytopic constraints on state, output and input.

4.2 Algebraic Formulation

Let ζ stand for x, f, $\bar{f}$, z or $\bar{z}$ and ξ stand for u or w, define stack vectors $\zeta = \left[\zeta_0^T, \zeta_1^T, \cdots, \zeta_N^T\right]^T \in \mathbb{R}^{(N+1)n_\zeta}$ and $\xi = \left[\xi_0^T, \xi_2^T, \cdots, \xi_{N-1}^T\right]^T \in \mathbb{R}^{Nn_\xi}$. Then the discrete dynamic system can be expressed in the form of (4.5):

$$\begin{bmatrix} x \\ f \\ z \end{bmatrix} = \begin{matrix} (N+1)n_x \\ (N+1)n_f \\ (N+1)n_z \end{matrix} \overset{\begin{matrix} Nn_x & Nn_u & Nn_w \end{matrix}}{\begin{bmatrix} \tilde{A} & \tilde{B}_u & \tilde{B}_w \\ \tilde{C}_f & \tilde{D}_{fu} & \tilde{D}_{fw} \\ \tilde{C}_z & \tilde{D_{zu}} & \tilde{D}_{zw} \end{bmatrix}} \begin{bmatrix} x_0 \\ u \\ w \end{bmatrix}, \tag{4.5}$$

where for states,

$$\tilde{A} = \left[\begin{array}{c} I_{n_x} \\ \hline A \\ A^2 \\ A^3 \\ \vdots \\ A^N \end{array}\right] \in \mathbb{R}^{(N+1)n_x \times n_x},$$

$$\tilde{B}_u = \left[\begin{array}{ccccc} 0 & 0 & 0 & \cdots & 0 \\ \hline B_u & 0 & 0 & \cdots & 0 \\ AB_u & B_u & 0 & \cdots & 0 \\ A^2B_u & AB_u & B_u & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B_u & A^{N-2}B_u & A^{N-3}B_u & \cdots & B_u \end{array}\right] \in \mathbb{R}^{(N+1)n_x \times Nn_u}$$

$$\tilde{B}_w = \left[\begin{array}{ccccc} 0 & 0 & 0 & \cdots & 0 \\ \hline B_w & 0 & 0 & \cdots & 0 \\ AB_w & B_w & 0 & \cdots & 0 \\ A^2B_w & AB_w & B_w & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B_w & A^{N-2}B_w & A^{N-3}B_w & \cdots & B_w \end{array}\right] \in \mathbb{R}^{(N+1)n_x \times Nn_w}$$

for constraints signal f and similarly for costs signal z,

$$\tilde{C}_f = \begin{bmatrix} C_f \\ C_f A \\ C_f A^2 \\ \vdots \\ C_f A^{N-1} \\ \hline \hat{C}_f A^N \end{bmatrix} \in \mathbb{R}^{(N+1)n_f \times n_x}, \tilde{C}_z = \begin{bmatrix} C_z \\ C_z A \\ C_z A^2 \\ \vdots \\ C_z A^{N-1} \\ \hline \hat{C}_z A^N \end{bmatrix} \in \mathbb{R}^{(N+1)n_z \times n_x},$$

$$\tilde{D}_{fu} = \begin{bmatrix} D_{fu} & 0 & 0 & \cdots & 0 \\ C_{fu}B_u & D_{fu} & 0 & \cdots & 0 \\ C_{fu}AB_u & C_{fu}B_u & D_{fu} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{fu}A^{N-2}B_u & C_{fu}A^{N-3}B_u & C_{fu}A^{N-4}B_u & \cdots & D_{fu} \\ \hline \hat{C}_{fu}A^{N-1}B_u & \hat{C}_{fu}A^{N-2}B_u & \hat{C}_{fu}A^{N-3}B_u & \cdots & \hat{C}_{fu}B_u \end{bmatrix} \in \mathbb{R}^{(N+1)n_f \times Nn_u},$$

$$\tilde{D}_{fw} = \begin{bmatrix} D_{fw} & 0 & 0 & \cdots & 0 \\ C_{fw}B_w & D_{fw} & 0 & \cdots & 0 \\ C_{fw}AB_u & C_{fw}B_u & D_{fw} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{fw}A^{N-2}B_w & C_{fw}A^{N-3}B_w & C_{fw}A^{N-4}B_w & \cdots & D_{fw} \\ \hline \hat{C}_{fw}A^{N-1}B_w & \hat{C}_{fw}A^{N-2}B_w & \hat{C}_{fw}A^{N-3}B_w & \cdots & \hat{C}_{fw}B_w \end{bmatrix} \in \mathbb{R}^{(N+1)n_f \times Nn_w},$$

$$\tilde{D}_{zu} = \left[\begin{array}{ccccc} D_{zu} & 0 & 0 & \cdots & 0 \\ C_{zu}B_u & D_{zu} & 0 & \cdots & 0 \\ C_{zu}AB_u & C_{zu}B_u & D_{zu} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{zu}A^{N-2}B_u & C_{zu}A^{N-3}B_u & C_{zu}A^{N-4}B_u & \cdots & D_{zu} \\ \hline \hat{C}_{zu}A^{N-1}B_u & \hat{C}_{zu}A^{N-2}B_u & \hat{C}_{zu}A^{N-3}B_u & \cdots & \hat{C}_{zu}B_u \end{array}\right] \in \mathbb{R}^{(N+1)n_z \times Nn_u},$$

$$\tilde{D}_{zw} = \left[\begin{array}{ccccc} D_{zw} & 0 & 0 & \cdots & 0 \\ C_{zw}B_w & D_{zw} & 0 & \cdots & 0 \\ C_{zw}AB_u & C_{zw}B_u & D_{zw} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{zw}A^{N-2}B_w & C_{zw}A^{N-3}B_w & C_{zw}A^{N-4}B_w & \cdots & D_{zw} \\ \hline \hat{C}_{zw}A^{N-1}B_w & \hat{C}_{zw}A^{N-2}B_w & \hat{C}_{zw}A^{N-3}B_w & \cdots & \hat{C}_{zw}B_w \end{array}\right] \in \mathbb{R}^{(N+1)n_z \times Nn_w},$$

4.3 Causal State-feedback Control Law

In this system, we consider a causal state-feedback control law as the controller scheme of RMPC problem, which means the input control signal is related to the previous states (based on the current prediction step k). With the feedback, the controller can take the advantage of current states to decrease the effect of uncertain item (here means disturbance) to some extent [12], therefore the constraints will be relaxed and the conservatism is decreased as a result [13]. We set

$$u = \tilde{K_0}x_0 + \tilde{K}x + \tilde{v} \tag{4.6}$$

where $\tilde{K_0} \in \mathbb{R}^{Nn_u \times n_x}$, $\tilde{K} \in \mathbb{R}^{Nn_u \times (N+1)n_x}$ are the current and future state-feedback gain matrices and $\tilde{v} \in \mathbb{R}^{Nn_u}$ is the control-perturbation sequence. The specific form can be written as the following:

$$
\begin{aligned}
u_0 &= K_0^1 x_0 \\
u_1 &= K_0^2 x_0 + K^{1,1} x_1 \\
u_2 &= K_0^3 x_0 + K^{2,1} x_1 + K^{2,2} x_2 \\
&\vdots \\
u_{N-1} &= K_{N-1}^3 x_0 + K^{N-1,1} x_1 + \cdots + K^{N-1,N-1} x_{N-1} + 0 \times x_N
\end{aligned}
$$

Therefore, the causality is apparent in that the matrix $\begin{bmatrix} \tilde{K_0} & \tilde{K} \end{bmatrix}$ is block lower triangular, which can be written as (4.7):

$$
\begin{bmatrix} \tilde{K_0} & \tilde{K} \end{bmatrix} = \left[\begin{array}{c|cccccc}
K_0^1 & 0 & 0 & 0 & 0 & 0 & 0 \\
K_0^2 & K^{1,1} & 0 & 0 & 0 & 0 & 0 \\
K_0^3 & K^{2,1} & K^{2,2} & 0 & 0 & 0 & 0 \\
K_0^4 & K^{3,1} & K^{3,2} & K^{3,3} & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
K_0^{N-1} & K^{N-1,1} & K^{N-1,2} & K^{N-1,3} & \cdots & K^{N-1,N-1} & 0
\end{array}\right]. \tag{4.7}
$$

By substituting (4.5) into (4.6), we can obtain the following expression for u as (4.8)

$$
u = \hat{K_0} x_0 + \hat{K}_w w + \hat{v}, \tag{4.8}
$$

where,

$$
\begin{bmatrix} \hat{K_0} & \hat{K}_w & \hat{v} \end{bmatrix} = \left(I_{Nn_u} - \tilde{K}\tilde{B}_u\right)^{-1} \left[\tilde{K_0} + \tilde{K}\tilde{A} \quad \tilde{K} \quad \tilde{v}\right]. \tag{4.9}
$$

Note that u is affine in the new variables $\left(\hat{K}_0, \hat{K}_w, \hat{v}\right)$ which have the same structure and dimension as $\left(\tilde{K}_0, \tilde{K}, \tilde{v}\right)$, then the triple can be recovered by (4.10):

$$
\begin{bmatrix} \tilde{K_0} & \tilde{K} & \tilde{v} \end{bmatrix} = \left(I_{Nn_u} + \hat{K}_w\tilde{B}_u\right)^{-1} \left[\hat{K_0} - \hat{K}_w\tilde{A} \quad \hat{K}_w \quad \hat{v}\right]. \tag{4.10}
$$

In the following step, the input sequence variable u is eliminated by substituting the triple $\left(\hat{K}_0, \hat{K}_w, \hat{v}\right)$ as the decision variables.

Substituting (4.6) into (4.5) gives,

$$\begin{aligned}\begin{bmatrix} f \\ z-\bar{z} \end{bmatrix} &= \left[\begin{array}{c|c} \tilde{D}_{fw}^{\hat{K}_w} & \tilde{D}_{f}^{\hat{K}_0,\hat{v}} \\ \hline \tilde{D}_{zw}^{\hat{K}_w} & \tilde{D}_{z}^{\hat{K}_0,\hat{v}} \end{array}\right] \begin{bmatrix} w \\ 1 \end{bmatrix}, \\ &:= \left[\begin{array}{c|c} \tilde{D}_{fw}+\tilde{D}_{fu}\hat{K}_w\tilde{B}_w & \tilde{D}_{fu}\hat{v}+\left(\tilde{C}_f+\tilde{D}_{fu}\hat{K}_0\right)x_0 \\ \hline \tilde{D}_{zw}+\tilde{D}_{zu}\hat{K}_w\tilde{B}_w & \tilde{D}_{zu}\hat{v}+\left(\tilde{C}_z+\tilde{D}_{zu}\hat{K}_0\right)x_0-\bar{z} \end{array}\right] \begin{bmatrix} w \\ 1 \end{bmatrix}.\end{aligned} \tag{4.11}$$

Until now, we can express the constraints and cost function by decision variables $\hat{K}_0$, $\hat{K}_w$ and $\hat{v}$ as

$$f = f\left(\hat{K}_0, \hat{K}_w, \hat{v}\right), \tag{4.12}$$

and

$$\sum_{k=1}^{n_z} \left(z_k - \bar{z}_k\right)^T \left(z_k - \bar{z}_k\right) = J\left(\hat{K}_0, \hat{K}_w, \hat{v}\right). \tag{4.13}$$

4.4 Cost Function

From (4.13), set the reference trajectory $\bar{z}$ to be zero, and set a upper bound γ^2 of the cost function as

$$\gamma^2 - z^T z \geq 0 \tag{4.14}$$

In the sequel, the problem can be transformed to minimize the upper bound γ^2 of cost function. And the equality form can be transformed into an inequality form as the standard SDPR form. Apply the Schur complement on (4.14), it is equivalent to

$$\begin{bmatrix} \gamma^2 & z^T \\ z & I_{(N+1)n_z} \end{bmatrix} \geq 0. \tag{4.15}$$

Expand the cost z in the form of (4.5), then (4.15) can be written in the form as

$$\begin{bmatrix} \gamma^2 & \star \\ \tilde{D}_{zw}^{\hat{K}_w} w + \tilde{D}_z^{\hat{K}_0,\hat{v}} & I_{(N+1)n_z} \end{bmatrix} \geq 0.$$

