Research Article

Getting a Suitable Terminal Cost and Maximizing the Terminal Region for MPC

Wang Ya-feng,[1,2] Sun Fu-chun,[2] Zhang You-an,[1] Liu Hua-ping,[2] and Min Haibo[2]

[1] *Department of Control Engineering, Naval Aeronautical and Astronautical University, Yantai 264001, China*
[2] *Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*

Correspondence should be addressed to Wang Ya-feng, wyfyxy@sina.com

The model predictive control (MPC) subject to control and state constraint is studied. Given a terminal cost, a terminal region is obtained through iterative estimation by using support vector machine (SVM). It is proved that the obtained terminal region is the largest terminal region when the terminal cost is given. The relationships between terminal cost and terminal region and between terminal cost and total cost are discussed, respectively. Based on these relationships, a simple method to get a suitable terminal cost is proposed and it can be adjusted according to our need. Finally, some experiment results are presented.

1. Introduction

Model predictive control also known as receding horizon control has become quite popular recently. The key advantage is its ability to handle control and state constraints. It was pointed out in [1] that what MPC solves is the standard optimal control (SOC) problem except that it uses the finite-horizon optimization (some paper used quasi-infinite horizon optimization, such as [2, 3]) to replace the infinite-horizon optimization of SOC and the control is computed online.

Generally, to say that an MPC is good or bad, it is contrasted with SOC from two aspects: the domain of attraction and the total cost (the wasted performance index from initial time to infinity). If an MPC has a larger domain of attraction and for any initial state point the total cost is fewer than those of another MPC, it is considered as a better MPC. In MPC, there are three factors playing important roles in its performances on the two aspects as just mentioned: the prediction horizon, the terminal region, and the terminal cost. As known to all, lengthening the prediction horizon, the domain of attraction will be enlarged and the

total cost will decrease, but the computation burden of online optimization will increase. Recently, many attentions have concentrated on the latter two factors: (I) how to get a large terminal region? (II) and how to get a suitable terminal cost? Here, only some typical papers are listed. Chen and Allgöwer [2] presented a terminal cost by using SOC method based on the linearized model of system and took an ellipsoidal set in which the state can be drived to the equilibrium point by linear feedback control as the terminal region. Cannon et al. [4] used a polytopic set to replace the ellipsoidal set. De Doná et al. [5] took the stabilizable region of using saturated control as the terminal region. Ong et al. [6] got a very large stabilizable region and a terminal cost of using linear feedback control via support vector machine. Limon et al. [7] proved that, for MPC without terminal state constraint in its on-line optimization, the terminal region will be enlarged by weighting the terminal cost. Most of these papers have a common shortage; the terminal region is computed under a precondition; some explicit controller was given in advance, like linear feedback controller and saturated controller. So, the computed terminal region is somewhat conservative, in other words, it is not the largest one.

In this paper, a novel method is proposed to get a terminal state region. Given a terminal cost, a set sequence is obtained by using one-step set contraction iteratively. It is proved that, when the iteration time goes to infinity, this set sequence will converge to the maximal terminal region. In this sequence, each set is estimated by using support vector machine (SVM, see [7, 8] for details). Next, the relationships between terminal cost and terminal region and between terminal cost and total cost are discussed, respectively. Then, a simple method to get a suitable terminal cost according to our need is given. Finally, some experiment results are presented.

2. The Relationship between SOC and MPC

As mentioned, MPC is an approximation to SOC, and SOC is the counterpoint to evaluate MPC. Here, the study on MPC begins with the comprehension of SOC. Consider the discrete-time system as follows:

$$x_{k+1} = f(x_k, u_k), \tag{2.1}$$

where $x_k \in R^n$, $u_k \in R^m$ are the state and the input of the system at sampling time $t = k$, respectively. $x_{k+1} \in R^n$ is the successor state and the mapping $f : R^{n+m} \mapsto R^n$ with $f(\mathbf{0}, \mathbf{0}) = \mathbf{0}$ is known. The system is subject to constraints on both state and control action, and they are given by $x_k \in X, u_k \in U$, where X is a closed set and U a compact set, both of them containing the origin. The control objective is usually to steer the state to the origin.

The optimization problem $P_\infty(x_0)$ of SOC at the initial state x_0 can be stated as follows:

$$\min_{u(i,x_0) \in U} J_\infty(\mathbf{u}, x_0) = \sum_{i=0}^{\infty} q(x(i, x_0), u(i, x_0)),$$

$$\text{s.t. } x(i + 1, x_0) = f(x(i, x_0), u(i, x_0)), \tag{2.2}$$

$$x(i + 1, x_0) \in X, \quad u(i, x_0) \in U,$$

where $x(0, x_0) = x_0$, $q(x, u)$ is the stage cost and its form is chosen as $q(x, u) = x^T Q x + u^T R u$ in which Q, R are positive definites.

It is well known that the stability is guaranteed if $P_\infty(x_0)$ has feasible solution. Here, assume that the solution of $P_\infty(x_0)$ is known if it has the following. $J_\infty^*(x_0)$ is defined as the solution, $u_\infty^*(i, x_0)$, $i = 1, 2, \ldots, \infty$, the optimal control trajectory and $x_\infty^*(i, x_0)$, the corresponding state trajectory. As is well known, $J_\infty^*(x_0)$ is the total cost of using $u_\infty^*(i, x_0)$ to drive x_0 to 0.

But solving $P_\infty(x_0)$ is not an easy job, especially when f is nonlinear. To avoid this problem, the infinite-horizon optimization of SOC can be approximated by the finite-horizon optimization of MPC (As mentioned, quasi-infinite horizon optimization was used in [2, 3]. For convenience, we consider it to belong to the frame of finite-horizon optimization).

