Chapter 6

Lecture 6: Practical MPC

6.1 How to Choose Terminal Set

From last chapter, we knows that in order to prevent finite-horizon MPC from limited horizon, we need a terminal set.

6.1.1 Infinite LQR Controller Solution

How to choose terminal set can be difficult in general, but one case is constructive: that is, suppose a **final infinite LQR controller**

$$\begin{gathered} f(x,u) = Ax + Bu \quad \mathbb{X} \text{ and } \mathbb{U} \text{ polytopes} \quad l(x,u) = x^T Q x + u^T R u \\ \kappa_f(x) = Kx \quad K = -\left(R + B^T P B\right)^{-1} B^T P A \\ P = Q + A^T P A - A^T P B \left(R + B^T P B\right)^{-1} B^T P A \\ V_f(x) := x^T P x = \sum_{i=0}^{\infty} x_i^T Q x_i + x_i^T K^T R K x_i \end{gathered} \tag{6.1}$$

By using a LQR controller, we can then define terminal invariant set: choose the terminal set $\mathcal{X}_f$ to be the maximum invariant set for the closed-loop system $x^+ = (A + BK)x$ subject to $\mathcal{X}_f \subset \mathbb{X}, K\mathcal{X}_f \subset \mathbb{U}$.

6.1.2 An Interesting Phenomenon

If the stage cost function outside terminal set has the same format as the cost function in terminal set, during the receding horizon process of an MPC, once in one optimization, we found the final step state is not at the edge of the terminal set, then the following optimization will keep on the same trajectory.

6.2 Practical Problem 1: Enlarging the Feasible Set

A significant problem take by terminal set is the terminal set reduces the feasible set. One example is shown in Figure 6.1.

Tow methods can be used to solve this problem: **Remove the terminal set** or **Soft Constrained MPC**.

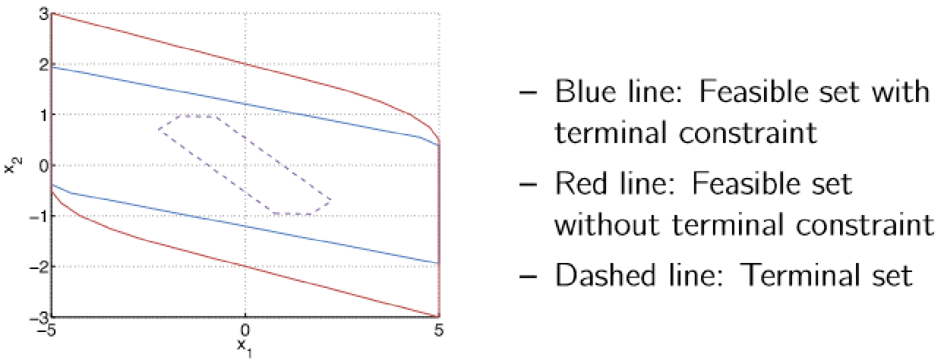


Figure 6.1: Terminal Set reduces the Feasible Set

6.2.1 MPC without Terminal Set

We can remove terminal constraints while maintaining stability if

- initial state lie in sufficiently small subset of feasible set
- N is sufficiently large

such that terminal state satisfies terminal constraint without enforcing it in the optimization. Solution of the finite horizon MPC problem corresponds to the infinite horizon solution.

6.2.2 Soft Constrained MPC

Motivation

1. State Constraints may lead to infeasibility
2. Controller must provide some input in every circumstances
3. Input constraints are always mandatory and state constraints can always be temporarily violated

Objectives

1. Minimize the duration of violation
2. Minimize the size of violation

Model: Self-Constrained Model

We can relax constraints by introducing so-called **slack variables** $\epsilon_i \in \mathcal{R}^p$. And then penalize the amount of constraint violation in the cost by means of penalty $p(\epsilon)$.

$$\begin{aligned} &\min_u \textstyle\sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i + \rho(\epsilon_i) + x_N^T P x_N + \rho(\epsilon_N) \\ \text{s.t.}\quad & x_{i+1} = Ax_i + Bu_i \\ & H_x x_i \leq k_x + \epsilon_i, \\ & H_u u_i \leq k_u, \\ & \epsilon_i \geq 0 \end{aligned} \tag{6.2}$$

Another question is how to choose penalty? There are two main trends:

- Quadratic penalty: $\rho(\epsilon_i) = \epsilon_i^T S \epsilon_i$
- Quadratic and linear norm penalty $\rho(\epsilon_i) = \epsilon_i^T S \epsilon_i + s\,\|\epsilon_i\|_{1/\infty}$

And some experience are as follows:

- For Quadratic Penalty: Increase in S leads to reduced size of violation but longer duration
- If weight s is large enough, constraints are satisfied if possible
- Increasing s results in increasing peak violations and decreasing duration

Notes: Standard methods for soft constraints MPC do not provide a stability guarantee for open-loop unstable systems

Model: Separation of Objectives

We can also separate the objectives into two steps.