And it can be decomposed as the form of

$$T_1 + \mathcal{H}(T_2 w T_3) \geq 0. \tag{4.16}$$

Where $\mathcal{H}(A) = A + A^T$, and the above inequality can be expressed as:

$$\overbrace{\begin{bmatrix} \gamma^2 & \star \\ \tilde{D}_z^{\hat{K}_0,\hat{v}} & I_{(N+1)n_z} \end{bmatrix}}^{T_1} + \overbrace{\begin{bmatrix} \mathbf{0} \\ \tilde{D}_{zw}^{\hat{K}_w} \end{bmatrix}}^{T_2} w \overbrace{\begin{bmatrix} \mathbf{1} & \mathbf{0} \end{bmatrix}}^{T_3} + \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix} w^T \begin{bmatrix} \mathbf{0} & \tilde{D}_{zw}^{\hat{K}_w T} \end{bmatrix} \geq 0,$$

where

$$\begin{bmatrix} T_1 & T_2 \\ T_3 & \mathbf{0} \end{bmatrix} = \begin{array}{c} \\ 1 \\ (N+1)n_z \\ \hline 1 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & (N+1)n_z & Nn_w \end{array} \\ \left[\begin{array}{ccc} \gamma^2 & \star & \mathbf{0} \\ \tilde{D}_z^{\hat{K}_0} & \boldsymbol{I} & \tilde{D}_{zw}^{\hat{K}_w} \\ \hline \mathbf{1} & \mathbf{0} & \mathbf{0} \end{array}\right] \end{array}.$$

Note that, if there exists positive semi-definite symmetric matrix D, which is marked as $D = D^T \geq 0$, then it has

$$T_3^T (w - \underline{w})^T D(\overline{w} - w) T_3 \geq 0. \tag{4.17}$$

The equation (4.16) can be rewritten as

$$\underbrace{T_3^T (w - \underline{w})^T D(\overline{w} - w) T_3}_{(4.17)} - \underbrace{T_3^T (w - \underline{w})^T D(\overline{w} - w) T_3}_{(4.17)} + \underbrace{T_1 + T_2 w T_3 + T_2^T w^T T_3^T}_{(4.16)} \geq 0$$

$$\underbrace{T_3^T (w - \underline{w})^T D(\overline{w} - w) T_3}_{(4.17)} + \begin{bmatrix} 1 & T_3^T w^T \end{bmatrix} \underbrace{\begin{bmatrix} T_1 + T_3^T \underline{w}^T D \overline{w} T_3 & \star \\ T_2^T - \frac{1}{2} D(\overline{w} + \underline{w}) & D \end{bmatrix}}_{\mathcal{L}(D, \tilde{K}_0, \tilde{K}_w, \tilde{v})} \begin{bmatrix} 1 \\ w T_3 \end{bmatrix} \geq 0.$$

As a result, (4.16) is equivalent to $\mathcal{L}(D, \hat{K}_0, \hat{K}_w, \hat{v}) \geq 0$.

4.5 Constraints

From (4.12), the constraints of the causal RMPC can be expressed as

$$\underline{f} \leq f = f\left(\hat{K}_0, \hat{K}_w, \hat{v}\right) \leq \overline{f}$$

where $\underline{f}$, $\overline{f} \in \mathbb{R}^{(N+1)n_f}$ are the lower and upper trajectory boundaries of the constraints signal, which can be applied on both the state and input signal at the same prediction step k, therefore, the problem can not be solved in the form of stack vectors. Instead, a unit select vector $e_i^T = [0, 0, \cdots, 1, \cdots, 0]^T, \forall i \in \{0, 1, 2, \cdots, (N+1)N_f\}$ is introduced to make the constraint be satisfied at each step k, which is

$$e_i^T \underline{f} \leq e_i^T f \leq e_i^T \overline{f}.$$

Discuss the left part inequality first, which is

$$e_i^T f - e_i^T \underline{f} \geq 0.$$

Similarly, it can also be written in the form of (4.16) as

$$\underline{T}_1^i + \mathcal{H}(\underline{T}_2^i w \underline{T}_3^i) \geq 0, \quad \forall i = 0, 1, \cdots, (N+1)n_f. \tag{4.18}$$

which is

$$\overbrace{e_i^T(\tilde{D}_f^{\hat{K}_0,\hat{v}} - \underline{f})}^{\underline{T}_1^i} + \overbrace{\frac{1}{2} e_i^T \tilde{D}_{f_w}^{\hat{K}_w}}^{\underline{T}_2^i} w \cdot \overbrace{1}^{\underline{T}_3^i} + \frac{1}{2} \cdot 1 \cdot w^T \tilde{D}_{f_w}^{\hat{K}_w T} e_i \geq 0,$$

where

$$\begin{bmatrix} \underline{T}_1^i & \underline{T}_2^i \\ \underline{T}_3^i & \mathbf{0} \end{bmatrix} = \begin{array}{c} \\ 1 \\ 1 \end{array} \overset{\begin{array}{cc} 1 & Nn_w \end{array}}{\left[\begin{array}{c|c} e_i^T(\tilde{D}_f^{\hat{K}_0,\hat{v}} - \underline{f}) & \frac{1}{2} e_i^T \tilde{D}_{f_w}^{\hat{K}_w} \\ \hline 1 & \mathbf{0} \end{array}\right]}.$$

Also, Note that if there exists positive semi-definite symmetric matrix $\underline{D}_i$, which is marked as $\underline{D}_i \geq 0$, then it has

$$\underline{T}_3^{iT}(w - \underline{w})^T \underline{D}_i (\overline{w} - w) \underline{T}_3^i \geq 0. \tag{4.19}$$

The equation (4.18) can be rewritten as

$$\underbrace{\underline{T}_3^{iT}(w-\underline{w})^T\underline{D}_i(\overline{w}-w)\underline{T}_3^i}_{(4.19)}-\underbrace{\underline{T}_3^{iT}(w-\underline{w})^T\underline{D}_i(\overline{w}-w)\underline{T}_3^i}_{(4.19)}+\underbrace{\underline{T}_1^i+\underline{T}_2^iw\underline{T}_3^i+\underline{T}_2^{iT}w^T\underline{T}_3^{iT}}_{(4.18)}\geq 0$$

$$\underbrace{\underline{T}_3^{iT}(w-\underline{w})^T\underline{D}_i(\overline{w}-w)\underline{T}_3^i}_{(4.19)}+\begin{bmatrix}1 & \underline{T}_3^{iT}w^T\end{bmatrix}\underbrace{\begin{bmatrix}\underline{T}_1^i+\underline{T}_3^{iT}\underline{w}^T\underline{D}_i\overline{w}\underline{T}_3^i & \star\\ \underline{T}_2^{iT}-\frac{1}{2}\underline{D}_i(\overline{w}+\underline{w}) & \underline{D}_i\end{bmatrix}}_{\underline{\mathcal{L}}_i(\underline{D}_i,\tilde{K}_0,\tilde{K}_w,\tilde{v})}\begin{bmatrix}1\\ w\underline{T}_3^i\end{bmatrix}\geq 0.$$

As a result, (4.18) is equivalent to $\underline{\mathcal{L}}_i(\underline{D}_i,\hat{K}_0,\hat{K}_w,\hat{v})\geq 0$. Similarly, for the right part inequality

$$e_i^T\overline{f}-e_i^Tf\geq 0,$$

it can be written in the form of (4.16) as

$$\overline{T}_1^i+\mathcal{H}(\overline{T}_2^iw\overline{T}_3^i)\geq 0,\quad \forall i=0,1,\cdots,(N+1)n_f. \tag{4.20}$$

where

$$\begin{bmatrix}\overline{T}_1^i & \overline{T}_2^i\\ \overline{T}_3^i & \mathbf{0}\end{bmatrix}=\begin{array}{c}\\ 1\\ 1\end{array}\begin{array}{c}\begin{array}{cc}1 & Nn_w\end{array}\\ \left[\begin{array}{c|c} e_i^T(\overline{f}-\tilde{D}_f^{\hat{K}_0,\hat{v}}) & -\frac{1}{2}e_i^T\tilde{D}_{f_w}^{\hat{K}_w}\\ \hline 1 & \mathbf{0}\end{array}\right]\end{array}.$$

Also, Note that if there exists positive semi-definite symmetric matrix $\overline{D}_i$, which is marked as $\overline{D}_i\geq 0$, then (4.20) is equivalent to

$$\overline{\mathcal{L}}_i(\overline{D}_i,\hat{K}_0,\hat{K}_w,\hat{v})=\begin{bmatrix}\overline{T}_1^i+\overline{T}_3^{iT}\underline{w}^T\overline{D}_i\overline{w}\overline{T}_3^i & \star\\ \overline{T}_2^{iT}-\frac{1}{2}\overline{D}_i(\overline{w}+\underline{w}) & \overline{D}_i\end{bmatrix}\geq 0.$$

As the result, the causal RMPC problem can be reduced as the following form based on the SDPR technique.

$$\min_{(\hat{K}_0,\hat{K},\hat{v})\in\mathcal{U}}\quad \gamma^2$$

$$\begin{aligned} & D \geq 0, \quad \mathcal{L}(D, \tilde{K}_0, \tilde{K}_w, \tilde{v}) \geq 0, \\ s.t. \quad & \overline{D}_i \geq 0, \quad \underline{\mathcal{L}}_i(\underline{D}_i, \hat{K}_0, \hat{K}_w, \hat{v}) \geq 0, \forall i \in \{1, 2, 3, \cdots, (N+1)N_f\}, \\ & \underline{D}_i \geq 0, \quad \overline{\mathcal{L}}_i(\overline{D}_i, \hat{K}_0, \hat{K}_w, \hat{v}) \geq 0, \forall i \in \{1, 2, 3, \cdots, (N+1)N_f\}. \end{aligned} \tag{4.21}$$

4.6 Numerical Case Study

In this section, a numerical example is designed to illustrate the performance of the LMI-based RMPC with causal state-feedback control law. The programs are running on the CVX platform with Mosek Solver.

The DTLI system is given by:

$$A = \begin{bmatrix} 1 & 0.8 \\ 0.5 & 1 \end{bmatrix}, \quad B_u = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad B_w = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}. \tag{4.22}$$

The system is suffered by bounded disturbance as

$$w_i \in \mathcal{W}_i := \{w \in \mathbb{R} : -1 \leq w \leq 1\}, \quad i \in \{1, 2, \cdots, N-1\} \tag{4.23}$$

The cost signal z gives the same penalty weight on state error and input, which is given as

$$z_i = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}}_{C_z} x_i + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{D_{zu}} u_i + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}_{D_{zd}} w_i \quad i \in \{0, 1, 2, \cdots, N-1\}$$

$$z_N = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\hat{C}_z} x_N$$

The constraint is a kind of stage constraint for each state and input, it is given as

$$f_i = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}}_{C_f} x_i + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{D_{fu}} u_i + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}_{D_{fw}} w_k, \quad i \in \{0, 1, 2, \cdots, N-1\}$$

$$f_N = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\hat{C}_f} x_N$$

The boundaries on them are kept as the same at each step and are given as

$$\begin{bmatrix} -7 \\ -7 \\ -7 \end{bmatrix} \leq \begin{bmatrix} x_k \\ u_i \end{bmatrix} \leq \begin{bmatrix} 7 \\ 7 \\ 7 \end{bmatrix}, \quad \begin{matrix} k \in \{0, 1, 2, \cdots, N\} \\ i \in \{0, 1, 2, \cdots, N-1\}. \end{matrix} \tag{4.24}$$

The trajectory of state and input signal U for CRMPC are shown in Fig.4.1 and Fig.4.2, respectively. It is observed that the performance is much better when compared with that shown in Fig.3.3 and Fig.3.4.

