INSA de Lyon

Master Thesis

Receding Horizon Optimal Control

Author:
Jie FAN

Supervisor:
Prof. ChengZhong XU

A thesis submitted in fulfillment of the requirements for the degree of Master

Lab. Ampere
Lab. LAGEP

June 2016

INSA de Lyon

Abstract

Lab. Ampere
Lab. LAGEP

Master

Receding Horizon Optimal Control

by Jie FAN

Receding Horizon Optimal Control (RHOC) is widely applied in the process industry. And RHOC is also the technique that is able to consider and handle model constraints and has no specif request to mathematic model. Even though almost real systems have nonlinear and dynamic feature, taking computational complex into consideration, most of RHOC applications are always used based on linear model.

In this thesis, we principally focus on the basic Receding Horizon Optimal Control strategy whose cost function respect state and input. We are interested in controller by solving a open-loop optimal problem. We especially give a dual-mode controller. The feasibility of this open-loop optimal problem and the stability of closed-loop controlled system over infinite horizon are proven.

Furthermore, we instantiate the basic theory to apply in the reference trajectory tracking problem, in consideration of disturbance and different constraints. What's more, we modifier the cost function including output, set reference and increment inputs. After that, the optimal problem have been transformed into the corresponding formulation of QP problem.

Several examples have been implemented that confirm a stable behavior of the controlled system with satisfactory results. From the experiments, RHOC algorithm is developed which is fast enough and all constraints are well satisfied.

Contents

Abstract i

List of Figures iv

List of Tables v

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Literature Review . . . 2
- 1.3 Organization of this Thesis . . . 2

2 RHOC for Linear System 4
- 2.1 RHOC for Continuous System . . . 4
  - 2.1.1 Problem Setup . . . 4
- 2.2 RHOC over Finite Horizon . . . 4
  - 2.2.1 Prediction Model . . . 4
  - 2.2.2 Open-loop Optimization without Constraints . . . 5
  - 2.2.3 Open-loop Optimization under Constraints . . . 7
- 2.3 RHOC over Infinite Horizon . . . 8
  - 2.3.1 Dual-Controller . . . 8
  - 2.3.2 Transform Horizon from Infinite to Finite . . . 9
  - 2.3.3 Closed-loop Properties . . . 10

3 Reference Tracking Linear RHOC 14
- 3.1 Nominal Problem Setup . . . 14
- 3.2 Regularization . . . 16
- 3.3 Disturbance . . . 17
- 3.4 Constraints . . . 19
  - 3.4.1 Input Constraints . . . 19
  - 3.4.2 Input Increment Constraints . . . 19
  - 3.4.3 Output Constraints . . . 21

4 Implentation 23
- 4.1 Illustrative Example 1 . . . 23
  - 4.1.1 Problem Setup . . . 23
  - 4.1.2 Numerical Results . . . 24
- 4.2 Illustrative Example 2 . . . 27

4.2.1 Problem Setup 27
4.2.2 Linearization 30
4.2.3 Numerical Results 30

5 Conclusion and Perspective **34**

Bibliography **35**

List of Figures

4.1 The evolution of each system state 25
4.2 The evolution of predictive inputs 25
4.3 The evolution of predictive outputs 26
4.4 Change of matrix Q_y. Column1: $Q_y = diag[0.0001\ 0.0001]$. Column2: $Q_y = diag[100\ 100]$. 27
4.5 Change of matrix S. Column1: $S = diag[100\ 100]$. Column2: $S = diag[0.001\ 0.001]$. 28
4.6 System of four tanks 28
4.7 Ex2: The evolution of each system state 31
4.8 Ex2: The evolution of predictive inputs 31
4.9 Ex2: The evolution of predictive outputs 32
4.10 Column1: disturbance and non feed-forward. Column2: disturbance and feed-forward. 33

List of Tables

4.1 Influence of prediction horizon 25
4.2 Influence of sampling time 25
4.3 Parameter values of Example 2 29

Chapter 1

Introduction

1.1 Motivation

Receding Horizon Optimal Control (RHOC) , or Model Predictive Control (MPC), becomes a kind of more and more popular control strategy in recent years which has been widely discussed in the domain of control.

In short, RHOC is an optimal control approach which uses an explicit mathematic model of the process to predictive and compute the control. The basic idea of RHOC [Mayne et al., 2000] requests that (1) at each sampling instant, solving a finite horizon open-loop optimal problem online by minimizer a cost function with the current state; (2) and obtaining an optimal predicted control sequence, the first element is applied; (3) at the next sampling time, the above-described process is repeated using the new measurement.

RHOC is wildly applied in various domains, especially in industry, and that is summarized in a very thoughtful survey paper [Qin and Badgwell, 2003]. From the formulation of RHOC, we conclude certain features as follows: RHOC allows various flexible plant models for prediction and do not need to understand the internal mechanism, for example, multi-variable, linear, nonlinear, discrete and continue-time models; it can systematically and explicitly handle the state and control constraints which is crucial for real physical system; the stability of controlled system is guaranteed by the choice of terminal cost and terminal set; system states must be measured before the prediction is implemented. However, Since online solving optimization problems, it will increase the computational complexity.

1.2 Literature Review

In [Morari and Lee, 1999], it was stated that the ideas of RHOC occurred in the 1960s, however, a real interest in this area started to sharply increase only in the 1980s after publication of the first paper on Dynamic Matrix Control (DMC) by Cutler and Ramaker in 1979. From then till now, the research on theory and application of RHOC have achieved many results.

A number of excellent literatures and papers have emerged, such as: [Ricker, 1993] [Bemporad et al., 2002] explain that, in absence of constraints, an infinite horizon problem can be regarded as a LQ control problem which offers a simple solution to unconstrained linear system. When taking constraints into consideration, [Rawlings and Muske, 1993] have shown that the receding horizon scheme is a powerful proof to guarantee the stability of closed-loop stable or instable system over finite or infinite horizon. In [Mayne et al., 2000], Mayne details the stability and optimality for constrained RHOC.

Furthermore, RHOC strategy is also used for nonlinear systems. [CHEN and ALLGWER, 1998] propose a kind quasi-infinite horizon scheme with input constraints to ensure the asymptotic stability by choosing an appropriate terminal matrix. Similarly, [Jadbabaie et al., 2001] provide a complete analysis of the stability and give the safe region which is based on a finite horizon optimal problem with a terminal cost.

Last few years, non-squared integrand function is regarded as cost function for constrained nonlinear systems in [Liu, 2014], the author discusses the proof of Recursive Feasibility and stability for robust RHOC based on the idea of [CHEN and ALLGWER, 1998].

Besides ordinary differential systems, RHOC is also applied to partial differential systems. [Pham, 2012] extends the RHOC method to infinite dimension systems, especially gives the stability proof of 2×2 hyperbolic system.

1.3 Organization of this Thesis

This thesis is organized as follows. The motivation and background of RHOC is shortly reviewed in Chapter 1. In Chapter 2, the basic theory of RHOC for linear system theory is reviewed, including RHOC over finite horizon and over infinite horizon, the dual-mode controller and the stability over infinite horizon are given. Based on the theory in Chapter 2, by starting with RHOC on basic form, the reference tracking problem is extended step by step in chapter 3. Also, different constraints (i.e. input constraints, input increment constraints and output constraints)and these corresponding

explicit controllers are described. The Chapter 4 is the implement of RHOC in matlab by illustrating two examples. Finally, in chapter 5, we conclude this thesis and describe some future work.

Chapter 2

RHOC for Linear System

2.1 RHOC for Continuous System

2.1.1 Problem Setup

The class of linear system to be controlled is described by the following ODE:

$$\begin{aligned} \dot{x} &= Ax(t) + Bu(t) \\ x(0) &= x_0 \\ s.t.\ \ u(t) &\in U,\ \forall t \geqslant 0 \end{aligned} \tag{2.1}$$

where state vector is denoted by $x(t) \in \mathcal{R}^n$ and input vector is denoted by $u(t) \in \mathcal{R}^m$. In this chapter, we just consider the representation in state-space and time-invariant form, since lots of analysis are related to Lyapunov framework. And we analyze RHOC problem in two parts: over finite horizon and over infinite horizon.

2.2 RHOC over Finite Horizon

2.2.1 Prediction Model

We always use a dynamic mathematic model to present the future response of the controlled system. For a linear system, we should convert it from continuous time to discrete time, so the first system we study is given by:

$$\begin{cases} x(k+i+1|k) = Ax(k+i|k) + Bu(k+i|k) & i = 0, 1, 2, \dots \\ x(k|k) = x(k) \end{cases} \tag{2.2}$$

where $x(k+i|k)$ and $u(k+i|k)$ denote the $k+i$ th step state and input which are predicted at time k, respectively. A and B are constant matrices. $x(k|k)$ presents initial condition. Generally, the predictive input and state sequences are formulated by

$$\underline{\mathbf{u}}(k) = \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \cdots \\ u(k+N-1|k) \end{bmatrix} \qquad \underline{\mathbf{x}}(k) = \begin{bmatrix} x(k+1|k) \\ x(k+2|k) \\ \cdots \\ x(k+N|k) \end{bmatrix} \tag{2.3}$$

And we can obtain certain equations yielded by Eq. 2.2 and 2.3 :

$$\begin{aligned} x(k|k) &= x(k) \\ x(k+1|k) &= Ax(k) + Bu(k|k) \\ x(k+2|k) &= Ax(k+1|k) + Bu(k+1|k) \\ &= A^2x(k) + ABu(k|k) + Bu(k+1|k) \\ &\vdots \end{aligned} \tag{2.4}$$

Summing these equalities in 2.4, we rewrite them as the formulation below:

$$\begin{aligned} x(k+i|k) &= A^i x(k) + M_2^i u(k) \qquad i = 0, 1, \cdots, N \\ \underline{\mathbf{x}}(k) &= M_1 x(k) + M_2 \underline{\mathbf{u}}(k) \end{aligned} \tag{2.5}$$

here,

$$M_1 = \begin{bmatrix} A \\ A^2 \\ \cdots \\ A^N \end{bmatrix}, \quad M_2 = \begin{bmatrix} B & 0 & 0 \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}$$

For open-loop optimization problem, already in [Bemporad et al., 2002], it was realized that in the unconstrained case, the solution is given by a linear quadratic(LQ) controller, and in the constraint case[GARCIA and MORSHEDI, 1986], the optimization problem is a quadratic programming (QP), i.e. minimization a cost function while taking into account linear constraints.

