Amirkabir
University of Technology
Tehran Polytechnic

Tehran Polytechnic

Final Project of Convex Optimization(2384213)

Convex Optimization in Model Predictive Control

Author:
Yousof Koohmaskan

Supervisor:
Dr. Behzad Samadi

Koohmaskan@gmail.com
Koohmaskan@aut.ac.ir

August 25, 2010

Contents

1 Introduction **3**
- 1.1 Introduction to predictive control . . . 3
- 1.2 Concept of MPC . . . 6
- 1.3 Filters in MPC . . . 9
- 1.4 Different type of predictive control . . . 10

2 Optimal Control for General Systems **12**
- 2.1 Optimal Control Based on Minimum Criterion . . . 12
  - 2.1.1 Dynamic Programming . . . 12
  - 2.1.2 Pontryagin's Minimum Principle . . . 13
- 2.2 Optimal Control Based on Minimax Criterion . . . 15
- 2.3 Linear Optimal Controls via LMI . . . 16
  - 2.3.1 LQR via LMI . . . 16
  - 2.3.2 H_∞ Control via LMI . . . 20

3 State Feedback in Model Predictive Control **23**
- 3.1 LQR via LMI in Predictive Form . . . 23
- 3.2 H_∞ control via LMI in Predictive Form . . . 29

4 Filters in MPC **31**
- 4.1 Kalman Filter . . . 31
- 4.2 Minimum Variance Finite Impulse Response Filters . . . 34

5 Quadratic Programming in MPC **43**
- 5.1 Introduction . . . 43
- 5.2 Example of QP . . . 43
- 5.3 QP as semi-definite programming (SDP) . . . 44
- 5.4 QP in Optimal Control . . . 45
  - 5.4.1 QP with equality constraints . . . 45
  - 5.4.2 QP with inequality constraints . . . 47
- 5.5 QP used in MPC . . . 50
- 5.6 Implementation . . . 52

A Appendix **56**

List of Figures

1 Control System 4
2 State feedback control 4
3 Output injection control 5
4 One-time long-term planning 6
5 Periodic short-term planning 7
6 Annual short-term planning 8
7 Investment planning 8
8 FIR filter with receding horizon 4 10
9 Optimal path from a to d 12
10 Optimal control and states in example 2 19
11 H_∞ optimal control and states in example 3 22
12 Tracking optimal control states in predictive form 27
13 Optimal control in predictive form 27
14 Tracking control in predictive form; new system 28
15 Optimal control in predictive form; new system 28
16 Tracking state using Kalman filter 35
17 Tracking optimal control using Kalman filter 35
18 Tracking state using MV filter 41
19 Tracking optimal control using MV filter 41
20 Error estimation in Kalman Filter and MV filter 42
21 States in QP problem with inequality constraints 53
22 Optimal control of QP problem with inequality constraints 53
23 States in QP problem with inequality constraints using CVXOPT 54
24 Optimal control of QP problem with inequality constraints using CVXOPT 54
25 States for QP problem in MPC 55
26 Optimal control for QP problem in MPC 55

Model predictive control is the only advanced control technology that has made a substantial impact on industrial control problems: its success is largely due to its almost unique ability to handle, simply and effectively, hard constraints on control and states. (D.Q.Mayne, Constrained optimal control. European Control Conference, Plenary Lecture, September 2001)

1 Introduction

1.1 Introduction to predictive control

Model Predictive Control (MPC) is a modern powerful control strategy which reached wide popularity in industry and process control. MPC is a form of control in which the current control action is obtained by solving on-line, at each sampling instant, a finite horizon open-loop optimal control problem, using the current state of the plant as the initial state; the optimization yields an optimal control sequence and the first control in this sequence is applied to the plant. MPC where sometimes called Receding Horizon Control(RHC), but MPC is better known , is based on the conventional optimal control that is obtained by minimization or mini-maximization of some performance criterion either for a fixed finite horizon or for an infinite horizon.
There are three types of well-known predictive control.

- Generalized predictive control(GPC)
- Receding Horizon Control(RHC)
- Model Predictive Control(MPC)

Historically, GPC and MPC has been investigated and implemented for industrial applications independently. Originally, RHC dealt with state-space models, while GPC and MPC dealt with I/O models. These three controls are equivalent to one another when the problem formulation is the same.

We would like to discuss about discrete-time systems, because of easy applying via computer aided design and also near subject to my final project!
A dynamical control system has input variable, state variable and output variable. Its model can be linear or nonlinear. It can be represented as a stochastic system with noises or a deterministic system with disturbance. In the Figure 1 a control system is depicted.

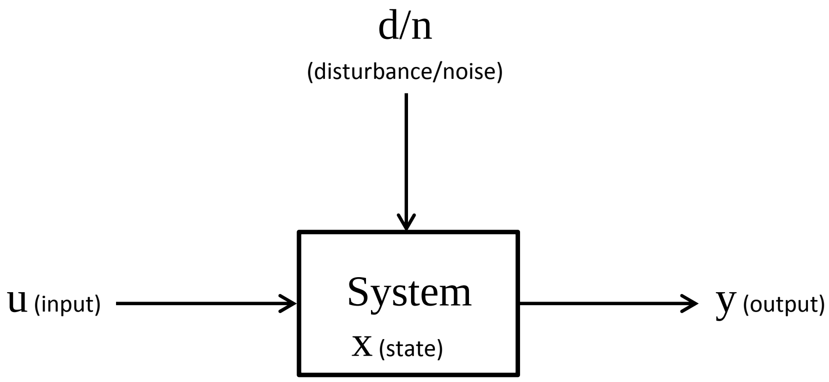

Figure 1: Control System

There are some objective for control system, *e.g.* output regulation, tracking reference signal, closed-loop stability and so on. If all the states be available, the state feedback can be used for control and if system be observable, output feedback (output injection) may be applied. Figure 2 illustrates the state feedback control and Figure 3 show output injection control.

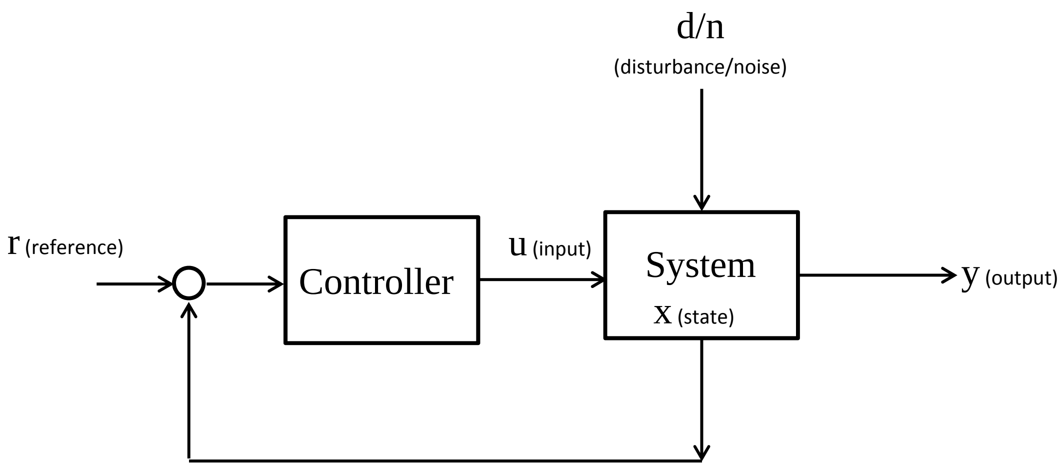

Figure 2: State feedback control

There are several approaches for control designs to meet control objectives. Optimal control has been one of the widely used methods. It is obtained by minimizing or maximizing a certain performance criterion or combination of them, *i.e.* mini-maximizing or maxi-minimizing a certain performance criterion.

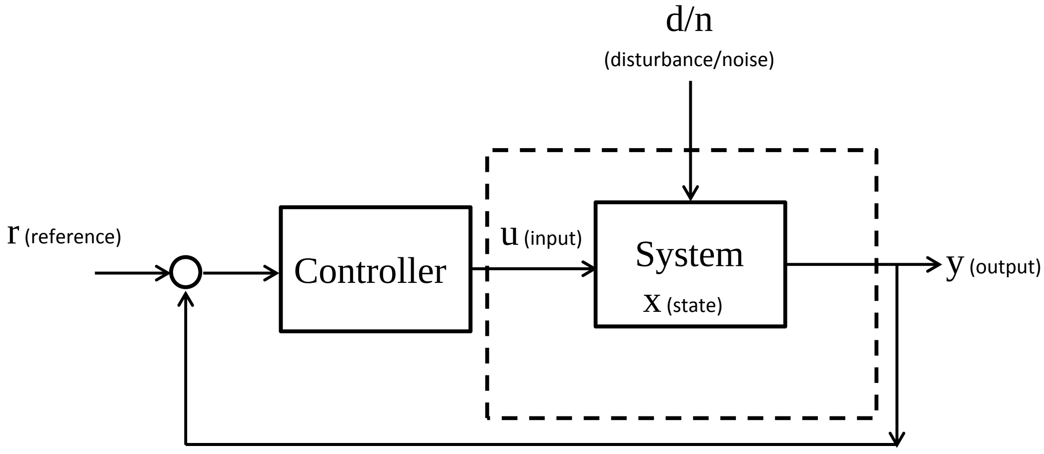

Figure 3: Output injection control

Some popular optimal controls:

- The LQ control for state feedback control based on minimizing
- The LQG control for output feedback control based on minimizing
- The H_∞ control based on mini-maximization

1.2 Concept of MPC

Often, feedback control systems must run for a sufficiently long period, as in electrical power generation plants and chemical processes. The basic concept of MPC is as follows;

1. At the current time, the optimal control is obtained, either closed-loop type, or open-loop type, on a finite fixed horizon from the current time k, say $[k, k+N]$.

2. Among the optimal controls on the entire fixed horizon $[k, k+N]$, only the first one is adopted as the current control law.

3. The procedure is then repeated at the next time, say $[k+1, k+1+N]$.

Here we refer to description of [1] about idea in receding horizon control. The concept of RHC can be easily explained by using a company's investment planning to maximize the profit. The investment planning should be continued for the years to come as in feedback control systems. There could be three policies for a company's investment planning:

(1) One-time long-term planning
Investment planning can be carried over a fairly long period, which is closer to infinity, as in Figure 4 This policy corresponds to the infinite horizon optimal control obtained over $[k, \infty]$.

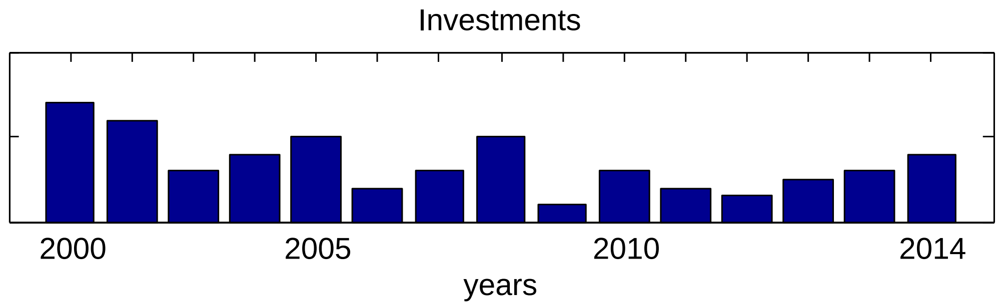

Figure 4: One-time long-term planning

(2) Periodic short-term planning Instead of the one-time long-term planning, we can repeat short-term investment planning, say investment planning every 5-years, which is given in Figure 5.

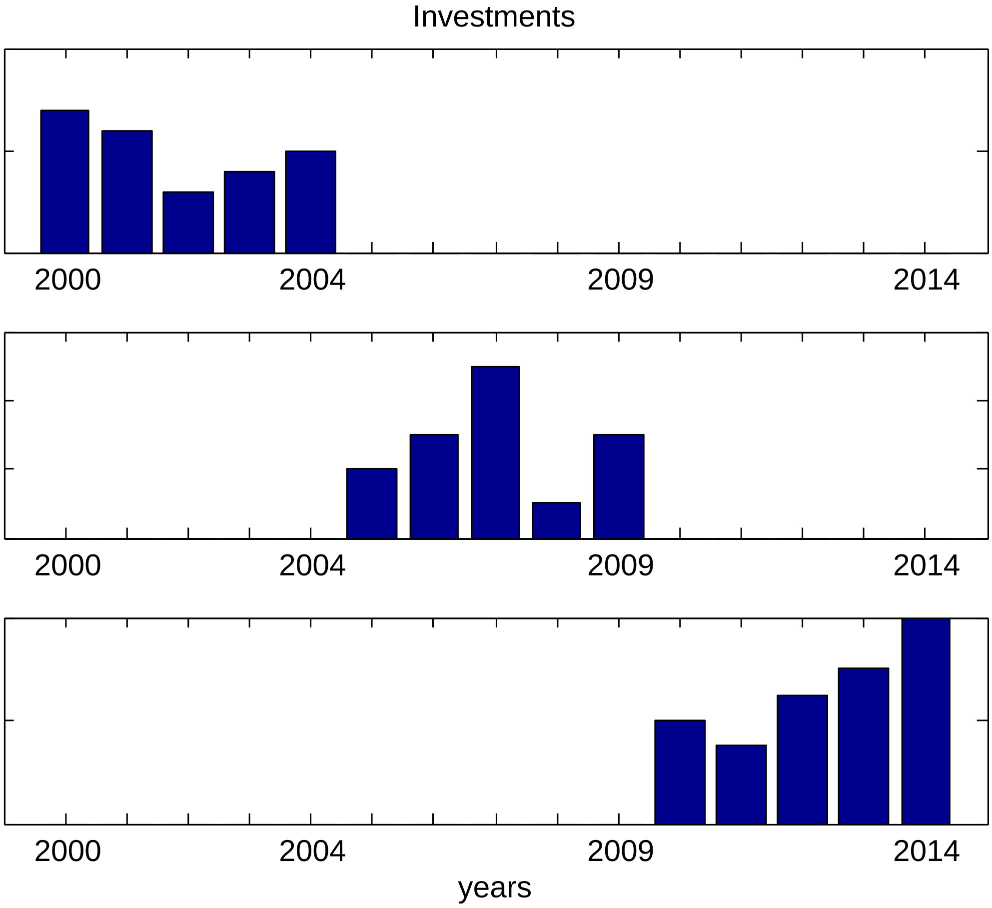

Figure 5: Periodic short-term planning

(3) Annual short-term planning For a new policy, it may be good to have a short-term planning every year and the first year's investment is selected for the current year's investment policy. This concept is depicted in Figure 6.

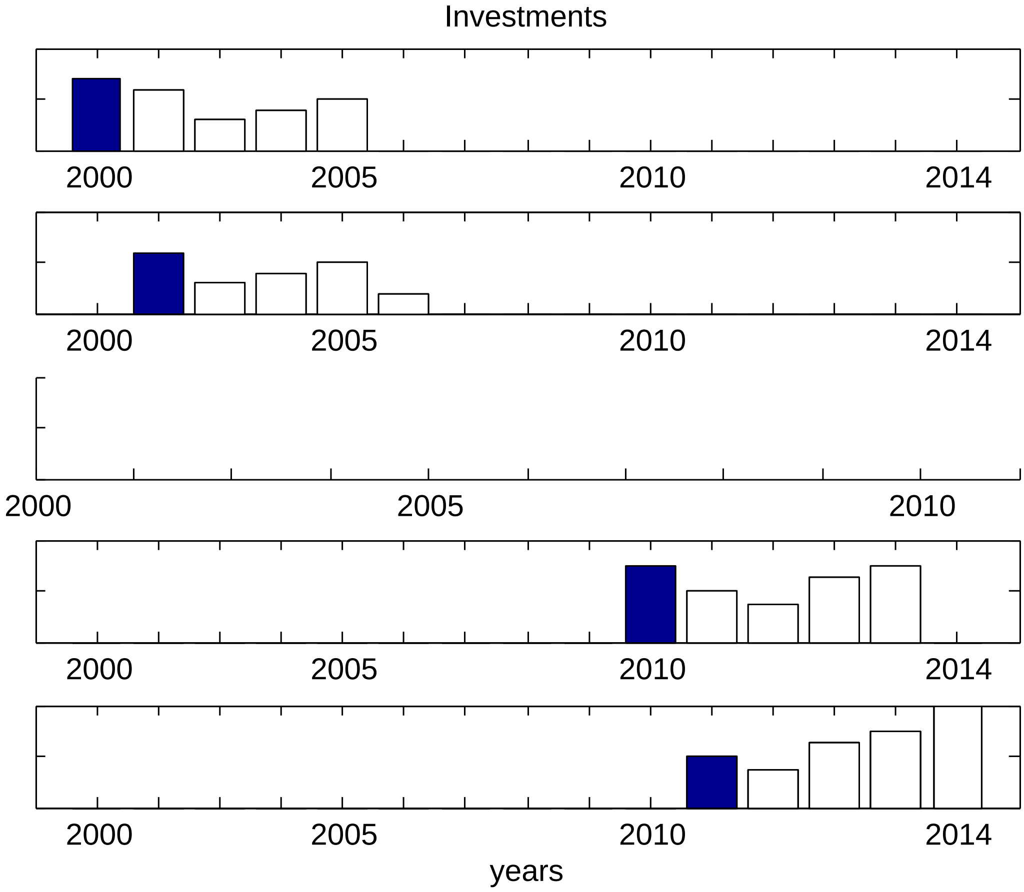

Figure 6: Annual short-term planning

It may be good to have a short-term planning every year and the first year's investment is selected for the current year's investment policy. This investment planning can be shown as in Figure 7.

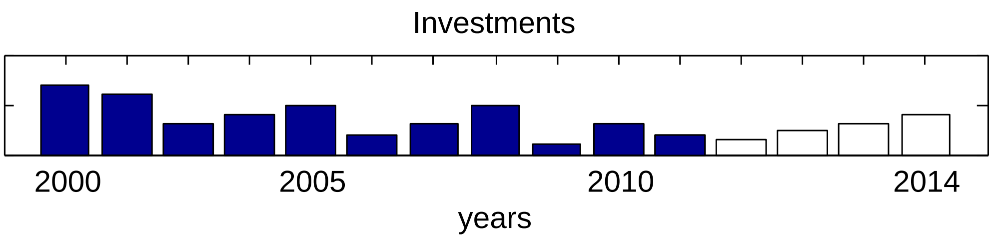

Figure 7: Investment planning

All the planning mentioned before have some advantages and some disadvantages. Whereas the long term planning gives us infinite horizon, but the computation may last much, the short-time planning gives us fast computation, but finite horizon. In the missile control systems, the second(short-time planning) is used.

1.3 Filters in MPC

If the states may not be available or the measurement of all states be expensive, we should estimate the states via measuring the inputs and outputs. This procedure can be performed by a state observer for deterministic systems or a filter for stochastic systems. Often, it is called a filter for both systems. The well-known Luenberger observer for deterministic state-space signal models and the Kalman filter for stochastic state-space signal models are infinite impulse response (IIR) type filters. This means that the state observer utilizes all the measured data up to the current time k from the initial time k_0. Equation (1) and (2) demonstrate state observer and state space realization, respectively.

$$\hat{x}[k+1] = A\hat{x}[k] + Bu[k] + L\left(y[k] - C\hat{x}[k] - Du[k]\right) \tag{1}$$

$$G = \left[\begin{array}{c|cc} A-LC & B-LD & L \\ \hline I & 0 & 0 \end{array}\right] \tag{2}$$

Where $\begin{bmatrix} u \\ y \end{bmatrix}$ is input and $\hat{x}$ is output.

Instead, we can utilize the measured data on the recent finite time $[kN_f, k]$ and obtain an estimated state by a linear combination of the measured inputs and outputs over the receding finite horizon with some weighting gains to be chosen so that the error between the real state and the estimated one is minimized. N_f is called the filter horizon size and is a design parameter.

Example 1 *As an example, consider the state(1-Dimensional) in this filter.*

$$\hat{x}[k] = 1.4750x[k-1] - 0.6787x[k-2] + 0.1191x[k-3] - 0.0065x[k-4]$$

Figure 8 shows that the filter used from 4 previous steps to estimate present step. The window which here, we call it horizon move to right each step, means doesn't employ more than 4 steps to estimate. It is a finite impulse response (FIR) type filter.