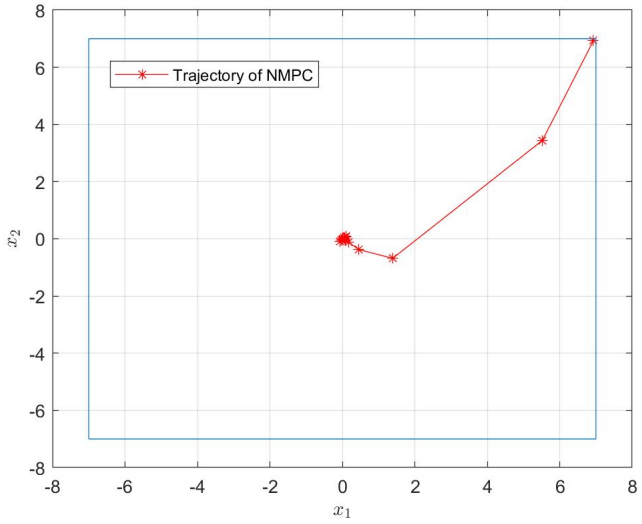


Figure 4.1: Trajectory of CRMPC

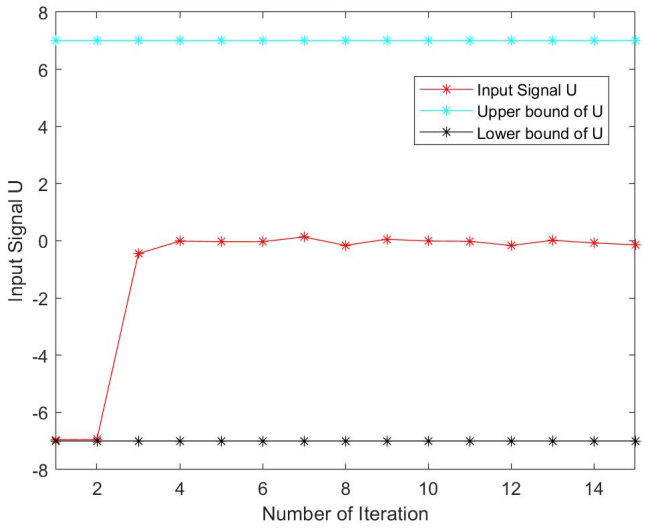


Figure 4.2: Input Signal of CRMPC

The actual cost and its upper bound is shown in Fig.4.3. We can see that the minimized upper bound is very close to the actual cost.

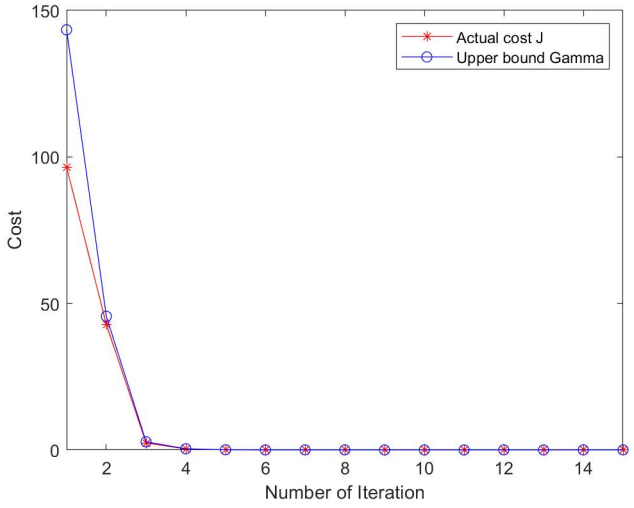


Figure 4.3: Cost of CRMPC

Chapter 5

Computational Efficient Methods for LMI-based CRMPC

THIS chapter introduces two efficient computational methods for LMI-based robust model predictive control with causal state-feedback (CRMPC) scheme.

In Section 5.1, an online method called Single LMI Method is introduced.

In Section 5.2, an offline method taking advantage of an offline lookup table is proposed and illustrated in detail with a numerical case.

In Section 5.3, the two methods are compared with the original problem and the performance in reducing online computation time is shown and discussed.

5.1 Online Method: Single LMI Method

In this section, instead of solving multiple matrix inequalities for the constraints (one matrix inequality for each one of the constraints, see (4.18) and (4.20)), an approach is proposed here to combine all the inequalities within a single inequality, which in turn results in reduced computational complexity because of the abandonment of circulation in the online calculation. Besides, the total number of matrix inequalities is primarily decreased, improving the simplicity of the constraints form.

5.1.1 Method Description

The approach takes advantage of an S-procedure to derive one LMI condition sufficient for a set of scalar inequalities shown in the last section. The details are shown as following: Let $\tilde{f} \in \mathbb{R}^{N_f}$ and let $e \in \mathbb{R}^{N_f}$ be the vector of ones. Then $\tilde{f} \geq 0$ is satisfied if there exist $\mu \in \mathbb{R}$ and $M \in \mathbb{D}^{N_f}$ such that,

$$\mathcal{L} := \begin{bmatrix} 2\mu & \star \\ \tilde{f} - Me - e\mu & M + M^T \end{bmatrix} \geq 0. \tag{5.1}$$

It can be proved by the following steps:

Let

$$\Omega := \left\{ \operatorname{diag}\left(\delta_1, \ldots, \delta_{N_f}\right) : \delta_i \in \{0,1\}, \sum_{i=1}^{N_f} \delta_i = 1 \right\}.$$

Then,

$$\tilde{f} \geq 0 \Leftrightarrow e^T \Delta \tilde{f} + \tilde{f}^T \Delta^T e \geq 0, \forall \Delta \in \mathbf{\Omega} \tag{5.2}$$

Suppose that $\Delta \in \Omega$, then, it follows from $\delta_i \in \{0,1\}$ and $\sum_{i=1}^{N_f} \delta_i = 1$ that

$$\begin{aligned} \Delta M + M^T \Delta^T - \Delta\left(M + M^T\right)\Delta^T = 0, \forall M \in \mathbb{D}^{N_f}, \\ e^T \Delta e \mu + \mu e^T \Delta^T e - 2\mu = 0, \forall \mu \in \mathbb{R}, \end{aligned} \tag{5.3}$$

respectively. It is easy to verify the identity,

$$\begin{aligned} e^T \Delta \tilde{f} + \tilde{f}^T \Delta^T e =& e^T \overbrace{\left(\Delta M + M^T \Delta^T - \Delta\left(M + M^T\right)\Delta^T\right)}^{0} e \\ &+ \overbrace{\left(e^T \Delta e \mu + \mu e^T \Delta^T e - 2\mu\right)}^{0} \\ &+ \begin{bmatrix} 1 & e^T \Delta \end{bmatrix} \mathcal{L} \begin{bmatrix} 1 \\ \Delta^T e \end{bmatrix} \end{aligned}$$

Therefore, with the addition of (5.3), (5.2) can be rewritten in a quadratic from and $\mathcal{L} \geq 0$ is sufficient for $\tilde{f} \geq 0$. In the sequel, we can replace $\tilde{f}$ by $\overline{f} - f$ and $f - \underline{f}$ to generate two

matrix inequalities for upper bound and lower bound of the constraint signal f respectively. It can also be written as a single LMI if the original inequality for constraint signal f is restructured as:

$$\overline{f}^{\star} - f^{\star} = \begin{bmatrix} \overline{f} \\ -\underline{f} \end{bmatrix} - \begin{bmatrix} f \\ -f \end{bmatrix} \geq 0$$

5.1.2 Constraint Signal Reformulation

Based on the method introduced in the last subsection, the original matrix inequalities for constraint signal (4.18) and (4.20) and can be reformulated as (5.4) and (5.5) respectively:

$$\underline{T}_1^s + \mathcal{H}(\underline{T}_2^s w \underline{T}_3^s) \geq 0 \tag{5.4}$$

$$\overline{T}_1^s + \mathcal{H}(\overline{T}_2^s w \overline{T}_3^s) \geq 0 \tag{5.5}$$

Where $e \in \mathbb{R}^{N_f}$ is the vector of ones and exists $\underline{\mu}, \overline{\mu} \in \mathbb{R}$ and $\underline{M}, \overline{M} \in \mathbb{D}^{N_f}$ that

$$\begin{bmatrix} \underline{T}_1^s & \underline{T}_2^s \\ \underline{T}_3^s & \mathbf{0} \end{bmatrix} = \begin{array}{c} \\ 1 \\ (N+1)n_f \\ \hline 1 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & (N+1)n_f & Nn_w \end{array} \\ \left[\begin{array}{c|cc} 2\underline{\mu} & \star & \mathbf{0} \\ \tilde{D}_f^{\hat{K}_0,\hat{v}} - \underline{f} - \underline{M}e - e\underline{\mu} & \underline{M} + \underline{M}^T & \tilde{D}_{zw}^{\hat{K}_w} \\ \hline \mathbf{1} & \mathbf{0} & \mathbf{0} \end{array}\right] \end{array},$$

$$\begin{bmatrix} \overline{T}_1^s & \overline{T}_2^s \\ \overline{T}_3^s & \mathbf{0} \end{bmatrix} = \begin{array}{c} \\ 1 \\ (N+1)n_f \\ \hline 1 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & (N+1)n_f & Nn_w \end{array} \\ \left[\begin{array}{c|cc} 2\overline{\mu} & \star & \mathbf{0} \\ \overline{f} - \tilde{D}_f^{\hat{K}_0,\hat{v}} - \overline{M}e - e\overline{\mu} & \overline{M} + \overline{M}^T & -\tilde{D}_{zw}^{\hat{K}_w} \\ \hline \mathbf{1} & \mathbf{0} & \mathbf{0} \end{array}\right] \end{array}.$$

Replace $(\underline{T}_1^i, \underline{T}_2^i, \underline{T}_3^i)$ and $(\overline{T}_1^i, \overline{T}_2^i, \overline{T}_3^i)$ by $(\underline{T}_1^s, \underline{T}_2^s, \underline{T}_3^s)$ and $(\overline{T}_1^s, \overline{T}_2^s, \overline{T}_3^s)$ respectively, $\underline{\mathcal{L}}_i(\underline{D}_i, \hat{K}_0, \hat{K}_w, \hat{v})$ and $\overline{\mathcal{L}}_i(\overline{D}_i, \hat{K}_0, \hat{K}_w, \hat{v})$ are in turn transformed into $\underline{\mathcal{L}}(\underline{D}, \hat{K}_0, \hat{K}_w, \hat{v})$ and $\overline{\mathcal{L}}(\overline{D}, \hat{K}_0, \hat{K}_w, \hat{v})$.

As a result, the optimization problem for CRMPC (4.21) can be restructured as

$$\min_{(\hat{K}_0,\hat{K},\hat{v})\in\mathcal{U}} \quad \gamma^2$$

$$\begin{aligned} & D \geq 0, \quad \mathcal{L}(D,\tilde{K}_0,\tilde{K}_w,\tilde{v}) \geq 0, \\ s.t. \quad & \overline{D} \geq 0, \quad \underline{\mathcal{L}}(\underline{D},\hat{K}_0,\hat{K}_w,\hat{v}) \geq 0, \forall i \in \{1,2,3,\cdots,(N+1)N_f\}, \\ & \underline{D} \geq 0, \quad \overline{\mathcal{L}}(\overline{D},\hat{K}_0,\hat{K}_w,\hat{v}) \geq 0, \forall i \in \{1,2,3,\cdots,(N+1)N_f\}. \end{aligned} \tag{5.6}$$

A case study of using single LMI method will be performed and analyzed in Chapter8.

5.2 Offline Method: Lookup Table

In this section, an offline method taking advantage of using an offline lookup table to reduce the online computation time is introduced and adjusted to the model with only disturbances.

Using an offline lookup table to turn a part of SDP online into offline is proposed in [4], whose idea is to use a lookup table to record the initial feasible solutions of a particular region to give a warm start to an online problem the model with uncertainty.

However, the model investigated in this project is linear with only addictive disturbances, so there is no need to store the initial solution because that could be optimal. In terms of not introducing too much conservatism, the application of the lookup table in this project is to store the future state-feedback gain matrix $\tilde{K}$(which can be restored from $\hat{K}_w$ by (4.10)), and compute the rest parameters $\tilde{K}_0(\hat{K}_0)$ and $\tilde{v}(\hat{v})$ online.

5.2.1 Formulation of the method

In order to guarantee a fixed control law that is suitable for enabling a set of initial states not to violate any constraints, we can use (4.5) to represent x in the form of x_0 and gives x_0 a tighter constraint to check if the constraint is satisfied for all states initialed in this small set. If it is satisfied, then the set of states along the controller can be recorded in the lookup table until all the subsets of the feasible region are checked and corresponding

controllers are designed and stored.