2.2.2 Open-loop Optimization without Constraints

For the predicted system, we define a performance cost function in terms of predicted state sequence $\underline{\mathbf{x}}(k)$ and predicted input sequence $\underline{\mathbf{u}}(k)$. In general, the quadratic finite

cost is used:

$$\begin{aligned}
&\underline{\mathbf{u}}^*(k) = \arg\min_{\underline{\mathbf{u}}} J(k) \\
&J(k) = \sum_{i=0}^{N-1} [x^T(k+i|k)Qx(k+i|k) + u^T(k+i|k)Ru(k+i|k)] + x^T(k+N|k)Px(k+N|k)
\end{aligned} \tag{2.6}$$

where the meaning of $x(k+i|k)$ and $u(k+i|k)$ are similar in 2.2. N is prediction horizon. Q and R denote positive definite, symmetric weighting matrices. P is a positive definite terminal matrix corresponding to the terminal state $x(k+N|k)$. For linear system, the predicted state linearly depends on the predicted inputs. Thus we can rewrite the cost function in 2.6 as a quadratic cost function in terms of $\underline{\mathbf{u}}(k)$ which is realized by Eq. 2.5.

$$J(k) = \underline{\mathbf{u}}^T(k)E\underline{\mathbf{u}}(k) + 2x^T(k)F\underline{\mathbf{u}} + x^T(k)Gx(k) \tag{2.7}$$

where

$$\begin{aligned}
E &= M_2^T P M_2 + \tilde{R} \\
F &= M_2^T \tilde{Q} M_1 \\
G &= M_1^T Q M_1 + Q
\end{aligned} \tag{2.8}$$

with $\tilde{Q} = diag(Q, \cdots, Q, P)$, and $\tilde{R} = diag(R, \cdots, R)$. From the above, we are able to off-line compute the matrices E, F and G. To solve the optimal problem in absence of constraints, we consider the gradient of $J(k)$ with respect to $\underline{\mathbf{u}}(k)$ to calculate the feedback solution.

$$\nabla_u J = 2E\underline{\mathbf{u}}(k) + 2Fx \tag{2.9}$$

And while E is nonsingular, the optimal predicted sequence $\underline{\mathbf{u}}^*$ is unique given by $\underline{\mathbf{u}}^*(k) = -E^{-1}Fx(k)$. After this, we implement the first element of it at each sampling time k, we call this as receding horizon control law. Due to matrices E and F are constant, this is also a linear time-invariant sate feedback controller. In the single-input case, we just take the first row as the gain matrix:

$$\begin{aligned}
u(k) &= u^*(k|k) = K_N x(k) \\
K_N &= -[I_{n_u}\ 0\ \cdots\ 0]E^{-1}F
\end{aligned} \tag{2.10}$$

2.2.3 Open-loop Optimization under Constraints

Generally, we should consider some constraints to solve the actual optimal problem, since almost all real system are under constraints. As we all know, constrains are classified as hard/soft constraint, input/state constraint and equality/inequality constraint etc. according to different standards. **Hard/Soft Constraint** If the infeasibility of a problem needs to be avoided, hard constraint must be satisfied but soft constraint sometimes may be violated.

Input/State Constraint The input and state constraint can be represented as the following form:

$$\begin{aligned} u_{min} \leq u(k) \leq u_{max} \\ x_{min} \leq x(k) \leq x_{max} \end{aligned} \tag{2.11}$$

here, the subscript max and min correspond the upper boundary and the lower boundary. Note that we just consider linear constraints in this thesis. For RHOC problem, we have transformed the quadratic cost function as a function of predicted inputs, so we should also rewrite the form of constraints as:

$$A_c u \leq b_c \tag{2.12}$$

i.e.

$$\begin{bmatrix} I \\ -I \end{bmatrix} \underline{\mathbf{u}}(k) \leq \begin{bmatrix} \bar{\mathbf{I}} u_{max} \\ -\bar{\mathbf{I}} u_{min} \end{bmatrix} \tag{2.13}$$

Here, if the input is single, $\bar{\mathbf{I}}$ is a vector of ones and if the input has dimension n_u, it is then a matrix (i.e. $\bar{\mathbf{I}} = [I_{n_u}\ I_{n_u}\ \cdots\ I_{n_u}]$). Similarly, we can transform state constraints in this form by using 2.13 and Eq. 2.5 :

$$\begin{bmatrix} {M_2}^i \\ -{M_2}^i \end{bmatrix} u(k) \leq \begin{bmatrix} x_{max} \\ -x_{min} \end{bmatrix} + \begin{bmatrix} -A^i \\ A^i \end{bmatrix} x(k) \qquad i = 1, 2, \cdots, N \tag{2.14}$$

Also, we can definite:

$$B_c \underline{\mathbf{u}} \leq b_0 + A_x x(k) \tag{2.15}$$

Therefore, combining this constraint above and the cost function 2.7, we can reset the optimal problem just involving the predicted inputs:

$$\begin{aligned} \min_{\underline{\mathbf{u}}} \ & \frac{1}{2}\ \underline{\mathbf{u}}^T E \underline{\mathbf{u}} + x^T(k) F^T \underline{\mathbf{u}} \\ s.t \quad & B_c \underline{\mathbf{u}} \leq b_0 + A_x x(k) \\ & E = M_2^T P M_2 + \tilde{R} \\ & F = M_2^T \tilde{Q} M_1 \end{aligned} \tag{2.16}$$

Because of $E > 0$ and linear constraint, this optimal problem is a convex optimization problem whose solution is unique. To minimize the cost function of input prediction, the term $x^T G x$ is therefore omitted. In this way, the problem above is also a quadratic programming problem. In order to solve it, there are a lot of algorithms, for example, Active set algorithms and Interior point QP algorithms. A common feature of them is to use the theory of Lagrange multiplier method, and which is based on the first-order Karush-Kuhn-Tucker (KKT) optimality conditions [Boyd and Vandenberghe, 2004].

2.3 RHOC over Infinite Horizon

From some papers such as [POLAK and YANG, 1993][Droge and Egerstedt, 2011], we have known that the more length of prediction horizon is reduced, the more differences between predicted response and closed-loop response we have. Furthemore, [Ricker, 1993] told us that a finite horizon cannot guarantee the closed-loop stability. In order to avoid this problem, an infinite-horizon cost should be chosen. And its form is given:

$$J(k) = \sum_{i=0}^{\infty} [x^T(k+i|k) Q x(k+i|k) + u^T(k+i|k) R u(k+i|k)] \tag{2.17}$$

Here, we define a predicted input sequence over the infinite prediction horizon to hold the number of free variables finite in the optimization process and to solve the optimal problem by minimizing the infinite cost function.

2.3.1 Dual-Controller

To conveniently achieve the minimization of Eq. 2.17, a dual-mode prediction is considered (which will be proven to be optimal in the section 2.3.3 under several conditions):

$$u(k+i|k) = \begin{cases} u^*(k|k) & i = 0, 1, \cdots, N-1 \quad \text{(mode1)} \\ Kx(k+i|k) & i = N, N+1, \cdots \quad \text{(mode2)} \end{cases} \tag{2.18}$$

For this form prediction law, the input predictions are divided into two parts: *mode*1 and *mode*2. The *mode*1 shows that the predicted inputs are obtained by solving the open-loop optimal problem over an N-sampling horizon, and the *mode*2 shows a stabilizing state-feedback control law over the remaining horizon.

2.3.2 Transform Horizon from Infinite to Finite

In fact, for the infinite-horizon cost function Eq. 2.17, we can turn it into a finite-horizon cost function Eq. 2.6 by choosing an appropriate terminal weighting matrix P, and which will allow the cost function over the *mode*2 horizon to equal the terminal cost $x^T(k+N|k)Px(k+N|k)$.
In addition, we note that the matrix P is the solution of the Lyapunov equation:

$$P-(A+BK)^TP(A+BK)=Q+K^TRK \tag{2.19}$$

Now, to prove this transform, we firstly introduce the Theorem 2.1.

Theorem 2.1 (**Terminal weighting matrix**). *Along closed-loop trajectories of the model $x(k+1)=Ax(k)+Bu(k)$ under the feedback law $u(k)=Kx(k)$, the infinite horizon quadratic cost is given by:*

$$\sum_{i=0}^{\infty}[x^T(i)Qx(i)+u^T(i)Ru(i)]=x^T(i)Px(i) \tag{2.20}$$

where P satisfies the Eq. 2.19.

Proof. Given Eq. 2.19 left and right multiply the terms $x^T(i)$ and $x(i)$, and then set $V(x)=x^TPx$, according to those conditions in Theorem 2.1, we obtain:

$$V(x(i))-V(x(i+1))=x^T(i)Qx(i)+u^TQu(i)$$

Summing the equations in previous step from $i=0$ to ∞:

$$V(x(0))-\lim_{k\to\infty}V(x(k))=\sum_{i=0}^{\infty}[x^T(i)Qx(i)+u^T(i)Ru(i)]$$

We assume that $(A+BK)$ is stable, and then $V(x(k))=x^T(0)(A+BK)^{kT}P(A+BK)^kx(0)\to 0$ as $k\to\infty$. So we have it proved. □

According to the Theorem 2.1, we have:

$$\sum_{i=N}^{\infty}[x^T(i)Qx(i) + u^T(i)Ru(i)] = x^T(N)Px(N) \tag{2.21}$$

Replacing Eq. 2.17 by 2.21, we can obtain a finite cost function with the terminal cost Eq. 2.6.

2.3.3 Closed-loop Properties

This section focuses on the performance of the feedback controller in the closed-loop system. Without the disturbance, the modelling error and constraints, we can extend the prediction horizon from infinite horizon to finite horizon by choosing an appropriate terminal weighting matrix P Which has been proven by the previous section. Furthermore, this infinite horizon can ensure that the trajectories of predicted input and state are same as the actual's.

- **Lyapunov stability**

To prove the feedback control law that we have obtained makes the closed-loop system stable, we need to consider a positive definite Lyapunov function $V(x)$, and we assume that RHOC optimal problem is always feasible in this section (i.e. for all time k, there always exist the predicted states and inputs which satisfy the constraints). Furthermore, the definitions of equilibrium points and stable equilibrium are given so as to support the proof of Lyapunov stability.