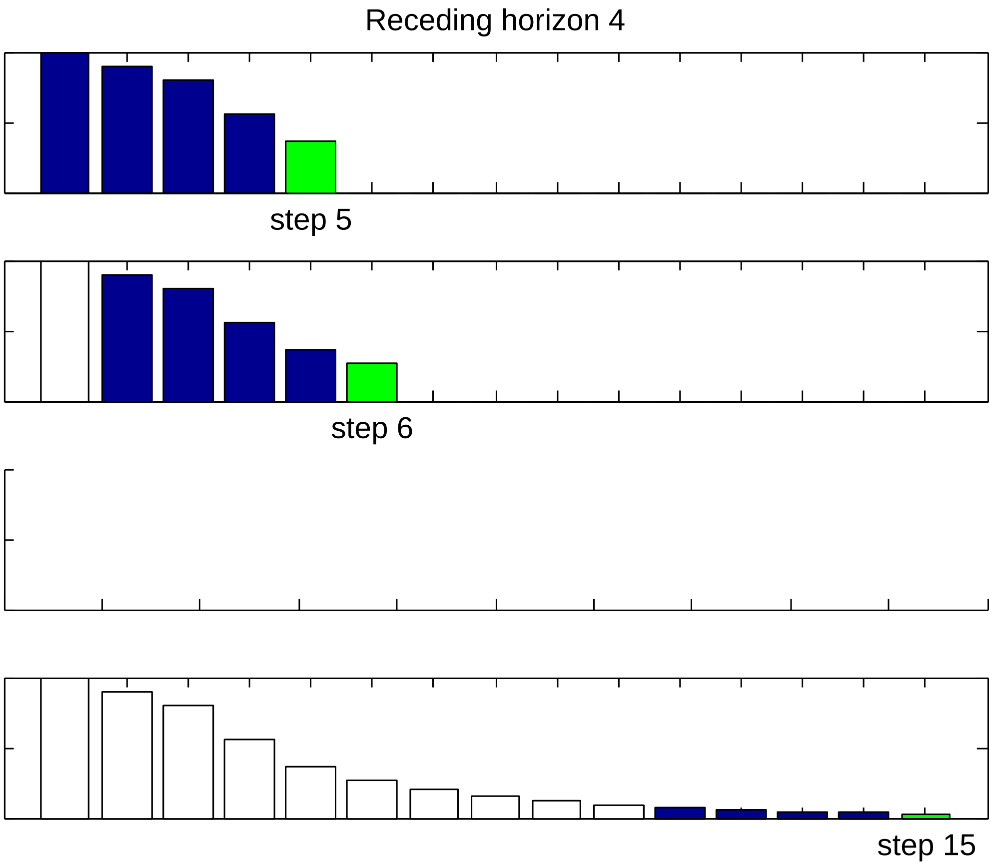

Figure 8: FIR filter with receding horizon 4

1.4 Different type of predictive control

After introduction to MPC, we present some difference between types of predictive control as mentioned before.

GPC was developed in the self-tuning and adaptive control area. Some control strategies that achieve minimum variance were adopted in the self-tuning control. GPC is based on the single input and single output (SISO) models such as auto regressive moving average (ARMA) or controlled auto regressive integrated moving average (CARIMA) models which have been widely used for most adaptive controls.

MPC has been developed on a model basis in the process industry area as an alternative algorithm to the conventional proportional integrate derivative (PID) control that does not utilize the model. The original version of MPC was developed for truncated I/O models, such as FIR models or finite step response (FSR) models. Model algorithmic control (MAC) was developed for FIR models and the dynamic matrix control (DMC) was developed for FSR models . These two control strategies coped with I/O constraints. Since I/O models such as the FIR model or the FSR model are physically intuitive, they are widely accepted in the process industry. However, these early control strategies were somewhat heuristic, limited to the FIR or the FSR models, and not applicable to unstable systems. Thereafter, lots of extensions have been made for state-space models. Some of applications are as follows:

- Distillation column
- Pulp and paper plant
- Servo mechanism
- Robot arm

RHC has been developed in academia as an alternative control to the celebrated LQ controls. RHC is based on the state-space framework. The stabilizing property of RHC has been shown for case of both continuous and discrete systems using the terminal equality constraint. In addition, it has been extended to tracking controls, output feedback controls, and nonlinear controls.

What makes MPC successful in industry are:

- It handles multivariable control problems naturally
- It can take account of actuator limitations
- It allows operation closer to constraints, hence increased profit
- It has plenty of time for on-line computations
- It can handle non-minimal phase and unstable processes
- It is an easy to tune method and
- It handles structural changes.

2 Optimal Control for General Systems

2.1 Optimal Control Based on Minimum Criterion

A discrete time system is represented by:

$$x_{i+1} = f(x_i, u_i, i), \quad x_{i_0} = x_0 \tag{3}$$

where $x_i \in \mathbb{R}^n$ and $u_i \in \mathbb{R}^m$ are state and input respectively.
A performance criterion with the free terminal state is given by

$$J(x_{i_0}, i_0, u) = \sum_{i=i_0}^{i_f-1} g(x_i, u_i, i) + h(x_{i_f}, i_f) \tag{4}$$

The minimization problem is then

$$\begin{array}{ll} \text{minimize} & J(x_{i_0}, i_0, u) \\ \text{subject to} & \text{constraints} \end{array} \tag{5}$$

The constraints may be for example terminal constraints, i.e. $x_{i_f} = x_f$. So the cost function will has small change. The function $h(.,.)$ then is constant and new constraint should be satisfied. In other word the problem now is

$$\begin{array}{ll} \text{minimize} & \sum_{i=i_0}^{i_f-1} g(x_i, u_i, i) \\ \text{subject to} & x_{i_f} = x_f \end{array} \tag{6}$$

2.1.1 Dynamic Programming

Bellman's principle of optimality tells: *An optimal policy has the property that whatever the initial state and initial decision are,the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.* Figure 9 depicts the method of selecting optimal path from a to d. In

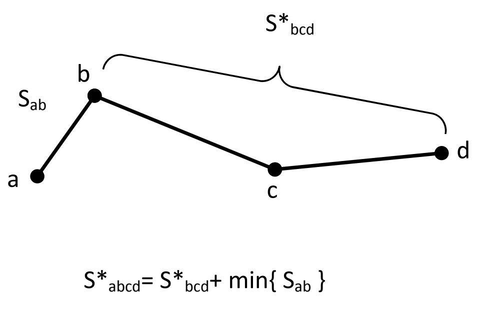

Figure 9: Optimal path from a to d

fact the best path is chosen from domain that minimize S_{ab} cost function, i.e. $p^* = arg(\min S_{ab}(u))$. It may be interesting about choice of the name dynamic programming by Richard Bellman. He said *"I spent the Fall quarter (of 1950) at RAND. My first task was to find a name for multistage decision processes. An interesting question is, Where did the name, dynamic programming, come from? The 1950s were not good years for mathematical research. We had a very interesting gentleman in Washington named Wilson. He was Secretary of Defense, and he actually had a pathological fear and hatred of the word, research. I'm not using the term lightly; I'm using it precisely. His face would suffuse, he would turn red, and he would get violent if people used the term, research, in his presence. You can imagine how he felt, then, about the term, mathematical. The RAND Corporation was employed by the Air Force, and the Air Force had Wilson as its boss, essentially. Hence, I felt I had to do something to shield Wilson and the Air Force from the fact that I was really doing mathematics inside the RAND Corporation. What title, what name, could I choose? In the first place I was interested in planning, in decision making, in thinking. But planning, is not a good word for various reasons. I decided therefore to use the word, programming. I wanted to get across the idea that this was dynamic, this was multistage, this was time-varying. I thought, let's kill two birds with one stone. Let's take a word that has an absolutely precise meaning, namely dynamic, in the classical physical sense. It also has a very interesting property as an adjective, and that is it's impossible to use the word, dynamic, in a pejorative sense. Try thinking of some combination that will possibly give it a pejorative meaning. It's impossible. Thus, I thought dynamic programming was a good name. It was something not even a Congressman could object to. So I used it as an umbrella for my activities".*

2.1.2 Pontryagin's Minimum Principle

By definition, the optimal control u^* makes the performance criterion J be a local minimum if

$$J(u) - J(u^*) = \Delta J \geq 0 \tag{7}$$

Define the function $\mathcal{H}$, called the *Hamiltonian*

$$\mathcal{H}(x_i, u_i, p_{i+1}, i) \triangleq g(x_i, u_i, i) + p_{i+1}^T f(x_i, u_i, i) \tag{8}$$

where p_{i+1} is Lagrange multiplier.
In terms of the Hamiltonian, the necessary conditions for u_i^* to be an optimal control are

$$x_{i+1}^* = \frac{\partial \mathcal{H}}{\partial p_{i+1}}(x_i^*, u_i^*, p_{i+1}^*, i) \tag{9}$$

$$p_i^* = \frac{\partial \mathcal{H}}{\partial x}(x_i^*, u_i^*, p_{i+1}^*, i) \tag{10}$$

$$\mathcal{H}(x_i^*, u_i^*, p_{i+1}^*, i) \leq \mathcal{H}(x_i^*, u_i, p_{i+1}^*, i) \tag{11}$$

for all admissible u_i on the $i \in [i_0, i_f 1]$, and two boundary conditions

$$\begin{aligned} x_{i_0} &= x_0 \\ p_{i_f}^* &= \frac{\partial h}{\partial x_{i_f}}(x_{i_f}^*, i_f) \end{aligned} \tag{12}$$

In this case, for u_i^* to minimize the Hamiltonian it is necessary that

$$\frac{\partial \mathcal{H}}{\partial u_i}(x_i^*, u_i^*, p_{i+1}^*, i) = 0, \quad i \in [i_0, i_f 1] \tag{13}$$

If (14) is satisfied and the matrix Hessian be positive definite, i.e.

$$\frac{\partial^2 \mathcal{H}}{\partial u_i^2}(x_i^*, u_i^*, p_{i+1}^*, i) > 0$$

then, the problem is *convex* and u_i^* is global optimal solution which makes cost function J to be minimized.

The above phrase point out a fundamental property of convex optimization problems that say, *any locally optimal point is also globally optimal*[2].

2.2 Optimal Control Based on Minimax Criterion

When the term minimax is mentioned, we usually think about something that causes undesired treatment. Disturbance may be a good instance. First we find the disturbance w where maximize the cost function J and then find the control law u where minimize it. The system that we should consider is as following:

$$x_{i+1} = f(x_i, u_i, w_i, i), \quad x_{i_0} = x_0 \tag{14}$$

with a performance criterion

$$J(x_{i_0}, i_0, u, w) = \sum_{i=i_0}^{i_f-1} [g(x_i, u_i, w_i, i)] + h(x_{i_f}, i_f) \tag{15}$$

where $x_i \in \mathbb{R}^n$, $u_i \in \mathbb{R}^m$ and $w_i \in \mathbb{R}^l$ are state, input and disturbance respectively.

We want to minimize the performance criterion, while disturbances try to maximize one. We may think that u^* is the best control, while w^* is the worst disturbance. The existence of these u^* and u^* is guaranteed by specific conditions. Hence the below inequality is satisfied:

$$J(x_{i_0}, i_0, u^*, w) \leq J(x_{i_0}, i_0, u^*, w^*) \leq J(x_{i_0}, i_0, u, w^*) \tag{16}$$

The control law u^* makes the performance criterion (15) a local minimum if

$$J(u, w) - J(u^*, w) = \Delta J \geq 0 \tag{17}$$

and the disturbance w^* makes the performance criterion (15) a local maximum if

$$J(u, w) - J(u, w^*) = \Delta J \leq 0 \tag{18}$$

and finally, in terms of the Hamiltonian, the necessary conditions for u_i^* to be an optimal control are

$$x_{i+1}^* = \frac{\partial \mathcal{H}}{\partial p_{i+1}}(x_i^*, u_i^*, w_i^*, p_{i+1}^*, i) \tag{19}$$

$$p_i^* = \frac{\partial \mathcal{H}}{\partial x}(x_i^*, u_i^*, w_i^*, p_{i+1}^*, i) \tag{20}$$

$$\mathcal{H}(x_i^*, u_i^*, w_i, p_{i+1}^*, i) \leq \mathcal{H}(x_i^*, u_i^*, w_i^*, p_{i+1}^*, i) \leq \mathcal{H}(x_i^*, u_i, w_i^*, p_{i+1}^*, i) \tag{21}$$

for all admissible u_i and w_i on the $i \in [i_0, i_f 1]$, and two boundary conditions

$$x_{i_0} = x_0$$

$$p_{i_f}^* = \frac{\partial h}{\partial x_{i_f}}(x_{i_f}^*, i_f)$$

The cost function for the dynamic programming in minimaximization criterion can be obtained from equation below:

$$J^*(x_i, i) = \min_{u \in \mathcal{U}} \max_{w \in \mathcal{W}} \Big[g(x_i, u_i, w_i, i) + J^*(f(x_i, u_i, w_i, i), i+1) \Big] \tag{22}$$

2.3 Linear Optimal Controls via LMI

Optimal control problems for discrete LTI systems are reformulated in terms of linear matrix inequalities (LMIs). Since LMI problems are convex, it can be solved very efficiently and the global minimum is always found.

2.3.1 LQR via LMI

Infinite horizon LQR problem in discrete time system has the cost function as follows:

$$J_\infty = \sum_{i=0}^{\infty} \left(x_i^T Q x_i + u_i^T R u_i \right) \tag{23}$$

where $Q > 0$ and $R > 0$.
We focus on designing a linear optimal state feedback control, *i.e.* $u_i = H x_i$. Assume that exist $V(x_i) = x_i^T K x_i$, which $K > 0$ and satisfies the following inequality:

$$V(x_{i+1}) - V(x_i) \leq -\Psi_i \tag{24}$$

where $\Psi_i = x_i^T Q x_i + u_i^T R u_i$. If we write inequality (24) in following manner:

$$\begin{aligned} V(x_1) - V(x_0) &\leq -\Psi_0 \\ V(x_2) - V(x_1) &\leq -\Psi_1 \\ V(x_3) - V(x_2) &\leq -\Psi_2 \\ &\vdots \\ V(x_{n+1}) - V(x_n) &\leq -\Psi_n \\ &\vdots \end{aligned}$$

sum of both sides of inequality with attention that V is decreasing yields:

$$-V(x_0) \quad \leq \quad -\Psi_0 - \Psi_1 - \Psi_2 - \cdots - \Psi_n - \cdots$$

if $n \to \infty$ yields:

$$V(x_0) \geq J_\infty \tag{25}$$

substituting $x_{i+1} = Ax_i + Bu_i$ and $u_i = Hx_i$ in (24)

$$\Big(Ax_i + BHx_i\Big)^T K \Big(Ax_i + BHx_i\Big) - x_i^T K x_i \leq -\Big(x_i^T Q x_i + (Hx_i)^T R (Hx_i)\Big)$$

$$x_i^T \Big(A + BH\Big)^T K \Big(A + BH\Big) x_i - x_i^T K x_i \leq -x_i^T \Big(Q + H^T R H\Big) x_i \tag{26}$$

if all the x_i satisfies (26), the matrices inequality is:

$$(A + BH)^T K (A + BH) - K + (Q + H^T R H) \leq 0 \tag{27}$$

We should minimize cost function J_∞. But another approach can be minimizing its upper bound as in (25). This approach can be repeated in one other step, *i.e.* instead of directly minimizing $V(x_0)$, we take an approach where its upper bound is minimized. For this purpose, assume that there exists $\gamma > 0$ such that

$$x_0^T K x_0 \leq \gamma \tag{28}$$

The inequality (27) can be written:

$$-K + Q + H^T R H + (A + BH)^T K (A + BH) \leq 0$$

$$-K + \begin{bmatrix} H^T & (A+BH)^T & I \end{bmatrix} \begin{bmatrix} R & 0 & 0 \\ 0 & K & 0 \\ 0 & 0 & Q \end{bmatrix} \begin{bmatrix} H \\ A+BH \\ I \end{bmatrix} \leq 0$$

$$-K + \begin{bmatrix} H^T & (A+BH)^T & I \end{bmatrix} \begin{bmatrix} R^{-1} & 0 & 0 \\ 0 & K^{-1} & 0 \\ 0 & 0 & Q^{-1} \end{bmatrix}^{-1} \begin{bmatrix} H \\ A+BH \\ I \end{bmatrix} \leq 0 \tag{29}$$

Note that R must be nonsingular. Applying Schur complement to (29) we have

$$\begin{bmatrix} K & H^T & (A+BH)^T & I \\ H & R^{-1} & 0 & 0 \\ A+BH & 0 & K^{-1} & 0 \\ I & 0 & 0 & Q^{-1} \end{bmatrix} \geq 0 \tag{30}$$

Also for the inequality (28):

$$\gamma - x_0^T \left(K^{-1}\right)^{-1} x_0 \geq 0$$

$$\begin{bmatrix} \gamma & x_0^T \\ x_0 & K^{-1} \end{bmatrix} \geq 0 \tag{31}$$

Several LMIs can be representable as one single LMI. Combining (30) and (31) yields

$$\left[\begin{array}{cccc|cc} K & H^T & (A+BH)^T & I & 0 & 0 \\ H & R^{-1} & 0 & 0 & 0 & 0 \\ A+BH & 0 & K^{-1} & 0 & 0 & 0 \\ I & 0 & 0 & Q^{-1} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \gamma & x_0^T \\ 0 & 0 & 0 & 0 & x_0 & K^{-1} \end{array}\right] \geq 0 \tag{32}$$

Pre and post multiplying (33) by a positive definite matrix $diag(K^{-1}, I, I, I, I, I)$,

$$\left[\begin{array}{cccc|cc} K^{-1} & K^{-1}H^T & K^{-1}(A+BH)^T & K^{-1} & 0 & 0 \\ HK^{-1} & R^{-1} & 0 & 0 & 0 & 0 \\ (A+BH)K^{-1} & 0 & K^{-1} & 0 & 0 & 0 \\ K^{-1} & 0 & 0 & Q^{-1} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \gamma & x_0^T \\ 0 & 0 & 0 & 0 & x_0 & K^{-1} \end{array}\right] \geq 0 \tag{33}$$

Changing variable $Y = K^{-1}$ and $L = HK^{-1}$, the problem summarize below.

minimize γ

subject to

$$\left[\begin{array}{cccc|cc} Y & L^T & (AY+BL)^T & Y^T & 0 & 0 \\ L & R^{-1} & 0 & 0 & 0 & 0 \\ (AY+BL) & 0 & Y & 0 & 0 & 0 \\ Y & 0 & 0 & Q^{-1} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \gamma & x_0^T \\ 0 & 0 & 0 & 0 & x_0 & Y \end{array}\right] \geq 0 \tag{34}$$

After finding Y and L, the state feedback obtains; $H = LY^{-1}$

Example 2 *Consider the following LQR problem:*

$$x[k+1] = \begin{bmatrix} 0.8 & 0.75 \\ 0 & 1 \end{bmatrix} x[k] + \begin{bmatrix} 0 & -0.2 \\ 1 & 0.5 \end{bmatrix} u[k]$$

with the weighting matrices $Q = I$ and $R = 10I$. Suppose that the initial condition be $x_0 = [-3 \quad 5]^T$. We want to find control which minimizes the cost function J_∞, mentioned before.
Here we obtain matrices Y and L and parameter γ as below:

$$Y = \begin{bmatrix} 0.5799 & -0.1570 \\ -0.1570 & 0.1678 \end{bmatrix} \qquad L = \begin{bmatrix} 0.0237 & -0.0685 \\ 0.0223 & -0.0303 \end{bmatrix} \qquad \gamma = 155.5400$$

Consequently,

$$H = LY^{-1} = \begin{bmatrix} -0.0934 & -0.4958 \\ -0.0140 & -0.1938 \end{bmatrix}$$

The control and states is represented in Figure (10)

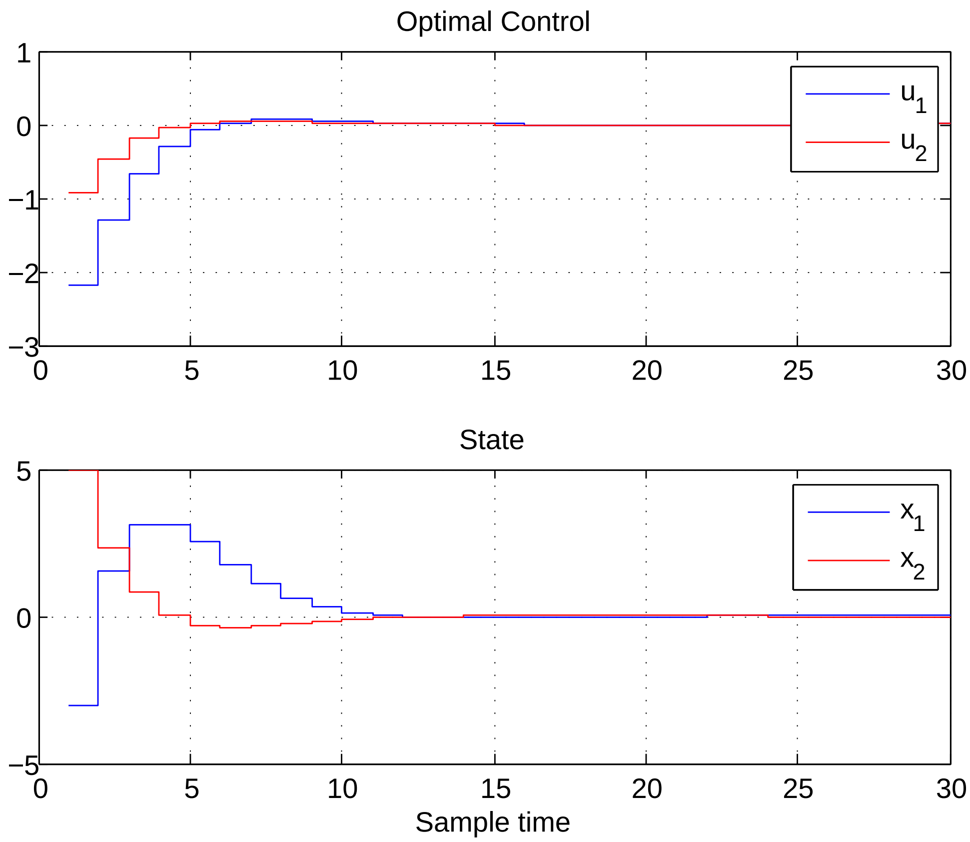

Figure 10: Optimal control and states in example 2

2.3.2 H_∞ Control via LMI

The system is modeled below:

$$x_{i+1} = Ax_i + Bu_i \tag{35}$$
$$z_i = C_z x_i + D_{zu} u_i \tag{36}$$

For this system, the well-known bounded real lemma (BRL) is stated as follows:

Lemma 1 (Bounded Real Lemma) *Let $\gamma > 0$. If there exists $X > 0$ such that*

$$\begin{bmatrix} -X^{-1} & A & B & 0 \\ A^T & -X & 0 & C_z^T \\ B^T & 0 & -\gamma W_u & D_{zu}^T \\ 0 & C_z & D_{zu} & -\gamma W_z^{-1} \end{bmatrix} < 0 \tag{37}$$

then

$$\frac{\sum_{i=i_0}^{\infty} z_i^T W_z z_i}{\sum_{i=i_0}^{\infty} u_i^T W_u u_i} < \gamma^2 \tag{38}$$

where u_i and z_i are input and output of system (35), (36).