With tighter constraints on the initial state, the origin constraints (5.4) and (5.5) can be expressed by (5.7) and (5.8) as

$$\underline{T}_1 + \mathcal{H}(\underline{T}_2^w w \underline{T}_3^w) + \mathcal{H}(\underline{T}_2^{x_0} x_0 \underline{T}_3^{x_0}) \geq 0. \tag{5.7}$$

and

$$\overline{T}_1 + \mathcal{H}(\overline{T}_2^w w \overline{T}_3^w) + \mathcal{H}(\overline{T}_2^{x_0} x_0 \overline{T}_3^{x_0}) \geq 0. \tag{5.8}$$

Where $e \in \mathbb{R}^{N_f}$ is the vector of ones and exist $\underline{\mu}, \overline{\mu} \in \mathbb{R}$ and $\underline{M}, \overline{M} \in \mathbb{D}^{N_f}$ that

$$\begin{bmatrix} \underline{T}_1 & \underline{T}_2^{x_0} & \underline{T}_2^w \\ \underline{T}_3^{x_0} & \mathbf{0} & \star \\ \underline{T}_3^w & \star & \mathbf{0} \end{bmatrix} = \left[\begin{array}{cc|c|c} 2\underline{\mu} & \star & \mathbf{0} & \mathbf{0} \\ \tilde{D}_f^{\hat{K}_0,\hat{v}} - \underline{f} - \underline{M}e - e\underline{\mu} & \underline{M} + \underline{M}^T & \tilde{C}_f + \tilde{D}_{f_u}\hat{K}_0 & \tilde{D}_{zw}^{\hat{K}_w} \\ \hline \mathbf{1} & \mathbf{0} & \mathbf{0} & \star \\ \hline \mathbf{1} & \mathbf{0} & \star & \mathbf{0} \end{array}\right] \tag{5.9}$$

and

$$\begin{bmatrix} \overline{T}_1 & \overline{T}_2^{x_0} & \overline{T}_2^w \\ \overline{T}_3^{x_0} & \mathbf{0} & \star \\ \overline{T}_3^w & \star & \mathbf{0} \end{bmatrix}, = \left[\begin{array}{cc|c|c} 2\overline{\mu} & \star & \mathbf{0} & \mathbf{0} \\ \overline{f} - \tilde{D}_f^{\hat{K}_0,\hat{v}} - \overline{M}e - e\overline{\mu} & \overline{M} + \overline{M}^T & \tilde{C}_f + \tilde{D}_{f_u}\hat{K}_0 & \tilde{D}_{zw}^{\hat{K}_w} \\ \hline \mathbf{1} & \mathbf{0} & \mathbf{0} & \star \\ \hline \mathbf{1} & \mathbf{0} & \star & \mathbf{0} \end{array}\right]. \tag{5.10}$$

For the lower bound constraint (5.7), given a subset $x_0 \in \mathcal{X}_0 := \{x_0 \in \mathbb{R}^n : \underline{c}_0 \leq C_0 x_0 \leq \overline{c}_0\}$, if there exist $0 \leq \underline{D}_0^{x_0} \in \mathbb{D}^{n_x}$ and $0 \leq \underline{D}_0^w \in \mathbb{D}^{n_w}$ such that

$$\underline{L} := \begin{bmatrix} \underline{T}_1 + \underline{T}_3^T \underline{c}_0^T \underline{D}_0^{x_0} \overline{c}_0 \underline{T}_3 + \underline{T}_3^T \underline{c}_0^T D_0^{x_0} \overline{c}_0 \underline{T}_3 & * & \star \\ \underline{T}_2^{x_0 T} - \frac{1}{2} C_0^T D_0 (\underline{c}_0 + \overline{c}_0) \underline{T}_3^{x_0} & C_0^T \underline{D}_0^{x0} C_0 & 0 \\ \underline{T}_2^{w T} - \frac{1}{2} D_0^w (\underline{w} + \overline{w}) \underline{T}_3^w & 0 & \underline{D}_0^w \end{bmatrix} \geq 0$$

Then

$$\begin{aligned}
&\underline{T}_1 + \mathcal{H}(\underline{T}_2^w w \underline{T}_3^w) + \mathcal{H}(\underline{T}_2^{x_0} x_0 \underline{T}_3^{x_0}) \\
=&\underline{T}_3^{x_0 T} \left(C_0 x_0 - \underline{c}_0\right)^T D_0^{x_0} \left(\overline{c}_0 - C_0 x_0\right) \underline{T}_3^{x_0} + \underline{T}_3^{w T} (\overline{w} - w)^T D_0^w (w - \underline{w}) T_3^w \\
&+ \begin{bmatrix} I_{(N+1)n_f+1} & \underline{T}_3^{x_0 T} x_0^T & \underline{T}_3^{w T} w^T \end{bmatrix} \underline{L} \begin{bmatrix} I_{(N+1)n_f+1} \\ x_0 \underline{T}_3^{x_0} \\ w \underline{T}_3^w \end{bmatrix} \geq 0
\end{aligned}$$

is satisfied, and the procedure for (5.8) is the same.

Algorithm 1: CRMPC with Lookup Table

Offline Algorithm:

Step1: Divide original feasible region for state into several small grids.

Step2: Replace $\overline{f}$ and $\underline{f}$ in (5.9) and (5.10) by $\beta \underline{f}$ and $\beta \overline{f}$ respectively, where $1 \leq \beta \in \mathbb{R}$. Try to minimize β subject to a tighter constraint (5.7) and (5.8).
If $\beta = 1$, store the $\tilde{K}(\hat{K}_w)$ in lookup table and then repeat **step2** on the next grid until all the grids have been checked and corresponding parameters have been stored.
If $\beta > 1$, subdivide the grid and go to **step2** to check if the smaller grid satisfy the constraint.

Online Algorithm:

Step1: Define current iteration flag $i = 1$ and maximum iteration number i_{max}.

Step2: Check the number of grid in which the current state is and read according $\tilde{K}(\hat{K}_w)$. Solve online SDP to compute the rest parameters, namely $\tilde{K}_0(\hat{K}_0)$ and $\tilde{v}(\hat{v})$. Form the causal state-feedback control law and use it to update state.
If $i < i_{max}$, go to **step2**, otherwise, go to **step3**.

Step3: End algorithm

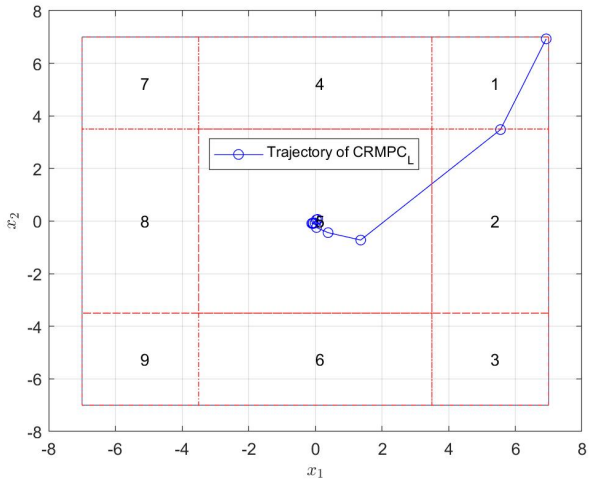


Figure 5.1: Trajectory of $CRMPC_L$

5.2.2 Numerical Case study

Simulate the method of using lookup table on LMI-based CRMPC scheme on the same numerical case as (4.22), (4.23) and (4.24). The offline lookup table generated and the trajectory of state is shown in Fig. 5.1. Where it is observed that 9 small grids are divided from the original feasible region, the state moves from grid 1 to grid 2 and enter grid 5 to the origin point. The the upper bound γ of cost function and the cost of actual state is shown in Fig. 5.2.

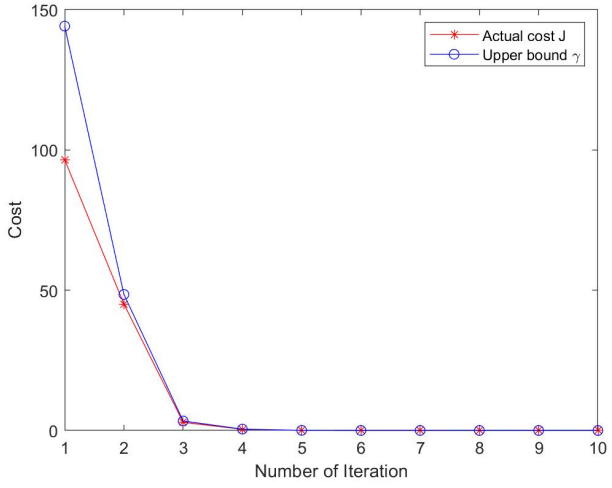
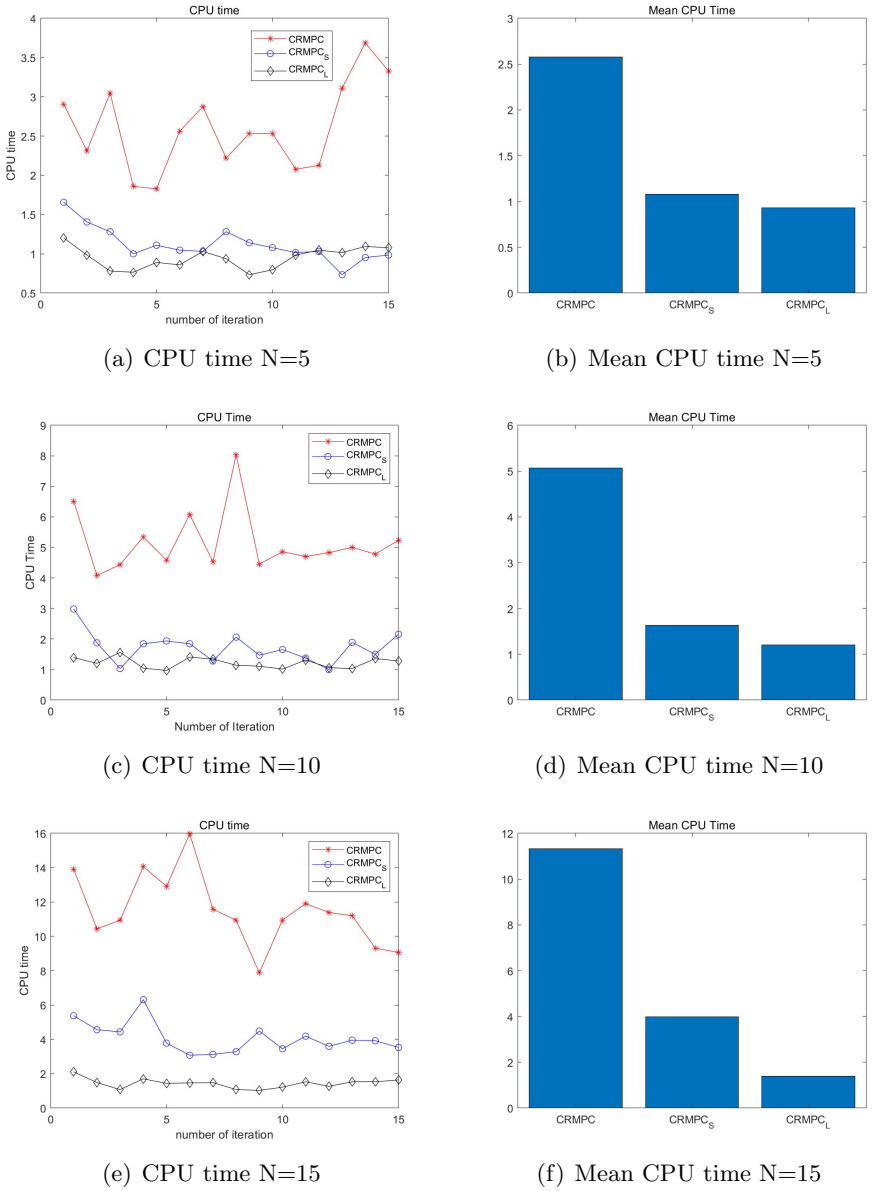


Figure 5.2: Cost of $CRMPC_L$

5.3 Comparisons of Methods for LMI-based CRMPC

Compare $\#CRMPC$(CRMPC without single LMI method), $\#CRMPC_S$(CRMPC with single LMI method) and $\#CRMPC_L$ (CRMPC with single LMI method and application of offline lookup table) together on the same numerical case as shown before in this chapter. The simulation results in terms of online CPU time of CVX toolbox are shown in Fig. 5.3 and Table. 5.1 with specific data.