Definition 2.2 (**Equilibrium Point**). x_0 is an equilibrium point of the system $x(k+1) = f(x(k))$ if and only if $f(x_0) = x_0$. As usual $x = 0$ is assumed to be the equilibrium of interest, and $f(0) = 0$.

Definition 2.3 (**Stable Equilibrium**). $x = 0$ is a stable equilibrium if, for all $k > 0$, the state $x(k)$ remains within an arbitrarily small neighborhood of $x = 0$ whenever the initial condition $x(0)$ is sufficiently close to $x = 0$. i.e. for all $R > 0$ there exists $r > 0$ such that:

$$\|x(0)\| < r \Longrightarrow \|x(k)\| < R \qquad \forall k > 0. \tag{2.22}$$

Theorem 2.4 (**Stability**). *The equilibrium $x = 0$ is stable if there exists a continuously differentiable scalar function $V(x)$ such that:*
(a) $V(x)$ is positive definite
(b) $V(f(x)) - V(x) \leq 0$

Theorem 2.5 (**Asymptotic Stability**). *The equilibrium $x = 0$ is asymptotic stable, which is equivalent if it is stable and, in addition, if there is a neighborhood V_0 of $x = 0$ such that $x(k) \longrightarrow 0$ as $k \longrightarrow \infty$, $\forall x(0) \in V_0$ and:*
(c) $V(x)$ is positive definite
(d) $V(f(x)) - V(x) \leq -l(x) \leq 0$
then $l(x(k)) \longrightarrow 0$ as $k \longrightarrow \infty$.

For the finite horizon case, we can easily prove its stability by considering closed-loop system poles, so we do not mention here. For the infinite horizon case, the cost function $J^*(k)$ is regarded as a Lyapunov function, since it is also a function of $x(k)$:

$$J^*(k) = V(x(k))$$

and the weighting matrices Q, R, P in the cost function are all non-negative, $J^*(k)$ is a positive definite function for all $x(k)$.

Lemma 2.6 (**Convergence of $J^*(k)$**). *If both of the following conditions hold:*
(e) the terminal weight P in Eq.2.6 is chosen so that $J(k)$ is equivalent to the infinite horizon cost Eq.2.17
(f) the optimal predicted input sequence computed at time k is feasible (i.e.satisfies constraints) for the optimization problem at time $k + 1$, then the optimal predicted cost is non-increasing and satisfies:

$$J^*(k+1) - J^*(k) \leq -[x^T(k)Qx(k) + u^T Ru(k)]$$

along closed-loop trajectories.

Proof. According to the condition (f), we definite an input sequence $\tilde{u}(k+1)$ at time $k+1$ which is suitable for the predicted inputs at time k because we assume that the optimal predicted input sequence computed at time k is feasible for the optimization problem at time $k+1$:
at time t:

$$\tilde{u}(k) = \{u^*(k|k), u^*(k+1|k), \cdots, u^*(k+N-1|k)\} \tag{2.23}$$

at time $t+1$:

$$\tilde{u}(k+1) = \{u^*(k+1|k), u^*(k+2|k), \cdots, K_N x^*(k+N|k)\} \tag{2.24}$$

Note that the last element of Eq. 2.24 is equal to $u^*(k+N|k)$ by using the state feedback control law over the horizon of $mode2$. By the condition (e), we have:

$$\begin{aligned}
\tilde{J}(K+1) =& \sum_{i=1}^{N}[x^T(k+i|k)Qx(k+i|k) + u^T(k+i|k)Ru(k+i|k)] + x^T(k+N+1|k) \\
& Px(k+N+1|k) \\
=& \sum_{i=1}^{\infty}[x^T(k+i|k)Qx(k+i|k) + u^T(k+i|k)Ru(k+i|k)] \\
=& \sum_{i=0}^{\infty}[x^T(k+i|k)Qx(k+i|k) + u^T(k+i|k)Ru(k+i|k)] - x^T(k|k)Qx(k|k) - \\
& u^T(k|k)Ru(k|k) \\
=& J^*(k) - [x^T(k|k)Qx(k|k) + u^T(k|k)Ru(k|k)]
\end{aligned}$$

And $J(k+1)$ is a suboptimal cost function at time $k+1$, therefore,

$$\tilde{J}(K+1) \geq J^*(k+1)$$

From the above, we obtain:

$$J^*(k+1) - J^*(k) \leq -[x^T(k|k)Qx(k|k) + u^T(k|k)Ru(k|k)]$$

□

By lemma 2.6 and theorem 2.5, and we know the matrices Q, R are positive definite symmetric, we can therefore know that the system is asymptotically stable over the infinite horizon around the equilibrium point by using the dual-mode control law.

- **Recursive feasibility**

From the proof of stability, we assume Eq. 2.24 yielded by Eq. 2.23 which satisfies constraints $u_{min} \leq u(k) \leq u_{max}$ at each time $i = 1, 2, \cdots$. So it is simple to know that the first $N-1$ elements of Eq. 2.24 at time $k+1$ satisfy constraints since they are equal to the control elements $u^*(k+1|k), \cdots, u^*(k+N-1|k)$ at time k. However, if the prediction input sequence at time $k+1$ also requires to satisfy constraints, the Nth element $u^*(k+N+1|k) = K_N x^*(k+N|k)$ must satisfy an extra constraint, and this constraint is called terminal constraint which guarantees the feasibility of Eq. 2.24 by applying a fixed $mode2$ feedback controller. Therefore, the terminal constraints are equivalent to the constraints on terminal state prediction. By convention we describe this safe region as the terminal constraint set as Ω. To guarantee the closed-loop stability of the controlled system and the existence of predicted inputs which depends on the

assumption of the definition Eq. 2.24 and condition (d) in the lemma 2.6, we should first of all decide to the terminal constraints set for recursive feasibility (i.e. satisfy the constraints). Consider the input and state constraints $u_{min} \leq u(k) \leq u_{max}$ and $x_{min} \leq x(k) \leq x_{max}$, we desire to ensure the N elements of Eq. 2.24 at time $k+1$, we must satisfy the following constrains:

$$x(k+|N) \in \Omega \Longrightarrow \begin{cases} x_{min} - \leq x(k+N|k) \leq x_{max} \\ u_{min} \leq Kx(k+N|k) \leq u_{max} \end{cases} \tag{2.25}$$

However, we must also make sure the predicted state $x(k+N+1k+1) \in \Omega$ yielded by Eq. 2.24 at time $k+1$ which is equivalent to $x(k+N+1k) \in \Omega$. Therefore, if the recursive feasibility is guaranteed, the necessary and sufficient condition 2.25 must be satisfied and this terminal constraints Ω is an invariant set:

$$x(k+|N) \in \Omega \Longrightarrow (A+BK)x(k+N|k) \in \Omega$$

From the above, we conclude that this dual-controller over infinite horizon can not only guarantee the system asymptotically stable but also make all states entre a safe region called terminal constraints set Ω. When the system states are out of this terminal constraint set, we use the mode1 controller and its goal is to all system states outside enter this set Ω.

Chapter 3

Reference Tracking Linear RHOC

So far the exposition has been interested solely in the theory of RHOC problem, the goal of which is to drive the state of controlled system to the origin or a safe region. Furthermore, we will develop this basic theory to overcome a specific problem (i.e. reference tracking problem)which is commonly referred to as set trajectory tracking.

However, we find that, on the one hand, the general theoretical results on the performance like stability, feasibility, and robustness for tracking an arbitrary set reference is extremely hard to obtain [Balandat, 2010], so in [Limon et al., 2016], the tracking periodic references is implemented. On the other side, many papers like [Limon et al., 2016] generally use the state-input form to express the cost function. As these reasons, we modifier the cost function by using the difference between output and a set reference and the changes of inputs, and while guarantee that input and output constraints are satisfied at each time.

Hence, the following sections in this chapter will step by step give the development from a nominal reference tracking problem by adding different limits and modifications, and we will give some implements in the next chapter.

3.1 Nominal Problem Setup

In this section, we will start from the basic model in the absence of constraints for the reference tracking problem. For this specific problem, our ideal goal is to make the minimal changes between actual output and set reference which can be done by

minimizing a cost function, it is set up as follows:

$$\begin{aligned} & min\, J(k) = \frac{1}{2}\sum_{i=1}^{N} \|y(k+i|k) - r(k+i|k)\|^2_{\mathbf{Q}_y} \\ s.t. \quad & x(k+i+1|k) = Ax(k+i|k) + Bu(k+i|k) \quad k = 0, 1, \cdots, N-1 \\ & y(k+i|k) = Cx(k+i|k) \qquad\qquad\qquad\quad k = 0, 1, \cdots, N \\ & x(k|k) = x(k) \end{aligned} \tag{3.1}$$

where $y(k)$ is the actual output, $r(k)$ is the constant reference. In cost function, when $k = 0$, there is no influence on output, the term $\|y(0) - r(0)\|^2_{\mathbf{Q}_y}$ is omitted. Using Eq. 2.5, it can be rewritten as:

$$\begin{aligned} \mathbf{Y} &= CM_1 x(k) + CM_2 \underline{\mathbf{u}}(k) \\ &= \mathbf{H}x(k) + \mathbf{G}\mathbf{U} \\ y(k|k) &= Cx(k|k) \end{aligned}$$

$$\mathbf{Y} = \begin{bmatrix} y(k+1|k) \\ y(k+2|k) \\ \vdots \\ y(k+N|k) \end{bmatrix}, \ \mathbf{H} = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix}, \mathbf{U} = \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+N-1|k) \end{bmatrix} \tag{3.2}$$

$$\mathbf{G} = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \cdots & CB \end{bmatrix}$$

Adding a vector $\mathbf{R}$ as the reference:

$$\mathbf{R} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix} \tag{3.3}$$

So, the cost function can be:

$$J(k) = \frac{1}{2}\sum_{k=1}^{N} \|y(k+i|k) - r(k+i|k)\|^2_{\mathbf{Q}_y} = \frac{1}{2}\|\mathbf{Y} - \mathbf{R}\|^2_{\mathbf{Q}_y} \tag{3.4}$$

where $\mathbf{Q}_y = diag\{Q_y, Q_y, \cdots, Q_y\}$. Combining Eq. 3.2 and Eq. 3.4, we have:

$$J(k) = \frac{1}{2}\|\mathbf{G}\mathbf{U} - (\mathbf{R} - \mathbf{H}x(k))\|^2_{\mathbf{Q}_y} = \frac{1}{2}\|\mathbf{G}\mathbf{U} - b\|^2_{\mathbf{Q}_y}, \qquad b = \mathbf{R} - \mathbf{H}x(k) \tag{3.5}$$