First, we try to minimize the violation over the horizon.

$$\begin{aligned} \epsilon^{\min} = \operatorname{argmin}_{\mathrm{u},\epsilon} \epsilon_i^T S \epsilon_i + s^T \epsilon_i \\ \text{s.t. } x_{i+1} = Ax_i + Bu_i \\ H_x x_i \leq K_x + \epsilon_i \\ H_u u_i \leq K_u \\ \epsilon_i \geq 0 \end{aligned} \tag{6.3}$$

Then after get the ϵ_i^{min}, we optimize the controller performance:

$$\begin{aligned} &\min_u \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i + x_N^T P x_N \\ &\text{s.t. } x_{i+1} = Ax_i + Bu_i \\ &\quad H_x x_i \leq k_x + \epsilon_i^{\min} \\ &\quad H_u u_i \leq k_u \end{aligned} \tag{6.4}$$

6.3 Tracking Problem

In tracking problem, we try to track given reference r such that $y_k \to r$ as $k \to \infty$

We can also use tracking problem method in tracking piecewise constant reference, but the change of reference can render the optimization problem infeasible.

6.3.1 Model

Target Condition

$$\begin{aligned} x_s = Ax_s + Bu_s \\ Cx_s = r \end{aligned} \Rightarrow \underbrace{\begin{bmatrix} 1-A & -B \\ C & 0 \end{bmatrix}}_{(n_x+n_y)\times(n_x+n_u)} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} \tag{6.5}$$

When implementing, we can solve this problem by solving an optimization problem.

- If target problem is feasible:

$$\begin{aligned}
&\min u_s^T R_s u_s \\
&\text{s.t.} \begin{bmatrix} I-A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} \\
&H_x x_s \le k_x \\
&H_u u_s \le k_u
\end{aligned} \tag{6.6}$$

- If target problem in infeasible:

$$\begin{aligned}
&\min \quad (Cx_s - r)^T Q_s (Cx_s - r) \\
&\text{s.t. } x_s = Ax_s + Bu_s \\
&H_x x_s \le k_x \\
&H_u u_s \le k_u
\end{aligned} \tag{6.7}$$

$$\begin{aligned} \Delta x &= x - x_s \\ \Delta u &= u - u_s \end{aligned} \Rightarrow \begin{aligned} \Delta x_{k+1} &= x_{k+1} - x_s \\ &= Ax_k + Bu_k - (Ax_s + Bu_s) \\ &= A\Delta x_k + B\Delta u_k \end{aligned} \tag{6.8}$$

Delta Formation

After get the suitable target, we can generate a MPC model problem in Delta Formation

$$\begin{aligned}
\min \quad & \textstyle\sum_{i=0}^{N-1} \Delta x_i^T Q \Delta x_i + \Delta u_i^T R \Delta u_i + V_f(\Delta x_N) \\
\text{s.t.} \quad & \Delta x_0 = \Delta x \\
& \Delta x_{i+1} = A\Delta x_i + B\Delta u_i \\
& H_x \Delta x_i \le k_x - H_x x_s \\
& H_u \Delta u_i \le k_u - H_u u_s \\
& \Delta x_N \in \mathcal{X}_f
\end{aligned} \tag{6.9}$$

The control framework is shown as 6.2. The control framework then is divided into two steps:

- Find optimal sequence of Δu^*
- Input applied to the system $u_0^* = \Delta u_0^* + u_s$

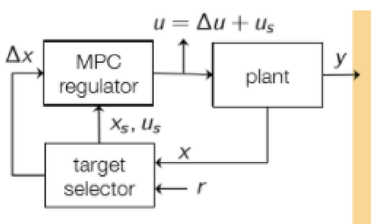


Figure 6.2: Tracking Problem

6.3.2 Convergence Analysis

Assume target is feasible with $x_s \in \mathbb{X}, u_s \in \mathbb{U}$ and choose terminal weight $V_f(x)$ and constraint $\mathcal{X}_f$ as in the regulation case satisfying:

- $\mathcal{X}_f \subseteq \mathbb{X}, Kx \in \mathbb{U}$ for all $x \in \mathcal{X}_f$
- $V_f(x^+) - V_f(x) \leq -I(x, Kx)$ for all $x \in \mathcal{X}_f$

If in addition the target reference x_s, u_s is such that

- $x_s \oplus \mathcal{X}_f \subseteq \mathbb{X}, K\Delta x + u_s \in \mathbb{U}$ for all $\Delta x \in \mathcal{X}_f$

then the closed-loop system converges to the target reference, i.e. $x_k \to x_s$ and therefore $y_k = Cx_k \to r$ for $k \to \infty$

6.3.3 Terminal Set Problem

Based on previous convergence analysis, because of shifted terminal set, if we want to guarantee convergence, the set of feasible targets is really small. Which is shown in Figure 6.3.

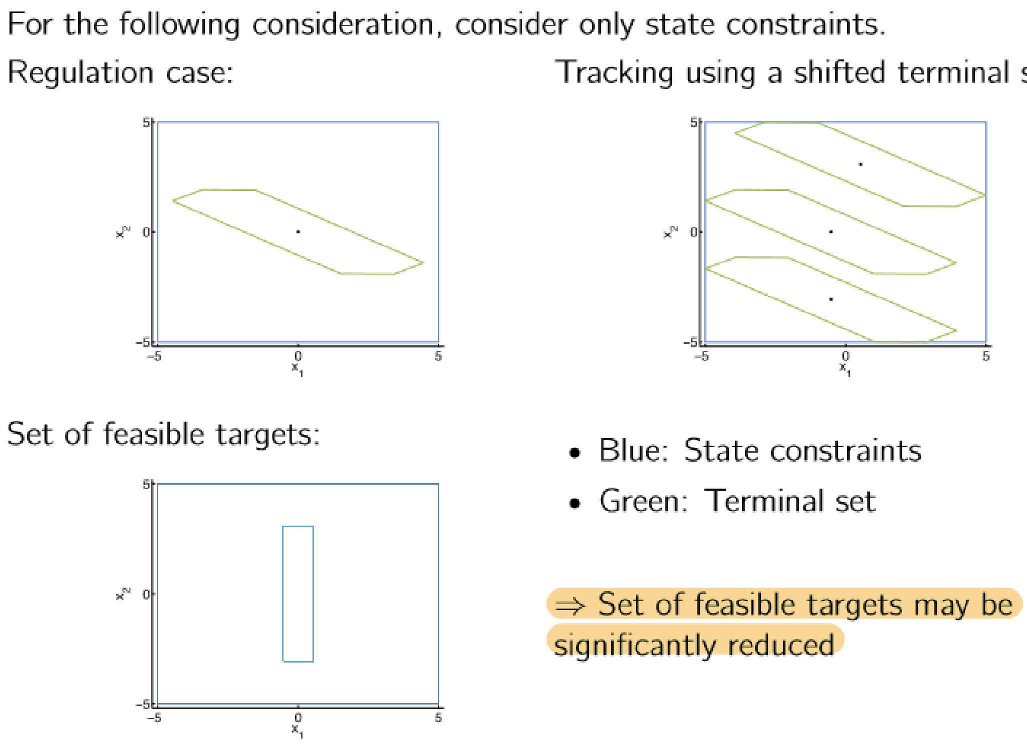


Figure 6.3: Track Terminal Set Problem

A possible solution is enlarging set of feasible targets by scaling terminal set for regulation. The idea is squeeze the terminal set when the setting point get close to the boundary of the state constraints. This method is illustrated in Figure 6.4.

6.4 Offset Free Control (with Constant Disturbance)

If a system is in the presence of a disturbance, then we want it converges to set-point with zero offset.