(a) CPU time N=5

(b) Mean CPU time N=5

(c) CPU time N=10

(d) Mean CPU time N=10

(e) CPU time N=15

(f) Mean CPU time N=15

Figure 5.3: CPU time of CVX

Table 5.1: CPU time of Mosek solver

Horizon length N =5

Method	Mean ± Std deviation	Maximum CPU time
CRMPC without single LMI	2.5781 ± 0.5631 s	3.6875 s
CRMPC with Single LMI	1.0781 ± 0.1648 s	1.4062 s
CRMPC with Lookup Table	0.9286 ± 0.1233 s	1.0938 s

Horizon length N =10

Method	Mean ± Std deviation	Maximum CPU time
CRMPC without single LMI	5.0670 ± 0.9810 s	8.0312 s
CRMPC with Single LMI	1.6373 ± 0.3682 s	2.1562 s
CRMPC with Lookup Table	1.2031 ± 0.1774 s	1.5625 s

Horizon length N =15

Method	Mean ± Std deviation	Maximum CPU time
CRMPC without single LMI	11.4029 ± 2.5816 s	15.8906 s
CRMPC with Single LMI	3.9922 ± 1.2205 s	7.2656 s
CRMPC with Lookup Table	1.4554 ± 0.1962 s	1.8438 s

In Fig. 5.3, the left column shows the line graph of CPU time of CVX box of each iteration for prediction horizon $N = 5$, $N = 10$, and$N = 15$ respectively, and the right column shows the bar graphs of their mean CPU time. To prevent the influence of the extra time caused by the program's launch, the CPU time at the first iteration is not counted when computing the mean value of the data.

It is observed that with the application of the single LMI method, the time of CVX calculation for each iteration is significantly reduced, which can be represented by the significant gaps between red lines and the others in the figures of the left column. Meanwhile, with the prediction horizon N getting longer, the gaps between blue and black lines go larger. Therefore $\#CRMPC_L$ can primarily relieve the computational burden of the long horizon MPC scheme. The code for the above can be found by the URL in Appendix, named as `CRMPC_S_final.m` and `CRMPC_L_final.m`.

Chapter 6

Robust Control Invatiant Set

IN this chapter, the robust control invariant(RCI) set which is an important base is introduced.

In the following subsections, a low-complexity RCI set for model with bounded disturbance will be designed and illustrated in detail.

6.1 Design of low-complexity RCI set

The RCI set can be defined as a set in which arbitrary initial state will not exceed the set and violate the initial constraints with a fixed control law despite the influence of uncertainty and disturbance.

Given a DLTI system with bounded disturbance as defined in (4.3) and (4.2). It is also constrained by

$$x_k \in \mathcal{X} = \{x \in \mathbb{R}^{n_x} : V_x x \leq \bar{x}\} \tag{6.1}$$

and

$$u_k \in \mathcal{U} = \{u \in \mathbb{R}^{n_u} : V_u u \leq \bar{u}\} \tag{6.2}$$

where $V_x \in \mathbb{R}^{n_x \times n_{fx}}$, $0 < \overline{x} \in \mathbb{R}^{n_{fx}}$, $V_u \in \mathbb{R}^{n_{fu} \times n_u}$, $0 < \overline{u} \in \mathbb{R}^{n_{fu}}$ are given and define the state and input constraints in (6.1) and (6.2). It also has $n_f = n_{fx} + n_{fu}$, which is used to separate constraint signal f aforementioned into state constraint and input constraint for clearer definition of RCI set. Usually, a low complexity RCI set is considered as the

form of (6.3):

$$\mathcal{Z} := \{x \in \mathbb{R}^{n_x} : -b \leq Px \leq b\} \tag{6.3}$$

where $b \in \mathbb{R}^{n_x}$ is a vector of ones and $P \in \mathbb{R}^{n_x \times n_x}$ is a square matrix with $\det(C) \neq 0$. The set $\mathcal{Z} \subset \mathbb{R}^n$ is an RCI set [14] under linear state-feedback for the system in (4.3) subject to the constraints in (6.1) and (6.2) if there exists a control law $u = Kx$ such that

$$A_K \mathcal{Z} \oplus B_w \mathcal{W} \subseteq \mathcal{Z}, \forall w \in \mathcal{W} \tag{6.4}$$

$$\mathcal{Z} \subseteq \mathcal{X} \tag{6.5}$$

$$K\mathcal{Z} \subseteq \mathcal{U} \tag{6.6}$$

where $\oplus$ denotes the Minkowski sum and $A_K := A + B_u K$. An RCI set $\mathcal{Z}$ of the form (6.3) and state feedback matrix K are admissible if (6.4)-(6.6) are satisfied.

6.2 Algebraic Formulation of LC-RCI set

In this section, the formulation of constraints and cost function for LC-RCI will be shown in detail.

6.2.1 Constraints

In this subsection, the definition of (6.4)-(6.6) will be shown as conditions in matrix inequalities form, and some nonlinearities will be handled afterwards.

Condition for Invariance Constraint

Due to $\mathcal{Z}$ and $\mathcal{W}$ are symmetric, the invariance constraint (6.4) can be written by one side as

$$e_i^T P A_K x + B_w w - e_i^T b \leq 0, \quad \forall i \in \mathcal{N}_n, \forall x \in \mathcal{Z}, \forall w \in \mathcal{W}, \tag{6.7}$$

where e_i^T is a selecting row vector as defined before and $\mathcal{N}_{n_x} := \{1, \cdots, n_x\}$. Then for any D^i and D_w^i, (6.7) can be written as

$$-(b - Px)^T D^i (Px + b) - (\overline{w} - w)^T D_w^i (w + \overline{w}) - y^T \mathcal{L}_i(D^i, D_w^i, P, K) y \leq 0,$$

where $y^T := \begin{bmatrix} x^T w^T 1 \end{bmatrix}$, and

$$\mathcal{L}_i(D^i, D_w^i, P, K) := \begin{bmatrix} P^T D^i P & 0 & -\frac{1}{2} A_K^T P^T e_i \\ \star & D_w^i & -\frac{1}{2} B_w^T P^T e_i \\ \star & \star & e_i^T b - b^T D^i b - \overline{w}^T D_w^i \overline{w}, \end{bmatrix}$$

Using S-procedure, the sufficient and necessary condition for (6.7) is derived as if there exists $D^i \in \mathbb{D}_+^{n_x \times n_x}$ and $D_x^i \in \mathbb{D}_+^{n_w \times n_w}$ that

$$\mathcal{L}_i(D^i, D_w^i, P, K) \geq 0, \tag{6.8}$$

then (6.7) is satisfied.

Condition for State and Input constraint

The state and input constraints (6.5) and (6.6) can be written as

$$e_j^T V_u K x - e_j^T \overline{u} \leq 0, \quad \forall j \in \mathcal{N}_{n_{fu}}, \forall x \in \mathcal{Z}, \quad \forall w \in \mathcal{W} \tag{6.9}$$

$$e_m^T V_x x - e_m^T \overline{x} \leq 0, \quad \forall m \in \mathcal{N}_{n_{fx}}, \forall x \in \mathcal{Z}, \quad \forall w \in \mathcal{W} \tag{6.10}$$

where e_j^T and e_m^T are selecting row vectors and $\mathcal{N}_{n_{fx}} := \{1, \cdots, n_{fx}\}$, $\mathcal{N}_{n_{fu}} := \{1, \cdots, n_{fu}\}$.

Similarly, using S-procedure, the sufficient and necessary condition for (6.9) and (6.10) is that if there exist $D_x^m \in \mathbb{D}_+^{n_{fx} \times n_{fx}}$ and $D_u^j \in \mathbb{D}_+^{n_{fu} \times n_{fu}}$ such that

$$\begin{aligned} \mathcal{L}_x^m(D_x^m, P) &\geq 0, \\ \mathcal{L}_u^j(D_u^j, P, K) &\geq 0, \end{aligned} \tag{6.11}$$

where

$$\mathcal{L}_x^m(D_x^m, P) = \begin{bmatrix} P^T D_x^m P & -\frac{1}{2} V_x^T e_m \\ \star & e_m^T \bar{x} - b^T D_x^m b \end{bmatrix},$$

$$\mathcal{L}_u^j(D_u^j, P, K) = \begin{bmatrix} P^T D_u^j P & -\frac{1}{2} K^T V_u^T e_j \\ \star & e_j^T \bar{u} - b^T D_u^j b \end{bmatrix},$$

then (6.9) and (6.10) are satisfied.

Linearization

Noted that (6.8) and (6.11) are nonlinear matrix inequalities that can not be solved directly online with CVX toolbox, the nonlinear items namely $P^T \star P$($\star$ denotes D^i, D_x^m or D_u^j) and $A_K^T P^T$ should be approximately linearized by taking advantage of two theorems that proposed and proved in [11].

Firstly, we linearize the item in $\mathcal{L}_i(D^i, D_w^i, P, K)$ in (6.7). Applying a congruence transformation $\mathrm{diag}(I_{n_x}, I_{n_w}, -2I)$ and taking a Schur complement on $\mathcal{L}_i(D^i, D_w^i, P, K)$, then (6.7) is sufficient and necessary by

$$\begin{bmatrix} P^T D^i P & 0 \\ \star & D_w^i \end{bmatrix} - \begin{bmatrix} A_K^T \\ B_w^T \end{bmatrix} P^T e_i r_i^{-1} (P^T e_i)^T \begin{bmatrix} A_K & B_w \end{bmatrix} \geq 0, \quad \forall i \in \mathcal{N}_{n_x}. \tag{6.12}$$

where $r_i = 4(e_i^T b - b^T D^i b - \overline{w}^T D_w^i \overline{w})$.

Applying extended lemma of Schur complement introduced in chapter 2 on (6.12), if there exists $X_i = X_i^T \in \mathbb{R}^{n_x \times n_x}, \forall i \in \mathcal{N}_{n_x}$, (6.12) is sufficient and necessary by proving

$$\begin{bmatrix} P^T D^i P & 0 & A_K^T \\ \star & D_w^i & B_w^T \\ \star & \star & X_i^{-1} \end{bmatrix} \geq 0, \quad \begin{bmatrix} X_i & P^T e_i \\ \star & r_i \end{bmatrix} \geq 0 \tag{6.13}$$

Noted that P is a square matrix with $\det(P) \neq 0$, meaning P exists inverse matrix P^{-1}. Applying a congruence transformation $\mathrm{diag}(P^{-T}, I_{n_w}, I_{n_x})$ on the first inequality in (6.13)

yields (with $\hat{K} := KP^{-1}$) :

$$\begin{bmatrix} D^i & 0 & P^{-T}A^T + \hat{K}^T B_u^T \\ \star & D_w^i & B_w^T \\ \star & \star & X_i^{-1} \end{bmatrix} \geq 0, \quad \forall i \in \mathcal{N}_{n_x} \tag{6.14}$$

For the second inequality in (6.13), use the congruence transformation $\texttt{diag}\left(C^{-T}, I\right)$ yields,

$$\begin{bmatrix} P^{-T}X_iP^{-1} & e_i \\ \star & 4\left(e_i^T b - b^T D_x^i b - \overline{w}^T D_w^i \overline{w}\right) \end{bmatrix} \geq 0, \quad \forall i \in \mathcal{N}_{n_x} \tag{6.15}$$

Until now, due to the model is only influenced by additive disturbance, the necessary and sufficient conditions [11] for the invariance constraint (6.4) can now be given by (6.14)-(6.15).