The proof isn't difficult. It is utilizing Lyapunov function ($V(x_i) = x_i^T K x_i$) and in the proof procedure X will be equivalent to $\frac{1}{\sqrt{\gamma}}K$.
Now, we consider the system with disturbance.

$$x_{i+1} = Ax_i + Bu_i + B_w w_i, \quad x_0 = 0 \tag{39}$$
$$z_i = C_z x_i + D_{zu} u_i \tag{40}$$

The control law has to be satisfy $u_i = Hx_i$ (one of the constraints). Substituting in (39), (40)

$$x_{i+1} = (A + BH)x_i + B_w w_i, \quad x_0 = 0$$
$$z_i = (C_z + D_{zu}H)x_i$$

According to the BRL, H which guarantees $\|G(z)\|_\infty < \gamma$ should satisfy, for some $X > 0$,

$$\begin{bmatrix} -X^{-1} & (A+BH) & B_w & 0 \\ (A+BH)^T & -X & 0 & (C_z + D_{zu}H)^T \\ B_w^T & 0 & -\gamma I & 0 \\ 0 & (C_z + D_{zu}H) & 0 & -\gamma I \end{bmatrix} < 0 \tag{41}$$

Pre- and post-multiplying (41) by $diag(I, X^{-1}, I, I)$:

$$\begin{bmatrix} -X^{-1} & (A+BH)X^{-1} & B_w & 0 \\ X^{-1}(A+BH)^T & -X^{-1} & 0 & X^{-1}(C_z + D_{zu}H)^T \\ B_w^T & 0 & -\gamma I & 0 \\ 0 & (C_z + D_{zu}H)X^{-1} & 0 & -\gamma I \end{bmatrix} < 0 \qquad (42)$$

A change of variables such that:

$$Y = X^{-1} \text{ and } L = HX^{-1}$$

$$\begin{bmatrix} -Y & (AY+BL) & B_w & 0 \\ (AY+BL)^T & -Y & 0 & (C_zY + D_{zu}L)^T \\ B_w^T & 0 & -\gamma I & 0 \\ 0 & (C_zY + D_{zu}L) & 0 & -\gamma I \end{bmatrix} < 0 \qquad (43)$$

If the above problem be feasible, the H is a state feedback which guarantee $\|G(z)\|_\infty < \gamma$. Infinite horizon H_∞ control via LMI, can be summarized as following optimization problem:

$$\begin{aligned} &\text{minimize} \quad \gamma \\ &\text{subject to} \quad \begin{bmatrix} -Y & (AY+BL) & B_w & 0 \\ (AY+BL)^T & -Y & 0 & (C_zY + D_{zu}L)^T \\ B_w^T & 0 & -\gamma I & 0 \\ 0 & (C_zY + D_{zu}L) & 0 & -\gamma I \end{bmatrix} < 0 \end{aligned} \qquad (44)$$

and control obtained $H = LY^{-1}$.

Example 3 *Consider the problem Example 2, but already H_∞ problem :*

$$\begin{aligned} x[k+1] &= \begin{bmatrix} 0.8 & 0.75 \\ 0 & 1 \end{bmatrix} x[k] + \begin{bmatrix} 0 & -0.2 \\ 1 & 0.5 \end{bmatrix} u[k] + \begin{bmatrix} 0.01 & -0.032 \\ 0.101 & -0.05 \end{bmatrix} w[k] \\ y[k] &= \begin{bmatrix} 1 & -0.1 \end{bmatrix} x[k] + \begin{bmatrix} 0.1 & -0.05 \end{bmatrix} u[k] \end{aligned}$$

Suppose that the initial condition be $x_0 = [-3 \quad 5]^T$. We want to find control which minimizes the $\|G(z)\|_\infty$.

Here we obtain matrices Y and L and parameter γ as below:

$$Y = \begin{bmatrix} 959.0 & 735.4 \\ 735.4 & 1183.2 \end{bmatrix} \qquad L = \begin{bmatrix} -4888.0 & -3530.8 \\ 7933.0 & 5279.0 \end{bmatrix} \qquad \gamma = +1.63456e - 005$$

Consequently,

$$H = LY^{-1} = \begin{bmatrix} -5.3661 & 0.3509 \\ 9.2678 & -1.2983 \end{bmatrix}$$

The state feedback:

$$u_i = Hx_i$$

The control and states is represented in Figure (11)

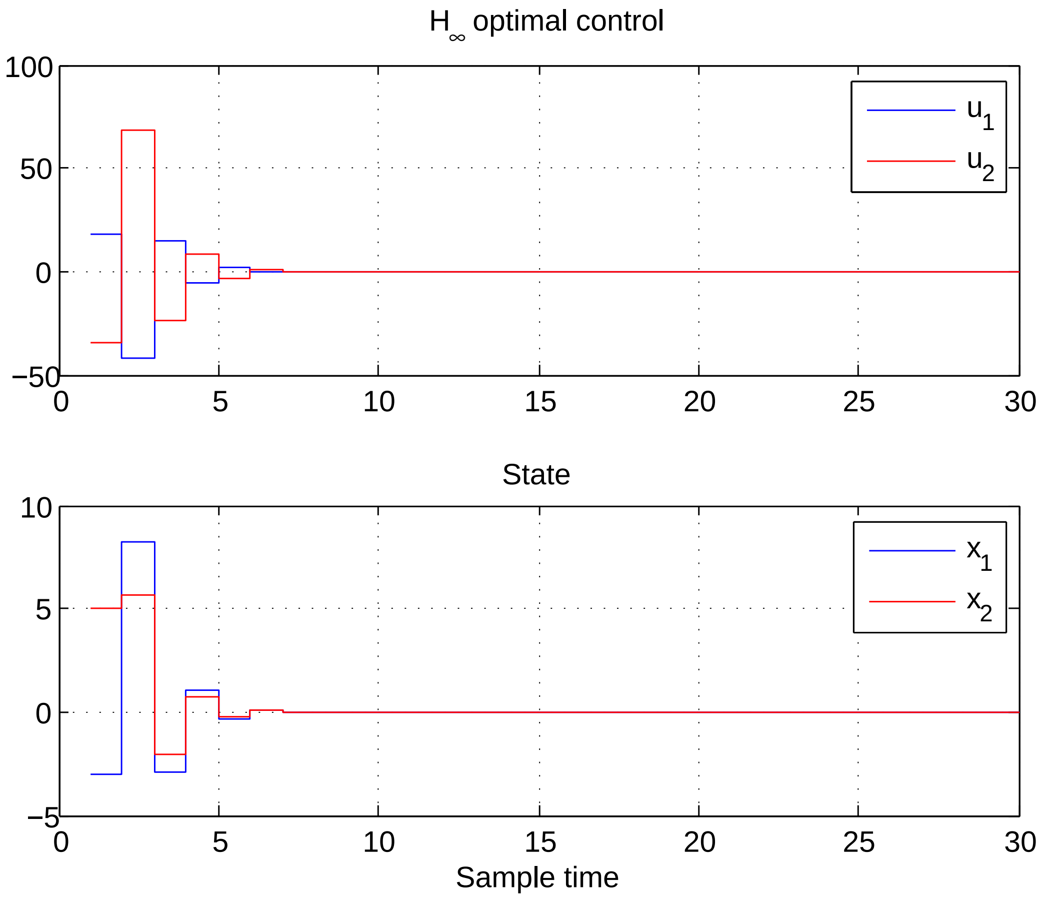

Figure 11: H_∞ optimal control and states in example 3

3 State Feedback in Model Predictive Control

3.1 LQR via LMI in Predictive Form

First we focus on free terminal state. From discrete time system relations, *i.e.*

$$\begin{aligned} x_{k+j+1|k} &= Ax_{k+j|k} + Bu_{k+j|k} \\ z_{k+j|k} &= C_z x_{k+j|k} \qquad\qquad j \in [0, N-1] \end{aligned}$$

we can write the system in the time interval $[k, k+N]$.

$$\begin{aligned} x_{k|k} &= x_{k|k} \\ x_{k+1|k} &= Ax_{k|k} + Bu_{k|k} \\ x_{k+2|k} &= Ax_{k+1|k} + Bu_{k+1|k} \\ &\vdots \\ x_{k+N|k} &= Ax_{k+N-1|k} + Bu_{k+N-1|k} \end{aligned}$$

Substituting first line in second, then second in third, ...

$$\begin{aligned} x_{k|k} &= x_{k|k} \\ x_{k+1|k} &= Ax_{k|k} + Bu_{k|k} \\ x_{k+2|k} &= A(Ax_{k|k} + Bu_{k|k}) + Bu_{k+1|k} \\ &= A^2 x_{k|k} + ABu_{k|k} + Bu_{k+1|k} \\ x_{k+3|k} &= A(A^2 x_{k|k} + ABu_{k|k} + Bu_{k+1|k}) + Bu_{k+2|k} \\ &= A^3 x_{k|k} + A^2 Bu_{k|k} + ABu_{k+1|k} + Bu_{k+2|k} \\ &\vdots \\ x_{k+N-1|k} &= A^{N-1} x_{k|k} + A^{N-2} Bu_{k|k} + \cdots + Bu_{k+N-2|k} \\ x_{k+N|k} &= A^N x_{k|k} + A^{N-1} Bu_{k|k} + \cdots + Bu_{k+N-1|k} \end{aligned}$$

There is one state which should be considered; $x_{k|k}$. If we introduce new matrices:

$$F = \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^{N-1} \end{bmatrix} \quad \text{and} \quad H = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ A^{N-2}B & A^{N-3}B & \cdots & B & 0 \end{bmatrix} \tag{45}$$

Then we can use transformations

$$U_k = \begin{bmatrix} u_{k|k} \\ u_{k+1|k} \\ u_{k+2|k} \\ \vdots \\ u_{k+N-1|k} \end{bmatrix} \quad \text{and} \quad X_k = \begin{bmatrix} x_{k|k} \\ x_{k+1|k} \\ x_{k+2|k} \\ \vdots \\ x_{k+N-1|k} \end{bmatrix} \tag{46}$$

Now, we can write:

$$X_k = Fx_{k|k} + HU_k \tag{47}$$

The terminal state is given by

$$x_{k+N|k} = A^N x_{k|k} + A^{N-1}Bu_{k|k} + \cdots + Bu_{k+N-1|k} \tag{48}$$

where can be represented by

$$\begin{aligned} x_{k+N|k} &= A^N x_{k|k} + [A^{N-1}B, A^{N-2}B, \cdots, B] \begin{bmatrix} u_{k|k} \\ u_{k+1|k} \\ u_{k+2|k} \\ \vdots \\ u_{k+N-1|k} \end{bmatrix} \\ &= A^N x_{k|k} + \bar{B}U_k \end{aligned} \tag{49}$$

$$\bar{B} = [A^{N-1}B, A^{N-2}B, \cdots, B]$$

Let us define

$$\bar{Q}_N = diag\{\underbrace{Q, \cdots, Q}_{N}\}, \quad \text{and} \quad \bar{R}_N = diag\{\underbrace{R, \cdots, R}_{N}\} \tag{50}$$

The cost function can be rewritten by

$$\begin{aligned} J(x_k, U_k) = {} & [X_k - X_k^r]^T \bar{Q}_N [X_k - X_k^r] + U_k^T \bar{R}_N U_k \\ & + (x_{k+N|k} - x_{k+N}^r)^T Q_f (x_{k+N|k} - x_{k+N}^r) \end{aligned} \tag{51}$$

where

$$X_k^r = \begin{bmatrix} x_k^r \\ x_{k+1}^r \\ x_{k+2}^r \\ \vdots \\ x_{k+N-1}^r \end{bmatrix}$$

If we put the equation (47) and (49) in (51), then

$$\begin{aligned} J(x_k, U_k) &= [Fx_{k|k} + HU_k - X_k^r]^T \bar{Q}_N [Fx_{k|k} + HU_k - X_k^r] + U_k^T \bar{R}_N U_k \\ &\quad + (A^N x_{k|k} + \bar{B}U_k - x_{k+N}^r)^T Q_f (A^N x_{k|k} + \bar{B}U_k - x_{k+N}^r) \\ \\ &= U_k^T [H^T \bar{Q}_N H + \bar{R}_N] U_k + 2[Fx_{k|k} - X_k^r]^T \bar{Q}_N H U_k \\ &\quad + [Fx_{k|k} - X_k^r]^T \bar{Q}_N [Fx_{k|k} - X_k^r] \\ &\quad + (A^N x_{k|k} + \bar{B}U_k - x_{k+N}^r)^T Q_f (A^N x_{k|k} + \bar{B}U_k - x_{k+N}^r) \\ \\ &= U_k^T W U_k + w^T U_k + [Fx_{k|k} - X_k^r]^T \bar{Q}_N [Fx_{k|k} - X_k^r] \\ &\quad + (A^N x_{k|k} + \bar{B}U_k - x_{k+N}^r)^T Q_f (A^N x_{k|k} + \bar{B}U_k - x_{k+N}^r) \end{aligned} \tag{52}$$

where $W = H^T \bar{Q}_N H$ and $w^T = 2[Fx_{k|k} - X_k^r]^T \bar{Q}_N H$.
The optimal input can be obtained by taking $\frac{\partial J(x_k, U_k)}{\partial U_k} = 0$. So

$$\begin{aligned}
\frac{\partial J(x_k, U_k)}{\partial U_k} &= W^T U_k + W U_k + w + 2\bar{B}^T Q_f (A^N x_{k|k} + \bar{B} U_k - x_{k+N}^r) \\
&= 2W U_k + w + 2\bar{B}^T Q_f \bar{B} U_k + 2\bar{B}^T Q_f (A^N x_{k|k} - x_{k+N}^r) \\
&= 2[W + \bar{B}^T Q_f \bar{B}] U_k + 2H^T \bar{Q}_N [Fx_{k|k} - X_k^r] + 2\bar{B}^T Q_f (A^N x_{k|k} - x_{k+N}^r) \\
&= 0
\end{aligned}$$

$$U_k = -[W + \bar{B}^T Q_f \bar{B}]^{-1} \Big[H^T \bar{Q}_N (Fx_{k|k} - X_k^r) + \bar{B}^T Q_f (A^N x_{k|k} - x_{k+N}^r) \Big] \tag{53}$$

The optimal control can be obtained as

$$u_k = [1, 0, \cdots, 0] U_k \tag{54}$$

In order to obtain an LMI form, we decompose the cost function (52) into two parts

$$J(x_k, U_k) = J_1(x_k, U_k) + J_2(x_k, U_k)$$

where

$$\begin{aligned}
J_1(x_k, U_k) &= U_k^T W U_k + w^T U_k + [Fx_{k|k} - X_k^r]^T \bar{Q}_N [Fx_{k|k} - X_k^r] \\
J_2(x_k, U_k) &= (A^N x_{k|k} + \bar{B} U_k - x_{k+N}^r)^T Q_f (A^N x_{k|k} + \bar{B} U_k - x_{k+N}^r)
\end{aligned}$$

We assume that

$$\begin{aligned}
J_1(x_k, U_k) &\leq \gamma_1 \\
J_2(x_k, U_k) &\leq \gamma_2
\end{aligned}$$

or

$$U_k^T W U_k + w^T U_k + [Fx_{k|k} - X_k^r]^T \bar{Q}_N [Fx_{k|k} - X_k^r] \leq \gamma_1 \tag{55}$$
$$(A^N x_{k|k} + \bar{B} U_k - x_{k+N}^r)^T Q_f (A^N x_{k|k} + \bar{B} U_k - x_{k+N}^r) \leq \gamma_2 \tag{56}$$

Hence

$$J(x_k, U_k) \leq \gamma_1 + \gamma_2 \tag{57}$$

From Schur complement, (55) and (56) are equivalent to

$$\begin{bmatrix} \gamma_1 - w^T U_k - [Fx_{k|k} - X_k^r]^T \bar{Q}_N [Fx_{k|k} - X_k^r] & U_k^T \\ U_k & W^{-1} \end{bmatrix} \geq 0 \tag{58}$$

and

$$\begin{bmatrix} \gamma_2 & (A^N x_{k|k} + \bar{B} U_k - x_{k+N}^r) \\ (A^N x_{k|k} + \bar{B} U_k - x_{k+N}^r)^T & Q_f^{-1} \end{bmatrix} \geq 0 \tag{59}$$

The optimal solution U_k^* can be obtained by an LMI problem as follows:

minimize $\quad \gamma_1 + \gamma_2$

$$\text{subject to} \quad \begin{bmatrix} \gamma_1 - w^T U_k - [F x_{k|k} - X_k^r]^T \bar{Q}_N [F x_{k|k} - X_k^r] & U_k^T \\ U_k & W^{-1} \end{bmatrix} \geq 0$$

$$\begin{bmatrix} \gamma_2 & (A^N x_{k|k} + \bar{B} U_k - x_{k+N}^r) \\ (A^N x_{k|k} + \bar{B} U_k - x_{k+N}^r)^T & Q_f^{-1} \end{bmatrix} \geq 0$$

Then optimal control can be obtained by $u_k^* = [1, 0, \cdots, 0] U_k^*$.

For the fixed terminal state, the inequality (56) isn't needed and it changes to equality form

$$A^N x_{k|k} + \bar{B} U_k = x_{k+N}^r \tag{60}$$

Thus, we should solve following SDP:

minimize $\quad \gamma_1$

$$\text{subject to} \quad \begin{bmatrix} \gamma_1 - w^T U_k - [F x_{k|k} - X_k^r]^T \bar{Q}_N [F x_{k|k} - X_k^r] & U_k^T \\ U_k & W^{-1} \end{bmatrix} \geq 0$$

$$A^N x_{k|k} + \bar{B} U_k = x_{k+N}^r$$

Example 4 *Consider following system in free terminal state:*

$$x[k+1] = \begin{bmatrix} 1 & 0.75 \\ 0 & 0.8 \end{bmatrix} x[k] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u[k]$$

with cost function weighting matrices, $Q = I$, $R = 10$ and $Q_f = 5I$. We want optimal control to track the reference signals

$$x_1^r[k] = sin[k/10] \quad and \quad x_2^r[k] = sin[k/10+1]$$

Receding horizon is equal to 5, i.e. *N=5*
We use control (53) to solve the problem. The states and optimal control are depicted in Figures (12) and (13) respectively. Figure (12) illustrates that, the

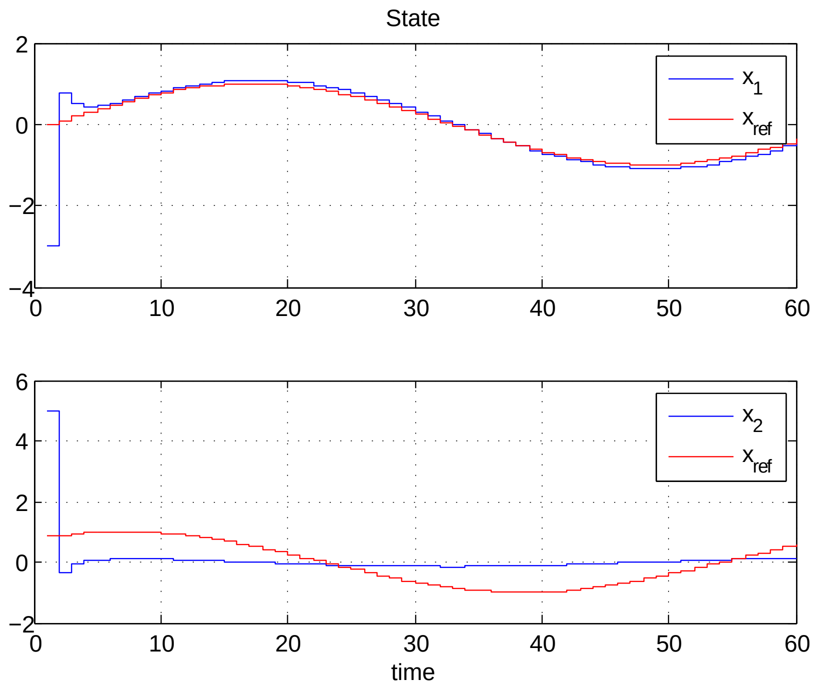

Figure 12: Tracking optimal control states in predictive form

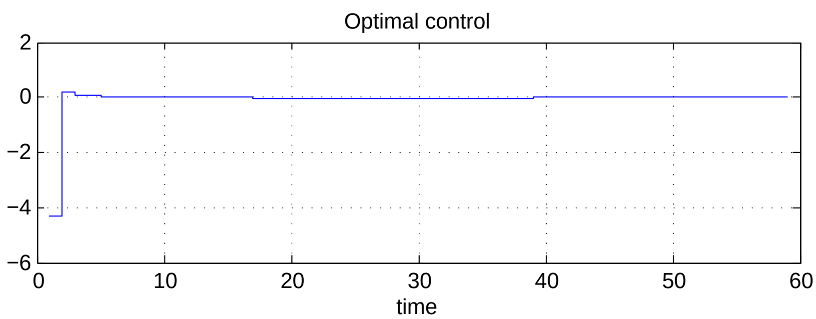

Figure 13: Optimal control in predictive form

state x_1 is tracked well, but x_2 isn't. I encountered an unexpected result. When we moved one of the eigenvalues of matrix A outside the unit circle, tracking result in state x_2 improved! For example the eigenvalues of following system are in 3.7 and 0.8.

$$x[k+1] = \begin{bmatrix} 3.7 & 0.75 \\ 0 & 0.8 \end{bmatrix} x[k] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u[k]$$

System has a unstable pole, but control law and state x_2 are bounded, but tracking in x_1 distorts.Figures (14) and (15) demonstrate the results.