This can be seen as a QP problem which is easy to solve:

$$\begin{aligned} J(k) &= \frac{1}{2}\|\mathbf{GU} - b\|^2_{\mathbf{Q}_y} \\ &= \frac{1}{2}(\mathbf{GU} - b)^T\mathbf{Q}_y(\mathbf{GU} - b) \\ &= \frac{1}{2}\mathbf{U}^T\mathbf{G}^T\mathbf{Q}_y\mathbf{GU} - (\mathbf{G}^T\mathbf{Q}_y b)^T\mathbf{U} + \frac{1}{2}b^T\mathbf{Q}_y b \\ &= \frac{1}{2}\mathbf{U}^T H_Y\mathbf{U} + {g_Y}^T\mathbf{U} + \varphi \end{aligned} \tag{3.6}$$

where φ, H_Y and g_Y are given by:

$$\begin{aligned} \varphi &= \frac{1}{2}b^T\mathbf{Q}_y b \\ H_Y &= \mathbf{G}^T\mathbf{Q}_y\mathbf{G} \\ g_Y &= -\mathbf{G}^T\mathbf{Q}_y b \\ &= \mathbf{G}^T\mathbf{Q}_y\mathbf{H}x(k) - \mathbf{G}^T\mathbf{Q}_y\mathbf{R} \\ &= \Gamma_x x(k) + \Gamma_R\mathbf{R} \end{aligned} \tag{3.7}$$

φ should be discarded since it has no influence on minimizing this problem. Therefore, this QP problem is:

$$\begin{aligned} \min_{\mathbf{U}} J(k) &= \frac{1}{2}\mathbf{U}^T H_Y\mathbf{U} + {g_Y}^T\mathbf{U} \\ H_Y &= \mathbf{G}^T\mathbf{Q}_y\mathbf{G} \\ g_Y &= \Gamma_x x(k) + \Gamma_R\mathbf{R} \end{aligned} \tag{3.8}$$

3.2 Regularization

In practice, we sometimes need to make sure the difference between two continuous inputs smaller possible, which can obtain a slow changing input. For this, adding a new term to cost function $J_{\Delta u}$, and $\Delta u(k) = u(k) - u(k-1)$. So the control problem becomes:

$$\begin{aligned} min\, J_R(k) &= J(k) + J_{\Delta u} \\ &= \frac{1}{2}\sum_{i=1}^{N}\|y(k+i|k) - r(k+i|k)\|^2_{\mathbf{Q}_y} + \frac{1}{2}\sum_{i=0}^{N-1}\|\Delta u(k+i|k)\|^2_{\mathbf{S}} \\ s.t. \quad x(k+i+1|k) &= Ax(k+i|k) + Bu(k+I|k) \quad k = 0, 1, \cdots, N-1 \\ y(k+i|k) &= Cx(k+i|k) \qquad\qquad\qquad\qquad k = 0, 1, \cdots, N \\ x(k|k) &= x(k) \end{aligned} \tag{3.9}$$

Comparing with the basic control problem, we just consider the term $J_{\Delta u}$.

$$\begin{aligned} J_{\Delta u} &= \frac{1}{2}\sum_{i=0}^{N-1} \|\Delta u(k+i|k)\|_{\mathbf{S}}^2 \\ &= \frac{1}{2}\sum_{i=0}^{N-1} \|u(k+i|k) - u(k+i-1|k)\|_{\mathbf{S}}^2 \\ &= \frac{1}{2}\mathbf{U}^T H_S \mathbf{U} + (H_{u-1}u(k-1|k))^T\mathbf{U} + \frac{1}{2}u(k-1|k)\mathbf{S}u(k-1|k) \end{aligned} \tag{3.10}$$

where H_S and H_{u-1} are given by:

$$H_S = \begin{bmatrix} 2S & -S & & & \\ -S & 2S & -S & & \\ & \ddots & \ddots & \ddots & \\ & & -S & 2S & -S \\ & & & -S & S \end{bmatrix}, \quad H_{u-1} = -[S \quad 0 \quad 0 \quad \cdots \quad 0]^T \tag{3.11}$$

From the view of QP problem, $H_{\Delta u} = H_S$ and $g_{\Delta u} = H_{u-1}u(k-1|k)$, the term $\frac{1}{2}u(k-1|k)\mathbf{S}u(k-1|k)$ like φ is discarded. Therefore, this QP formulation is as following:

$$\begin{aligned} \min_{\mathbf{U}} J(k) &= \frac{1}{2}\mathbf{U}^T H \mathbf{U} + g^T\mathbf{U} \\ H &= H_Y + H_{\Delta u} = \mathbf{G}^T\mathbf{Q}_y\mathbf{G} + H_S \\ g &= g_Y + g_{\Delta u} = \Gamma_x x(k) + \Gamma_R\mathbf{R} + H_{u-1}u(k-1|k) \end{aligned} \tag{3.12}$$

3.3 Disturbance

Disturbance is a factor that can not be completely avoided for controlled systems. So the RHOC problem will be developed by adding a disturbance term in state which is described by:

$$\begin{aligned} min\, J_D(k) &= \frac{1}{2}\sum_{i=1}^{N} \|y(k+i|k) - r(k+i|k)\|_{\mathbf{Q}_y}^2 + \frac{1}{2}\sum_{i=0}^{N-1} \|\Delta u(k+i|k)\|_{\mathbf{S}}^2 \\ s.t. \quad x(k+i+1|k) &= Ax(k+i|k) + Bu(k+i|k) + Dd(k+i|k) \quad i = 0, 1, \cdots, N-1 \\ y(k+i|k) &= Cx(k+i|k) \quad i = 0, 1, \cdots, N \\ x(k|k) &= x(k) \end{aligned} \tag{3.13}$$

Comparing with the regularization problem, the unique difference is the term disturbance, we can easily obtain the following output:

$$\mathbf{Y}_D = \mathbf{H}x(k) + \mathbf{G}\mathbf{U} + \mathbf{E}_d D \tag{3.14}$$

where $\mathbf{H}$ and $\mathbf{G}$ are not changed, $\mathbf{E}_d$ and D are given by:

$$\mathbf{E}_d = \begin{bmatrix} CE & & & \\ CAE & CE & & \\ CA^2E & CAE & CE & \\ \vdots & \vdots & \ddots & \\ CA^NE & CA^{N-1}E & \cdots & CE \end{bmatrix}, D = \begin{bmatrix} d(k|k) \\ d(k+1|k) \\ \vdots \\ d(k+N-1|k) \end{bmatrix} \tag{3.15}$$

Since the output has changed, the cost function involving term $\mathbf{Y}_D$ should be:

$$\begin{aligned} J_d &= \frac{1}{2}\sum_{i=1}^{N} \|y(k+i|k) - r(k+i|k)\|^2_{\mathbf{Q}_y} \\ &= \frac{1}{2}\|\mathbf{Y}_D - R\|^2_{\mathbf{Q}_y} \\ &= \frac{1}{2}\|\mathbf{G}\mathbf{U} - (R - \mathbf{H}x(k) - \mathbf{E}_d D)\|^2_{\mathbf{Q}_y} \\ &= \frac{1}{2}\|\mathbf{G}\mathbf{U} - b_d\|^2_{\mathbf{Q}_y} \\ &= \frac{1}{2}\mathbf{U}^T\mathbf{G}^T\mathbf{Q}_y\mathbf{G}\mathbf{U} - (\mathbf{G}^T\mathbf{Q}_y b_d)^T\mathbf{U} + \frac{1}{2}{b_d}^T\mathbf{Q}_y b_d \\ &= \frac{1}{2}\mathbf{U}^T H_Y\mathbf{U} + {g_d}^T\mathbf{U} + \varphi_d \end{aligned} \tag{3.16}$$

where

$$\begin{aligned} b_d &= R - \mathbf{H}x(k) - \mathbf{E}_d D \\ H_Y &= \mathbf{G}^T\mathbf{Q}_y\mathbf{G} \\ g_d &= -\mathbf{G}^T\mathbf{Q}_y b_d \\ &= -\mathbf{G}^T\mathbf{Q}_y R - \mathbf{H}x(k) - \mathbf{E}_d D \\ &= \Gamma_x x(k) + \Gamma_R\mathbf{R} + \Gamma_D D \end{aligned} \tag{3.17}$$

This is also a QP problem, which is extended by 3.12:

$$\begin{aligned} \min_{\mathbf{U}} J(k) &= \frac{1}{2}\mathbf{U}^T H\mathbf{U} + g^T\mathbf{U} \\ H &= H_Y + H_{\Delta u} = \mathbf{G}^T\mathbf{Q}_y\mathbf{G} + H_S \\ g &= \Gamma_x x(k) + \Gamma_R\mathbf{R} + H_{u-1}u(k-1|k) + \Gamma_D D \end{aligned} \tag{3.18}$$

3.4 Constraints

Having derived RHOC, taking regularization and disturbance into consideration, we continue to extend it under some constraints. In general, there does not exist a physical system which has no boundaries or no limits. This section will show that input constraints, increment of input constraints and output constraints in detail. Note that all constraints are for each step time i.