Noted that $P^{-T}X_iP^{-1}$ is still nonlinear, then multiply (6.15) by $\lambda_i\rho^{-1}$(with fixed and given $\rho = 1$), where $\lambda_i = e_i^T \Lambda e_i$, and then follow by a congruence transformation $\text{diag}(I_{n_x}, \rho I)$, it yields,

$$\begin{bmatrix} \lambda_i\rho^{-1}P^{-T}X_iP^{-1} & \lambda_i e_i \\ \star & 4\lambda_i\rho\left(e_i^T d - d^T D_x^i d - v^T D_w^i v\right) \end{bmatrix} \succ 0, \quad \forall i \in \mathcal{N}_{n_x}. \tag{6.16}$$

Redefine $\hat{X}_i^{-1} := \rho\lambda_i^{-1}X_i^{-1}$, $\hat{D}_w^i := \rho\lambda_i D_w^i$ and $\hat{D}_x^i := \rho\lambda_i D_x^i$. Noted that $\lambda_i e_i = \Lambda e_i$ and apply the congruence transformation $\text{diag}\left(Z_i^T\Lambda^{-1}, I\right)$ on (6.16), it has

$$\begin{bmatrix} (P^{-1}\Lambda^{-1}Z_i)^T\hat{X}_iP^{-1}\Lambda^{-1}Z_i & Z_i^T e_i \\ \star & l_i \end{bmatrix} \geq 0, \quad \forall i \in \mathcal{N}_{n_x}. \tag{6.17}$$

where $l_i := 4(\rho\lambda_i e_i^T b - \overline{w}^T\hat{D}_w^i\overline{w} - b^T\hat{D}_x^i b)$, $K := \hat{K}P$

Using slack-variable identity (16) on the $(1,1)$ entry of (6.17) gives

$$\begin{bmatrix} (P^{-1}\Lambda^{-1}Z_i)^T + P^{-1}\Lambda^{-1}Z_i + X_i^{-1} & Z_i^T e_i \\ \star & l_i \end{bmatrix} \geq 0, \quad \forall i \in \mathcal{N}_{n_x}. \tag{6.18}$$

Applying extended lemma of BMI introduced in chapter 2 on entry (1,1) of (6.18) with matrix

$$M_o := \begin{bmatrix} \Lambda & \Lambda \\ F_i & H_i \end{bmatrix}, \quad \forall i \in \mathcal{N}_{n_x},$$

and ignoring the positive term $P^{-1}\Lambda^{-1}E_i\Lambda^{-1}P^{-T}$ yields the LMIs as following:

$$\begin{bmatrix} E_i & Z_i \\ \star & Q_i \end{bmatrix} \geq 0, \begin{bmatrix} Q_i - X_i^{-1} & F_i - P^{-1} & H_i - P^{-1} & Z_i^T e_i \\ \star & 2\Lambda - E_i & \Lambda + F_i^T - Z_i & 0 \\ \star & \star & H_i^T + H_i - Q_i & 0 \\ \star & \star & \star & l_i \end{bmatrix} \geq 0, \quad \forall i \in \mathcal{N}_{n_x}. \tag{6.19}$$

Hence, (6.19) is sufficient and necessary condition for the second inequality in (6.13).

The next step is to deal with the nonlinear items in conditions for state and input constraints. Due to the projection P is exposed at the two sides of entry (1,1) in both $\mathcal{L}_x^m(D_x^m, P)$ and $\mathcal{L}_u^j(D_u^j, P, K)$, it is easy to linearize them by applying congruence transformation $\text{diag}\left(P^{-T}, I\right)$ on (6.11) and yields

$$\begin{bmatrix} D_x^m & -\frac{1}{2}P^{-T}V_x^T e_m \\ \star & e_m^T\bar{x} - b^T D_x^m b \end{bmatrix} \geq 0, \begin{bmatrix} D_u^j & -\frac{1}{2}\hat{K}^T V_u^T e_j \\ \star & e_j^T\bar{u} - b^T D_u^j b \end{bmatrix} \geq 0. \tag{6.20}$$

Until now, nonlinear matrix inequalities (6.8) and (6.11) obtained in last subsection have all be reformulated and satisfied by LMIs (6.14), (6.19) and (6.20), which are sufficient and necessary conditions for definitions (6.6), (6.5) and (6.6).

6.2.2 Cost Function

In this section, a cost function in order to compute the smallest volume constraint-admissible RCI set (also minimal volume RCI set approximations) is introduced. The cost function for computing maximal vlolume RCI set approximations can also be found in [11], [15]. The volume of $\mathcal{Z}$ is proportional to $\left|\det\left(P^{-1}\right)\right|$ [16] and noted that P^{-1} is a variable to be solved by online SDP, so we derive a upper bound matrix variable

$\overline{W} = \overline{W}^T \geq 0$ on the determinant and apply Schur complement on it as given below:

$$\overline{W} - P^{-1}P^{-T} \geq 0 \Leftrightarrow \begin{bmatrix} \overline{W} & P^{-1} \\ \star & I \end{bmatrix} \geq 0. \tag{6.21}$$

The trace of matrix variable $\overline{W}$ is a convex function and can be used to find the global minimum. After all, the approximated optimal problem (Because not all the conditions are sufficient and necessary) to find the minimal volume of LC-RCI is shown as following:

$$\begin{aligned} &\min_{P^{-1},\hat{K},\hat{X}_i^{-1}} \operatorname{trace}(\overline{W}) \\ &s.t. \quad (6.14) \\ &\qquad (6.19) \\ &\qquad (6.20) \end{aligned} \tag{6.22}$$

where projection matrix P, matrix variable $\hat{X}_i$ and coefficient of inner controller K can be restored by inverse matrix and $K := \hat{K}P$.

6.2.3 Update Algorithm

Noted that (6.19) is used to ensure (6.15), which will inevitably introduces conservatism to guarantee a feasible solution for optimal problem(6.22) and is not the optimal solution to the original problem. Therefore, once we obtain a initial feasible solution to (6.22), we can rewrite (6.15) by redefined variable $\hat{X}_i$ and l_i as

$$\begin{bmatrix} P^{-T}\hat{X}_i P^{-1} & \lambda_i e_i \\ \star & l_i \end{bmatrix} \geq 0, \quad \forall i \in \mathcal{N}_{n_x}. \tag{6.23}$$

We can take the advantage of linear approximation to replace the (1,1) entry of (6.23), which is

$$P^{-T}\hat{X}_i P^{-1} \approx P^{-T}\hat{X}_{i0}P_0^{-1} + P_0^{-T}\hat{X}_{i0}P^{-1} - P_0^{-T}\hat{X}_{i0}\hat{X}_i^{-1}\hat{X}_{i0}P_0^{-1}.$$

The matrices $\bullet_0$ represent the solution from the previous update solution of (6.24) or the initial solution calculated by the original optimal problem (6.22), Where the optimal problem for update is shown as

$$\begin{aligned} \min_{P^{-1},\hat{K},\hat{X}_i^{-1}} \quad & \text{trace}(\overline{W}) \\ s.t. \quad & (6.14) \\ & (6.23) \\ & (6.20) \end{aligned} \quad . \tag{6.24}$$

To summarize, the whole algorithm to compute LC-RCI is shown as following.

Algorithm 2: Generation of LC-RCI Set

Step1 Initial preparations: Define a linear discrete system, state constraint, input constraint, disturbance and LC-RCI as (4.3), (6.1), (6.2), (4.2) and (6.3) respectively. Define maximum number of iteration $iter$, current iteration label $i = 1$ and update tolerance $tol > 0$.

Step2 Initial solution: Solve optimal problem (6.22) and record the solution P, $\hat{X}_i$ and $\overline{W}$ as $\bullet_0$ items for later update.

Step3 Update solution: Solve optimal problem (6.24) and record the solution P, $\hat{X}_i$ and $\overline{W}$ as $\bullet_0$ items for later update. Calculate the approximate volume change rate $T = (trace(\overline{W}_0) - trace(\overline{W}))/trace(\overline{W}_0)$. Current iteration label $i = i + 1$.

If $T > tol$ and $i < iter$, go to **step3**, otherwise, go to **step4**.

Step4 End algorithm

6.3 Numerical Case Study

In this section, a numerical case is presented to calculate its minimal LC-RCI with the **Algorithm 2** proposed above. The system and bounded disturbance are kept as the same with (4.22) and (4.23) respectively, while the constraint signal is reformulated by state constraint and input constraint as:

$$x_k \in \mathcal{X} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -7 \\ -7 \end{bmatrix} \leq x \leq \begin{bmatrix} 7 \\ 7 \end{bmatrix} \right\}$$

and

$$u_k \in \mathcal{U} = \{u \in \mathbb{R}^{\vdots} - 7 \leq u \leq 7\}$$

Due to the symmetry of LC-RCI and the feasible region, we only need to consider one side boundary when solving SDP. Set the update tolerance $tol = 0.01$.

The update of the volume of LC-RCI is vividly illustrated in Fig. 6.1, where the light blue/purple parallelograms are LC-RCIs computed and updated at each iteration, the red parallelogram is the minimal(final) LC-RCI computed by **Algorithm 2** and the yellow point is the origin point (0,0). It is observed that the volume of LC-RCI is minimizing at each update iteration, and the LC-RCIs are centrosymmetric about the origin.

The initial projection at the first iteration P_0 and the final projection $P^\star$ for minimal RCI set by **Algorithm 2** are

$$P_0 = \begin{bmatrix} 12.5774 & -5.0310 \\ -2.6858 & 8.5157 \end{bmatrix}, \quad P^\star = \begin{bmatrix} 16.1483 & -6.1546 \\ -10.0697 & 14.5850 \end{bmatrix}.$$

The actual volume of the LC-RCI and the minimized cost, namely $trace(\overline{W})$, of each iteration is shown in Fig. 6.2.

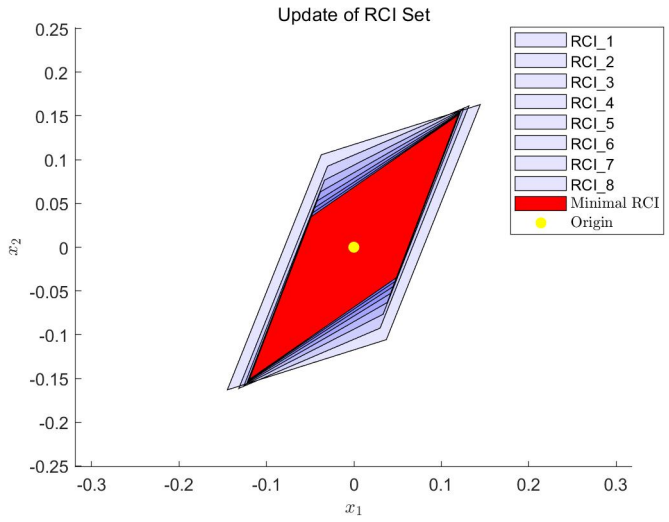


Figure 6.1: Minimizing Volume of LC-RCI Set

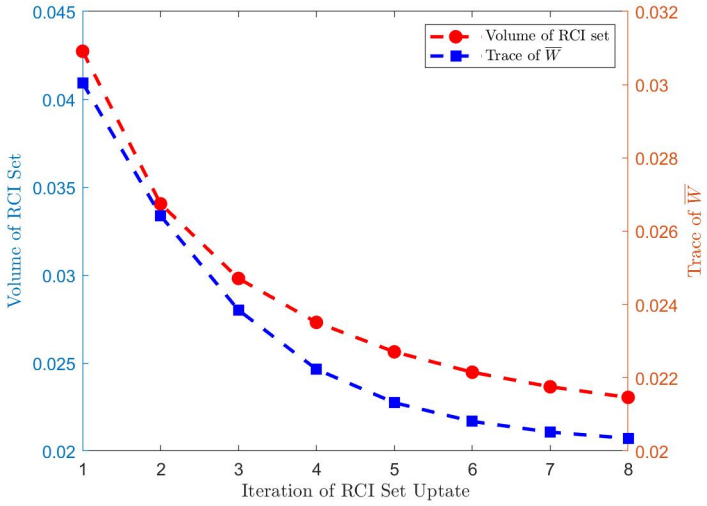


Figure 6.2: Update of minimal LC-RCI Set

Chapter 7

Tube-based Model Predictive Control

IN this chapter, a new offline method for reducing online computation time called tube-based Model predictive control (Tube-based MPC) is introduced.

In Section 7.1, a simple explanation of the idea of Tube-based MPC is given.

In Section 7.2, the specific formulation will be illustrated in detail.

In Section 7.3, the performance of this offline method will be demonstrated with a numerical case.

7.1 Ideas Description

The idea of this method is to separate the nominal system from the real system(with uncertainty and disturbance), then transfer the original problem from controlling the real system into controlling the nominal system and regulate the states of the system in a RCI set and the trajectory constraint of RCI sets for the states will look like a tube. Calculate their according controllers respectively and combine them as the final robust controllers to update the state. In this way, the online computation for designing a robust controller is transferred from solving SDP into solving QP, which is much less computationally expensive.