Similarly, the optimization problem $P_N(x_0)$ of MPC at the initial state x_0 can be stated as

$$
\min_{u(i,x_0)\in U} J_N(\mathbf{u}, x_0) = \sum_{i=0}^{N-1} q(x(i, x_0), u(i, x_0)) + F(x(N, x_0)),
$$

$$
\text{s.t. } x(i + 1, x_0) = f(x(i, x_0), u(i, x_0)),
$$

$$
x(i + 1, x_0) \in X, \quad u(i, x_0) \in U, \quad x(N, x_0) \in X_f,
$$

$$(2.3)$$

where N is prediction horizon, X_f is terminal region, and F is terminal cost satisfying $F(\mathbf{0}) = 0$ and $F(x) \geq \alpha(\|x\|)$, (the mapping $\alpha : R_+ \mapsto R_+$ satisfying $\alpha(0) = 0$ is continuous and strictly increasing, where $R_+ = \{x \in R \mid x > 0\}$).

There exist many optimization algorithms to compute the solution of $P_N(x_0)$. Let $J_N^*(x_0)$ be the solution, $\mathbf{u}_N^*(x_0) = \{u_N^*(0, x_0), \ldots, u_N^*(N-1, x_0)\}$, $\mathbf{x}_N^*(x_0) = \{x_N^*(0, x_0), \ldots, x_N^*(N-1, x_0)\}$ be the optimal control trajectory and corresponding predicted state trajectory of $P_N(x_0)$, respectively.

At sampling time $t = 0$, $u_N^*(0, x_0)$ is inputted to the system. At the next sampling time $t = 1$, $x_1 = f(x_0, u_N^*(0, x_0))$ is outputted, and the control input $u_N^*(0, x_1)$ can be computed by solving the optimization problem $P_N(x_1)$. By repeating this procedure, two trajectories can be obtained: $\mathbf{u}_{\text{RH}}(x_0) = \{u_N^*(0, x_0), u_N^*(0, x_1), \ldots\}$, $\mathbf{x}_{\text{RH}}(x_0) = \{x_0, x_1, \ldots\}$. Here, for convenience, they are called as the receding horizon control trajectory and the receding horizon state trajectory of MPC with $P_N(x_0)$, respectively.

The introduction of X_f and F in (2.3) is to guarantee the closed loop stability of system.

Lemma 2.1. *Define $\Gamma_N := \{x_0 \in X \mid x_N^*(N, x_0) \in X_f\}$. For any $x_0 \in \Gamma_N$ if X_f and F satisfy two conditions as follows:*

(C1) F being a Lyapunov function, more strictly, for any $x \in X_f$, there exists

$$
F(x) \geq \min_{u\in U}\{q(x, u) + F(f(x, u))\},
$$

$$(2.4)$$

(C2) X_f being an invariant set. In other words, one has $f(x, u) \in X_f$. Here, u is the control in (C1).

It is guaranteed that, x will be led to 0 by using $\mathbf{u}_{\text{RH}}(x_0)$. Γ_N is called the domain of attraction.

The proof can be found in [1].

Obviously, the optimal choice of $F(x(N, x_0))$ is $J_\infty(x(N, x_0))$. The total cost with this choice is $J_\infty(x_0)$, namely it is the least one. But, as mentioned, J_∞ can not be obtained generally, so other ways should be found to get F. Certainly, the closer it approaches to J_∞, the better it is. When F is given, X_f can be found to satisfy the conditions (C1) and (C2). There are many X_f satisfying the conditions and different X_f will be obtained by different methods. But, to make the domain of attraction the largest, it is wished that the largest X_f can be obtained. Here, define $X_{f,\max}$ as the largest terminal region when F is given, and in the next section, a novel method to get $X_{f,\max}$ will be proposed.

3. Maximizing the Terminal Region for MPC

Until now, there exist many methods to construct X_f. As mentioned, these methods have a common basic idea: some controller like linear feedback controller or saturation controller is given in advance, then a stabilizable domain of using this controller is computed and works as the terminal region of MPC. It is obvious that this kind of construction is somewhat conservative and the X_f computed by using this method does not approximate to $X_{f,\max}$ to the largest extent.

In this paper, a novel method is proposed in which X_f is constructed directly from conditions (C1) and (C2).

### 3.1. Approximating the Largest Terminal Region Asymptotically

Define X_f as

$$X_f := \left\{ x \in X \mid F(x) \geq F^*_{X_f}(x) \right\}, \tag{3.1}$$

where $F^*_{X_f}(x) = \min F_{X_f}(x)$ is the solution of the following optimization problem:

$$\min_{u \in U} F_{X_f}(x) = q(x, u) + F(f(x, u)), \tag{3.2}$$

$$\text{s.t. } f(x, u) \in X_f.$$

Obviously, for an $x \in X$, it cannot be decided whether x belongs to X_f from (3.1) and (3.2) when X_f is unknown. The difficulty is that the state constraint in the optimization (3.2) uses the X_f itself. To avoid it, the method of asymptotic approximation is adopted. Firstly, an initial set X_f^0 which can be obtained by the following discriminant is given:

$$X_f^0 := \left\{ x \in X \mid F(x) \geq F^*_{X_f^0}(x) \right\}, \tag{3.3}$$
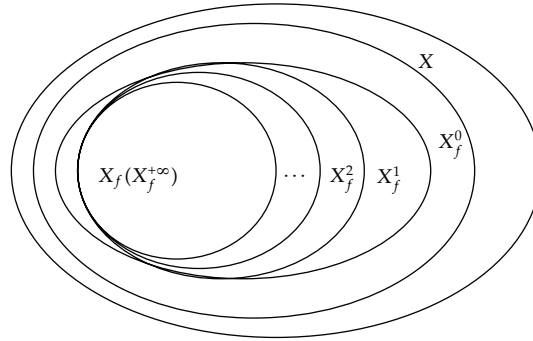
Figure 1: The approximation of $X_{f,\max}$.

where $F^*_{X^0_f}(x)$ is the solution of

$$\min_{u \in U} F_{X^0_f}(x) = q(x, u) + F(f(x, u)),$$

$$\text{s.t. } f(x, u) \in X. \tag{3.4}$$

Then, using X^0_f instead of X in the state constraint in optimization of (3.4), X^1_f can be obtained. One by one, $X^2_f X^3_f, \ldots$ will be obtained. The whole procedure can be pictured as in Figure 1.