6.4.1 Basic Idea and Augmented Model

In order to finish that, our basic idea is:

- Model the disturbance

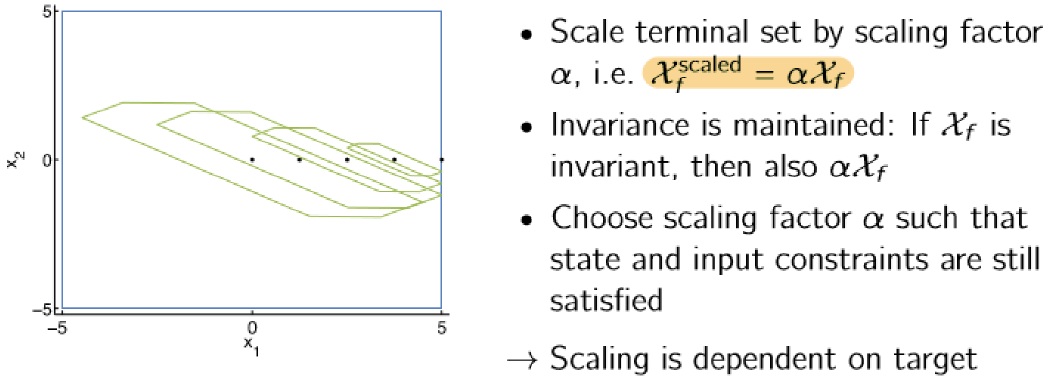


Figure 6.4: Scaling Terminal Set

- Use the output measurements and model to estimate the state and the disturbance
- Find control inputs that use the disturbance estimate to remove offset

We can then build an augmented model for the system

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + B_d d_k \\ d_{k+1} &= d_k \\ y_k &= Cx_k + C_d d_k \end{aligned} \tag{6.10}$$

6.4.2 Linear State Estimation

For estimating state and disturbances, we can use a linear estimator to estimate. The model is as following:

$$\begin{bmatrix} \hat{x}_{k+1} \\ \hat{d}_{k+1} \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{d}_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \begin{bmatrix} L_x \\ L_d \end{bmatrix} \left(C\hat{x}_k + C_d\hat{d}_k - y_k \right) \tag{6.11}$$

Then the error dynamic will become:

$$\begin{aligned} \begin{bmatrix} x_{k+1} - \hat{x}_{k+1} \\ d_{k+1} - \hat{d}_{k+1} \end{bmatrix} &= \begin{bmatrix} A & B_d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k \\ &\quad - \begin{bmatrix} A & B_d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{d}_k \end{bmatrix} - \begin{bmatrix} B \\ 0 \end{bmatrix} u_k - \begin{bmatrix} L_x \\ L_d \end{bmatrix} \left(C\hat{x}_k + C_d\hat{d}_k - Cx_k - C_d d_k \right) \\ &= \left(\begin{bmatrix} A & B_d \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} L_x \\ L_d \end{bmatrix} \begin{bmatrix} C & C_d \end{bmatrix} \right) \begin{bmatrix} x_k - \hat{x}_k \\ d_k - \hat{d}_k \end{bmatrix} \end{aligned} \tag{6.12}$$

Which means we can let the estimator stable by choosing appropriate $L = \begin{bmatrix} L_x \\ L_d \end{bmatrix}$, i.e. choosing L let error dynamics become stable and converges to zero. This can be done by a lot of methods, for example, pole configuration.

6.4.3 Deal With Noise

If instead of constant disturbance, we have some non-constant noise:

$$
\begin{aligned}
\begin{bmatrix} x_{k+1} \\ d_{k+1} \end{bmatrix} &= \begin{bmatrix} A & B_d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + w \\
y_k &= C x_k + C_d d_k + v
\end{aligned}
\tag{6.13}
$$

where w, v are white noise with covariance matrices Q_e R_e respectively.

There are mainly two ways to be used

- use Kalman filter to estimate both the state and the intergrating disturbance
- optimal estimator gain L that minimizes the variance of the estimation error.

6.5 OffSet Free Tracking

6.5.1 Model

The basic model of off-set free tracking problem is

$$
\begin{aligned}
x_s &= A x_s + B u_s + B_d d_s \\
y_s &= C x_s + C_d d_s = r
\end{aligned}
\tag{6.14}
$$

So, now we should set our target condition according to the disturbance.