7.2 Algebraic Formulation

Given a nominal system (3.1)(which will be labelled with $\overline{\bullet}$) and a real system (4.1) influenced by bounded disturbance (4.2) with the constraints on states (6.1) and input (6.2). The whole scheme of the tube-based MPC will be illustrated as shown in Fig. 7.1. So the

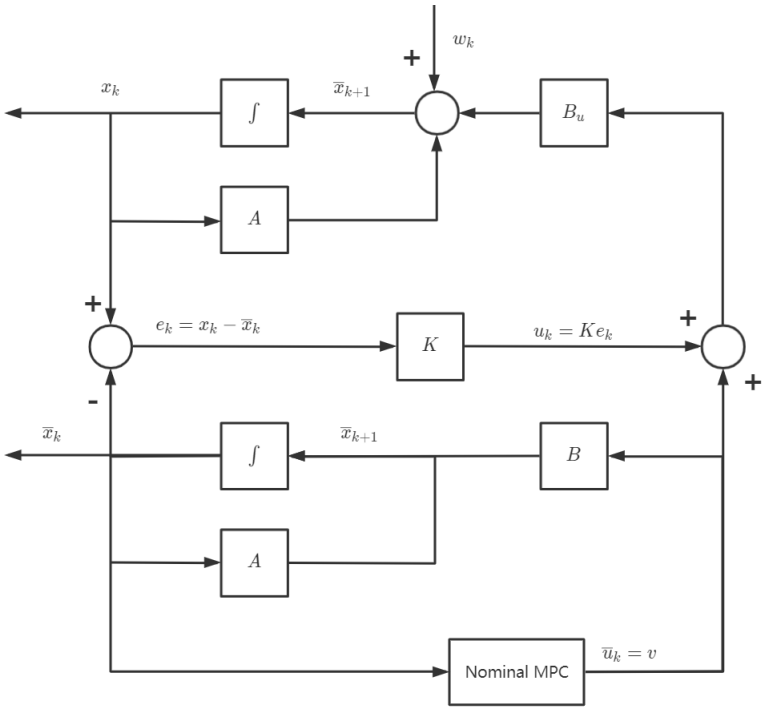


Figure 7.1: Tube-based MPC scheme

algorithm of the design of Tube-based MPC controller can be concluded as:

Algorithm 3: Tube-based MPC with LC-RCI set

Offline Algorithm:

Compute a LC-RCI set with **Algorithm 2** and get its controller $u = Kx$.

Online Algorithm:

Step1: Define current iteration flag $i = 1$ and maximum iteration number $iter$.

Step2: Design a controller for nominal system by solving QP online $\overline{u} = v$. Compute

the error between real state and nominal state $e_k = x_k - \overline{x_k}$, and combine the two controllers as $u^\star = \overline{u}_k + u_k = Ke_k + v$. Update the current state.

If $i < iter$, go to **step2**, otherwise, go to **step3**.

Step3: End algorithm

7.3 Numerical Case study for Tube-based MPC

Given the same numerical case as before, the trajectory of the Tube-based MPC is shown in Fig 7.2. The actual states are shown in blue-filled square points, and the blue dash line is the trajectory of the actual system. The nominal states are shown in red-filled circle points, and the red dash line is the trajectory of the nominal system. The parallelograms surrounded by black lines and filled with light blue/purple are LC-RCI sets computed offline. With the zoomed-in figure shown at the right corner, it is more straightforward

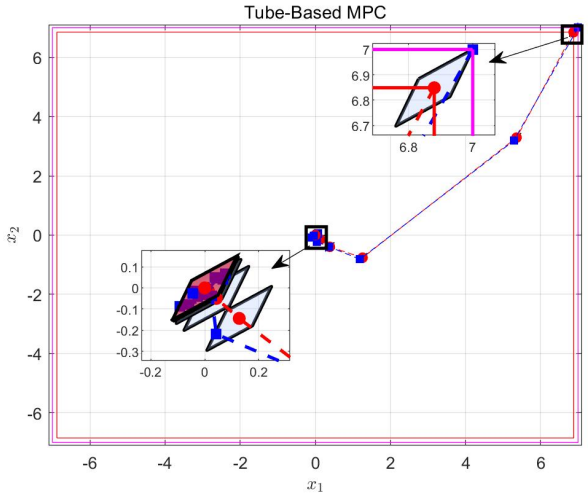


Figure 7.2: Trajectory of Tube-based MPC

for us to see that the initial state of the existing system is defined at the corner of an LC-RCI set, and the center of the set is defined as the initial state of nominal system. Therefore, the area with fuchsia/rose color borders is the original feasible region for the actual system, and the area with red borders is a tighter feasible region for the nominal system.

From another zoomed-in figure at the center, we can see that we can control the actual state to the origin by controlling the whole LC-RCI set whose center is the nominal system and guarantee the actual state will not exceed each LC-RCI set. A zoomed-in figure for

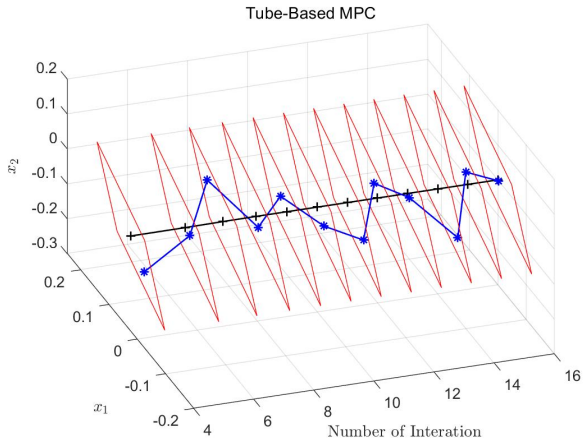


Figure 7.3: Trajectory of Tube-based MPC in 3-D

iteration 5 to 15 in 3-D view is shown in Fig 7.3. We can see that a sequence of LC-RCI sets look like a tube, and the motion of the actual states will not exceed this tube.

In Fig7.4, it is observed that with the designed controller, the cost of the natural system follows the nominal one, and both quickly decrease to zero in 2-3 iterations. The code for Tube-based MPC can be found by the URL in Appendix, named as `Tube_based_MPC_final.m`.

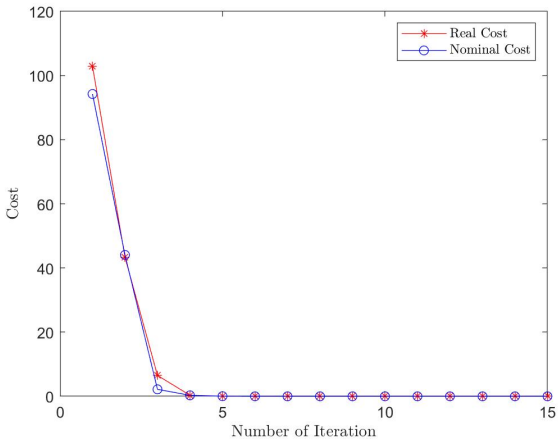


Figure 7.4: Cost of Tube-based MPC

Chapter 8

Comparison and Conclusion

THE performance for 4 methods introduced, which are $\#CRMPC$(short for RMPC with Causal Feedback Control Law), $\#CRMPC_S$(short for $\#CRMPC$ with Single LMI Method), $\#CRMPC_L$(short for $\#CRMPC_S$ with Offline Lookup Table Method) and $\#TBMPC$(short for Tube-based MPC), are analyzed and assessed, and the future work for this for project is also proposed.

In section 8.1 a comprehensive comparison for 4 methods introduced before in terms of online computation time, optimality(cost) and conservatism on the same numerical case is illustrated.

In section 8.2, 3 ideas about future work based on this project are proposed.

8.1 Comprehensive Comparisons

In this section, simulations with 4 methods are performed on the same numerical case introduced before using the CVX package with Mosek Solver, in MATLAB R2021a on a laptop with AMD Ryzen 7 4800U CPU with Radeon Graphics(1800 MHz) and 16.0 GB memory.

For the convenience of reading, the numerical case is described again in the following. The

unstable DTLI system is given as

$$A = \begin{bmatrix} 1 & 0.8 \\ 0.5 & 1 \end{bmatrix}, \quad B_u = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad B_w = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}.$$

The system is influenced by bounded disturbance as

$$w_i \in \mathcal{W} := \{w \in \mathbb{R} : -1 \leq w \leq 1\}, \quad i \in \{0, 1, 2, \cdots, N-1\}$$

The constraint imposed on state and input of the system is given as

$$\begin{bmatrix} -7 \\ -7 \\ -7 \end{bmatrix} \leq \begin{bmatrix} x_k \\ u_i \end{bmatrix} \leq \begin{bmatrix} 7 \\ 7 \\ 7 \end{bmatrix}, \quad \begin{array}{l} k \in \{0, 1, 2, \cdots, N\}, \\ i \in \{0, 1, 2, \cdots, N-1\}. \end{array}$$

The cost function is given as

$$\begin{aligned} J &= x_N^T P x_N + \sum_{k=0}^{N-1} (x_i^T Q_i x_i + u_i^T R_i u_i) \\ &= x^T \tilde{Q} x + u^T \tilde{R} u \end{aligned}, \qquad i \in \{0, 1, 2, \cdots, N-1\}.$$

Where penalty matrices are given by

$$Q_i = Q = P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R_i = R = 1, \quad i \in \{0, 1, 2, \cdots, N-1\}.$$

8.1.1 Online Computation Time

The first indicator for the performance of designed controllers to be evaluated is the online computation time, which is the leading property that is cared for most by this project. Run simulations of 4 methods one by one and use `'tic-toc'` to record their online computation time($\#TBMPC$ will not use CVX toolbox online, so we can not compare time in terms of CPU time of CVX toolbox). Simulate on the system with prediction horizon $N = 5$, $N = 10$ and $N = 15$ respectively, and the numerical results are shown in Table 8.1.

Table 8.1: Online computing time of 4 methods

Method	Online time N=5	Online time N=10	Online time N=15
$CRMPC$	15.8301s	30.8313s	66.9169s
$CRMPC_S$	5.8865s	8.939s	21.4306s
$CRMPC_L$	5.2084s	5.6659s	6.1977s
$TBMPC$	0.1727s	0.1744s	0.1777s

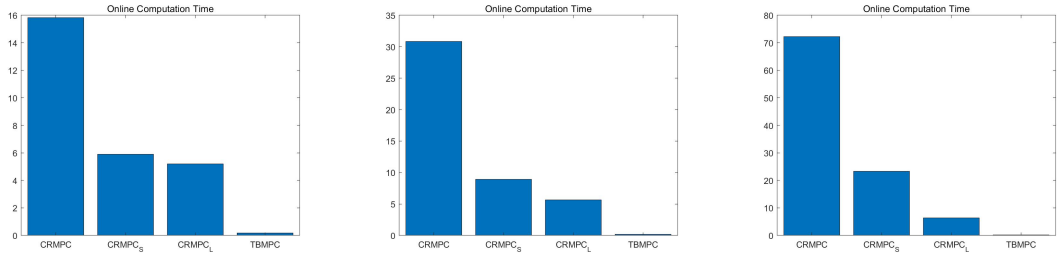
$CRMPC$: Robust Model Predictive Control with Causal Feedback Control Law

$CRMPC_S$: $CRMPC$ with Single Linear Matrix Inequality Method

$CRMPC_L$: $CRMPC_S$ with Offline Lookup Table Method

$TBMPC$: Tube-based Model Predictive Control

The results in Table 8.1 can be more clearly shown in bar graphs in Fig. 8.1, where it is observed that the online computation time is decreasing one by one. The gaps between methods increase by the length of prediction horizon N, which means the reduction of computation time goes more obvious.



(a) Code time N=5 (b) Code time N=10 (c) Code time N=15

Figure 8.1: Code Time of 4 Methods

The reduction of computation time between two specific methods can be shown more clearly in Table. 8.2. Where we can see that with the application of a single LMI method, the online computation time can be reduced by 62.81% against the original method without using it when $N = 5$, and the reduction increase to around 70% when $N = 10$ and $N = 15$, which has a generally good performance in reducing online computation time.

With the offline lookup table application based on $\#CRMPC_S$, it can be seen that the time reduction at $N = 5$ is only 11.52%, which is just a tiny improvement. However, it goes to 36.63% and 71.08% when $N = 10$ and $N = 15$ respectively, which shows that the offline lookup table method is suitable for the RMPC scheme with a long prediction horizon.

Last but not least, for the non-LMI-based method $\#TBMPC$, because the problem has been transformed from solving online SDP into solving online QP, the online computational burden has been highly reduced. As shown in the Table8.2, the reduction rates can reach over 96% compared with the first three LMI-based methods.