In Figure 1, X^1_f is defined as

$$X^1_f := \left\{ x \in X^0_f \mid F(x) \geq F^*_{X^1_f}(x) \right\}, \tag{3.5}$$

where $F^*_{X^1_f}(x)$ is the solution of

$$\min_{u \in U} F_{X^1_f}(x) = q(x, u) + F(f(x, u)),$$

$$\text{s.t. } f(x, u) \in X^0_f. \tag{3.6}$$

Similarly, X^j_f can be defined as?

$$X^j_f := \left\{ x \in X^{j-1}_f \mid F(x) \geq F^*_{X^j_f}(x) \right\}, \tag{3.7}$$

and $F^*_{X^j_f}(x)$ is the solution of

$$\min_{u \in U} F_{X^j_f}(x) = q(x, u) + F(f(x, u)),$$

$$\text{s.t. } f(x, u) \in X^{j-1}_f. \tag{3.8}$$

Obviously, as Figure 1 shows, there exists $X_f^0 \supseteq X_f^1 \supseteq \cdots \supseteq X_f^j \supseteq X_f$. As j increases, X_f^j will converge to a set denoted by $X_f^{+\infty}$. But, whether is $X_f^{+\infty}$ the terminal region we want? Theorem 3.1 provides the answer to this question.

Theorem 3.1. *For X_f^j defined in (3.7), when j goes to infinity, X_f^j will converge to $X_{f,\max}$, namely, as $j \rightarrow +\infty$, one has $X_f^j \rightarrow X_{f,\max}$.*

This theorem is proved by contradiction.

Proof. (A) Assume that there exists X_{spo} which satisfies $X_{\mathrm{spo}} \supset X_{f,\max}$ and $X_f^j \rightarrow X_{\mathrm{spo}}$, as $j \rightarrow +\infty$ then for any $x \in X_{\mathrm{spo}}$, we have $F(x) \geq \min_{u \in U}\{q(x,u) + F(f(x,u))\}$ and $f(x,u) \in X_{\mathrm{spo}}$. Obviously this is contradicted with $X_{f,\max}$ being the largest one satisfying (C1) and (C2).

(B) Similarly, assume that there exists X_{spo} which satisfies $X_{\mathrm{spo}} \subset X_{f,\max}$ and $X_f^j \rightarrow X_{\mathrm{spo}}$, as $j \rightarrow +\infty$, then there exists N with $0 \leq N < +\infty$ satisfying $X_f^N \supseteq X_{f,\max}$ and $X_{f,\max} \setminus X_f^{N+1} \neq \Phi$, where Φ denotes the empty set. Choose any $x \in X_{f,\max} \setminus X_f^{N+1}$, it is obviouse that, x satisfies $F(x) \geq \min_{u \in U}\{q(x,u) + F(f(x,u))\}$ and $f(x,u) \in X_{f,\max} \subseteq X_f^N$. On the other hand, we know that $x \in X_f^N$, so x meets the conditions in the definition of X_f^{N+1} and we have $x \in X_f^{N+1}$. This is contradicted with $x \in X_{f,\max} \setminus X_f^{N+1}$. $\square$

Remark 3.2. Generally, in the computation of $X_{f,\max}$, it is impossible to keep computation until $j \rightarrow +\infty$. So, when the iteration goes to $j = N$, if X_f^N is equal to X_f^{N-1} in principle, X_f^N can be taken as the terminal region we want.

Remark 3.3. The terminal region computed through the method in Remark 3.2 is not $X_{f,\max}$ itself, but its enclosing set. Then, the corresponding domain of attraction may include some points which should not be in the real domain of attraction. To avoid this problem, the outspread skill from smaller region to larger one can be used to turn¡?ehlt?¿ the contraction skill from larger region to smaller one in this paper. The concrete algorithm is not presented here, just the general idea is stated; giving a known subset of $X_{f,\max}$, denoted by $X_{f,0}$ in advance and using $X_{f,0}$ to serve as the state constraint in (3.4), then a larger region $X_{f,1}$ will be computed. By the same procedure as presented in Section 3.1, a terminal region which is a subset of $X_{f,\max}$ will be gotten.

To obtain X_f^j is not an easy job. The only tool is statistical learning method. Here, the SVM is used.

3.2. Support Vector Machine

SVM (see [8, 9]) is the youngest part of statistical learning theory. It is an effective approach for pattern recognition. In SVM approach, the main aim of an SVM classifier is obtaining a function, which determines the decision boundary or hyperplane. This hyperplane optimally separates two classes of input data points.

Take the example of separating X into A and $X \setminus A$. For each $x_i \in A$, an additional variable $y_i = +1$ is introduced. Similarly, for each $x_i \in X \setminus A$, $y_i = -1$ is introduced. Define

$I^+ := \{i : y_i = +1\}$, $I^- := \{i : y_i = -1\}$. SVM is used to find a separating hyperplane $O(x) := w \cdot \phi(x_i) + b = 0$, between A and $X \setminus A$. Then, we obtain an estimated set of A, $\widehat{A} = \{x \in X \mid O(x) \geq 0\}$. $O(x)$ can be obtained by solving the following problem:

$$\min_{\alpha} \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \ker(x_i, x_j) - \sum_i \alpha_i,$$

$$\text{s.t. } \sum_i \alpha_i y_i = 0, \tag{3.9}$$

$$0 \leq \alpha_i \leq C, \quad \forall i \in I^+, \quad \alpha_i \geq 0, \quad \forall i \in I^-,$$

where ker denotes the kernel function.