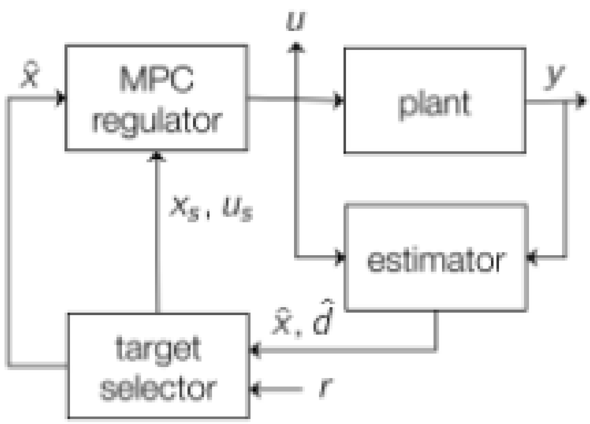
$$
\begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} B_d \hat{d} \\ r - C_d \hat{d} \end{bmatrix}
\tag{6.15}
$$

Normal Formation

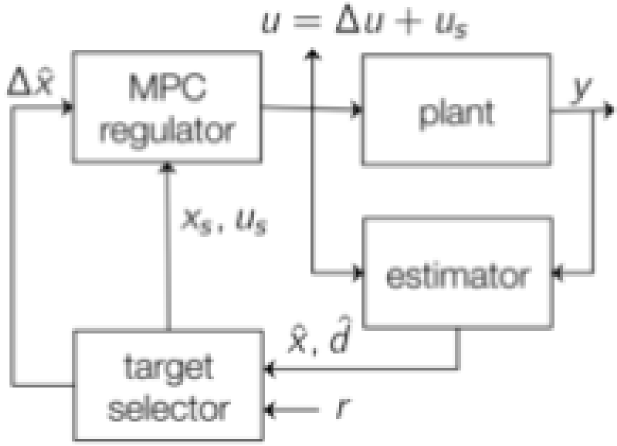
$$
\begin{aligned}
\min \quad & \sum_{i=0}^{N-1} (x_i - x_s)^T Q (x_i - x_s) + (u_i - u_s)^T R (u_i - u_s) + V_f (x_N - x_s) \\
s.t. \quad & x_0 = \hat{x} \\
& d_i = \hat{d} \\
& x_{i+1} = A x_i + B u_i + d_i \\
& H_x x_i \le k_x \\
& H_u u_i \le k_u \\
& x_N - x_s \in \mathcal{X}_f
\end{aligned}
\tag{6.16}
$$

The whole process is: at each sampling time

1. Estimate state and disturbance $\hat{x}$, $\hat{d}$
2. Obtain (x_s, u_s) from steady-state target problem using disturbance estimate
3. Solve MPC problem for tracking using disturbance estimate $\hat{d}$



(a) off set free tracking normal formation



(b) off set free tracking delta formation

Figure 6.5: off set free tracking

Delta Formation

$$\begin{array}{ll} \min & \sum_{i=0}^{N-1} \Delta x_i^T Q \Delta x_i + \Delta u_i^T R \Delta u_i + V_f\left(\Delta x_N\right) \\ \text{s.t.} & \Delta x_0 = \Delta \hat{x} \\ & \Delta x_{i+1} = A\Delta x_i + B\Delta u_i \\ & H_x \Delta x_i \leq k_x - H_x x_s \\ & H_u \Delta u_i \leq k_u - H_u u_s \\ & \Delta x_N \in \mathcal{X}_f \end{array} \tag{6.17}$$

The whole process is: at each sampling time

1. Estimate state and disturbance $\hat{x}$, $\hat{d}$
2. Obtain (x_s, u_s) from steady-state target problem using disturbance estimate
3. initial state $\Delta\hat{x} = \hat{x} - x_s$
4. Solve MPC problem for tracking in Delta Formation.

The structure of these two methods are shown in Figure 6.5.

6.5.2 Convergence Analysis

Assumption

- Consider the case $n_d = n_y$, i.e. number of disturbance states equal to number of measured outputs.
- Assume target steady-state problem is feasible and constraints are not active at steady-state.

Convergence

If closed-loop system converges to $\hat{x}_s, \hat{d}_s, y_s$, i.e. $\hat{x}_k \to \hat{x}_s, \hat{d}_k \to \hat{d}_s, y_k \to y_s$ as $k \to \infty$, then

$$y_k = Cx_k \to r \text{ for } k \to \infty$$

Summary

In this chapter, we first introduce a common method to determine the terminal controller and terminal set, that is use a "virtual LQR controller". Then we proposed that the terminal set may shrink the feasible set and we proposed two possible solution to solve this problem: 1. MPC without constraints and 2. MPC with soft constrained.

Then we introduce the tracking problem, in tracking problem, we first compute a steady state based on reference. Then we introduce how to due with constant disturbance in control, that is add a stable observer to estimate both the state and the disturbance. We call this problem the offset-free control problem. Finally we combine offset-free control and tracking problem and introduce how to solve offset-free tracking problem.