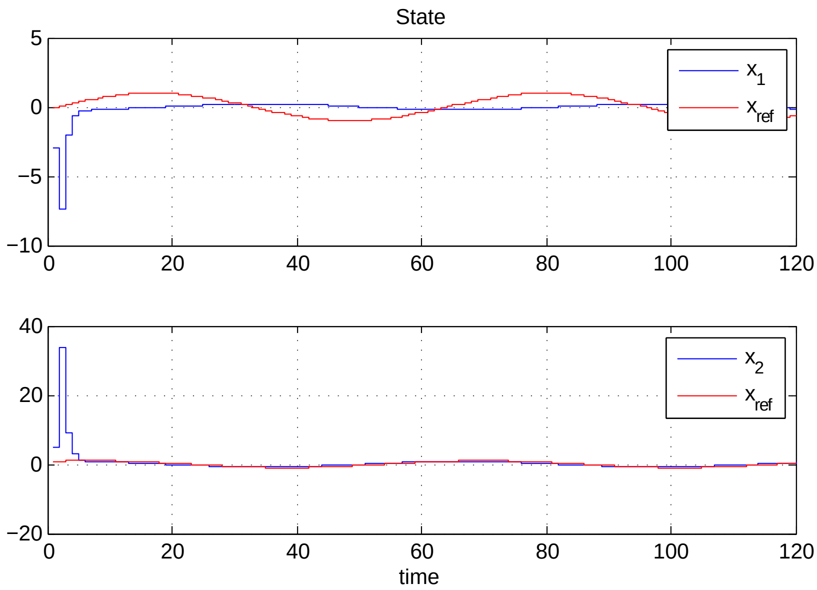

Figure 14: Tracking control in predictive form; new system

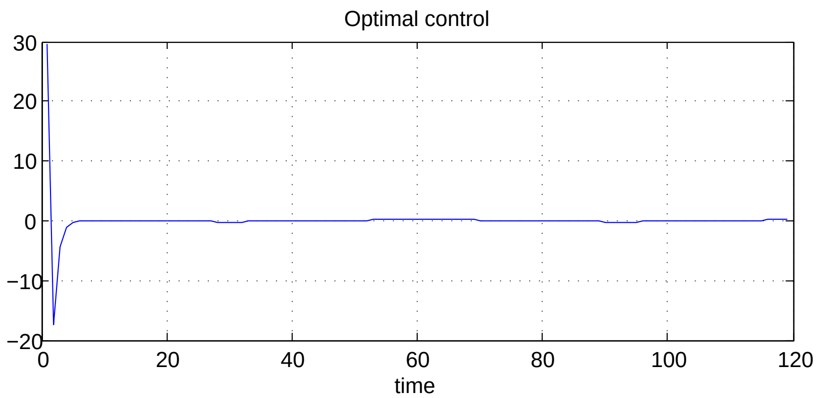

Figure 15: Optimal control in predictive form; new system

3.2 H_∞ control via LMI in Predictive Form

From discrete time system relations, *i.e.*

$$\begin{aligned} x_{k+j+1|k} &= Ax_{k+j|k} + Bu_{k+j|k} + B_w w_{k+j|k} \\ z_{k+j|k} &= C_z x_{k+j|k} \qquad\qquad j \in [0, N-1] \end{aligned}$$

we can have new representation:

$$\begin{aligned} x_{k|k} &= x_{k|k} \\ x_{k+1|k} &= Ax_{k|k} + Bu_{k|k} + B_w w_{k|k} \\ x_{k+2|k} &= A(Ax_{k|k} + Bu_{k|k} + B_w w_{k|k}) + Bu_{k+1|k} + B_w w_{k+1|k} \\ &= A^2 x_{k|k} + ABu_{k|k} + AB_w w_{k|k} + Bu_{k+1|k} + B_w w_{k+1|k} \\ x_{k+3|k} &= A(A^2 x_{k|k} + ABu_{k|k} + AB_w w_{k|k} + Bu_{k+1|k} + B_w w_{k+1|k}) + Bu_{k+2|k} + B_w w_{k+2|k} \\ &= A^3 x_{k|k} + A^2 Bu_{k|k} + A^2 B_w w_{k|k} + ABu_{k+1|k} + AB_w w_{k+1|k} + Bu_{k+2|k} + B_w w_{k+2|k} \\ &\vdots \\ x_{k+N-1|k} &= A^{N-1} x_{k|k} + A^{N-2} Bu_{k|k} + A^{N-2} B_w w_{k|k} + \cdots + Bu_{k+N-2|k} + B_w w_{k+N-2|k} \\ x_{k+N|k} &= A^N x_{k|k} + A^{N-1} Bu_{k|k} + A^{N-1} Bw_{k|k} + \cdots + Bu_{k+N-1|k} + B_w w_{k+N-1|k} \end{aligned}$$

$$X_k = Fx_{k|k} + HU_k + H_w W_k \tag{61}$$

where X_k, F, H and U_k are same as mentioned before, but

$$W_k = \begin{bmatrix} w_{k|k} \\ w_{k+1|k} \\ w_{k+2|k} \\ \vdots \\ w_{k+N-1|k} \end{bmatrix} \quad \text{and} \quad H_w = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ B_w & 0 & 0 & \cdots & 0 \\ AB_w & B_w & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ A^{N-2}B_w & A^{N-3}B_w & \cdots & B_w & 0 \end{bmatrix} \tag{62}$$

The H_∞ performance criterion can be written as

$$\begin{aligned} J(x_k, U_k, W_k) = {} & [Fx_{k|k} + HU_k + H_w W_k - X_k^r]^T \bar{Q}_N [Fx_{k|k} + HU_k + H_w W_k - X_k^r] \\ & + (A^N x_{k|k} + \bar{B}U_k + \bar{B}_w W_k - x_{k+N}^r)^T Q_f (A^N x_{k|k} + \bar{B}U_k + \bar{B}_w W_k - x_{k+N}^r) \\ & + U_k^T \bar{R}_N U_k - \gamma^2 W_k^T W_k \end{aligned} \tag{63}$$

Representing $J(x_k, U_k, W_k)$ in quadratic form

$$J(x_k, U_k, W_k) = [\mathcal{V}_1 W_k + \mathcal{V}_2]^T \mathcal{V}_1^{-1} [\mathcal{V}_1 W_k + \mathcal{V}_2] + U_k^T \mathcal{P}_1 U_k + 2U_k^T \mathcal{P}_2 + \mathcal{P}_3 \tag{64}$$

where

$$\mathcal{V}_1 \triangleq -\gamma^2 I + \bar{B}_w^T Q_f \bar{B}_w + H_w^T \bar{Q}_N H_w \tag{65}$$

$$\mathcal{V}_2 \triangleq H_w^T \bar{Q}_N^T [Fx_{k|k} + HU_k - X_k^r] + \bar{B}_w^T Q_f^T [A^N x_{k|k} + \bar{B}U_k - x_{k+N}^r] \tag{66}$$

$$\mathcal{P}_1 \triangleq -(H_w^T \bar{Q}_N^T H + \bar{B}_w^T Q_f^T \bar{B})^T \mathcal{V}_1^{-1} (H_w^T \bar{Q}_N^T H + \bar{B}_w^T Q_f^T \bar{B}) \tag{67}$$

$$+ H^T \bar{Q}_N H + \bar{B}_w^T Q_f \bar{B} + \bar{R}_N \tag{68}$$

$$\mathcal{P}_2 \triangleq -(H_w^T \bar{Q}_N^T H + \bar{B}_w^T Q_f^T \bar{B})^T \mathcal{V}_1^{-1} [H_w^T \bar{Q}_N^T (Fx_{k|k} - X_k^r) \tag{69}$$

$$+ \bar{B}_w^T Q_f^T (A^N x_{k|k} - x_{k+N}^r)] + H^T \bar{Q}_N F x_{k|k} + \bar{B}^T Q_f A^N x_{k|k} \tag{70}$$

$\mathcal{P}_3$ is a constant and hasn't any role in minimization or minimax. It can be shown as two cost function:

$$J(x_k, U_k, W_k) = J_1(x_k, U_k, W_k) + J_2(x_k, U_k, W_k)$$

where

$$\begin{aligned} J_1(x_k, U_k, W_k) &= [\mathcal{V}_1 W_k + \mathcal{V}_2]^T \mathcal{V}_1^{-1} [\mathcal{V}_1 W_k + \mathcal{V}_2] \\ J_2(x_k, U_k, W_k) &= U_k^T \mathcal{P}_1 U_k + 2U_k^T \mathcal{P}_2 + \mathcal{P}_3 \end{aligned}$$

we should try to maximize $[\mathcal{V}_1 W_k + \mathcal{V}_2]^T \mathcal{V}_1^{-1} [\mathcal{V}_1 W_k + \mathcal{V}_2]$ and minimize $U_k^T \mathcal{P}_1 U_k + 2U_k^T \mathcal{P}_2 + \mathcal{P}_3$.
In order to have an LMI, we have to try to minimize

$$-[\mathcal{V}_1 W_k + \mathcal{V}_2]^T \mathcal{V}_1^{-1} [\mathcal{V}_1 W_k + \mathcal{V}_2]$$

means, $\exists\ \gamma_2$

$$-[\mathcal{V}_1 W_k + \mathcal{V}_2]^T \mathcal{V}_1^{-1} [\mathcal{V}_1 W_k + \mathcal{V}_2] \leq \gamma_2 \tag{71}$$

and $\exists\ \gamma_1$

$$U_k^T \mathcal{P}_1 U_k + 2U_k^T \mathcal{P}_2 \leq \gamma_1 \tag{72}$$

Using Schur complement in the equations (71) and (72) yields

$$\begin{bmatrix} \gamma_1 - 2U_k^T \mathcal{P}_2 & U_k^T \\ U_k & \mathcal{P}_1^{-1} \end{bmatrix} \geq 0 \tag{73}$$

and

$$\begin{bmatrix} \gamma_2 & (\mathcal{V}_1 W_k + \mathcal{V}_2)^T \\ (\mathcal{V}_1 W_k + \mathcal{V}_2) & -\mathcal{V}_1 \end{bmatrix} \geq 0 \tag{74}$$

The SDP problem can be represented by:
minimize $\quad \gamma_1 + \gamma_2$

subject to $\begin{bmatrix} \gamma_1 - 2U_k^T \mathcal{P}_2 & U_k^T \\ U_k & \mathcal{P}_1^{-1} \end{bmatrix} \geq 0$

$$\begin{bmatrix} \gamma_2 & (\mathcal{V}_1 W_k + \mathcal{V}_2)^T \\ (\mathcal{V}_1 W_k + \mathcal{V}_2) & -\mathcal{V}_1 \end{bmatrix} \geq 0$$

What remains to do is just to pick up the first one among U_k as in (54)

4 Filters in MPC

4.1 Kalman Filter

Here, we consider the following stochastic model:

$$x_{i+1} = Ax_i + Bu_i + Gw_i \tag{75}$$

$$y_i = Cx_i + v_i \tag{76}$$

At the initial time i_0, the state x_{i_0} is a Gaussian random variable with a mean $\bar{x}_{i_0}$ and a covariance P_{i_0}. The system noise w_i and the measurement noise v_i are zero-mean white Gaussian and mutually uncorrelated. The covariances of w_i and v_i are denoted by Q_w and R_v respectively, which are assumed to be positive definite matrices. We assume that these noises are uncorrelated with the initial state x_{i_0}.

In practice, the state may not be available, so it should be estimated from measured outputs and known inputs. Thus, a state estimator, or filter is needed. This filter can be used for an output feedback control. We should estimate the state x_i from measured data and known inputs so that the error between the real state and the estimated state is minimized.

The Kalman filter, is derived for the following performance criterion:

$$E[(x_i - \hat{x}_{i|i})^T (x_i - \hat{x}_{i|i})|Y_i] \tag{77}$$

where $\hat{x}_{i|j}$ is denoted by the estimated value at time i based on the measurement up to j and $Y_i = [y_{i_0}, \cdots, y_i]^T$.
$\hat{x}_{i+1|i}$ and $\hat{x}_{i|i}$ are often called a predictive estimated value and a filtered estimated value respectively.

We want to find a relation which get us estimated $\hat{x}_{i+1|i}$ from previous inputs and outputs. By the definition of the conditional probability,

$$p(x_i|Y_i) = \frac{p(x_i, Y_i)}{p(Y_i)} = \frac{p(x_i, y_i, Y_{i-1})}{p(y_i, Y_{i-1})} \tag{78}$$

The numerator can be represented as

$$\begin{aligned} p(x_i, y_i, Y_{i-1}) &= p(y_i|x_i, Y_{i-1})p(x_i, Y_{i-1}) \\ &= p(y_i|x_i, Y_{i-1})p(x_i|Y_{i-1})p(Y_{i-1}) \\ &= p(y_i|x_i)p(x_i|Y_{i-1})p(Y_{i-1}) \end{aligned} \tag{79}$$

where the last equality comes from the fact that if x_i is given, then the Y_{i-1} is redundant information. Substituting (79) in (78) yields

$$\begin{aligned} p(x_i|Y_i) = \frac{p(x_i, y_i, Y_{i-1})}{p(y_i, Y_{i-1})} &= \frac{p(y_i|x_i)p(x_i|Y_{i-1})p(Y_{i-1})}{p(y_i|Y_{i-1})p(Y_{i-1})} \\ &= \frac{p(y_i|x_i)p(x_i|Y_{i-1})}{p(y_i|Y_{i-1})} \end{aligned} \tag{80}$$

Since the Y_i is given, the denominator is fixed, and about the nominator; it can be evaluated from statistical information. For the given x_i, y_i follows the normal distribution.

$$y_i \sim \mathcal{N}(Cx_i, R_v) \tag{81}$$

Since $E[x_i|Y_{i-1}] = \hat{x}_{i|i-1}$ and $E[(x_i - \hat{x}_{i|i-1})^T(x_i - \hat{x}_{i|i-1})|Y_{i-1}] = P_{i|i-1}$, the conditional probability $p(x_i|Y_{i-1})$ is normal, *i.e.* $\mathcal{N}(\hat{x}_{i|i-1}, P_{i|i-1})$. Thus, we have

$$\begin{aligned} p(y_i|x_i) &= \frac{1}{\sqrt{(2\pi)^m|R_v|}} exp\Big\{ -\frac{1}{2}[y_i - Cx_i]^T R_v^{-1}[y_i - Cx_i]\Big\} \\ p(x_i|Y_{i-1}) &= \frac{1}{\sqrt{(2\pi)^n|P_{i|i-1}|}} exp\Big\{ -\frac{1}{2}[x_i - \hat{x}_{i|i-1}]^T P_{i|i-1}^{-1}[x_i - \hat{x}_{i|i-1}]\Big\} \end{aligned}$$

From (80) we have

$$\begin{aligned} p(x_i|Y_i) = &\frac{1}{p(y_i|Y_{i-1})} \frac{1}{\sqrt{(2\pi)^m|R_v|}} \frac{1}{\sqrt{(2\pi)^n|P_{i|i-1}|}} \\ &\times exp\Big\{ -\frac{1}{2}[y_i - Cx_i]^T R_v^{-1}[y_i - Cx_i]\Big\} \\ &\times exp\Big\{ -\frac{1}{2}[x_i - \hat{x}_{i|i-1}]^T P_{i|i-1}^{-1}[x_i - \hat{x}_{i|i-1}]\Big\} \end{aligned} \tag{82}$$

Since the Gaussian probability density function has a peak value at the average, we will find x_i that sets the derivative of (82) to zero. Thus

$$\frac{dp(x_i|Y_i)}{dx_i} = 0 \quad \Rightarrow \quad C^T R_v^{-1}(y_i - Cx_i) - P_{i|i-1}^{-1}(x_i - \hat{x}_{i|i-1}) = 0$$

We denote x_i with $\hat{x}_{i|i}$, therefore

$$\begin{aligned} \hat{x}_{i|i} &= [P_{i|i-1}^{-1} + C^T R_v^{-1} C]^{-1} P_{i|i-1}^{-1}\hat{x}_{i|i-1} + [P_{i|i-1}^{-1} + C^T R_v^{-1} C]^{-1} C^T R_v^{-1} y_i && (83) \\ &= [I + P_{i|i-1} C^T R_v^{-1} C]^{-1}\hat{x}_{i|i-1} + [I + P_{i|i-1} C^T R_v^{-1} C]^{-1} P_{i|i-1} C^T R_v^{-1} y_i && (84) \\ &= \Big[I - P_{i|i-1} C^T (C P_{i|i-1} C^T + R_v)^{-1} C\Big]\hat{x}_{i|i-1} && (85) \\ &\quad + P_{i|i-1} C^T R_v^{-1}(I + C P_{i|i-1} C^T R_v^{-1})^{-1} y_i && (86) \\ &= \Big[I - P_{i|i-1} C^T (C P_{i|i-1} C^T + R_v)^{-1} C\Big]\hat{x}_{i|i-1} && (87) \\ &\quad + P_{i|i-1} C^T (R_v + C P_{i|i-1} C^T)^{-1} y_i && (88) \\ &= \hat{x}_{i|i-1} - P_{i|i-1} C^T (C P_{i|i-1} C^T + R_v)^{-1} C\hat{x}_{i|i-1} && (89) \\ &\quad + P_{i|i-1} C^T (R_v + C P_{i|i-1} C^T)^{-1} y_i && (90) \\ &= \hat{x}_{i|i-1} + P_{i|i-1} C^T (R_v + C P_{i|i-1} C^T)^{-1}(y_i - C\hat{x}_{i|i-1}) && (91) \\ &= \hat{x}_{i|i-1} + K_i(y_i - C\hat{x}_{i|i-1}) && (92) \end{aligned}$$

where

$$K_i \triangleq P_{i|i-1} C^T (R_v + C P_{i|i-1} C^T)^{-1} \tag{93}$$

The coefficient of $\hat{x}_{i|i-1}$ from (84) to (85) we used the matrix inversion lemma[1]. The coefficient of y_i from (84) to (86) we used another matrix lemma [2] $\hat{x}_{i+1|i}$ can be easily found

$$\hat{x}_{i+1|i} = E[x_{i+1}|Y_i] = AE[x_i|Y_i] + Bu_i + GE[w_i|Y_i] \tag{94}$$
$$= A\hat{x}_{i|i} + Bu_i \tag{95}$$
$$= A\Big[\hat{x}_{i|i-1} + K_i(y_i - C\hat{x}_{i|i-1})\Big] + Bu_i \tag{96}$$