In this paper, the following Gaussian kernel is used:

$$\ker(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right), \tag{3.10}$$

with σ being the positive Gaussian kernel width.

There are many software packages of SVM available on internet. They can be downloaded and used directly. By using SVM, the support vectors are extracted from $\{x_i\}$ and their relevant weights are exported. Denote P_s as the number of support vectors and X_s as the support vectors set, the optimal hyperplane is described as follows:

$$O(x) = \sum_{i=1}^{P_s} w_i \cdot \ker(x_i, x) + b, \tag{3.11}$$

where $x_i \in X_s$ is a support vector, and $w_i = \alpha_i y_i$ satifying $\sum_{i=1}^{P_s} w_i = 0$ is the relevant weight.

3.3. Estimating the Largest Terminal Region

In SVM classifier, the training data is inputted and the hyperplane will be outputted. To us, the training of data is the only job.

Take the separation of X_f^0 from X as an example. Firstly, choose arbitrary points $x_i \in X$, $i = 1, 2, \ldots, N_p$ (N_p is the number of training points) then decide the value of y_i corresponding to x_i by using the following procedure

If $F(x_i) \geq F_{X_f^0}^*(x_i)$,

$y_i = +1$; else $y_i = -1$

endif.

When all the y_i for all the x_i are gotten, they can be packed to constitute the training data. Then, by inputting the training data into SVM classifier, an optimal hyperplane $O^0(x) = 0$ and an estimated set of X_f^0, $\widehat{X}_f^0 = \{x \in X \mid O^0(x) \geq 0\}$ will be obtained.

When $\widehat{X}_f^0$ is known, the training data for separating X_f^1 from X_f^0 can be known by the similar procedure. By inputting them into SVM classifier, a hyperplane $O^1(x) = 0$ and an estimated set of X_f^1, $\widehat{X}_f^1 = \{x \in X \mid O^1(x) \geq 0\}$ will be gotten.

Repeatedly, a hyperplane series $O^j(x) = 0$, $j = 1, 2, \ldots$ will be obtained. When $j = N$ if it is satisfied that, for $x_i \in X_{\text{sup},N-1}$, $i = 1, 2, \ldots, N_{\text{sup},N-1}$, there exists

$$\sum_{i=1}^{N_{\text{sup},N-1}} \left\| O^N(x_i) - O^{N-1}(x_i) \right\| \leq \varepsilon N_{\text{sup},N-1}, \tag{3.12}$$

it is deemed that $\widehat{X}_f^N$ is equal to $\widehat{X}_f^{N-1}$ in principle and $\widehat{X}_f^N$ is taken as the final estimation of $X_{f,\max}$. Where $X_{\text{sup},N-1}$ is the support vectors set at $j = N - 1$, $N_{\text{sup},N-1}$ is the number of support vectors and ε is a tunable threshold. The smaller it is, the higher the precision of $\widehat{X}_f^N$ approximating to $X_{f,\max}$ is.

Remark 3.4. Here, we used the information that, in SVM classifier, the hyperplanes are decided just on the support vectors.

4. Choosing an Appropriate Terminal Cost

In the previous chapter, a method to maximize the terminal region was proposed, but the method has a premise: the terminal cost is given in advance. In this chapter, how to get a terminal cost will be shown. Before this, some properties of terminal cost will be analyzed.

4.1. Weighting Terminal Cost, the Domain of Attraction Will Be Enlarged

From conditions (C1) and (C2), it is known that the terminal region is based on the choice of the terminal cost. We want to know what the relationship between them is and this relationship will give us what messages when we choose terminal cost. Theorem 4.1 will give us the answer.

Denote $X_{f,\max}$, $X_{\lambda,\max}$ as the terminal regions of $P_N(x_0)$ and $P_{N,\lambda}(x_0)$ with a weighted terminal cost λF, $\lambda \geq 1$, respectively. Limon et al. [7] proved that the terminal region will be enlarged by weighting the terminal cost for MPC without terminal constraint in its on-line optimization. Here, we will show that, this property will also hold water in our case.

Theorem 4.1. *Consider F and $X_{f,\max}$ satisfying conditions (C1) and (C2). When a weighted terminal cost λF, $\lambda \geq 1$ is used, the corresponding terminal region $X_{\lambda,\max}$ is larger than $X_{f,\max}$, namely, $X_{f,\max} \subseteq X_{\lambda,\max}$.*

Proof. For any $x_0 \in X_{f,\max}$, the conditions of (C1) and (C2) are equivalent to the fact that there exists a control trajectory $\mathbf{u} = \{u_0, u_1, u_2, \ldots\}$ to make the following inequalities stand up:

$$F(x_i) \geq q(x_i, u_i) + F(x_{i+1}), \quad i = 0, 1, 2, \ldots, \tag{4.1}$$

where $x_{i+1} = f(x_i, u_i)$.

It is obvious that, when the terminal cost is λF, these inequalities also stand up using the same control trajectory

$$\lambda F(x_i) \geq q(x_i, u_i) + \lambda F(x_{i+1}). \tag{4.2}$$

So, we can see that $x_0 \in X_{\lambda,\max}$, namely, $X_{f,\max} \subseteq X_{\lambda,\max}$. $\square$

Remark 4.2. When the prediction horizon is given, the larger the terminal region is, the larger the domain of attraction is. So Theorem 4.1 shows that, giving the prediction horizon, by weighting terminal cost, the domain of attraction will be enlarged.

4.2. Weighting Terminal Cost, the Total Cost Will Increase

It is known from the last section that, when the terminal cost is weighted, the terminal region is enlarged. So, can we weight the terminal cost arbitrarily? The answer is no. This section will tell us the reason the total cost will be increased by weighting the terminal cost.