3.4.1 Input Constraints

Input Constraints give the limits to the input at each step time i, and its form like this:

$$\begin{aligned} & min\, J_D(k) = \frac{1}{2}\sum_{i=1}^{N} \|y(k+i|k) - r(k+i|k)\|^2_{\mathbf{Q}_y} + \frac{1}{2}\sum_{i=0}^{N-1} \|\Delta u(k+i|k)\|^2_{\mathbf{S}} \\ s.t. \quad & x(k+i+1|k) = Ax(k+i|k) + Bu(k+i|k) + Dd(k+i|k) \quad i = 0, 1, \cdots, N-1 \\ & y(k+i|k) = Cx(k+i|k) \quad i = 0, 1, \cdots, N \\ & x(k|k) = x(k) \\ & u_{min} \leq u(k+i|k) \leq u_{max} \quad i = 0, 1, \cdots, N-1 \end{aligned} \tag{3.19}$$

To convert into QP formulation, we change this constraint in matrix form:

$$u_{min} \leq u(k+i|k) \leq u_{max} \Leftrightarrow \begin{bmatrix} u_{min} \\ u_{min} \\ \vdots \\ u_{min} \end{bmatrix} \leq \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+N-1|k) \end{bmatrix} \leq \begin{bmatrix} u_{max} \\ u_{max} \\ \vdots \\ u_{max} \end{bmatrix} \tag{3.20}$$

Hence, the QP problem is summarized:

$$\begin{aligned} & \min_{\mathbf{U}} J(k) = \frac{1}{2}\mathbf{U}^T H \mathbf{U} + g^T \mathbf{U} \\ s.t. \quad & U_{min} \leq \mathbf{U} \leq U_{max} \end{aligned} \tag{3.21}$$

where the matrices Hessian H and gradient g are the same as in problem 3.18.

3.4.2 Input Increment Constraints

Input increment can possibly be regarded as constraints when our goal is to control the input rate within a limit. And this constraint is called $\Delta u(k)$ which varies between

two time steps. Therefore, this RHOC problem with input increment constraints is summarized as follows:

$$
\begin{aligned}
& min\, J_D(k) = \frac{1}{2}\sum_{i=1}^{N} \|y(k+i|k) - r(k+i|k)\|^2_{\mathbf{Q}_y} + \frac{1}{2}\sum_{i=0}^{N-1} \|\Delta u(k+i|k)\|^2_{\mathbf{S}} \\
s.t. \quad & x(k+i+1|k) = Ax(k+i|k) + Bu(k+i|k) + Dd(k+i|k) \quad i = 0, 1, \cdots, N-1 \\
& y(k+i|k) = Cx(k+i|k) \quad i = 0, 1, \cdots, N \\
& x(k|k) = x(k) \\
& u_{min} \le u(k+i|k) \le u_{max} \quad i = 0, 1, \cdots, N-1 \\
& \Delta u_{min} \le \Delta u(k) \le \Delta u_{max}
\end{aligned}
\tag{3.22}
$$

To get the corresponding formulation of QP problem, we rewrite this constraint $\Delta u(k)$:

$$
\Delta u_{min} \le \Delta u(k) \le \Delta u_{max} \Leftrightarrow \begin{bmatrix} \Delta u_{min} \\ \Delta u_{min} \\ \vdots \\ \Delta u_{min} \end{bmatrix} \le \begin{bmatrix} u(k|k) - u(k-1|k) \\ u(k+1|k) - u(k|k) \\ \vdots \\ u(k+N|k) - u(k+N-1|k) \end{bmatrix} \le \begin{bmatrix} \Delta u_{max} \\ \Delta u_{max} \\ \vdots \\ \Delta u_{max} \end{bmatrix} \Leftrightarrow
$$

Taking the term $u(k-1|k)$ into consideration, we have:

$$
\Delta u_{min} + u(k-1|k) \le u(k|k) \le \Delta u_{max} + u(k-1|k) \bigwedge \tag{3.23}
$$

$$
\underbrace{\begin{bmatrix} \Delta u_{min} \\ \Delta u_{min} \\ \vdots \\ \Delta u_{min} \end{bmatrix}}_{\Delta U_{min}{}'} \le \underbrace{\begin{bmatrix} -I & I & & \\ & -I & I & \\ & & \ddots & \ddots \\ & & & -I \quad I \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} u(k+1|k) \\ u(k+2|k) \\ \vdots \\ u(k+N-1|k) \end{bmatrix}}_{U'} \le \underbrace{\begin{bmatrix} \Delta u_{max} \\ \Delta u_{max} \\ \vdots \\ \Delta u_{max} \end{bmatrix}}_{\Delta U_{max}{}'} \tag{3.24}
$$

However, we cannot ignore constraints for $u(k|k)$, combining 3.23 with 3.20, we have the input constraints:

$$
\underbrace{\begin{bmatrix} max(u_{min}, \Delta u_{min} + u(k-1|k)) \\ u_{min} \\ \vdots \\ u_{min} \end{bmatrix}}_{U_{min}{}'} \le \mathbf{U} \le \underbrace{\begin{bmatrix} min(u_{max}, \Delta u_{max} + u(k-1|k)) \\ u_{max} \\ \vdots \\ u_{max} \end{bmatrix}}_{U_{max}{}'} \tag{3.25}
$$

Hence, the corresponding QP formulation is extended as bellow and note that U'_{min} and U'_{max} must be update at each steptime :

$$\begin{aligned} \min_{\mathbf{U}} J(k) &= \frac{1}{2}\mathbf{U}^T H\mathbf{U} + g^T\mathbf{U} \\ s.t. \qquad U'_{min} &\leq \mathbf{U} \leq U'_{max} \\ \Delta U'_{min} &\leq \Lambda U' \leq \Delta U'_{max} \end{aligned} \tag{3.26}$$

3.4.3 Output Constraints

Constraints on output are obviously limitations to maximum and minimum output at each steptime which are similar to input constraints. And its form is as below:

$$\begin{aligned} min\, J_D(k) &= \frac{1}{2}\sum_{i=1}^{N} \|y(k+i|k) - r(k+i|k)\|^2_{\mathbf{Q}_y} + \frac{1}{2}\sum_{i=0}^{N-1} \|\Delta u(k+i|k)\|^2_{\mathbf{S}} \\ s.t. \qquad x(k+i+1|k) &= Ax(k+i|k) + Bu(k+i|k) + Dd(k+i|k) \quad i = 0,1,\cdots,N-1 \\ y(k+i|k) &= Cx(k+i|k) \quad i = 0,1,\cdots,N \\ x(k|k) &= x(k) \\ u_{min} &\leq u(k+i|k) \leq u_{max} \quad i = 0,1,\cdots,N-1 \\ \Delta u_{min} &\leq \Delta u(k) \leq \Delta u_{max} \quad i = 0,1,\cdots,N-1 \\ y_{min} &\leq y(k+i|k) \leq y_{max} \quad i = 1,2,\cdots,N \end{aligned} \tag{3.27}$$

We ignore the output constraint at steptime $k = 0$, since it has no influence, so we just consider from $i = 1$ to $i = N$, which yields:

$$y_{min} \leq y(k+i|k) \leq y_{max} \Leftrightarrow \underbrace{\begin{bmatrix} y_{min} \\ y_{min} \\ \vdots \\ y_{min} \end{bmatrix}}_{Y_{min}} \leq \underbrace{\begin{bmatrix} y(k+1|k) \\ y(k+2|k) \\ \vdots \\ y(k+N|k) \end{bmatrix}}_{Y} \leq \underbrace{\begin{bmatrix} y_{max} \\ y_{max} \\ \vdots \\ y_{max} \end{bmatrix}}_{Y_{max}} \tag{3.28}$$

Considering Eq. 3.14 , we have:

$$\begin{aligned} &Y_{min} \leq \mathbf{H}x(k) + \mathbf{G}\mathbf{U} + \mathbf{E}_d D \leq Y_{max} \Leftrightarrow \\ &Y_{min} - \mathbf{H}x(k) - \mathbf{E}_d D \leq \mathbf{G}\mathbf{U} \leq Y_{max} - \mathbf{H}x(k) - \mathbf{E}_d D \Leftrightarrow \\ &Y_{min}{}' \leq \mathbf{G}\mathbf{U} \leq Y_{max}{}' \end{aligned} \tag{3.29}$$

Therefore, in this case, the control problem transformed in the QP formulation is like :

$$\begin{aligned} \min_{\mathbf{U}} J(k) &= \frac{1}{2}\mathbf{U}^T H\mathbf{U} + g^T\mathbf{U} \\ s.t. \qquad U'_{min} &\leq \mathbf{U} \leq U'_{max} \\ \Delta U'_{min} &\leq \Lambda U' \leq \Delta U'_{max} \\ {Y_{min}}' &\leq \mathbf{G}\mathbf{U} \leq {Y_{max}}' \end{aligned} \tag{3.30}$$

Chapter 4

Implentation

For the propose of this chapter, we are interested in the application of RHOC, more precisely, we give two examples to illustrate the theory in the last two chapter, including an example of dual-input dual-output linear system without specific physical meaning, the other is about four tanks which is described in [Johansson, 2000].

4.1 Illustrative Example 1

4.1.1 Problem Setup

The first example is a dual-input dual-output linear system with six states given by:

$$\begin{cases} \dot{x_1} = -0.8x_1 + 2x_5 + 0.5u_1 + 0.5w_1 \\ \dot{x_2} = -0.7x_2 + 0.5u_1 + 0.5w_2 \\ \dot{x_3} = 0.8x_1 + 0.7x_2 - 0.5x_3 \\ \dot{x_4} = 0.15x_3 - x_4 \\ \dot{x_5} = x_4 - 2x_5 \\ \dot{x_6} = 0.35x_3 - 0.8x_6 + u_2 \end{cases} \tag{4.1}$$

and output is given by:

$$\begin{cases} y_1 = x_3 \\ y_2 = x_1 + x_6 \end{cases}$$

s.t.

$$\begin{cases} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq u \leq \begin{bmatrix} 2 \\ 2 \end{bmatrix} \\ \begin{bmatrix} -0.2 \\ -0.2 \end{bmatrix} \leq \Delta u \leq \begin{bmatrix} 0.2 \\ 0.2 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq y \leq \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix} \end{cases} \tag{4.2}$$

$$A = \begin{bmatrix} -0.8 & 0 & 0 & 0 & 2 & 0 \\ 0 & -0.7 & 0 & 0 & 0 & 0 \\ 0.8 & 0.7 & -0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.15 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 0 \\ 0 & 0 & 0.35 & 0 & 0 & -0.8 \end{bmatrix}, \quad B = \begin{bmatrix} 0.5 & 0 \\ 0.5 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \tag{4.3}$$

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

It is easy to verify that this MIMO system is stable, as all the eigenvalues of A have negative real parts, i.e., $\sigma(A) = \{-0.8, -0.21, -0.99 \pm 0.52i, -2.10, -0.70\}$. We assume that initial condition is $x_0 = [1.2; 0.6; 1.8; 0.9; 0; 0.6]^T$, the tracking reference is $y_{ref} = [1.2 \quad 1.1]^T$.