$P_{i+1|i}$ can be obtained recursively from the error dynamic equations. We define error as $\tilde{x}_{i|i} \triangleq \hat{x}_{i|i} - x_i$ and $\tilde{x}_{i|i-1} = \hat{x}_{i|i-1} - x_i$. Substituting $\hat{x}_{i|i}$ and $\hat{x}_{i|i-1}$ with $\tilde{x}_{i|i} + x_i$ and $\tilde{x}_{i|i-1} + x_i$ in the equation (92) respectively and with regard to (76), yields

$$\tilde{x}_{i|i} = [I - K_iC]\tilde{x}_{i|i-1} + K_iv_i \tag{97}$$

and in equation (96) with regard to (75)

$$\tilde{x}_{i+1|i} = A\tilde{x}_{i|i} - Gw_i \tag{98}$$

From (97)

$$\begin{aligned}
P_{i|i} = E[\tilde{x}_{i|i}\tilde{x}_{i|i}^T] &= E\Big[([I - K_iC]\tilde{x}_{i|i-1} + K_iv_i)([I - K_iC]\tilde{x}_{i|i-1} + K_iv_i)^T\Big] \\
&= (I - K_iC)E[\tilde{x}_{i|i-1}\tilde{x}_{i|i-1}^T](I - K_iC)^T + K_iE[v_iv_i^T]K_i \\
&= (I - K_iC)P_{i|i-1}(I - K_iC)^T + K_iR_vK_i \\
&= (I - K_iC)P_{i|i-1} - (I - K_iC)P_{i|i-1}C^TK_i + K_iR_vK_i \\
&= (I - K_iC)P_{i|i-1} - P_{i|i-1}C^TK_i + K_iCP_{i|i-1}C^TK_i + K_iR_vK_i \\
&= (I - K_iC)P_{i|i-1} - P_{i|i-1}C^TK_i + K_i(CP_{i|i-1}C^T + R_v)K_i \\
\overset{K_i = P_{i|i-1}C^T(R_v + CP_{i|i-1}C^T)^{-1}}{\Longrightarrow} &= (I - K_iC)P_{i|i-1} - P_{i|i-1}C^TK_i + P_{i|i-1}C^TK_i \\
&= (I - K_iC)P_{i|i-1}
\end{aligned} \tag{99}$$

And from (98) and (99)

$$\begin{aligned}
P_{i+1|i} = E[\tilde{x}_{i+1|i}\tilde{x}_{i+1|i}^T] &= E\Big[(A\tilde{x}_{i|i} - Gw_i)(A\tilde{x}_{i|i} - Gw_i)^T\Big] \\
&= AE[\tilde{x}_{i|i}\tilde{x}_{i|i}^T]A^T + GE[w_iw_i^T]G^T \\
&= AP_{i|i}A^T + GQ_wG^T \\
&= A(I - K_iC)P_{i|i-1}A^T + GQ_wG^T \\
&= AP_{i|i-1}A^T + GQ_wG^T \\
&\quad - AP_{i|i-1}C^T(R_v + CP_{i|i-1}C^T)^{-1}CP_{i|i-1}A^T
\end{aligned} \tag{100}$$

[1] Matrix Inversion Lemma: $(A + BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1}$ in this case $A = I$, $B = P_{i|i-1}C^T$, $C = R_v^{-1}$ and $D = C$

[2] $A(I + BA)^{-1} = (I + AB)^{-1}A$

The initial values $\hat{x}_{i_0|i_0-1}$ and $P_{i_0|i_0-1}$ are given by $E[x_{i_0}]$ and $E[(\hat{x}_{i_0} - x_{i_0})(\hat{x}_{i_0} - x_{i_0})^T]$, which are a priori knowledge. If we represent the matrix P_i instead of $P_{i|i-1}$, then we can summarize Kalman filter as following:

- $$\hat{x}_{i+1|i} = A\hat{x}_{i|i-1} + AP_iC^T(R_v + CP_iC^T)^{-1}(y_i - C\hat{x}_{i|i-1})$$

- $$P_{i+1} = AP_iA^T + GQ_wG^T - AP_iC^T(R_v + CP_iC^T)^{-1}CP_iA^T$$

-Note that we didn't consider the term 'Bu_i' in the first line of summarized Kalman filter, because it is not a part of our estimation.

Example 5 *Consider the state feedback control with fixed terminal state and problem is same as expressed before, but in this case we compute the states from Kalman filter,*

$$x[k+1] = \begin{bmatrix} -0.3252 & 0.6504 & 0.6098 \\ 0.8130 & 0 & 0.8130 \\ 0.3659 & 0.2439 & -0.0813 \end{bmatrix} x[k] + \begin{bmatrix} 0.1 & -0.2 \\ 1 & 0.5 \\ 0 & 1 \end{bmatrix} u[k] + \begin{bmatrix} 0.9 \\ 0 \\ -0.255 \end{bmatrix} w[k]$$

$$y[k] = \begin{bmatrix} 1 & 0 & 0.15 \\ 0 & -1 & 0.2 \end{bmatrix} x[k] + v[k]$$

Suppose that the initial condition be $x_0 = [-3 \quad 5 \quad 2]^T$*. The weighting matrices are given as following:*

$$Q = I, \qquad Q_f = 5I, \qquad R = 10I$$

We want control to track our desired signal. Reference signals are

$$x_1^r[k] = \sin(k/10), \qquad x_2^r[k] = \sin(k/10 + 1), \qquad x_3^r[k] = \sin(k/10)$$

We generated the w_i *and* v_i *using MATLAB command* **`normrnd`** *and their means were zero and their variances were* 0.265 *and* 0.720 *respectively,* i.e.

$$w_i \sim \mathcal{N}(0, 0.265)$$
$$v_i \sim \mathcal{N}(0, 0.720)$$

The tracking states are represented in Figure (16) and control can be expressed in Figure (17).
The state x_1 *has tracked its reference signal,well. In the state* x_2 *there is a delay nearly* 15 *sample and the state* x_3 *has some decreasing in its amplitude.*

4.2 Minimum Variance Finite Impulse Response Filters

There are some discussion about the the FIR filter where matrix A is nonsingular for simplicity. The FIR filter can be represented by

$$\hat{x}_{k|k-1} = \sum_{i=k-N}^{k-1} H_{k-1}y_i + \sum_{i=k-N}^{k-1} L_{k-i}u_i \tag{101}$$

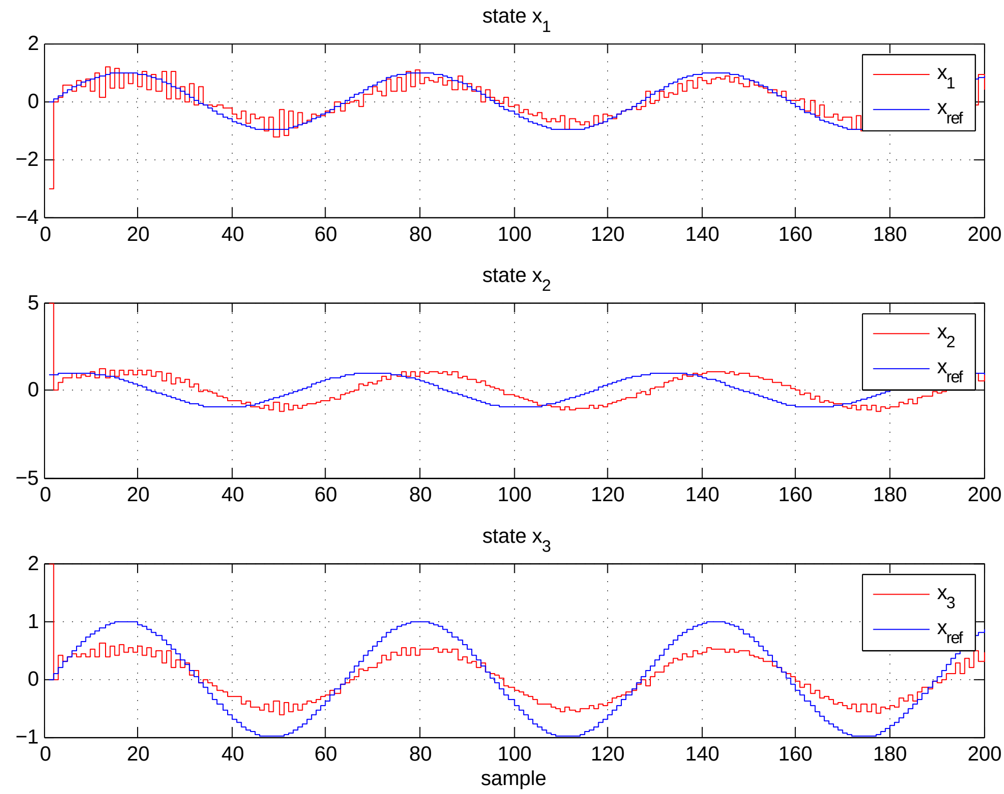

Figure 16: Tracking state using Kalman filter

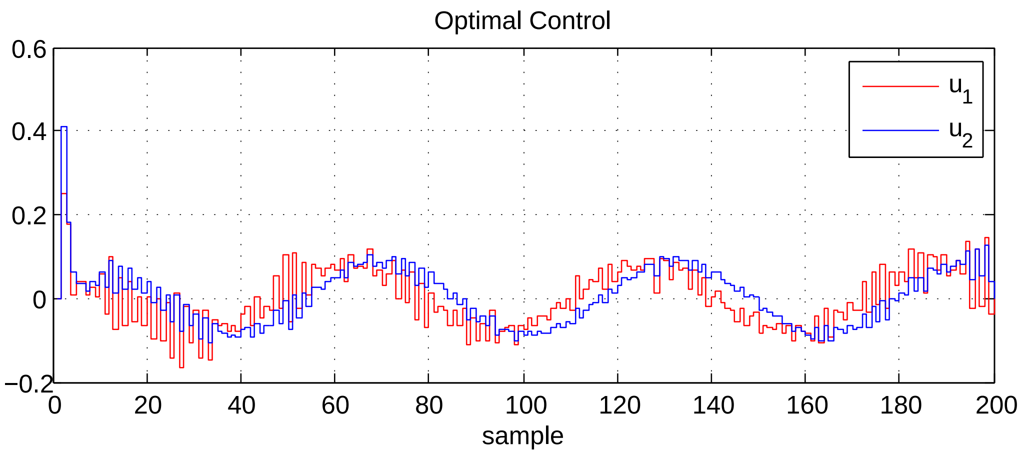

Figure 17: Tracking optimal control using Kalman filter

for discrete-time systems, where N is a filter horizon. The FIR filter does not have an initial state term and the filter gain must be independent of the initial state information. It is noted that a standard Kalman filter has an initial state term and the filter gain depends on the initial state information, such as

$$\hat{x}_{k|k-1} = M_{k-i_0} x_{i_0} + \sum_{i=i_0}^{k-1} H_{k-1} y_i + \sum_{i=i_0}^{k-1} L_{k-i} u_i \tag{102}$$

We can observe that FIR filters make use of finite measurements of inputs and outputs on the most recent time interval $[k-N, k]$, called the receding horizon or horizon. Since filters need to be unbiased as a basic requirement, it is desirable that the linear FIR filter must be unbiased. The unbiased condition for the FIR filter can be

$$E[\hat{x}_{k|k-1}] = E[x_k] \tag{103}$$

Among linear FIR filters with the unbiased condition, optimal filters will be obtained to minimize the estimation error variance. These filters are called minimum variance FIR (MVF) filters.
We define the output(measurement), control, system noise and measurement noise in batch form as

$$\begin{aligned}
Y_{k-1} &\triangleq [y_{k-N}^T \quad y_{k-N+1}^T \quad \cdots \quad y_{k-1}^T]^T & (104)\\
U_{k-1} &\triangleq [u_{k-N}^T \quad u_{k-N+1}^T \quad \cdots \quad u_{k-1}^T]^T & (105)\\
W_{k-1} &\triangleq [w_{k-N}^T \quad w_{k-N+1}^T \quad \cdots \quad w_{k-1}^T]^T & (106)\\
V_{k-1} &\triangleq [v_{k-N}^T \quad v_{k-N+1}^T \quad \cdots \quad v_{k-1}^T]^T & (107)
\end{aligned}$$

then measurement can be represent as

$$Y_{k-1} = \bar{C}_N x_k + \bar{B}_N U_{k-1} + \bar{G}_N W_{k-1} + V_{k-1} \tag{108}$$

where $\bar{C}_N$, $\bar{B}_N$ and $\bar{G}_N$ can be obtained from (75) and (76) assuming nonsingular A matrix

$$\begin{aligned}
x_i = A^{-1}x_{i+1} - A^{-1}Bu_i - A^{-1}Gw_i \quad &\Rightarrow \quad y_i = Cx_i + v_i\\
\Rightarrow \quad y_i &= C\Big[A^{-1}x_{i+1} - A^{-1}Bu_i - A^{-1}Gw_i\Big] + v_i\\
\Rightarrow \quad y_i &= CA^{-1}x_{i+1} - CA^{-1}Bu_i - CA^{-1}Gw_i + v_i
\end{aligned}$$

step by step

$$\begin{aligned}
y_{k-1} &= CA^{-1}x_k - CA^{-1}Bu_{k-1} - CA^{-1}Gw_{k-1} + v_{k-1}\\
y_{k-2} &= CA^{-1}x_{k-1} - CA^{-1}Bu_{k-2} - CA^{-1}Gw_{k-2} + v_{k-2}\\
&= CA^{-1}\Big[A^{-1}x_k - A^{-1}Bu_{k-1} - A^{-1}Gw_{k-1}\Big] - CA^{-1}Bu_{k-2} - CA^{-1}Gw_{k-2} + v_{k-2}\\
&= CA^{-2}x_k - CA^{-2}Bu_{k-1} - CA^{-2}Gw_{k-1} - CA^{-1}Bu_{k-2} - CA^{-1}Gw_{k-2} + v_{k-2}\\
\vdots \; &= \; \vdots\\
y_{k-N} &= CA^{-N}x_k - CA^{-N}Bu_{k-1} - CA^{-N}Gw_{k-1}\\
&\quad - CA^{-N+1}Bu_{k-2} - CA^{-N+1}Gw_{k-2}\\
&\quad - \cdots\\
&\quad + v_{k-N}
\end{aligned}$$

Hence, the matrices $\bar{C}_N$, $\bar{B}_N$ and $\bar{G}_N$ can be presented by

$$\bar{C}_N = \begin{bmatrix} CA^{-N} \\ \vdots \\ CA^{-2} \\ CA^{-1} \end{bmatrix}, \quad \bar{B}_N = \begin{bmatrix} -CA^{-1}B & -CA^{-2}B & \cdots & -CA^{-N}B \\ 0 & -CA^{-1}B & \cdots & -CA^{-N+1}B \\ 0 & 0 & \cdots & -CA^{-N+2}B \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -CA^{-1}B \end{bmatrix}$$

$$\bar{G}_N = \begin{bmatrix} -CA^{-1}G & -CA^{-2}G & \cdots & -CA^{-N}G \\ 0 & -CA^{-1}G & \cdots & -CA^{-N+1}G \\ 0 & 0 & \cdots & -CA^{-N+2}G \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -CA^{-1}G \end{bmatrix}$$

The noise term $\bar{G}_N W_{k-1} + V_{k-1}$ in equation (108) can be shown to be zero mean with covariance Ψ_N given by

$$\Psi_N \triangleq \bar{G}_N[diag(\underbrace{Q_w \; Q_w \; \cdots \; Q_w}_{N})]\bar{G}_N^T + [diag(\underbrace{R_v \; R_v \; \cdots \; R_v}_{N})] \tag{109}$$

An FIR filter with a batch form for the current state x_k can be expressed as a linear function of the finite measurements Y_{k-1} (104) and inputs U_{k-1} (105) on the horizon $[k-N, k]$ as follows:

$$\hat{x}_{k|k-1} = HY_{k-1} + LU_{k-1} \tag{110}$$

where

$$H \triangleq [H_N \; H_{N-1} \; \cdots \; H_1]$$
$$L \triangleq [L_N \; L_{N-1} \; \cdots \; L_1]$$

and matrices H and L will be chosen to minimize a given performance criterion later.
Equation (111) can be written as

$$\hat{x}_{k|k-1} = H(\bar{C}_N x_k + \bar{B}_N U_{k-1} + \bar{G}_N W_{k-1} + V_{k-1}) + LU_{k-1} \tag{111}$$

Taking the expectation on the both side yields

$$E[\hat{x}_{k|k-1}] = H\bar{C}_N E[x_k] + (H\bar{B}_N + L)U_{k-1} \tag{112}$$

Satisfying (103); the unbiased condition $E[\hat{x}_{k|k-1}] = E[x_k]$, we have

$$H\bar{C}_N = I, \quad H\bar{B}_N + L = 0 \tag{113}$$

substituting in (111)

$$\hat{x}_{k|k-1} = x_k + H\bar{G}_N W_{k-1} + HV_{k-1} \tag{114}$$

If we define error as

$$e_k \triangleq \hat{x}_{k|k-1} - x_k$$

then it is apparent

$$e_k = H\bar{G}_N W_{k-1} + HV_{k-1} \tag{115}$$

we should choose the matrix H in such a way that the estimation error e_k has minimum variance. We call this matrix H_B, so

$$\begin{aligned} H_B &= \arg\min_H E[e_k^T e_k] = \arg\min_H E\ tr[e_k e_k^T] \\ &= \arg\min_H tr[H\bar{G}_N Q_N \bar{G}_N^T H^T + HR_N H^T] \end{aligned} \tag{116}$$

where $Q_N = [diag(\underbrace{Q_w\ Q_w\ \cdots\ Q_w}_{N})]$ and $R_N = [diag(\underbrace{R_v\ R_v\ \cdots\ R_v}_{N})]$.

Solution to (116) involves following lemma.

Lemma 2 *Suppose that the following general trace optimization problem is given:*

$$\begin{array}{ll} minimmize & tr[(HA-B)C(HA-B)^T + HDH^T] \\ subject\ to & HE = F \end{array} \tag{117}$$

where $C = C^T > 0$, $D = D^T > 0$, and A, B, C, D, E, and F are constant matrices and have appropriate dimensions. The solution to the optimization problem (117) is as follows:

$$H = \begin{bmatrix} F & B \end{bmatrix} \begin{bmatrix} (E^T\Pi^{-1}E)^{-1}E^T\Pi^{-1} \\ CA^T\Pi^{-1}(I - E(E^T\Pi^{-1}E)^{-1}E^T\Pi^{-1}) \end{bmatrix} \tag{118}$$

where $\Pi \triangleq ACA^T + D$. The matrix H can be shown also

$$H = \begin{bmatrix} F & B \end{bmatrix} \begin{bmatrix} W_{11} & W_{12} \\ W_{12}^T & W_{22} \end{bmatrix}^{-1} \begin{bmatrix} E^T \\ A^T \end{bmatrix} D^{-1} \tag{119}$$

where

$$W_{11} = E^T D^{-1} E \tag{120}$$

$$W_{12} = E^T D^{-1} A \tag{121}$$

$$W_{22} = A^T D^{-1} A + C^{-1} \tag{122}$$

The optimization problem (116) can be matched with this lemma when

$$\begin{array}{lll} A & \longleftarrow & \bar{G}_N \\ B & \longleftarrow & 0 \\ C & \longleftarrow & Q_N \\ D & \longleftarrow & R_N \\ E & \longleftarrow & \bar{C}_N \\ F & \longleftarrow & I \end{array}$$

In the following theorem, the optimal filter gain H_B is represented in an explicit form.

Theorem 1 *When (A, C) is observable and $N \geq n$, the MVF filter $\hat{x}_{k|k-1}$ with a batch form on the horizon $[k-N, k]$ is given as follows*

$$\hat{x}_{k|k-1} = H_B(Y_{k-1} - \bar{B}_N U_{k-1}) \tag{123}$$

with the optimal gain matrix H_B determined by

$$H_B = (\bar{C}_N^T \Psi_N^{-1} \bar{C}_N)^{-1} \bar{C}_N^T \Psi_N^{-1} \tag{124}$$

From this theorem , it can be known that the MVF filter (123) processes the finite measurements and inputs on the horizon $[k-N, k]$ linearly and has the properties of *unbiasedness* and *minimum variance* by design. Note that the optimal gain matrix H_B (124) requires computation only on the interval $[0, N]$ once and is time-invariant for all horizons. This means that the MVF filter is time-invariant. It is a general rule of thumb that, due to the FIR structure, the MVF filter may also be robust against temporary modeling uncertainties or round-off errors. An MVF filter may have a faster tracking ability than an IIR filter even with noises.

An MVF filter can be used in many problems, such as fault detection and diagnosis of various systems, maneuver detection and target tracking of flying objects, and model-based signal processing.