Let $J^*_{N,\lambda}(x_0)$ be the solution of $P_{N,\lambda}(x_0)$, $\mathbf{u}^*_{N,\lambda}(x_0) = \{u^*_{N,\lambda}(0,x_0),\ldots,u^*_{N,\lambda}(N,x_0)\}$, $\mathbf{x}^*_{N,\lambda}(x_0) = \{x^*_{N,\lambda}(0,x_0),\ldots,x^*_{N,\lambda}(N,x_0)\}$ be the optimal control trajectory and corresponding state trajectory, respectively, and let $\mathbf{u}_{\mathrm{RH},\lambda}(x_0) = \{u^*_N(0,x_0),u^*_N(0,x_{1,\lambda}),\ldots\}$, $\mathbf{x}_{\mathrm{RH},\lambda}(x_0) = \{x_0,x_{1,\lambda},x_{2,\lambda},\ldots\}$ be the receding horizon control trajectory and receding horizon state trajectory of using MPC with $P_{N,\lambda}(x_0)$, respectively. Define $J_{\mathrm{RH},\lambda}(x_0)$, $J_{\mathrm{RH}}(x_0)$ as the total costs of using MPC with $P_{N,\lambda}(x_0)$ and with $P_N(x_0)$, respectively.

For convenience, consider an assumption.

Assumption 1. For any $x_0 \in \Gamma_N$, where Γ_N is the domain of attraction of MPC with $P_N(x_0)$, the terminal state by solving $P_{N,\lambda}(x_0)$ belongs to $X_{f,\max}$, that is to say $x^*_{N,\lambda}(N,x_0) \in X_{f,\max}$.

Remark 4.3. Assumption 1 means, for any $x_0 \in \Gamma_N$, that the solution of $P_{N,\lambda}(x_0)$ with $X_{\lambda,\max}$ as its terminal region is equal to that with $X_{f,\max}$ as its terminal region. A few points in Γ_N may not satisfy this assumption, for convenience, their influence is neglected. Under this assumption, it is obviouse that $J^*_{N,\lambda}(x_0) \geq J^*_N(x_0)$ and the following lemma holds water.

Lemma 4.4. *For any $x_0 \in \Gamma_N$, there exists*

$$\left(J^*_{N-1,\lambda} - J^*_{N-1}\right)\left(x^*_N(1,x_0)\right) \geq \left(J^*_{N-1,\lambda} - J^*_{N-1}\right)\left(x^*_{N,\lambda}(1,x_0)\right). \tag{4.3}$$

Proof. From the view of optimality, $J^*_N(x_0)$ can be expressed as

$$J^*_N(x_0) = q(x_0,u^*_N(0,x_0)) + J^*_{N-1}(x^*_N(1,x_0)). \tag{4.4}$$

Considering Assumption 1 and by optimality, there exists

$$J^*_N(x_0) \leq q\left(x_0,u^*_{N,\lambda}(0,x_0)\right) + J^*_{N-1}\left(x^*_{N,\lambda}(1,x_0)\right). \tag{4.5}$$

Similarly, $J^*_{N,\lambda}(x_0)$ can be expressed as

$$J^*_{N,\lambda}(x_0) = q\left(x_0,u^*_{N,\lambda}(0,x_0)\right) + J^*_{N-1,\lambda}\left(x^*_{N,\lambda}(1,x_0)\right). \tag{4.6}$$

And by optimality, there exists

$$J^*_{N,\lambda}(x_0) \leq q(x_0,u^*_N(0,x_0)) + J^*_{N-1,\lambda}(x^*_N(1,x_0)). \tag{4.7}$$

Obviously, the result of subtracting $J_N^*(x_0)$ from the right hand of (4.7) is bigger than the result of subtracting the right hand of (4.5) from $J_{N,\lambda}^*(x_0)$, in other words,

$$
\begin{aligned}
&\left(q(x_0, u_N^*(0, x_0)) + J_{N-1,\lambda}^*(x_N^*(1, x_0))\right) - J_N^*(x_0) \\
&\geq J_{N,\lambda}^*(x_0) - \left(q\left(x_0, u_{N,\lambda}^*(0, x_0)\right) + J_{N-1}^*\left(x_{N,\lambda}^*(1, x_0)\right)\right).
\end{aligned}
\tag{4.8}
$$

Finally, the following result can be obtained:

$$
\left(J_{N-1,\lambda}^* - J_{N-1}^*\right)\left(x_N^*(1, x_0)\right) \geq \left(J_{N-1,\lambda}^* - J_{N-1}^*\right)\left(x_{N,\lambda}^*(1, x_0)\right).
\tag{4.9}
$$

$\square$

Define $F_{\mathcal{K}}$ as a kind of $\mathcal{K}$ functions satisfying the following: for any $F_1, F_2 \in F_{\mathcal{K}}$, and $x_1, x_2 \in X$, there exists the following: if $F_1(x_1) \geq F_1(x_2)$, the inequality $F_2(x_1) \geq F_2(x_2)$ stands up.

To continue discussion, another assumption is needed.

Assumption 2. All of the positive cost functions used in this paper like J_{N-1}^*, J_N^*, $J_{N-1,\lambda}^*$, and $J_{N,\lambda}^*$ and the results of the addition or subtraction between them like $J_{N-1,\lambda}^* - J_{N-1}^*$, $J_{N-1,\lambda}^* + J_N^*$ belong to $F_{\mathcal{K}}$.

Based on Assumption 2 and Lemma 4.4, it is known that, for any $x_0 \in \Gamma_N$ and $F_{\mathcal{K}}$, there exists

$$
F_{\mathcal{K}}\left(x_N^*(1, x_0)\right) \geq F_{\mathcal{K}}\left(x_{N,\lambda}^*(1, x_0)\right).
\tag{4.10}
$$

Then, by using (4.10) and Assumptions 1 and 2, another lemma which is a key for our study on this issue can be gotten.