4.1.2 Numerical Results

Our object is to follow the reference output $y_{ref} = [1.2 \quad 1.1]^T$. We choose the prediction horizon is $N = 15$, the sampling time is $T_S = 4$. Additionally, assume that, the weighting matrices in the cost function $Q_y = diag[1\ 1]$, $S = diag[0.01\ 0.01]$. The tracking trajectory is shown in Fig. 4.1, 4.2 and 4.3. From these results, we know that the outputs can be well tracked while all the constraints are satisfied and all system states and inputs can reach a equilibrium in a short time. Next, we have mentioned that computational complexity is a shortcoming of RHOC. Here, we discuss which factors have influence on the computational complexity.

Firstly, we just change the prediction horizon N from 5 to 40 every five steps, and others are kept. The experimental results are shown in Table. 4.1.

Secondly, we change the sampling time T_s from 2 to 12 every two sampling intervals. The corresponding result is given by Table. 4.2.

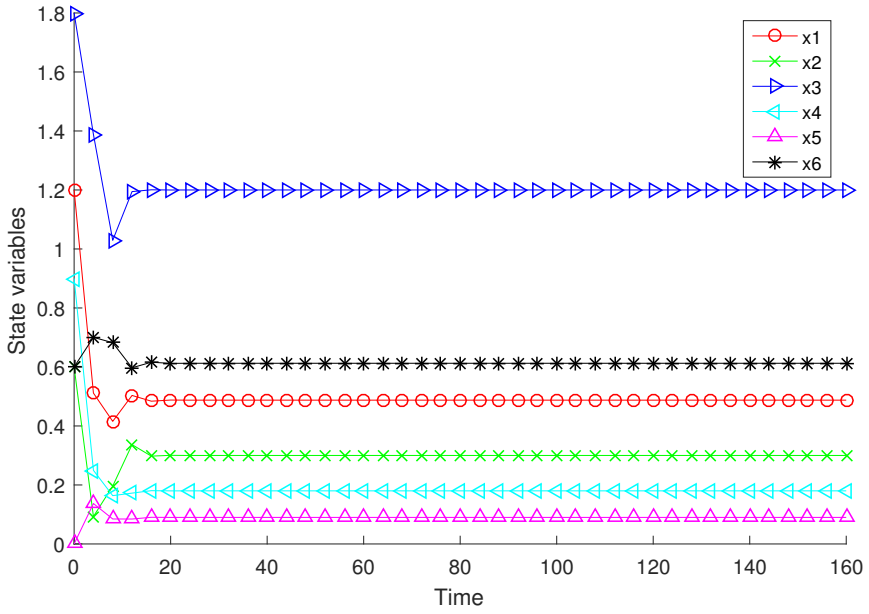


FIGURE 4.1: The evolution of each system state

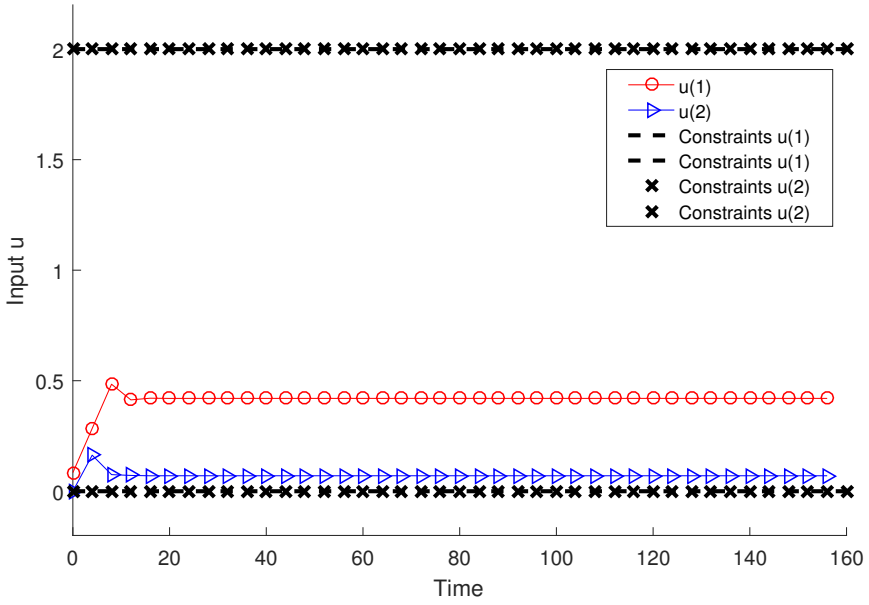


FIGURE 4.2: The evolution of predictive inputs

TABLE 4.1: Influence of prediction horizon

Horizon	N=5	N=10	N=15	N=20	N=25	N=30	N=35	N=40
Time(s)	1.61	2.60	4.27	5.71	8.39	11.56	14.15	17.90

TABLE 4.2: Influence of sampling time

Sampling Time(s)	T_S=2	T_S=4	T_S=6	$T_S = 8$	$T_S = 10$	$T_S = 12$
Time(s)	9.11	4.27	2.96	2.22	1.90	1.56

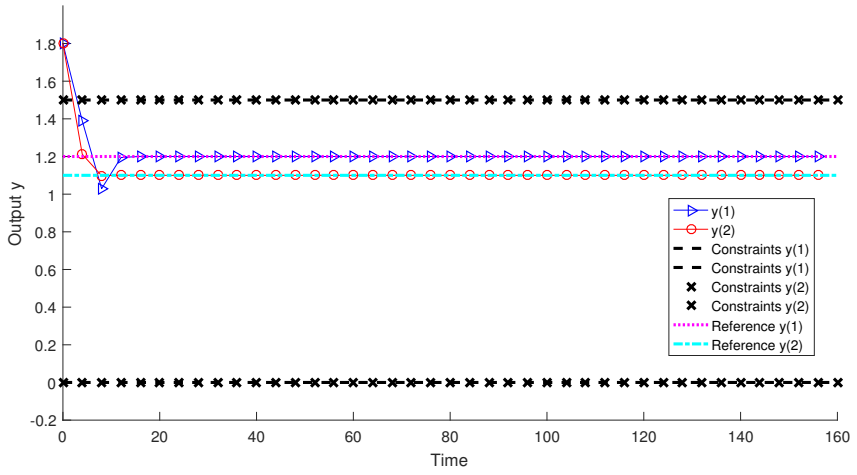


FIGURE 4.3: The evolution of predictive outputs

As the above tables, we shift our attention to the experimental results. we have seen that the tend of computational complexity for open-loop problem while the prediction horizon N or the sampling time is increased for the system in $example1$. And we confirm that the prediction horizon and the sampling time are the important factors on computing speed. When the samplIng time or prediction horizon is changed, the feasibility and stability are ensured, the output can reach the set reference as well. Therefore, the choice of a short or a long prediction horizon is a trade off problem between a better precision and a smaller computational complexity.

In addition, we concern about the change of matrices Q_y and S on the evaluation. In Figure 4.4, the column1 is the evolution of states, inputs and outputs as $Q_y = diag[0.0001\ 0.0001]$, and column2 is the evolution as $Q_y = diag[100\ 100]$. We can clearly see that the convergence rate of each variable becomes faster as Q_y increases, but all curves of variables become less smooth. Besides, there is a little deviation between set reference and y_1 . In Figure 4.5, the column1 is the evolution of states, inputs and outputs as $S = diag[100\ 100]$, and column2 is the evolution as $S = diag[0.001\ 0.001]$. It is easy to see that the convergence rate of each variable becomes faster as S reduces, but all curves of variables become more smooth. Hence, we have realized that matrices Q_y and S have entirely opposite influence on performance.

From the view of these results, if we desire to change the speed of reaching the set reference trajectory, we have several choices, for example, modifier the prediction horizon, sampling time, weighting matrix Q_y and S.

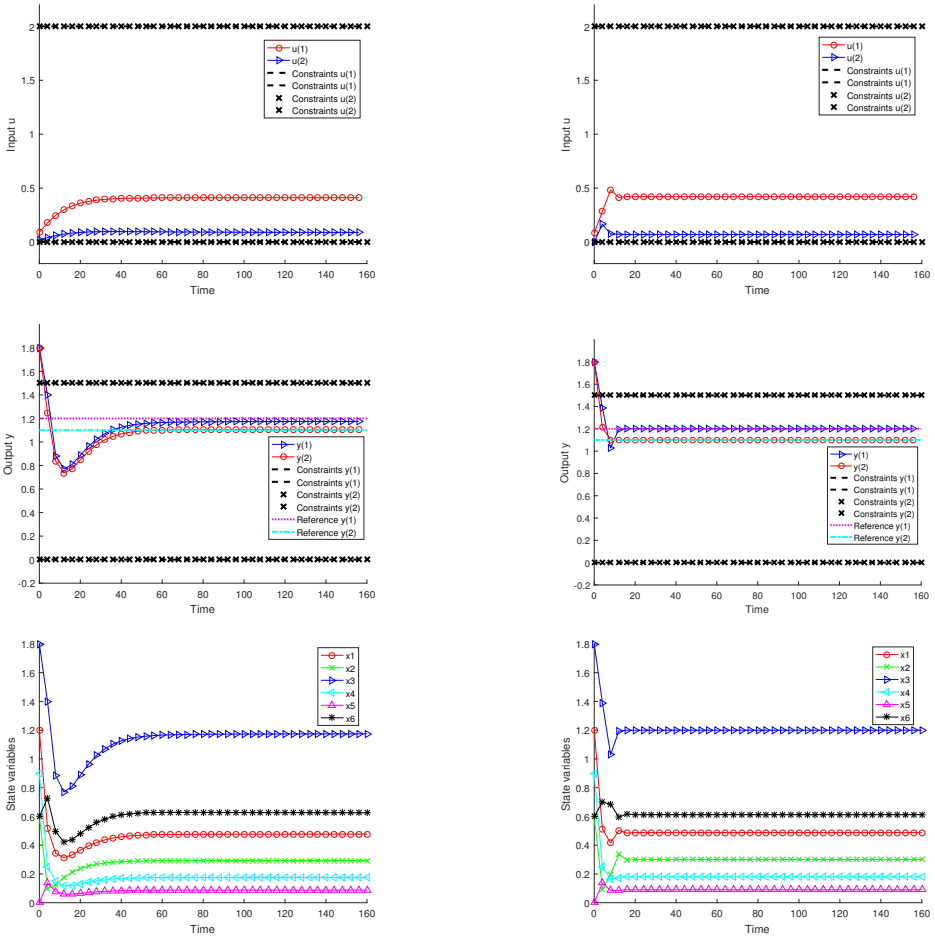


FIGURE 4.4: Change of matrix Q_y. Column1: $Q_y = diag[0.0001\ 0.0001]$. Column2: $Q_y = diag[100\ 100]$.