We can summarize minimum variance filter(MVF) as following:

- $$\hat{x}_{k|k-1} = H_B(Y_{k-1} - \bar{B}_N U_{k-1})$$
- $$H_B = (\bar{C}_N^T \Psi_N^{-1} \bar{C}_N)^{-1} \bar{C}_N^T \Psi_N^{-1}$$
- $$\Psi_N = \bar{G}_N[diag(\underbrace{Q_w \; Q_w \; \cdots \; Q_w}_{N})]\bar{G}_N^T + [diag(\underbrace{R_v \; R_v \; \cdots \; R_v}_{N})]$$
- $$\bar{C}_N = \begin{bmatrix} CA^{-N} \\ \vdots \\ CA^{-2} \\ CA^{-1} \end{bmatrix}, \quad \bar{B}_N = \begin{bmatrix} -CA^{-1}B & -CA^{-2}B & \cdots & -CA^{-N}B \\ 0 & -CA^{-1}B & \cdots & -CA^{-N+1}B \\ 0 & 0 & \cdots & -CA^{-N+2}B \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -CA^{-1}B \end{bmatrix}$$
- $$\bar{G}_N = \begin{bmatrix} -CA^{-1}G & -CA^{-2}G & \cdots & -CA^{-N}G \\ 0 & -CA^{-1}G & \cdots & -CA^{-N+1}G \\ 0 & 0 & \cdots & -CA^{-N+2}G \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -CA^{-1}G \end{bmatrix}$$

Example 6 *Consider the state feedback control with fixed terminal state and problem is same as expressed in previous example, using Kalman filter. Now we'd like to estimate the state using FIR filter; MVF. The system is modeled as*

$$x[k+1] = \begin{bmatrix} -0.3252 & 0.6504 & 0.6098 \\ 0.8130 & 0 & 0.8130 \\ 0.3659 & 0.2439 & -0.0813 \end{bmatrix} x[k] + \begin{bmatrix} 0.1 & -0.2 \\ 1 & 0.5 \\ 0 & 1 \end{bmatrix} u[k] + \begin{bmatrix} 0.9 \\ 0 \\ -0.255 \end{bmatrix} w[k]$$

$$y[k] = \begin{bmatrix} 1 & 0 & 0.15 \\ 0 & -1 & 0.2 \end{bmatrix} x[k] + v[k]$$

Suppose that the initial condition be $x_0 = [-3 \quad 5 \quad 2]^T$. *The weighting matrices are given as following:*

$$Q = I, \qquad Q_f = 5I, \qquad R = 10I$$

We want control to track our desired signal. Reference signals are

$$x_1^r[k] = \sin(k/15), \qquad x_2^r[k] = \sin(k/15+1), \qquad x_3^r[k] = \sin(k/15)$$

We generated the w_i *and* v_i *using MATLAB command* `normrnd` *and their means were zero and their variances were* 0.5 *and* 0.2 *respectively,* i.e.

$$w_i \sim \mathcal{N}(0, 0.5)$$
$$v_i \sim \mathcal{N}(0, 0.2)$$

The horizon length is $N = 10$. *The tracking states are represented in Figure (18) and control can be expressed in Figure (19).*

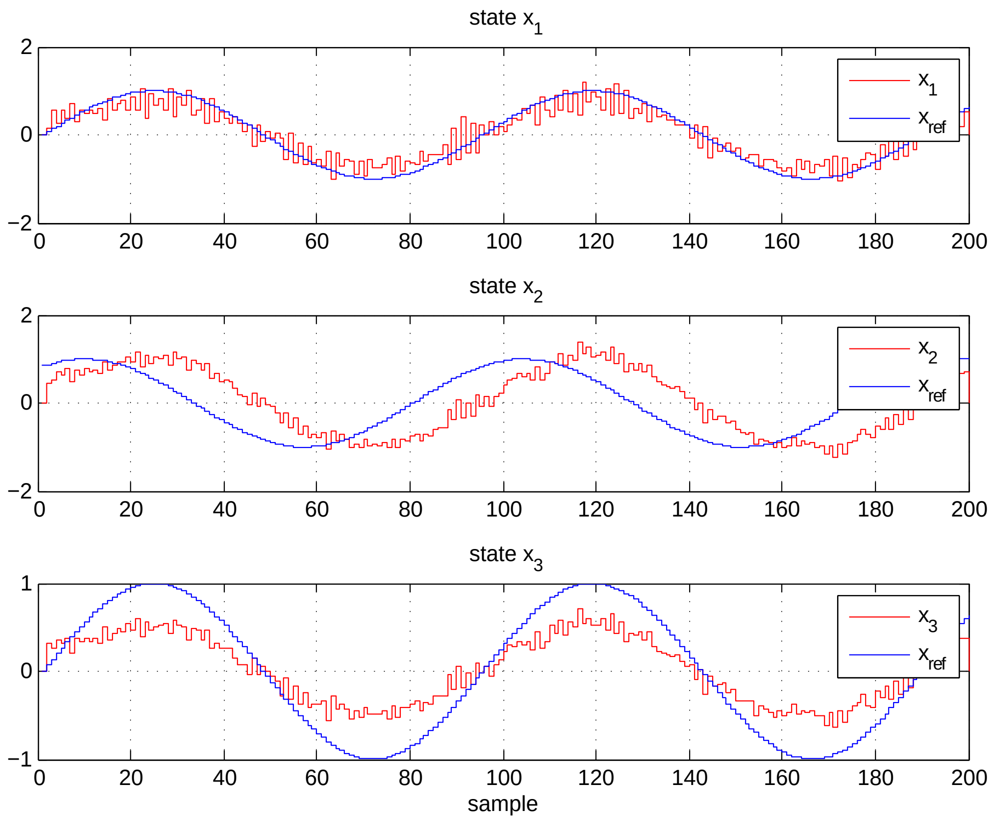

Figure 18: Tracking state using MV filter

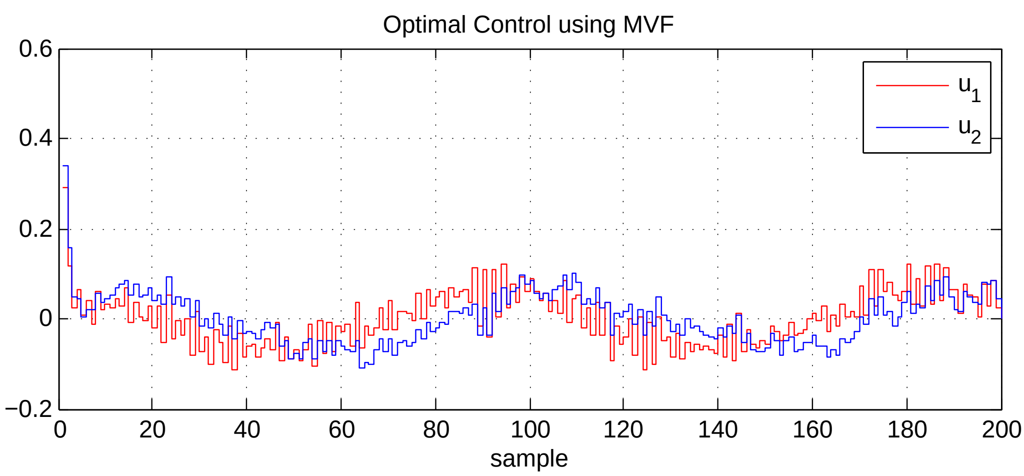

Figure 19: Tracking optimal control using MV filter

We compare the error estimation between Kalman filter and MVF in the Figure (20) for the recent example.

They are very close to themselves, but we can observe that the variation in Kalman filter is more than MV filter.

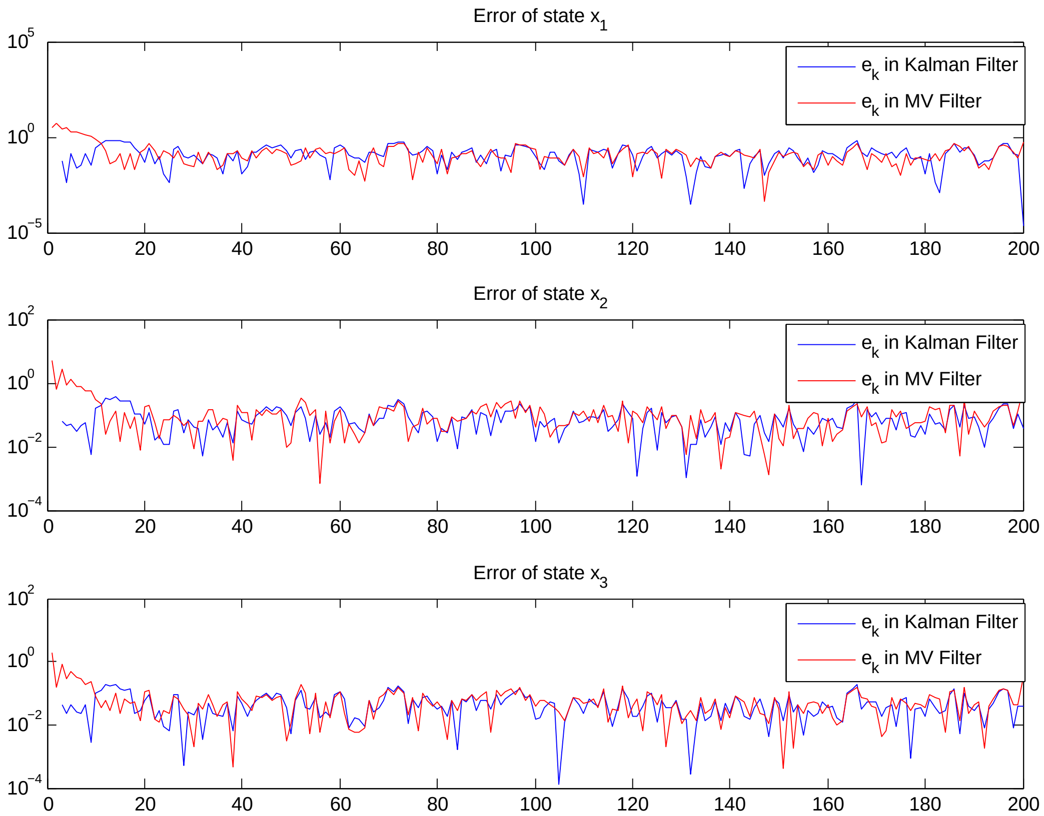

Figure 20: Error estimation in Kalman Filter and MV filter

5 Quadratic Programming in MPC

5.1 Introduction

A convex optimization problem is one of the form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \dots, m \\ & a_i^T x = b_i, \quad i = 1, \dots, p \end{array} \tag{125}$$

where $f_0, \dots, f_m$ are convex functions. If the objective is quadratic, and the constraint functions are affine, it is called *quadratic programming*(QP). Hence a quadratic program can be expressed in the form

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P x + q^T x + r \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array} \tag{126}$$

where P is a symmetric[3] positive semi-definite matrix, $G \in \mathbb{R}^{m \times n}$ and $A \in \mathbb{R}^{p \times n}$.

5.2 Example of QP

We present two examples as application of QP.First example says a least squares problem can be represented as a QP.

$$\|Ax - b\|_2^2 = x^T A^T A x - 2b^T A x + b^T b \tag{127}$$

The analytic solution to this simple problem is

$$x^* = (A^T A)^{-1} A^T b \tag{128}$$

If we have some constraints, the problem change to *constrained least squares* and there is no longer a simple analytical solution.

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2^2 \\ \text{subject to} & l_i \leq x_i \leq u_i \quad i = 1, \dots, n \end{array} \tag{129}$$

Now, the second example which was introduced by Markowitz is presented:
we consider a classical portfolio problem with n assets or stocks held over a period of time. We let x_i denote the amount of asset i held throughout the period, with x_i in dollars, at the price at the beginning of the period. A wide variety of constraints on the portfolio can be considered. The simplest set of constraints is that $x_i \geq 0$ and $\mathbf{1}^T x = B$ (the budget to be invested and usually

[3]Our discussion is about the condition when P is symmetric, if it was not symmetric we compose the symmetric and anti-symmetric part of P and pre and post multiplying by vector x^T and x respectively, causes the part anti-symmetric be zero, *i.e.*
$P = P_{Symmetric} + P_{Anti-symmetric} \quad \Rightarrow \quad x^T P_{Anti-symmetric} x = 0.$

it is taken to be unit.)
If we take a stochastic model for price changes with $p \in \mathbb{R}^n$ is a random vector, with known mean $\bar{p}$ and covariance Σ.
This classical portfolio optimization problem, is the QP

$$\begin{array}{ll} \text{minimize} & x^T \Sigma x \\ \text{subject to} & \bar{p}^T x \geq r_{min} \\ & \mathbf{1}^T x = 1, \quad x \succeq 0 \end{array} \tag{130}$$

We find the portfolio that minimizes the return variance (which is associated with the risk of the portfolio) subject to achieving a minimum acceptable mean return r_{min}, and satisfying the portfolio budget.

5.3 QP as semi-definite programming (SDP)

Semi-definite programming (SDP) can effectively solve many optimization problems involving LMI. The general form of SDP is an optimization problem of minimizing a linear function of a variable $x \in \mathbb{R}^n$ subject to a matrix inequality and a matrix equality:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & F(x) \preceq 0 \\ & Ax = b \end{array} \tag{131}$$

where

$$F(x) = F_0 + \sum_{i=1}^{n} x_i F_i \tag{132}$$

and F_i $\forall i = 1, \ldots, n$ are symmetric matrices. This SDP is a convex optimization problem. The QP can be transform to SDP as following

$$\begin{array}{ll} \text{minimize} & (1/2) x^T P x + q^T x + r \\ \text{subject to} & Gx \preceq h \end{array}$$

we consider an upper bound for the objective function, *e.g.* t;

$$(1/2) x^T P x + q^T x + r < t$$

Using Schur complement

$$t - r - q^T x - x^T (2P^{-1})^{-1} x > 0 \quad \Rightarrow \quad \begin{bmatrix} t - r - q^T x & x^T \\ x & 2P^{-1} \end{bmatrix} \succ 0$$

-Note that P is positive definite
Then the QP can be represented by

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \begin{bmatrix} t - r - q^T x & x^T \\ x & 2P^{-1} \end{bmatrix} \succ 0 \\ & Gx \preceq h \end{array}$$

which is a SDP. Note that the equality constraint for each type (either QP or SDP) is the same, *i.e.* $Ax = b$.

5.4 QP in Optimal Control

The objective in optimal control usually is a minimization on a quadratic cost function. The linear time invariant system where is modeled by a linear equality treats as the equality constraint of QP.

5.4.1 QP with equality constraints

An optimal control problem can be represented by

$$\begin{array}{ll} \text{minimize} & J(x^r, u) \\ \text{subject to} & x_{k+1} = Ax_k + Bu_k \quad k = 1, 2, \ldots, N-1 \end{array} \tag{133}$$

where

$$J(x^r, u) = \frac{1}{2} \sum_{k=0}^{N-1} \Big[(x_k - x_k^r)^T Q (x_k - x_k^r) + u_k^T R u_k \Big] + \frac{1}{2} (x_N - x_N^r)^T Q_f (x_N - x_N^r) \tag{134}$$

Here $x \in \mathbb{R}^n$, $x^r \in \mathbb{R}^n$ and $u \in \mathbb{R}^l$ are state, reference state and input respectively. N express the horizon for the problem.

We assume that the cost function $J(x^r, u)$ is strictly convex in the term of input and state *i.e.* $Q, R > 0$. The variables in the problem are $u_0, \ldots, u_{N-1}$, and $x_1, \ldots, x_N$. The initial state x_0 is given. The linear equality constraints are called the state equations. We define the overall optimization variable z in a batch form as

$$z = [u_0^T, x_1^T, \ldots, u_{N-1}^T, x_N^T]^T \in \mathbb{R}^{(n+l)N} \tag{135}$$

We try to obtain a relation like $Ax = b$.

$$\begin{array}{lll}
x_1 = Ax_0 + Bu_0 & \Rightarrow -Bu_0 + x_1 = Ax_0 & \begin{bmatrix} -B & I \end{bmatrix} \begin{bmatrix} u_0 \\ x_1 \end{bmatrix} = Ax_0 \\
x_2 = Ax_1 + Bu_1 & \Rightarrow -Ax_1 - Bu_1 + x_2 = 0 & \begin{bmatrix} -A & -B & I \end{bmatrix} \begin{bmatrix} x_1 \\ u_1 \\ x_2 \end{bmatrix} = 0 \\
\vdots = \ \vdots & \vdots & \vdots \\
x_N = Ax_{N-1} + Bu_{N-1} & \Rightarrow -Ax_{N-1} - Bu_{N-1} + x_N = 0 & \begin{bmatrix} -A & -B & I \end{bmatrix} \begin{bmatrix} x_{N-1} \\ u_{N-1} \\ x_N \end{bmatrix} = 0
\end{array}$$

When we collect all the equality constraints we'll have

$$\begin{bmatrix} -B & I & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -A & -B & I & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & I & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & -A & -B & I \end{bmatrix} \begin{bmatrix} u_0 \\ x_1 \\ \vdots \\ u_{N-1} \\ x_N \end{bmatrix} = \begin{bmatrix} Ax_0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{136}$$

The equation (136) can be shown as

$$\tilde{A}z = \tilde{b} \tag{137}$$

where $\tilde{A} \in \mathbb{R}^{Nn \times N(n+l)}$ and $\tilde{b} \in \mathbb{R}^{Nn}$.

We first suppose the regulation problem for simplicity, *i.e.*

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\sum_{k=0}^{N-1}\left[x_k^T Q x_k + u_k^T R u_k\right] + \frac{1}{2}x_N^T Q_f x_N \\ \text{subject to} & x_{k+1} = Ax_k + Bu_k \quad k = 1, 2, \ldots, N-1 \end{array} \tag{138}$$

then the optimization problem can be represented as

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}z^T H z \\ \text{subject to} & \tilde{A}z = \tilde{b} \end{array} \tag{139}$$

where

$$H = \begin{bmatrix} R & 0 & 0 & \cdots & 0 \\ 0 & Q & 0 & \cdots & 0 \\ 0 & 0 & R & & \vdots \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & \cdots & & Q_f \end{bmatrix} \tag{140}$$

and $H \in \mathbb{R}^{N(n+l) \times N(n+l)}$.
Using Lagrange function to solve this optimization problem we can write

$$L(z, v) = \frac{1}{2}z^T H z + v^T(\tilde{A}z - \tilde{b}) \tag{141}$$

and hence

$$\begin{gathered} \nabla_x L(z, v) = Hz + \tilde{A}^T v = 0 \\ \tilde{A}z = \tilde{b} \end{gathered} \tag{142}$$

A point $z^\star \in \mathbf{dom} f$ is optimal for (139) if and only if there is a $v^\star \in \mathbb{R}^{Nn}$ such that

$$\tilde{A}z^\star = \tilde{b}, \quad Hz^\star + \tilde{A}^T v^\star = 0 \tag{143}$$

This conditions are called *Karush-Kuhn-Tucker* (KKT) conditions. The set of $n+p$ linear equations in the $n+p$ variables $z^\star$ and $v^\star$ is called the KKT *system* for the equality constrained quadratic optimization problem (139).
We can write the equation (143) as

$$\begin{bmatrix} H & \tilde{A}^T \\ \tilde{A} & 0 \end{bmatrix} \begin{bmatrix} z^\star \\ v^\star \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{b} \end{bmatrix} \tag{144}$$

The coefficient matrix in (144) is called the KKT *matrix*.
There are three statements:

- When the KKT matrix is nonsingular, there is a unique optimal pair $(z^\star, v^\star)$.
- If the KKT matrix is singular, but the KKT system is solvable, any solution yields an optimal pair $(z^\star, v^\star)$.
- If the KKT system is not solvable, the quadratic optimization problem is unbounded below or infeasible.

There are several conditions equivalent to nonsingularity of the KKT matrix:

- P and A have no nontrivial common nullspace
- P is positive definite on the nullspace of A

5.4.2 QP with inequality constraints

When there is an inequality constraints the optimization problem is

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}z^T H z \\ \text{subject to} & Gz \preceq h \\ & \tilde{A}z = \tilde{b} \end{array} \tag{145}$$

The inequality constraints $Gz \preceq h$ can be for example an upper or lower bounds on the variables, *e.g.* $x_{min} \leq x_k \leq x_{max}$ and $u_{min} \leq u_k \leq u_{max}$.

Interior point methods is one of the efficient method for solving QP (or in general convex optimization) problem that include inequality constraints. We consider again the optimization problem (125) and present the full KKT conditions as following,

$$\begin{aligned} Ax^\star = b, \quad f_i(x^\star) &\leq 0, \quad i = 1, \dots, m \\ \lambda^\star &\succeq 0 \\ \nabla f_0(x^\star) + \textstyle\sum_{i=1}^m \lambda_i^\star \nabla f_i(x^\star) + A^T v^\star &= 0 \\ \lambda_i^\star f_i(x^\star) &= 0, \quad i = 1, \dots, m \end{aligned} \tag{146}$$

There is two important interior point methods:

- Barrier Method
- Primal-Dual Method

Primal-dual interior-point methods are often more efficient than the barrier method, especially when high accuracy is required, since they can exhibit better than linear convergence.

We represent the primal-dual method here.
The modified KKT conditions can be expressed as $r_t(x, \lambda, v) = 0$ where

$$r_t(x, \lambda, v) = \begin{bmatrix} \nabla f_0(x) + Df(x)^T\lambda + A^T v \\ -\mathbf{diag}(\lambda) f(x) - (1/t)\mathbf{1} \\ Ax - b \end{bmatrix} \tag{147}$$

and $t > 0$. Here $f : \mathbb{R}^n \to \mathbb{R}^m$ and its derivative matrix Df are given by

$$f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{bmatrix}, \quad Df(x) = \begin{bmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_m(x)^T \end{bmatrix}$$

In particular, x is primal feasible, and λ, v are dual feasible, with duality gap m/t. We call the first, second and third block componnents of r_t

$$r_{dual} = \nabla f_0(x) + Df(x)^T\lambda + A^T v \tag{148}$$
$$r_{cent} = -\mathbf{diag}(\lambda) f(x) - (1/t)\mathbf{1} \tag{149}$$
$$r_{pri} = Ax - b \tag{150}$$

as dual, central and primal residual.