Lemma 4.5. *Under Assumptions 1 and 2, for any $x_0 \in \Gamma_N$ and any positive cost function $F_{\text{tra}} \in F_{\mathcal{K}}$ satisfying $F_{\text{tra}} \leq J_{N-1}^*$, there exists*

$$
q(x_0, u_N^*(0, x_0)) + F_{\text{tra}}(x_1) \leq q\left(x_0, u_{N,\lambda}^*(0, x_0)\right) + F_{\text{tra}}(x_{1,\lambda}).
\tag{4.11}
$$

Proof. Here, x_1 means $x_N^*(1, x_0)$ and $x_{1,\lambda}$ means $x_{N,\lambda}^*(1, x_0)$. From Assumption 2, it is known that $(J_{N-1}^* - F_{\text{tra}}) \in F_{\mathcal{K}}$, so there exists

$$
\begin{aligned}
&q(x_0, u_N^*(0, x_0)) + F_{\text{tra}}(x_1) \\
&= q(x_0, u_N^*(0, x_0)) + J_{N-1}^*(x_1) - (J_{N-1}^* - F_{\text{tra}})(x_1) \\
&\leq q\left(x_0, u_{N,\lambda}^*(0, x_0)\right) + J_{N-1}^*(x_{1,\lambda}) - (J_{N-1}^* - F_{\text{tra}})(x_{1,\lambda}) \\
&= q\left(x_0, u_{N,\lambda}^*(0, x_0)\right) + F_{\text{tra}}(x_{1,\lambda}).
\end{aligned}
\tag{4.12}
$$

Here, we used the fact that $J_N^*(x_0) \leq J_{N,\lambda}^*(x_0)$ and $(J_{N-1}^* - F_{\text{tra}})(x_1) \geq (J_{N-1}^* - F_{\text{tra}})(x_{1,\lambda})$.

$\square$

Then, the reason why we cannot weight the terminal cost arbitrarily can be presented.

Theorem 4.6. *Under Assumptions 1 and 2, for any $x_0 \in \Gamma_N$, the following inequality stands up:*

$$J_{\mathrm{RH}}(x_0) \leq J_{\mathrm{RH},\lambda}(x_0). \tag{4.13}$$

Proof. It is obvious that J_N^* meets the condition in Lemma 4.5 because of $J_N^* \leq J_{N-1}^*$. Choose $F_{\mathrm{tra}} = J_N^*$ in Lemma 4.5. There exists

$$q(x_0, u_N^*(0, x_0)) + J_N^*(x_1) \leq q\left(x_0, u_{N,\lambda}^*(0, x_0)\right) + J_N^*(x_{1,\lambda}). \tag{4.14}$$

Similarly, for any $x \in \Gamma_N$, the following result can be obtained:

$$\begin{aligned}
q(x, u_N^*(0, x)) &+ J_N^*(x_N^*(1, x)) \\
&\leq q(x, u_N^*(0, x)) + J_{N-1}^*(x_N^*(1, x)) = J_N^*(x) \leq J_{N-1}^*(x).
\end{aligned} \tag{4.15}$$

So, choose

$$F_{\mathrm{tra}}(x) = q(x, u_N^*(0, x)) + J_N^*(x_N^*(1, x)). \tag{4.16}$$

By using x_1, $x_{1,\lambda}$ to replace x, respectively, there exists

$$\begin{aligned}
q(x_0, u_N^*(0, x_0)) &+ q(x_1, u_N^*(0, x_1)) + J_N^*(x_2) \\
&= q(x_0, u_N^*(0, x_0)) + F_{\mathrm{tra}}(x_1) \\
&\leq q\left(x_0, u_{N,\lambda}^*(0, x_0)\right) + F_{\mathrm{tra}}(x_{1,\lambda}) \\
&= q\left(x_0, u_{N,\lambda}^*(0, x_0)\right) + q(x_{1,\lambda}, u_N^*(0, x_{1,\lambda})) + J_N^*(x_N^*(1, x_{1,\lambda})).
\end{aligned} \tag{4.17}$$

From Assumption 1, it is known that, for $x_0 \in \Gamma_N$, there exists $x_{1,\lambda} \in \Gamma_N$. Replacing x_0 with $x_{1,\lambda}$ in (4.14), the following inequality can be gotten:

$$q(x_{1,\lambda}, u_N^*(0, x_{1,\lambda})) + J_N^*(x_N^*(1, x_{1,\lambda})) \leq q\left(x_{1,\lambda}, u_{N,\lambda}^*(0, x_{1,\lambda})\right) + J_N^*(x_{2,\lambda}). \tag{4.18}$$

So, there exists

$$\begin{aligned}
q(x_0, u_N^*(0, x_0)) &+ q(x_1, u_N^*(0, x_1)) + J_N^*(x_2) \\
&\leq q\left(x_0, u_{N,\lambda}^*(0, x_0)\right) + q\left(x_{1,\lambda}, u_{N,\lambda}^*(0, x_{1,\lambda})\right) + J_N^*(x_{2,\lambda}) \\
&\leq q\left(x_0, u_{N,\lambda}^*(0, x_0)\right) + q\left(x_{1,\lambda}, u_{N,\lambda}^*(0, x_{1,\lambda})\right) + J_{N,\lambda}^*((x_{2,\lambda})).
\end{aligned} \tag{4.19}$$

Repeating this procedure, there exists

$$
\begin{aligned}
& q(x_0, u_N^*(0, x_0)) + \cdots + q(x_j, u_N^*(0, x_j)) + J_N^*(x_{j+1}) \\
& \quad \leq q\left(x_0, u_{N,\lambda}^*(0, x_0)\right) + \cdots + q\left(x_{j,\lambda}, u_{N,\lambda}^*(0, x_{j,\lambda})\right) + J_{N,\lambda}^*((x_{j+1,\lambda})).
\end{aligned}
\tag{4.20}
$$

Let $j \to +\infty$, the final result can be obtained as follows:

$$
J_{RH}(x_0) \leq J_{RH,\lambda}(x_0).
\tag{4.21}
$$

□

Theorem 4.6 shows that, weighting the terminal cost, the total cost will be increased. So, when choosing a terminal cost, people should not only take into account the need of enlarging the terminal region.