4.2 Illustrative Example 2

4.2.1 Problem Setup

Consider a nonlinear system extracted from [Johansson, 2000], which consists of four connected tanks shown as Fig. 4.6. Our goal is to control the water level in the tank 1 and tank 2. This is done by controlling the pump flow, qa and qb. Pump A draws water from the basin below and pours it to tank 1 and tank 4, similarly, pump B draws water to tank 2 and 3. The lever of water in each tank is h_i, $i = 1, 2, 3, 4$. There is a relation between the flows at each outlet and the total flow which depends on the flow parameters θ_1 and θ_2, this relation is described in the follwing form:

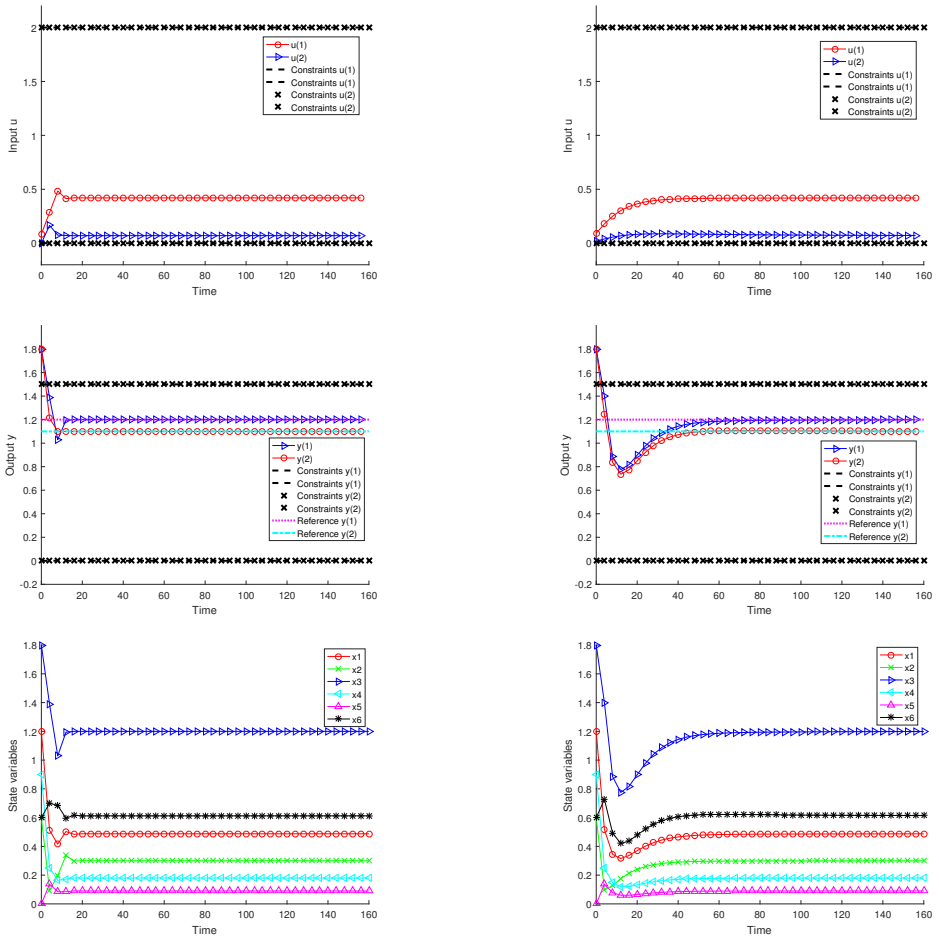


FIGURE 4.5: Change of matrix S. Column1: $S = diag[100\ 100]$. Column2: $S = diag[0.001\ 0.001]$.

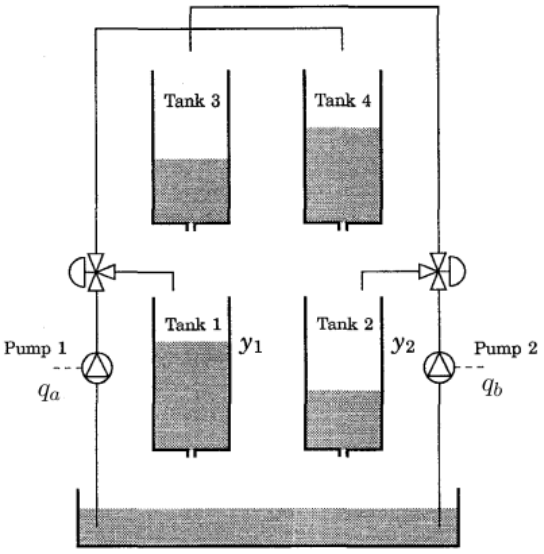


FIGURE 4.6: System of four tanks

$$\begin{cases} q_1 = \theta_1 q_a \\ q_2 = \theta_2 q_b \\ q_3 = (1-\theta_2) q_b \\ q_4 = (1-\theta_1) q_a \end{cases} \tag{4.4}$$

where,$0 \leq \theta_1 \leq 1$ and $0 \leq \theta_2 \leq 1$. $q_1,\ q_2,\ q_3,\ q_4$ are the each corresponding outlet, q_a and q_b are the pump flow.

We just consider a simplified example is given:

$$\begin{cases} \dot{h_1} = -\dfrac{a_1}{A_1}\sqrt{2gh_1} + \dfrac{a_3}{A_1}\sqrt{2gh_3} + \dfrac{\theta_1}{A_1} q_a \\ \dot{h_2} = -\dfrac{a_2}{A_2}\sqrt{2gh_2} + \dfrac{a_4}{A_2}\sqrt{2gh_4} + \dfrac{\theta_2}{A_2} q_b \\ \dot{h_3} = -\dfrac{a_3}{A_3}\sqrt{2gh_3} + \dfrac{(1-\theta_2)}{A_3} q_b \\ \dot{h_4} = -\dfrac{a_4}{A_4}\sqrt{2gh_4} + \dfrac{(1-\theta_1)}{A_4} q_a \end{cases} \tag{4.5}$$

$$\begin{cases} y_1 = h_1 \\ y_2 = h_2 \end{cases} \tag{4.6}$$

where the parameter values of controlled system are shown in Table. 4.3. And inputs are q_a, q_b, outpus are h_1 and h_2.

TABLE 4.3: Parameter values of Example 2

Cross section of the tanks$[cm^2]$	Value
A_1,A_2,A_3,A_4	28,32,28,32
Cross section of the outlet hole$[cm^2]$	Value
a_1	0.071
a_2	0.057
a_3	0.071
a_4	0.057
Flow parameters	Value
θ_1	0.3
θ_2	0.4
Gravity constant$[cm/s^2]$	Value
g	981

4.2.2 Linearization

In order to reduce the compute burden and make problem easier, we transform the non-linear system to a linear system model. By a Taylor expansion around the equilibrium point (i.e. $(x_i = h_i - {h_i}^0, u_j = q_j - {q_j}^0)$, i=1,2,3,4; j=a,b), we have:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -\frac{1}{T_1} & 0 & \frac{A_3}{A_1 T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{A_4}{A_2 T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{bmatrix} x + \begin{bmatrix} \frac{\theta_1}{A_1} & 0 \\ 0 & \frac{\theta_2}{A_2} \\ 0 & \frac{1-\theta_2}{A_3} \\ \frac{1-\theta_1}{A_4} & \end{bmatrix} \begin{bmatrix} q_a \\ q_b \end{bmatrix} \\ y &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x \end{aligned} \tag{4.7}$$

where,

$$T_i = \frac{A_i}{a_i}\sqrt{\frac{2{h_i}^0}{g}}, \qquad i = 1, 2, 3, 4$$

and then applying the forward Euler approximation on the Continuous model, considering a sampling instant k and a sampling time T_S,

$$x(k+i+1|k) = A(k)x(k+i|k) + B(k)u(k+i|k) \qquad i = 0, 1, 2, \ldots x(k|k) = x(k)$$

4.2.3 Numerical Results

We choose the sampling time is $T_S = 10$, the total time is $t = 800s$ and the prediction horizon is $N = 10$, the weighting matrices in the cost function $Q_y = diag[1\ 1]$, $S = diag[0.01\ 0.01]$. The constraints are given as follows:

$$\begin{cases} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq u \leq \begin{bmatrix} 12 \\ 12 \end{bmatrix} \\ \begin{bmatrix} -1 \\ -1 \end{bmatrix} \leq \Delta u \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq y \leq \begin{bmatrix} 14 \\ 14 \end{bmatrix} \end{cases} \tag{4.8}$$

In the absence of disturbance, our result of inputs, inputs and system states is shown in Fig. 4.9. As the result, we can see that each system state does not change any more after at time $t = 300$ and reaches their corresponding equilibrium. At the beginning,

the inputs are sharply increasing, reach the constraints, but do not beyond the upper boundary, and then they are reduced and stay steady. The tends of outputs are the same as the tends of *state*1 and *state*2, dual-output finally arrives steady.