Now consider the Newton step for solving the nonlinear equations $r_t(x, \lambda, v) = 0$, for fixed t. We will denote the current point and Newton step as

$$y = (x, \lambda, v), \quad \Delta y = (\Delta x, \Delta\lambda, \Delta v)$$

The Newton step is characterized by the linear equations

$$r_t(y + \Delta y) \approx r_t(y) + Dr_t(y)\Delta y = 0 \tag{151}$$

In the matrix form we can write

$$\begin{bmatrix} \nabla^2 f_0(x) + \sum_{i=1}^m \lambda_i \nabla^2 f_i & Df(x)^T & A^T \\ -\mathbf{diag}(\lambda) Df(x) & -\mathbf{diag}(f(x)) & 0 \\ A & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta v \end{bmatrix} = - \begin{bmatrix} r_{dual} \\ r_{cent} \\ r_{pri} \end{bmatrix} \tag{152}$$

We define duality gate mentioned before as

$$\hat{\eta} = -f(x)^T\lambda \tag{153}$$

We can now represent the basic primal-dual interior-point algorithm from [2] as

Algorithm *Primal-dual interior-point method.*

given x that satisfies $f_1(x) < 0, \dots, f_m(x) < 0, \lambda \succ 0, \mu > 1, \epsilon_{feas} > 0, \epsilon > 0$.
repeat

1. Determine t. Set $t := \mu m/\hat{\eta}$.
2. Compute primal-dual search direction Δy.
3. Line search and update. *i.e.* Determine step length $s > 0$ and set $y := y + s\Delta y$.

until $\|r_{pri}\|_2 \leq \epsilon_{feas}$, $\|r_{dual}\|_2 \leq \epsilon_{feas}$ and $\hat{\eta} \leq \epsilon$.

In step 1, values of the parameter μ on the order of 10 appear to work well. In the step 3, we first compute the largest positive step length, not exceeding one, that gives the next λ or λ^+ greater than or equal to zero, *i.e.* $\lambda^+ = \lambda + s\Delta\lambda \succeq 0$, hence

$$s^{\max} = \sup_s \{s \in [0,1] | \lambda + s\Delta\lambda \succeq 0\} \tag{154}$$

$$= \min\{1, \min\{-\lambda_i/\Delta\lambda_i | \Delta\lambda_i < 0\}\} \tag{155}$$

We start the backtracking with $s = 0.99s^{\max}$, and multiply s by $\beta \in (0,1)$ until we have

$$\|r_t(x^+, \lambda^+, v^+)\|_2 \leq (1 - \alpha s)\|r_t(x, \lambda, v)\|_2. \tag{156}$$

where the x^+, λ^+ and v^+ are the next iteration of x, λ and v respectively as for λ mentioned before.
α is typically chosen in the range 0.01 to 0.1, and β is typically chosen in the range 0.3 to 0.8.

Now we apply the equation (152) for (145), with this constraints:

$$x_{min} \leq x_k \leq x_{max}, \quad u_{min} \leq u_k \leq u_{max}$$

First we change this constraints to standard form in z:

$$\begin{aligned} u_{\min} - u_0 &\preceq 0 & x_{\min} - x_1 &\preceq 0 \\ u_0 - u_{\max} &\preceq 0 & x_1 - x_{\max} &\preceq 0 \\ &\vdots & &\vdots \\ u_{\min} - u_{N-1} &\preceq 0 & x_N - x_{\max} &\preceq 0 \\ u_{N-1} - u_{\max} &\preceq 0 & x_N - x_{\max} &\preceq 0 \end{aligned}$$

since $z = [u_0^T, x_1^T, \ldots, u_{N-1}^T, x_N^T]^T$

$$
\begin{aligned}
u_{\min_0} - u_0 &\preceq 0\\
u_0 - u_{\max_0} &\preceq 0\\
x_{\min_1} - x_1 &\preceq 0\\
x_1 - x_{\max_1} &\preceq 0\\
&\vdots\\
u_{\min_{N-1}} - u_{N-1} &\preceq 0\\
u_{N-1} - u_{\max_{N-1}} &\preceq 0\\
x_{\min_N} - x_N &\preceq 0\\
x_N - x_{\max_N} &\preceq 0
\end{aligned}
\Rightarrow
\begin{bmatrix}
-I & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0\\
I & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0\\
0 & -I & 0 & 0 & \cdots & 0 & 0 & 0 & 0\\
0 & I & 0 & 0 & \cdots & 0 & 0 & 0 & 0\\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots\\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & -I & 0\\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & I & 0\\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & -I\\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & I
\end{bmatrix}
\begin{bmatrix}
u_0\\ x_1\\ \vdots\\ u_{N-1}\\ x_N
\end{bmatrix}
\preceq
\begin{bmatrix}
-u_{\min_0}\\ u_{\max_0}\\ -x_{\min_1}\\ x_{\max_1}\\ \vdots\\ -u_{\min_{N-1}}\\ u_{\max_{N-1}}\\ -x_{\min_N}\\ x_{\max_N}
\end{bmatrix}
$$

so the left side matrix is $G \in \mathbb{R}^{2N(n+l)\times N(n+l)}$ and the right side matrix is $h \in \mathbb{R}^{2N(n+l)}$ and the inequality can be written as $Gz \preceq h$.
Applying equation (152),since the constraints are linear (or as a suitable word; affine), $\nabla^2 f_i = 0$. The inequality constraints can be shown as

$$
\begin{aligned}
f_1(z) &= u_{\min} + \begin{bmatrix}-I & 0 & \cdots & 0\end{bmatrix} z \preceq 0\\
f_2(z) &= -u_{\max} + \begin{bmatrix}I & 0 & \cdots & 0\end{bmatrix} z \preceq 0\\
f_3(z) &= x_{\min} + \begin{bmatrix}0 & -I & \cdots & 0\end{bmatrix} z \preceq 0\\
f_4(z) &= -u_{\max} + \begin{bmatrix}0 & I & \cdots & 0\end{bmatrix} z \preceq 0\\
\vdots \quad & \quad \vdots\\
f_{2N-1}(z) &= x_{\min} + \begin{bmatrix}0 & 0 & \cdots & -I\end{bmatrix} z \preceq 0\\
f_{2N}(z) &= -x_{\max} + \begin{bmatrix}0 & 0 & \cdots & I\end{bmatrix} z \preceq 0
\end{aligned}
$$

In a compact form

$$f(z) = Gz - h \tag{157}$$

Then, the term $Df(z)$ is the matrix G which mentioned above. The rest of variables:

$$r_{dual} = Hz + G^T\lambda + \tilde{A}^T v \tag{158}$$

$$r_{cent} = -\mathbf{diag}(\lambda)f(z) - (1/t)\mathbf{1} \tag{159}$$

$$r_{pri} = \tilde{A}z - \tilde{b} \tag{160}$$

5.5 QP used in MPC

All the discussion in previous subsections will used in model predictive control. In the model predictive control we should minimize a quadratic cost function subject to affine equality and inequality constraints. The only difference is that we should use the optimization on line and correspond to each sample. As described in the section 1.2 (Concept of MPC), the horizon is finite and all our decision applies to the first sample. In MPC, at each time k we solve the QP

$$\begin{array}{lll} \text{minimize} & \frac{1}{2}x_{k+N}^T Q_f x_{k+N} + \frac{1}{2}\sum_{i=k}^{k+N-1}[x_i^T Q x_i + u_i^T R u_i] & \\ \text{subject to} & x_{min} \leq x_i \leq x_{max}, \quad i = k+1, \ldots, k+N & \\ & u_{min} \leq u_i \leq u_{max}, \quad i = k, \ldots, k+N-1 & \\ & x_{i+1} = Ax_i + Bu_i, \quad i = k, \ldots, k+N-1 & \end{array} \tag{161}$$

where N is the finite horizon.

There are several software packages that include QP solvers

CVXOPT	source language: C, Python; API: Python
OpenOp	universal cross-platform Python-written numerical optimization framework;
OOQP	C++, interior point algorithm implementation by Gertz and Wright
QuadProg	source language: R (a port from S), algorithm of Goldfarb and Idnani (1982, 1983)
MOSEK	convex problems only
AMPL Modeling Language	AMPL (free for students for problems with up to 300 variables and 300 constraints
Optimization Toolbox	MATLAB

The first four solvers are free and others are commercial.
We use CVXOPT solver `cvx` toolbox to solve QP problems.

5.6 Implementation

We first summarize the required relation here:

- $\tilde{A} = \begin{bmatrix} -B & I & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -A & -B & I & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & I & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & -A & -B & I \end{bmatrix}, \qquad \tilde{b} = \begin{bmatrix} Ax_0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$

- $H = \begin{bmatrix} R & 0 & 0 & \cdots & 0 \\ 0 & Q & 0 & \cdots & 0 \\ 0 & 0 & R & & \vdots \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & \cdots & & Q_f \end{bmatrix}, \qquad h = \begin{bmatrix} -u_{\min_0} \\ u_{\max_0} \\ -x_{\min_1} \\ x_{\max_1} \\ \vdots \\ -u_{\min_{N-1}} \\ u_{\max_{N-1}} \\ -x_{\min_N} \\ x_{\max_N} \end{bmatrix}$

- $r_{dual} = Hz + G^T\lambda + \tilde{A}^T v$
- $r_{cent} = -\mathbf{diag}(\lambda) f(z) - (1/t)\mathbf{1}$
- $r_{pri} = \tilde{A}z - \tilde{b}$

- $f(z) = Gz - h$

- $\begin{bmatrix} H & G^T & \tilde{A}^T \\ -\mathbf{diag}(\lambda)G & -\mathbf{diag}(f(z)) & 0 \\ \tilde{A} & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta\lambda \\ \Delta v \end{bmatrix} = - \begin{bmatrix} r_{dual} \\ r_{cent} \\ r_{pri} \end{bmatrix}$

- $\begin{bmatrix} z^+ \\ \lambda^+ \\ v^+ \end{bmatrix} = \begin{bmatrix} z \\ \lambda \\ v \end{bmatrix} + s \begin{bmatrix} \Delta z \\ \Delta\lambda \\ \Delta v \end{bmatrix}$

- $s = 0.99 s^{\max}, \quad s^{\max} = \min\{1, \min\{-\lambda_i/\Delta\lambda_i | \Delta\lambda_i < 0\}\}\ , \quad s := \beta s$

First I present my results of finding the optimal control using this proposed algorithm, then I compare it with CVXOPT solver which solves QP problems. Indeed, my code is appended in appendix as Example QP solver.

Example 7 *We choose a single input system which is modeled as*

$$x[k+1] = \begin{bmatrix} 1 & 0.8 \\ 0 & -0.7 \end{bmatrix} x[k] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u[k]$$

Our performance criterion is

$$minimize \quad \frac{1}{2}x_N^T \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} x_N + \frac{1}{2}\sum_{i=0}^{N-1}[x_i^T x_i + 10u_i^T u_i]$$

This example does not use a model predictive control, but this is a LQR problem. We can extend it and exploit as in MPC. The given initial condition is in $\begin{bmatrix} 3 & -1 \end{bmatrix}^T$ *and the time horizon is equal to 50 sample. Inequality constraints can be express as*

$$-5 \le x_i \le 5, \forall i = 1,2 \qquad -0.8 \le u \le 0.8$$

Figures (21) and (22) illustrate the states and optimal control respectively.

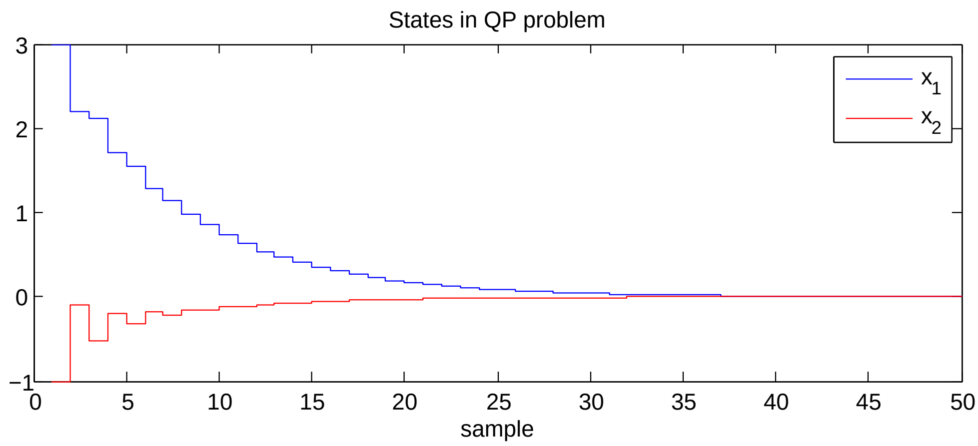

Figure 21: States in QP problem with inequality constraints

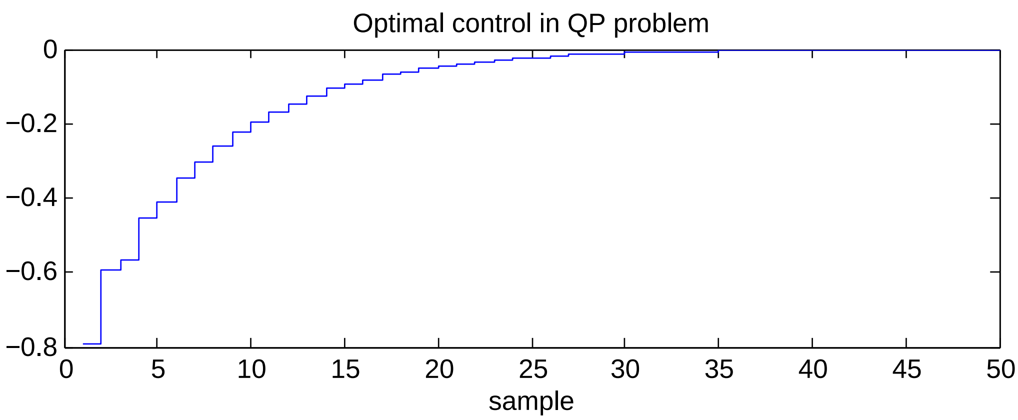

Figure 22: Optimal control of QP problem with inequality constraints

Also we use CVXOPT solver to finding optimal control, then Figures (23) and (24) demonstrate the states and optimal control respectively. We observe that the result are very close, and the difference may be return to used algorithms or bad programming.

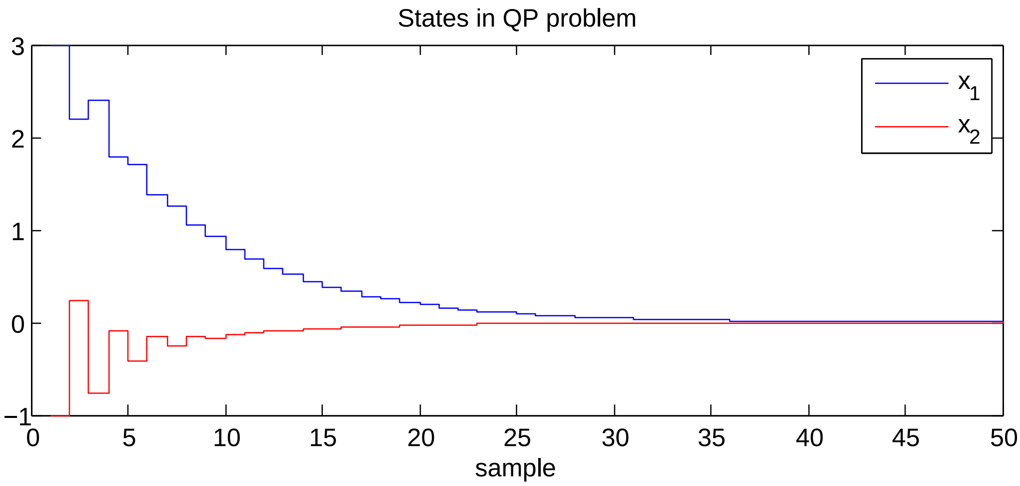

Figure 23: States in QP problem with inequality constraints using CVXOPT

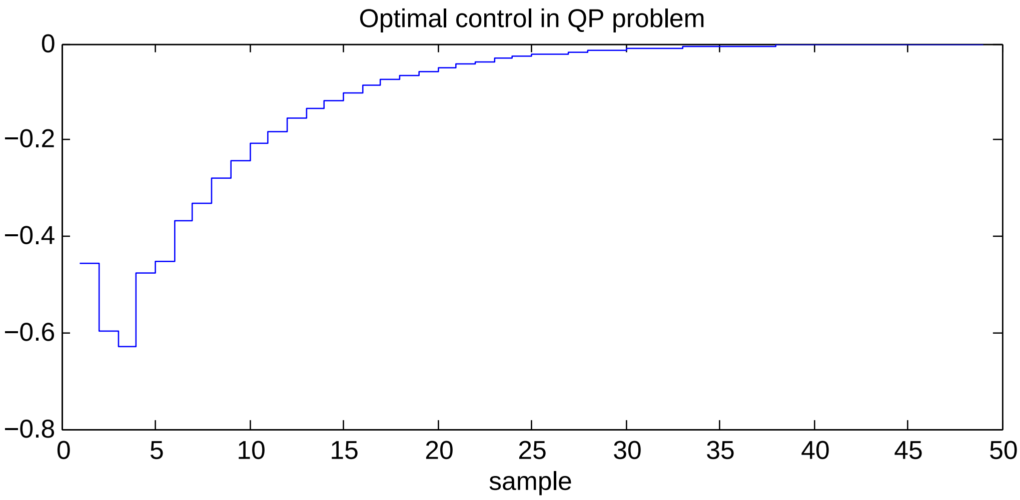

Figure 24: Optimal control of QP problem with inequality constraints using CVXOPT

Finally, the aim of last example is to find optimal control solution utilizing model predictive control concept.
If we take receding horizon $N = 10$ and all the previous example conditions are satisfied.

The states and optimal control are shown in Figures (25) and (26) respectively. For some consideration we use the standard solver CVXOPT for this problem.

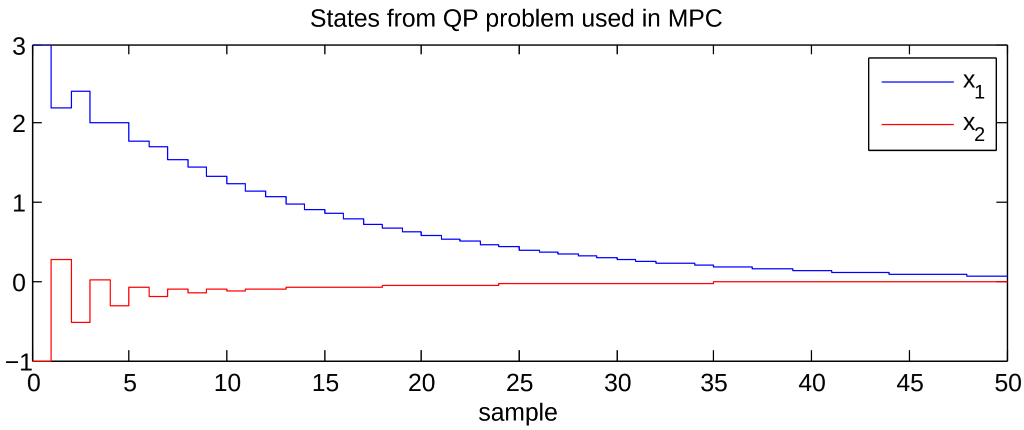

Figure 25: States for QP problem in MPC

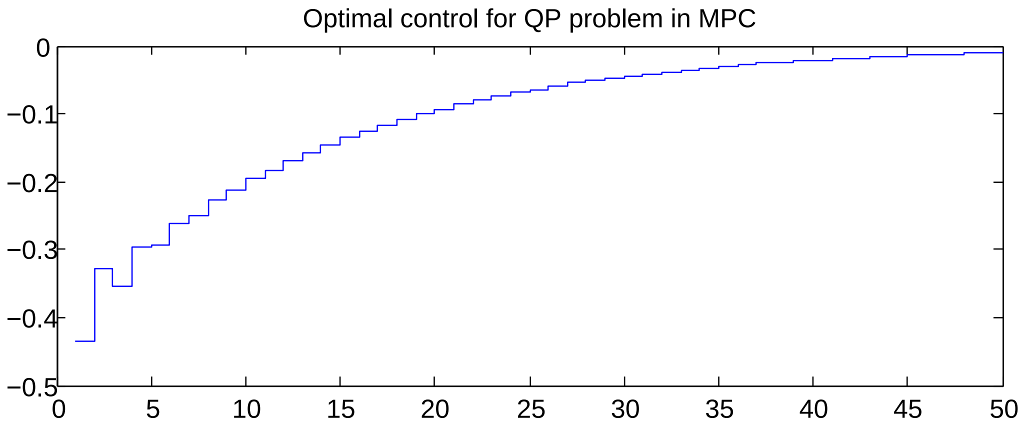

Figure 26: Optimal control for QP problem in MPC

The computation is less than infinite horizon, but the settling time is more than infinite horizon.