4.3. Getting an Appropriate Terminal Cost

It was pointed out from Theorems 4.1 and 4.6 that the terminal cost is a double-edged sword. On its choice, two factors, the terminal region and the total cost, must be considered. With different emphasis, different terminal cost should be chosen.

Here, a simple method to get a terminal cost is presented, whose basic idea is getting an initial terminal cost in advance then adjusting it according to our need.

As mentioned, a good terminal cost should approximate to J_∞^* as close as possible. People can only achieve it in a small neighborhood around the origin by using SOC method, see [2] for continuous-time system.

Consider the linearization of the system (2.1) at the origin

$$
x_{k+1} = Ax_k + Bu_k,
\tag{4.22}
$$

with $A = (\partial f / \partial x)(0, 0)$ and $B = (\partial f / \partial u)(0, 0)$.

Here, assume that (4.22) is stabilizable, then a terminal cost which serves as an initial candidate can be found through the following procedure.

Step 1. Solving the Riccati equation to get a preparatory G_0,

$$
G_0 = A^T G_0 A - \left(A^T G_0 B\right)\left(B^T G_0 B + R\right)^{-1}\left(B^T G_0 A\right) + Q.
\tag{4.23}
$$

Step 2. Getting a locally stabilizing linear state feedback gain K,

$$
K = -\left(B^T G_0 B + R\right)^{-1}\left(B^T G_0 A\right).
\tag{4.24}
$$

Step 3. Computing G_K by solving the following Riccati equation:

$$(\alpha A_K)^T G_K (\alpha A_K) - G_K = -Q_K, \tag{4.25}$$

where $A_K = A + BK$, $Q_K = Q + K^T RK$, and $\alpha \in [1, +\infty)$ is an adjustable parameter satisfying $\alpha |\lambda_{\max}(A_K)| < 1$.

Then, $F(x) = x^T G_K x$ can serve as an initial terminal cost. According to our need and the properties of terminal cost, the initial one can be adjusted to get an appropriate one, $F(x) = x^T(\lambda G_K)x$, $\lambda > 0$. For example, if a larger terminal region is wanted and the total cost is not cared, λ can be set to be a larger number; otherwise, if a lower total cost is demanded and the domain of attraction already covers the operating region of system, a small one can be used.

5. Simulation Experiment

The model is an approximate discrete-time realization from a continuous-time system used in [2] as follows:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T \\ T & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} T\mu \\ T\mu \end{bmatrix} u_k + \begin{bmatrix} T(1-\mu) & 0 \\ 0 & -4T(1-\mu) \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} u_k, \tag{5.1}$$

where $\mu = 0.5$, $T = 0.1$ s, and the state constraint and control constraint are $X = \{x \mid \|x\|_1 \leq 4\}$, $U = \{u \mid |u| \leq 2\}$, respectively.

The stage cost is chosen as $q(x, u) = x^T Q x + u^T R u$ with $Q = 0.5I$ and $R = 1$. By using SOC method, the locally linear feedback gain is adopted as $K = -[2.0107\ 2.0107]$ and $|\lambda_{\max}(A_K)| = 0.9000$ is obtained. Then, choose $\alpha = 1.11$ and get the terminal cost as $F(x) = x^T G x$ with $G = [1107.356\ 857.231; 857.231\ 1107.356]$.

To estimate each X_f^j, 4000 training points are generated. Set $\varepsilon = 1$, when $j = 15$, there exists

$$\sum_{i=1}^{N_{\sup,14}} \left\| O^{15}(x_i) - O^{14}(x_i) \right\| \leq \varepsilon N_{\sup,14}, \tag{5.2}$$

where $x_i \in X_{\sup,14}$, $X_{\sup,14}$ is the support vectors set at $j = 14$, and $N_{\sup,14}$ is the number of support vectors. Then, it is deemed that $\widehat{X}_f^{15}$ is equal to $\widehat{X}_f^{14}$ in principle and $\widehat{X}_f^{15}$ can be taken as the final estimation of $X_{f,\max}$. Figure 2 shows the approximation process of $X_{f,\max}$. The blue line is the hyperplane at $j = 1$, the black dot line is that at $j = 15$, and the red lines between them are those at $j = 2, 3, \ldots, 14$. Let the prediction horizon be $N = 3$. Figure 3 shows the closed-loop trajectories of some points chosen from the domain of attraction arbitrarily.

When the terminal cost is enlarged to $F(x) = x^T(10G)x$, a new terminal region larger than the old one can be obtained. Figure 4 shows it. The red line is the new hyperplane and the black dot line is the old one.