Table 8.2: Online Computing Time Reduction

N=5 Method / Method	$CRMPC$	$CRMPC_S$	$CRMPC_L$	$TBMPC$
$CRMPC$	0%	*	*	*
$CRMPC_S$	62.81%	0%	*	*
$CRMPC_L$	67.10%	11.52%	0%	*
$TBMPC$	98.91%	97.07%	96.68%	0%

N=10 Method / Method	$CRMPC$	$CRMPC_S$	$CRMPC_L$	$TBMPC$
$CRMPC$	0%	*	*	*
$CRMPC_S$	71.01%	0%	*	*
$CRMPC_L$	81.62%	36.62%	0%	*
$TBMPC$	99.43%	98.05%	96.92%	0%

N=15 Method / Method	$CRMPC$	$CRMPC_S$	$CRMPC_L$	$TBMPC$
$CRMPC$	0%	*	*	*
$CRMPC_S$	67.97%	0%	*	*
$CRMPC_L$	90.74%	71.08%	0%	*
$TBMPC$	99.73%	99.1%	97.13%	0%

To stage summary in terms of online computation time reduction, using $\#CRMPC_S$ is generally good and the performance of $\#CRMPC_L$ is more suitable for long prediction horizon scheme while $\#TBMPC$ has the best performance.

8.1.2 Optimality

In this section, the state trajectory, input signal, and cost(optimality) of the 4 methods are shown and analyzed as follows.

The simulation result for state trajectories of 4 methods is shown in Fig 8.2. The red line with red star markers represents the trajectory and states of $\#CRMPC$, the blue line with blue circle markers represent the trajectory and states of $\#CRMPC_S$, the black line with black diamond markers represent the trajectory and states of $\#CRMPC_L$ and green line with green square markers represent the trajectory and states of $\#TBMPC$. The region surrounded with the blue line is the feasible region(satisfy state constraints).

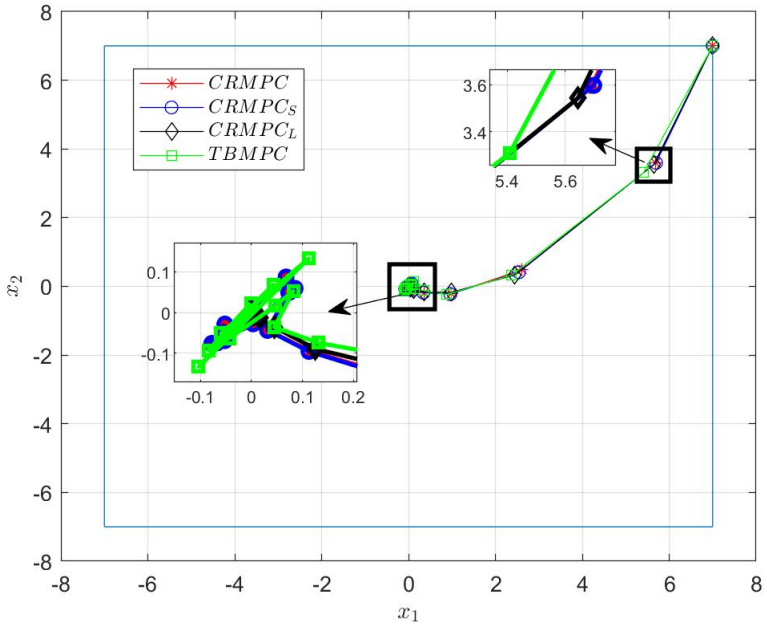


Figure 8.2: Trajectory for 4 Methods

The input signals of designed controllers for four methods respectively are shown in Fig. 8.3, and the optimality in the form of cost signal, which is calculated with actual states and inputs for four methods are shown in Fig. 8.4. All the states satisfy the state constraints, and the system can be controlled to the origin point from the initial state at

the boundary.

With the help of two zoomed-in pictures, we can observe that the trajectories and states of $\#CRMPC$ and $\#CRMPC_S$ are very similar because the scheme of the problem to be solved is nearly the same. The only difference might have resulted from computation error when using a single LMI method, introducing more variables into the program. The effectiveness of using sufficient conditions is not evident in this case.

The trajectory and states for $\#CRMPC_L$ at the first several steps have slight differences compared with the first two LMI-based methods. After all, three controller parameters are not computed simultaneously, which will more or less introduce conservatism into the performance because the three parameters are not well balanced and optimized when solving the SDP program in theoretical analysis. However, the optimality of using $\#CRMPC_L$ seems improved instead of decreased in this case(which will also be shown in Fig. 5.3).

At last, the trajectory and states for $\#TBMPC$ are very different from another 3 LMI-based methods' because it solves online QP without considering uncertain parts.

The optimality comparisons will be further discussed in the following.

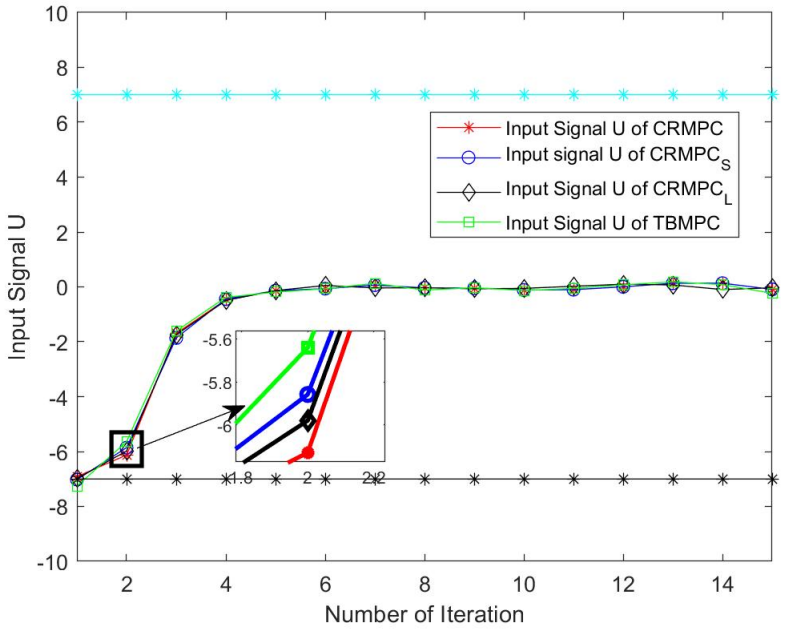


Figure 8.3: Input Signal for 4 Methods

In Fig. 8.3, the horizontal black line with black star markers and the cyan/bright blue line

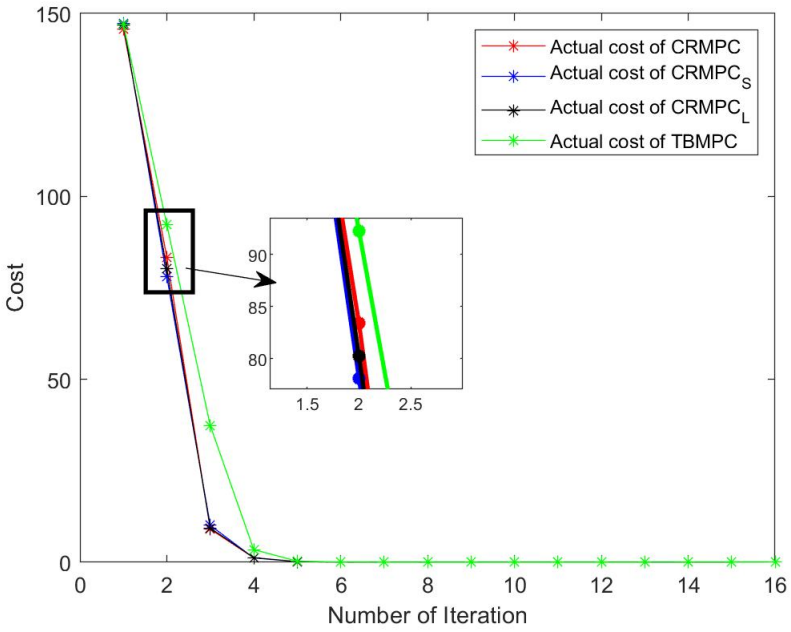


Figure 8.4: Optimality(Cost) for 4 Methods

with cyan/bright markers are the lower and upper bounds of the input signal, respectively, and the area inside is the feasible region for the input signal. It is observed that the input signals of the 4 methods all satisfy the input constraint. It is also noted that the positive and negative signs for input signal mean the direction instead of value.

Combined with Fig. 8.4 we can observe that the costs of three LMI-based methods are nearly the same while the cost of tube-based MPC is not minimized as quickly as theirs because it has the minor input signal in the first several steps. This phenomenon mainly results from the conservatism of $\#TBMPC$ for the nominal system is solved with tighter constraints.

To conclude, the optimality signals for 3 LMI-based methods are similar and that for tube-based MPC is the worst, but it is still acceptable in this case. The code for the above can be found by the URL in Appendix, named as `Compare_4_methods.m`.

8.2 Future Work

This project introduces, proposes, and illustrates four different methods in detail and make comparisons among them based on a simple numerical case, and some ideas can still improve it as following:

Idea 1 The comparisons for four methods are based on a simple system scheme, which is ideal. It would be beneficial if the system could be extended to higher dimensions.

Idea 2 The project only focuses on addictive bounded disturbance. At the same time, some natural systems are also influenced by structural uncertainty, which can be further investigated and compared with according methods proposed.

Idea 3 The algorithms and methods introduced and proposed in this project are only analyzed on the theoretical level, and they require being realized and verified on specific real cases in some proper ways to figure out whether those methods are usable or not, which is very important for an engineer. Moreover, a lack of this is the main drawback of this project.

Bibliography

[1] D. Hrovat, S. Di Cairano, H. E. Tseng, and I. V. Kolmanovsky, "The development of model predictive control in automotive industry: A survey," in *2012 IEEE International Conference on Control Applications*. IEEE, 2012, pp. 295–302.

[2] Y.-G. Xi, D.-W. Li, and S. Lin, "Model predictive control status and challenges," *Acta Automatica Sinica*, vol. 39, no. 3, pp. 222–236, 2013.

[3] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.

[4] A. Georgiou, F. Tahir, and S. A. Evangelou, "Computationally efficient robust model predictive control for uncertain system using causal state-feedback parameterization," *IEEE Transactions on Automatic Control*, 2021.

[5] F. Blanchini, "Control synthesis for discrete time systems with control and state bounds in the presence of disturbances," *Journal of optimization theory and applications*, vol. 65, no. 1, pp. 29–40, 1990.

[6] Y. I. Lee, B. Kouvaritakis, and M. Cannon, "Constrained receding horizon predictive control for nonlinear systems," *Automatica*, vol. 38, no. 12, pp. 2093–2102, 2002.

[7] D. Q. Mayne, M. M. Seron, and S. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[8] S. Raković, E. Kerrigan, K. Kouramas, and D. Mayne, *Invariant approximations of robustly positively invariant sets for constrained linear discrete-time systems subject*

to bounded disturbances. University of Cambridge, Department of Engineering Cambridge, 2004.

[9] D. Limón, I. Alvarado, T. Alamo, and E. F. Camacho, "Robust tube-based mpc for tracking of constrained linear systems with additive disturbances," *Journal of Process Control*, vol. 20, no. 3, pp. 248–260, 2010.

[10] C. Helmberg, "Semidefinite programming," *European Journal of Operational Research*, vol. 137, no. 3, pp. 461–482, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221701001436

[11] F. Tahir and I. M. Jaimoukha, "Low-complexity polytopic invariant sets for linear systems subject to norm-bounded uncertainty," *IEEE Transactions on Automatic Control*, vol. 60, no. 5, pp. 1416–1421, 2015.

[12] F. Tahir and I.M. Jaimoukha, "Causal state-feedback parameterizations in robust model predictive control," *Automatica*, vol. 49, no. 9, pp. 2675–2682, 2013.

[13] L.-T. Xie, L. Xie, and H.-Y. Su, "A comparative study on algorithms of robust and stochastic mpc for uncertain systems," *Acta Automatica Sinica*, vol. 43, no. zdhxb-43-6-969, p. 969, 2017.

[14] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[15] C.-Y. Liu, "Robust model predictive control: Robust control invariant sets and efficient implementation," Ph.D. dissertation, Department of Electrical and Electronic Engineering, Imperial College London, 2017.

[16] M. Cannon, V. Deshmukh, and B. Kouvaritakis, "Nonlinear model predictive control with polytopic invariant sets," *Automatica*, vol. 39, no. 8, pp. 1487–1494, 2003.

Appendix A

Code for Project

The code for this project can be find at the following website.

https://github.com/Ethani97/Approximate-robust-model-predictive-control-design-using-off-line-computational-methods.git