A Appendix

There are some MATLAB code of examples used in this project.
Example 2:

```
clear all
clc


A=[0.8 0.75; 0 1];
B=[0 -0.2 ;1 0.5];
x0=[-3;5];
R=[10 0; 0 10];
Q=eye(2);
cvx_begin sdp
variables g Y(2,2) L(2,2)
const=[Y, L', (A*Y+B*L)', Y', zeros(2,3);...
     L, R^(-1) ,zeros(2,7); ...
     (A*Y+B*L) , zeros(2,2), Y , zeros(2,5) ;...
     Y , zeros(2,4) , Q^(-1) ,zeros(2,3);...
     zeros(3,8),[g, x0';x0,Y] ];
minimize (g)
subject to
const>=0;
cvx_end

H=L*Y^(-1);

x=zeros(2,101);
x(:,1)=x0;
for j=1:100
    u(:,j)=H*x(:,j);
    x(:,j+1)=A*x(:,j)+B*u(:,j);
end


subplot(2,1,1)
stairs([1:30],u(1,1:30),'b')
hold on
stairs([1:30],u(2,1:30),'r')
grid on
title('H_\infty optimal control')
legend('u_1','u_2')
subplot(2,1,2)
stairs([1:30],x(1,1:30),'b')
hold on
stairs([1:30],x(2,1:30),'r')
grid on
title('State')
xlabel('Sample time')
legend('x_1','x_2')
```

```
%norm infty
clear all
clc


A=[0.8 0.75; 0 1];
B=[0 -0.2 ;1 0.5];
Bw=[0.01, -0.032; 0.101, -0.05];
C=[1, -0.1];
D=[0.1, -0.05];
x0=[-3;5];
Wz=eye(2);
cvx_begin sdp
variables g Y(2,2) L(2,2)
const=[-Y, (A*Y+B*L), Bw, zeros(2,1);...
     (A*Y+B*L)', -Y ,zeros(2,2), (C*Y+D*L)'; ...
     Bw' , zeros(2,2), -g*Wz , zeros(2,1) ;...
    zeros(1,2),(C*Y+D*L), zeros(1,2) ,-g];
minimize (g)
subject to
const<0;
cvx_end
```

Example 4:

```
clear all
clc


N=5; %receding horizon
NN=120;
t=0:NN+N-1;

A=[1,0.75;0,0.8];
B=[0;1];

x0=[-3; 5];

[nQ mQ] =size(A);
[nR mR] =size(B);

%weight matrices
Q=eye(mQ);
Qf=5*eye(mQ);
R=10*eye(mR);


Bbar=[A^4*B , A^3*B, A^2*B, A*B, B];
Q_N=blkdiag(Q,Q,Q,Q,Q);
H=[ zeros(2,5)
    B , zeros(2,4);
    A*B, B , zeros(2,3);
    A^2*B, A*B, B , zeros(2,2);
    A^3*B, A^2*B, A*B, B , zeros(2,1)];

W=H'*Q_N*H;
F=[eye(2);A;A^2;A^3;A^4];


u=zeros(1,NN-1);
x=zeros(2,NN);
xr=zeros(2,NN+N);
xr(1,:)=sin(t/10);
xr(2,:)=sin(t/10+1);
x(:,1)=x0;

for k=1:NN-1
    Xr=[xr(:,k);xr(:,k+1);xr(:,k+2);xr(:,k+3);xr(:,k+4)];
    U=-(W+Bbar'*Qf*Bbar)^(-1)*(H'*Q_N*(F*x(:,k)-Xr)+Bbar'*Qf*(A^5*x(:,k)-xr(:,k+N) ));
    u(:,k)=[1 0 0 0 0]*U;
    x(:,k+1)=A*x(:,k)+B*u(:,k);
end



plot(t(1:NN),x(1,:),'r',t(1:NN),xr(1,1:NN),'b')
```

Example 5:

```
% Kalman Filter
clear all
clc

% A, C, G, Q_w, R_v

%x_i+1=Ax_i+Bu_i+Gw_i
%y_i=Cx_i+v


%fixed terminal state
i_f=200; %step
t=0:(i_f-1);

%problem data
A =[
    -0.3252    0.6504    0.6098;
    0.8130         0    0.8130;
    0.3659    0.2439   -0.0813];

B =[
    0.1000   0.2000;
    1.0000    0.5000;
         0    1.0000];

G=[0.9; 0; -0.255];
C=[1 0 0.15;0 -1 0.2];

x0=[-3;5;2];

[nQ mQ] =size(A);
[nR mR] =size(B);
[nC mC] =size(C);
[nG mG] =size(G);


Q=eye(mQ);
Q_f=5*eye(mQ);
R=10*eye(mR);

w=normrnd(0,0.265,i_f,mG);
v=normrnd(0,0.720,i_f,nC);

Q_w=cov(w);
R_v=cov(v);
%all the x_i are the estimaed state
P=zeros(mQ);

xr=zeros(3,i_f);
xr(1,:)=sin(t/10);
xr(2,:)=sin(t/10+1);
xr(3,:)=sin(t/10);
```

```
K=Q_f;
g=-Q_f*xr(:,i_f);

x=zeros(mQ,i_f);
u=zeros(mR,i_f);
y=zeros(nC,i_f);

x(:,1)=x0;

for k=1:(i_f-1)
    y(:,k)=C*x(:,k)+v(k);
    u(:,k)=-R^(-1)*B'*(eye(mQ)+K*B*R^(-1)*B')^(-1)...
        *(K*A*x(:,k)+g);
    x(:,k+1)=A*x(:,k)+A*P*C'*(R_v+C*P*C')^(-1)...
        *(y(:,k)-C*x(:,k))+B*u(:,k);

    g=-Q*xr(:,k)+A'*(eye(mQ)+K*B*R^(-1)*B')^(-1)*g;
    K=Q+A'*(eye(mQ)+K*B*R^(-1)*B')^(-1)*K*A;
    P=A*P*A'+G*Q_w*G'-A*P*C'*(R_v+C*P*C')^(-1)*C*P*A';

end

%[Mean, Covariance] = ecmnmle(Data);
subplot(3,1,1)
stairs(x(1,:),'r')
hold on
stairs(xr(1,:),'b')
grid on
legend('x_1','x_r_e_f')
title('state x_1')

subplot(3,1,2)
stairs(x(2,:),'r')
hold on
stairs(xr(2,:),'b')
grid on
legend('x_2','x_r_e_f')
title('state x_2')

subplot(3,1,3)
stairs(x(3,:),'r')
hold on
stairs(xr(3,:),'b')
grid on
legend('x_3','x_r_e_f')
title('state x_3')
xlabel('sample')


stairs(u(1,:),'r')
```

Example 6:

```
% Minimum Variance Filter (MVF)
clear all
clc

% A, C, G, Q_w, R_v

%x_i+1=Ax_i+Bu_i+Gw_i
%y_i=Cx_i+v


%fixed terminal state
i_f=200; %step
t=0:(i_f-1);

N=10; %Predictive horizon

%problem data
A =[
    -0.3252    0.6504    0.6098;
    0.8130         0    0.8130;
    0.3659    0.2439   -0.0813];

B =[
    0.1000   0.2000;
    1.0000    0.5000;
         0    1.0000];

G=[0.9; 0; -0.255];
C=[1 0 0.15;0 -1 0.2];

%initial condition
x0=[-3;5;2];

[nQ mQ] =size(A);
[nR mR] =size(B);
[nC mC] =size(C);
[nG mG] =size(G);

%constructing \bar{B}_N, \bar{C}_N and \bar{G}_N
BB=B;
for k=1:N-1
    BB=blkdiag(B,BB); % constracting diag(B,B,...,B)
end

CC=C;
for k=1:N-1
    CC=blkdiag(C,CC); % constracting diag(C,C,...,C)
end

GG=G;
for k=1:N-1
    GG=blkdiag(G,GG); % constracting diag(G,G,...,G)
```

```
end

AAinv=zeros(mQ*N);
Achange_inv=A\eye(mQ); % inv(A)
for k=1:N-1
    Achange_inv=[A\eye(mQ) ,A\Achange_inv ]; %[A^(-1), ... , A^(-N)]
end

for k=1:N
    AAinv((k-1)*mQ+1:k*mQ,(k-1)*mQ+1:mQ*N)=Achange_inv(:,1:(N-k+1)*mQ);
end  %[A^(-1), A^(-2), ... , A^(-N)]
     %[   0  , A^(-1), ...,A^(-N+1)]
     %[                           ]
     %[   0  ,   0   , ... , A^(-2)]
     %[   0  ,   0   , ... , A^(-1)]


B_N=-CC*AAinv*BB;
C_N=CC*AAinv(:,(N-1)*mQ+1:N*mQ);
G_N=-CC*AAinv*GG;



%objective function weighting
Q=eye(mQ);
Q_f=5*eye(mQ);
R=10*eye(mR);


%Normal distribution generation
w=normrnd(0,0.5,i_f,mG);
v=normrnd(0,0.20,i_f,nC);

%covariance matrix
Q_w=cov(w);
R_v=cov(v);

%finding \Psi_N
QQw=Q_w;
for k=1:N-1
    QQw=blkdiag(Q_w,QQw); % constracting diag(Q_w,Q_w,...,Q_w)
end
RRv=R_v;
for k=1:N-1
    RRv=blkdiag(R_v,RRv); % constracting diag(R_v,R_v,...,R_v)
end
Psi_N=G_N*QQw*G_N'+RRv;

%MVF matrix
H_B=((C_N'/Psi_N)*C_N)\(C_N'/Psi_N);

%reference signal
```

```
xr=zeros(3,i_f);
xr(1,:)=sin(t/15);
xr(2,:)=sin(t/15+1);
xr(3,:)=sin(t/15);



%initial state, control, measurement
x=zeros(mQ,i_f);

u=zeros(mR,i_f);
y=zeros(nC,i_f);

Y=zeros(nC*N,1);
U=zeros(mR*N,1);

x(:,1)=x0;


%initialization Y and U
for k=1:N
    Y((k-1)*nC+1:k*nC,1)=y(:,k);
    U((k-1)*mR+1:k*mR,1)=u(:,k);
end


K=Q_f;
g=-Q_f*xr(:,i_f);

for k=1:(i_f-1)



    x(:,k)=H_B*(Y-B_N*U);
    y(:,k)=C*x(:,k)+v(k,:)';

    u(:,k)=-R^(-1)*B'*(eye(mQ)+K*B*R^(-1)*B')^(-1)*(K*A*x(:,k)+g);


    Y(1:(N-1)*nC)=Y(nC+1:N*nC);
    Y((N-1)*nC+1:N*nC)=y(:,k);
    U(1:(N-1)*mR)=U(mR+1:N*mR);
    U((N-1)*mR+1:N*mR)=u(:,k);

    g=-Q*xr(:,k)+A'*(eye(mQ)+K*B*R^(-1)*B')^(-1)*g;
    K=Q+A'*(eye(mQ)+K*B*R^(-1)*B')^(-1)*K*A;

end

%[Mean, Covariance] = ecmnmle(Data);
subplot(3,1,1)
stairs(x(1,:),'r')
```

Example 7a:

```
% QP solver
%
clear all
clc

% x_{k+1}=Ax_k+Bu_k;
% cost function; J=x_k^TQx_k+u_k^TRu_k

%i_f=20; %step

N=50; %Predictive horizon

%problem data
A=[1  0.8; 0 -0.7];
B=[0; 1];

%initial condition
x0=[3;-1];


[nQ mQ] =size(A);
[nR mR] =size(B);

%objective function weighting
Q=eye(mQ);
Q_f=5*eye(mQ);
R=10*eye(mR);


%%%%% consrtucting \tilde{A} and \tilde{b}  %%%%%
Atil=zeros(N*mQ,N*(mQ+mR));
btil=zeros(N*mQ,1);
btil(1:mQ)=A*x0; % just for one step optimization

first_row_Atil=[-B,eye(mQ)];
second_row_Atil=[-A,-B,eye(mQ)]; %Atil=[-B I 0 0 0 0;
                                   % 0 -A -B I 0 0;
                                   % 0 0 0 -A -B I ];

Atil(1:nR,1:(mQ+mR))=first_row_Atil;
for k=1:N-1
    Atil(k*nR+1:(k+1)*nR, ...
        k*mR+(k-1)*mQ+1:(k+1)*mR+(k+1)*mQ)=second_row_Atil;
end

%%%%%   consrtucting H  %%%%%
H=zeros(N*(mQ+mR));
for k=1:N
    H((k-1)*mR+(k-1)*mQ+1:k*mR+(k-1)*mQ,...
        (k-1)*mR+(k-1)*mQ+1:k*mR+(k-1)*mQ)=R;
    H(k*mR+(k-1)*mQ+1:k*mR+k*mQ,...
        k*mR+(k-1)*mQ+1:k*mR+k*mQ)=Q;
```

```
    if k==N
        H(k*mR+(k-1)*mQ+1:k*mR+k*mQ,...
        k*mR+(k-1)*mQ+1:k*mR+k*mQ)=Q_f;
    end
end

%%%%%  consrtucting G  %%%%%
first_col_G=[-eye(mR);eye(mR)];
second_col_G=[-eye(mQ);eye(mQ)]; %G=[-I 0 ;
                                    % I 0 ;
                                    % 0 -I ;
                                    % 0 I  ];
G_block=blkdiag(first_col_G,second_col_G);
G=G_block;
for k=1:N-1
    G=blkdiag(G_block,G); %constracting diag(G_block,...,G_block)
end


%%%%%  consrtucting h  %%%%%
h=zeros(2*N*(mQ+mR),1);
%
% for k=1:N
% bnds = randn(mR,2);
% u_low = min( bnds, [], 2 );
% u_up = max( bnds, [], 2 );
%
% bnds = 5*randn(mQ,2);
% x_low = min( bnds, [], 2 );
% x_up = max( bnds, [], 2 );
%
%
% h(2*(k-1)*mR+2*(k-1)*mQ+1:2*(k-1)*mR+2*(k-1)*mQ+mR)=-u_low;
% h(2*(k-1)*mR+2*(k-1)*mQ+1+mR:2*k*mR+2*(k-1)*mQ)=u_up;
% h(2*k*mR+2*(k-1)*mQ+1:2*k*mR+2*(k-1)*mQ+mQ)=-x_low;
% h(2*(k-1)*mR+2*(k-1)*mQ+1+2*mR+mQ:2*k*mR+2*k*mQ)=x_up;
% end

%another deterministic selection for u and x.
for k=1:N
u_low=-0.8*ones(mR,1);
u_up=0.8*ones(mR,1);

x_low=-5*ones(mQ,1);
x_up=5*ones(mQ,1);
h(2*(k-1)*mR+2*(k-1)*mQ+1:2*(k-1)*mR+2*(k-1)*mQ+mR)=-u_low;
h(2*(k-1)*mR+2*(k-1)*mQ+1+mR:2*k*mR+2*(k-1)*mQ)=u_up;
h(2*k*mR+2*(k-1)*mQ+1:2*k*mR+2*(k-1)*mQ+mQ)=-x_low;
h(2*(k-1)*mR+2*(k-1)*mQ+1+2*mR+mQ:2*k*mR+2*k*mQ)=x_up;
end

% initialization z, \lambda and v
```

```
z=zeros(N*(mQ+mR),1);
lamb=0.1*rand(2*N*(mQ+mR),1);  % zeros(2*N*(mQ+mR),1);
v=zeros(N*mQ,1);
y=zeros(N*(mQ+mR)+2*N*(mQ+mR)+N*mQ,1);
deltalamb=-0.001*rand(2*N*(mQ+mR),1);%technical attention!
beta=0.1;
mu=10;
m=length(h);

s=0.999*min(1,min(-lamb./deltalamb));
%nor=zeros(2,N);
%loop
for k=1:40  % may be changed, but we take is instead 'while'
f=G*z-h;
eta_hat=-f'*lamb;

t=mu*m/eta_hat;

r_dual=H*z+G'*lamb+Atil'*v;
r_cent=-diag(lamb)*f-(1/t)*ones(2*N*(mQ+mR),1);
r_pri=Atil*z-btil;

r_t=[r_dual;r_cent;r_pri];

KKT_mat=[H,G',Atil';
    -diag(lamb)*G,-diag(f),zeros(2*N*(mQ+mR),N*mQ);
    Atil,zeros(N*mQ,2*N*(mQ+mR)), zeros(N*mQ)];

deltaY=-KKT_mat\r_t;
y=y+s*deltaY;

z=y(1:N*(mQ+mR));
lamb=y(N*(mQ+mR)+1:3*N*(mQ+mR));
v=y(3*N*(mQ+mR)+1:3*N*(mQ+mR)+N*mQ);

s=beta*s;
%nor(1,k)=norm(r_pri);
%nor(2,k)=norm(r_dual);
end


x=zeros(mQ,N);
u=zeros(mR,N);
%pull out u and x from z;
for k=1:N
u(:,k)=z((k-1)*mR+(k-1)*mQ+1:(k-1)*mR+(k-1)*mQ+mR);
x(:,k)=z((k-1)*mR+(k-1)*mQ+1+mR:(k-1)*mR+(k-1)*mQ+mQ+mR);
end
x(:,2:N+1)=x;
x(:,1)=x0;
plot(1:N+1,x(1,:),1:N+1,x(2,:))
```

Example 7b:

```
clear all
clc


N=50;
%A=[1  0.8; 0 0.7];
%B=[0; 1];

A=[1  0.8; 0 -0.7];
B=[0; 1];


x0=[3;-1];



cvx_begin
variables x(2,N) u(N-1)
J=x(:,1)'*x(:,1) +10* u(1)'*u(1);
for j=1: (N-1)
    J=J+x(:,j)'*x(:,j) + 10*u(j)'*u(j);
end

J=J+x(:,N)'*[5,0;0,5]*x(:,N);

minimize ( J )
subject to
x(:,1)==x0;

for j=1:(N-1)
x(:,j)>= -5*ones(2,1);
x(:,j)<= 5*ones(2,1);
u(j)>= -0.8;
u(j)<= 0.8;
end

for j=1:(N-1)
x(:,j+1)==A*x(:,j)+B*u(j);
end

cvx_end


stairs(1:N,x(1,1:N),'b')
hold on
stairs(1:N,x(2,1:N),'r')
title('States in QP problem')
xlabel('sample')
legend('x_1','x_2')
```

Example 7c:

```
clear all
clc


i_f=50;
N=10;

A=[1  0.8; 0 -0.7];
B=[0; 1];


x0=[3;-1];
x00=x0; %save x0
xx=zeros(2,i_f);
uu=zeros(1,i_f);

for k=1:i_f

    cvx_begin
    variables x(2,N) u(N-1)
    J=x(:,1)'*x(:,1) +10* u(1)'*u(1);
    for j=1: (N-1)
        J=J+x(:,j)'*x(:,j) + 10*u(j)'*u(j);
    end

    J=J+x(:,N)'*[5,0;0,5]*x(:,N);

    minimize ( J )
    subject to
    x(:,1)==x0;

    for j=1:(N-1)
    x(:,j)>= -5*ones(2,1);
    x(:,j)<= 5*ones(2,1);
    u(j)>= -0.8;
    u(j)<= 0.8;
    end

    for j=1:(N-1)
    x(:,j+1)==A*x(:,j)+B*u(j);
    end
    cvx_end

    uu(k)=u(1);
    xx(:,k)=x(:,2);
    x0=x(:,2);
end

xx(:,2:i_f+1)=xx;
xx(:,1)=x00;

stairs(0:i_f,xx(1,:),'b')
```

References

[1] W. H. Kwon and S. Han. *Receding Horizon Control.* Springer-Verlag, 2005.

[2] Stephan Boyd *Convex Optimization* Cambridge University Press, 2004

[3] Y. Wang and S. Boyd. *Fast model predictive control using online optimization.* In Proceedings IFAC World Congress, pages 6974–6997, July 2008.

[4] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming (web page and software). `http://www.stanford.edu/ boyd/cvx/`, July 2008.

[5] Eduardo F. Camacho and Carlos Bordons *Model Predictive Control* Springer-Verlag London, 1999.

[6] Ton J.J. van den Boom and Anton A. Stoorvogel *Model Predictive Control* DISC Course, Lecture Notes, Delft Center for Systems and Control Delft University of Technology, 2010

[7] P.E. Orukpe *Basics of Model Predictive Control* Imperial College,London,2005

[8] Donald E. Kirk *Optimal Control Theory* Mineola, New York, 1970

[9] Carsten Scherer and Siep Weiland *Linear Matrix Inequalities in Control* Delft Center for Systems and Control Delft University of Technology, 2009

[10] H. E. Krogstad *Quadratic Programming Basics* TMA 4180 Optimeringsteori Spring 2005/Rev. 2008