For convenience, let (A) denote the MPC using $x^T G x$ as its terminal cost and (B) the MPC using $x^T(10G)x$. For some points chosen from Γ_3 of (A) arbitrarily, Figure 5 shows their
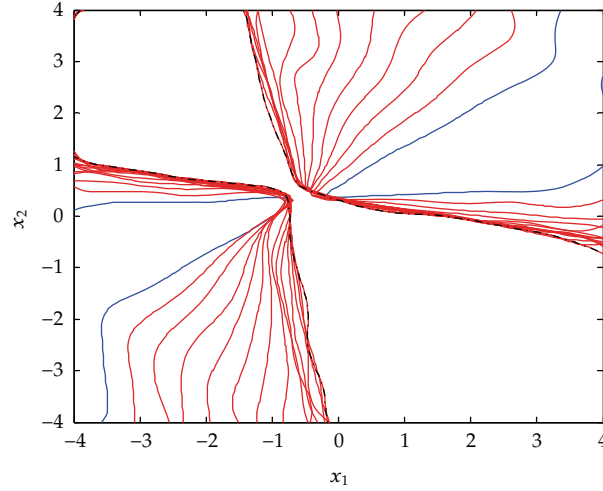
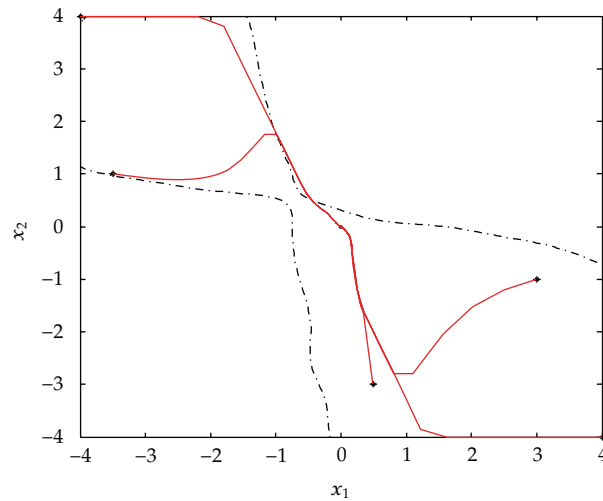**Figure 2:** The approximation process of terminal region.

Figure 3: The closed-loop trajectories of points in Γ_3.

Table 1: Comparison of the total costs.

Initial points	$\begin{bmatrix} 2 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -2 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 2 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 1 \end{bmatrix}$
(A)	34.7915	18.2460	16.4528	7.9936
(B)	37.0120	20.6714	23.2854	23.8467

closed-loop trajectories of using (A) and (B), respectively, where red lines denote the results of using (A), and blue dash-dotted lines denote the results of using (B). Table 1 shows the comparison of the total costs. Obviously, for the same point, the total cost of using (A) is smaller than that of using (B).
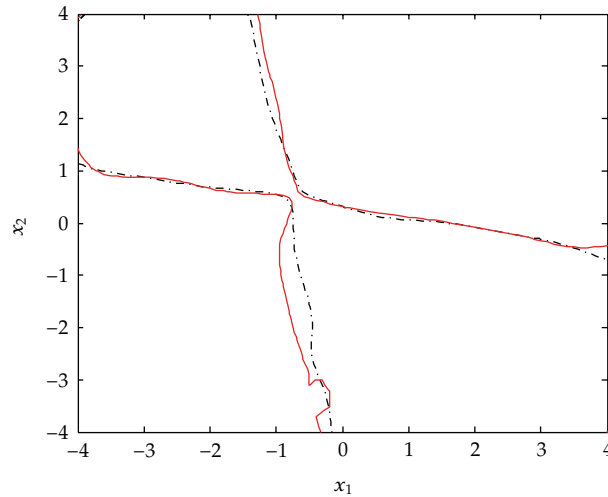
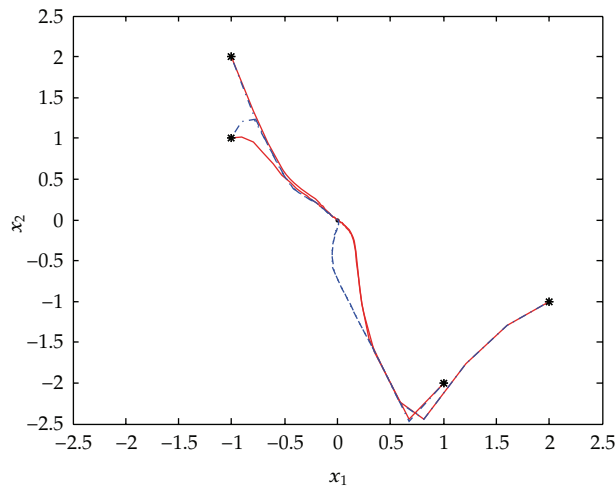**Figure 4:** Comparison of the terminal regions.

Figure 5: Comparison of the closed-loop trajectories.

6. Conclusion

This paper discussed the relationships between terminal cost and terminal region and between terminal cost and total cost, respectively. It showed that, by enlarging the terminal cost, terminal region will be enlarged, but the total cost will be increased too. A simple method to get a suitable terminal cost was proposed, it can be adjusted according to our need. For example, to get a larger terminal region, it can be weighted; to reduce the total cost, it can be unweighted. When a terminal cost was given, a novel method was proposed to receive a maximal terminal region by using SVM. With the same prediction horizon, its corresponding domain of attraction is the largest one.

References

[1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[2] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.

[3] V. T. Minh and N. Afzulpurkar, "A comparative study on computational schemes for nonlinear model predictive control," *Asian Journal of Control*, vol. 8, no. 4, pp. 324–331, 2006.

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

[5] J. A. De Doná, M. M. Seron, D. Q. Mayne, and G. C. Goodwin, "Enlarged terminal sets guaranteeing stability of receding horizon control," *Systems & Control Letters*, vol. 47, no. 1, pp. 57–63, 2002.

[6] C. J. Ong, D. Sui, and E. G. Gilbert, "Enlarging the terminal region of nonlinear model predictive control using the support vector machine method," *Automatica*, vol. 42, no. 6, pp. 1011–1016, 2006.

[7] D. Limon, T. Alamo, F. Salas, and E. F. Camacho, "On the stability of constrained MPC without terminal constraint," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 832–836, 2006.

[8] C. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[9] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.