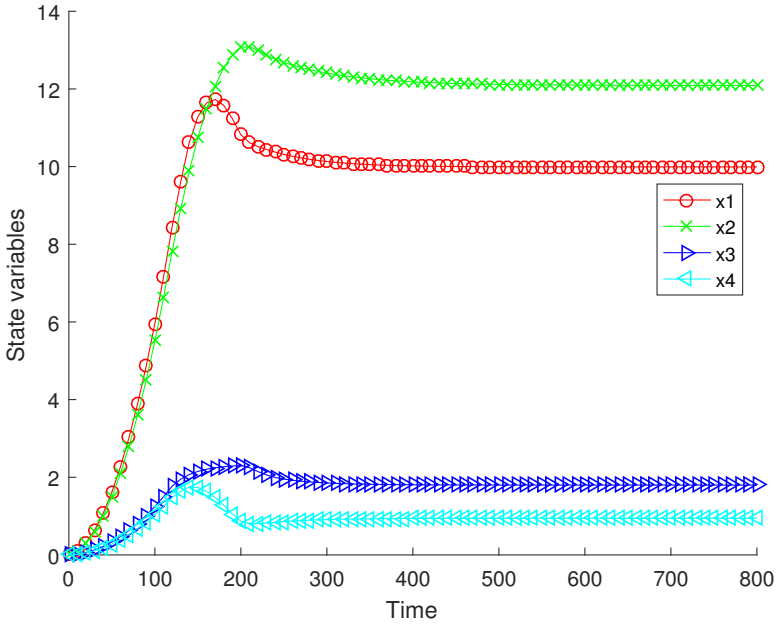


FIGURE 4.7: Ex2: The evolution of each system state

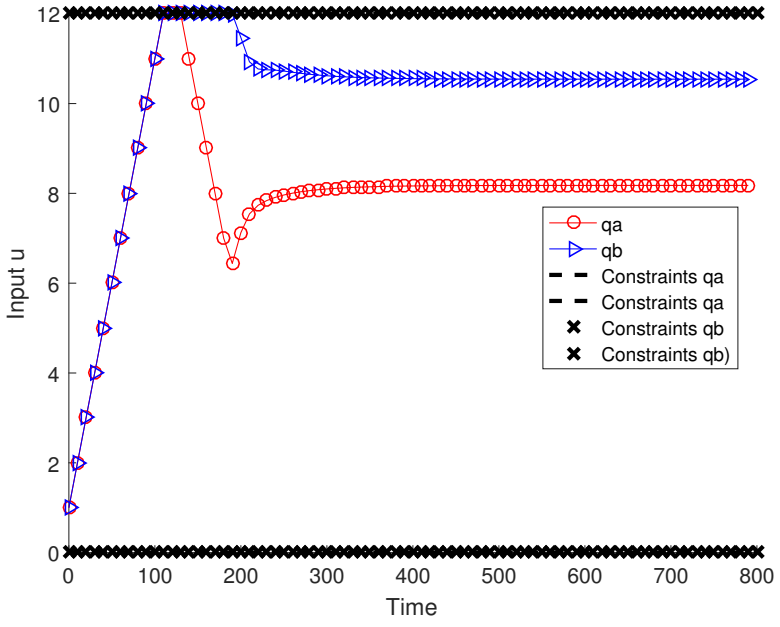


FIGURE 4.8: Ex2: The evolution of predictive inputs

Taking disturbance, soft constraints and feed-forward into consideration, other conditions unchanged, the results are shown in Fig. 4.10. We add a disturbance in two

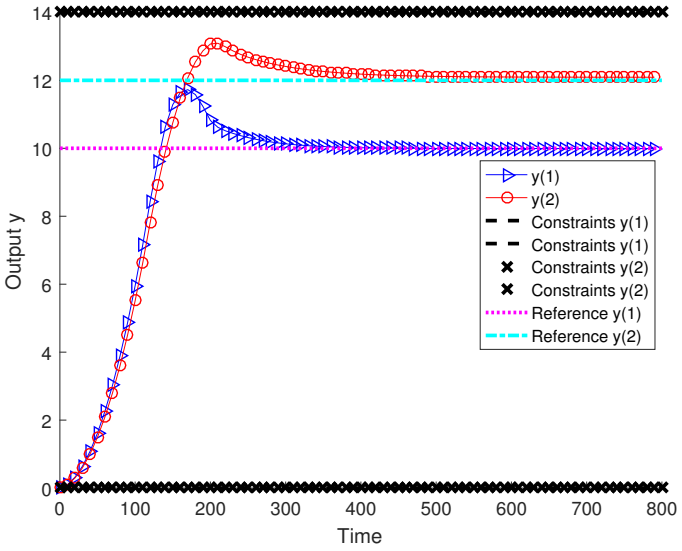


FIGURE 4.9: Ex2: The evolution of predictive outputs

intervals from $t = 400$ to $t = 420$ and from $t = 500$ to $t = 520$. In column 1, we just add a disturbance and the controller can steer the system to recover the equilibrium. Comparing with column 1, adding feed-forward in column 2 leads to a respond to inputs, outputs and states.

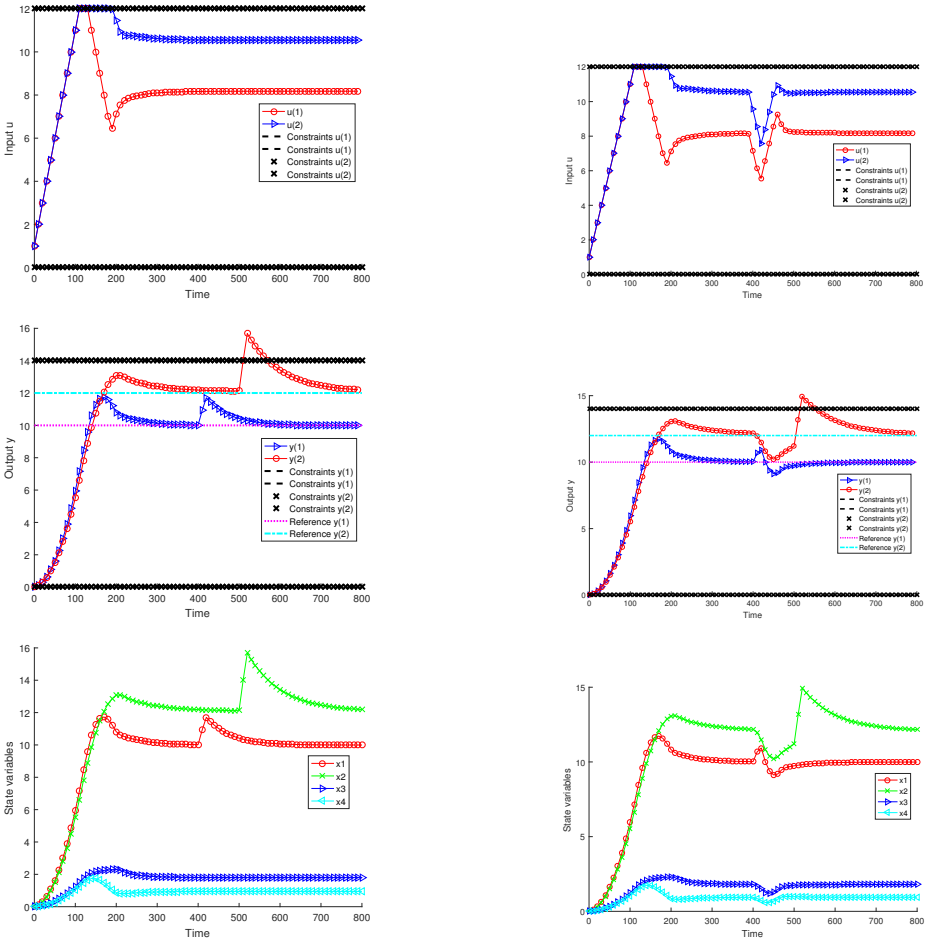


FIGURE 4.10: Column1: disturbance and non feed-forward. Column2: disturbance and feed-forward.

Chapter 5

Conclusion and Perspective

In this thesis, we studied the basic theory of RHOC based on discrete time linear system whose cost function respect state and input. We focus on controller by solving a open-loop optimal problem including over finite horizon and over infinite horizon. And then the proof of feasibility of open-loop problem and the stability of closed-loop controlled system over infinite horizon are given. For the reference trajectory tracking problem, we instantiate the basic theory, in consideration of disturbance and different constraints, and deduce the cost function of system model respecting output and input, and transform into the corresponding formulation of QP problem. Finally, Simulations confirm the theory by two examples.

For the future work, the first problem is to study nonlinear Receding Horizon Optimal Control method since nonlinear system problem is common in our life and almost all industrial processes have nonlinear dynamics features. Another problem to study is the conditions and proof of feasibility and stability for hyperbolic differential systems using RHOC method. Furthermore, apply RHOC method to partial differential systems, especially hyperbolic differential systems to analyze its stability and robust performance.

Bibliography

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

S.Joe Qin and Thomas A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733 – 764, 2003.

M. Morari and J.H. Lee. Model predictive control: Past, present and future. *Computers and Chemical Engineering*, 23(4):667–682, 1999.

N.L. Ricker. Adaptive optimal control: The thinking man's gpc. *Automatica*, 29(3):798 – 800, 1993.

A. Bemporad, F. Borrelli, and M. Morari. Model predictive control based on linear programming - the explicit solution. *IEEE Transactions on Automatic Control*, 47 (12):1974–1985, Dec 2002.

J.B. Rawlings and K. R. Muske. The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, 38(10):1512–1516, Oct 1993.

H. CHEN and F. ALLGWER. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205 – 1217, 1998.

A. Jadbabaie, J. Yu, and J. Hauser. Unconstrained receding-horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 46(5):776–783, May 2001.

X.T. Liu. Robust model predictive control and distributed model predictive control:feasibility and stability. *Phd thesis*, 2014.

V. Thang Pham. *Contribution to predictive control for systems of conservation laws.* Theses, Université de Grenoble, September 2012.

C. E. GARCIA and A. M. MORSHEDI. QUADRATIC PROGRAMMING SOLUTION OF DYNAMIC MATRIX CONTROL (QDMC). *Chemical Engineering Communications*, 46:73–87, 1986.

S. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

E. POLAK and T. H. YANG. Moving horizon control of linear systems with input saturation and plant uncertainty part 1. robustness. *International Journal of Control*, 58(3):613–638, 1993.

G. Droge and M. Egerstedt. Adaptive time horizon optimization in model predictive control. In *Proceedings of the 2011 American Control Conference*, pages 1843–1848, June 2011.

M. Balandat. Constrained robust optimal trajectory tracking: Model predictive control approaches. *Technische Universitat Darmstadt*, 2010.

D. Limon, M. Pereira, D. Muoz de la Pea, T. Alamo, C. N. Jones, and M. N. Zeilinger. Mpc for tracking periodic references. *IEEE Transactions on Automatic Control*, 61 (4):1123–1128, April 2016.

K. H. Johansson. The quadruple-tank process: a multivariable laboratory process with an adjustable zero. *IEEE Transactions on Control Systems Technology*, 8(3):456–465, May 2000.