Automatica 183 (2026) 112574

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

A characterization method of terminal ingredients for nonlinear MPC using value-based reinforcement learning[☆]

Jinghan Cui[a,b], Jinwu Gao[a,b], Xiangjie Liu[c], Yuqi Liu[d], Shuyou Yu[a,b,*]

[a] College of Communications Engineering, Jilin University, China
[b] National Key Laboratory of Automotive Chassis Integration and Bionics, Jilin University, China
[c] The State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, North China Electric Power University, China
[d] Shenyang Institute of Automation, Chinese Academy of Sciences, China

ARTICLE INFO

Article history:
Received 28 March 2024
Received in revised form 8 April 2025
Accepted 9 July 2025

Keywords:
Model predictive control
Reinforcement learning
Terminal conditions
Stability

ABSTRACT

The stability of nonlinear model predictive control (MPC) relies significantly on stabilizing factors such as the terminal region and cost. A larger terminal region not only expands the region of attraction for the closed-loop system but also contributes to reducing online computation costs. However, existing methods in the literature often impose limitations on the degrees of freedom available for characterizing terminal ingredients. This limitation arises from the reliance on either a predetermined linear local controller or a preset control Lyapunov function. This paper introduces an innovative approach to terminal ingredient characterization leveraging value-based reinforcement learning (RL). This method provides ample degrees of freedom for expanding the terminal region. To achieve this, a deep neural network is employed to learn the parametric state value function, serving as the terminal cost for MPC. The local controller adopts a one-step MPC instead of a predetermined linear or nonlinear feedback controller. Subsequently, a terminal set sequence is constructed iteratively through the one-step set expansion. The proposed approach's effectiveness is validated through simulations.

1. Introduction

Model predictive control (MPC) has garnered considerable success as an advanced optimal control technique, enabling the systematic handling of multi-variable systems and operating constraints. This success has led to its widespread adoption in various industrial applications (Qin & Badgwell, 2003). Generally, MPC ensures stability via approximating the infinite horizon optimal control which aims to optimize an infinite horizon objective function while considering system dynamics. The MPC transfers the infinite horizon optimization problem to a finite horizon optimization problem with additional stability conditions.

Significant progress has been made in establishing the stability of nonlinear model predictive control (NMPC) (Mayne, Rawlings, Rao, & Scokaert, 2000). In the field of NMPC, a pivotal stability result is the quasi-infinite horizon approach with the terminal region and terminal cost serving as crucial stabilizing elements (Chen & Allgöwer, 1998). The objective is to guide the system trajectory into a terminal region around the origin within a finite number of steps. Subsequently, a stable local controller steers the system trajectory towards the origin. The terminal cost constrains the infinite horizon cost of the system, starting from the terminal region. In Chen and Allgöwer (1998), an explicit approach for characterizing terminal region and terminal cost is provided which are computed based on the linearization of the nonlinear dynamics around an equilibrium point. A linear state feedback controller is then used to compute a quadratic terminal cost, with the terminal region as a sub-level set of this cost. An outstanding concern for stable NMPC is how to expand the terminal region, as the size of the terminal region directly impacts both the region of attraction and the online computation costs.

Several research findings have proposed methodologies for enhancing the terminal region by increasing degrees of freedom in the characterization of terminal ingredients. More tuning parameters can be provided to shape the terminal region by integrating tuning matrices (Rajhans, Patwardhan, & Pillai, 2017). Besides, a support vector machine (SVM) is employed to characterize the terminal region, resulting in a significant enlargement (Ong, Sui, & Gilbert, 2006). However, these approaches assumed a stable linear state feedback controller for nonlinear dynamics, limiting the terminal region size in nonlinear cases. Highlighting the significance of employing a nonlinear controller for terminal region characterization, it emphasized that the terminal region's size can

[☆] The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Martin Monnigmann under the direction of Editor Florian Dorfler.

[*] Corresponding author at: College of Communications Engineering, Jilin University, China.

E-mail addresses: jinghan@jlu.edu.cn (J. Cui), gaojw@jlu.edu.cn (J. Gao), liuxj@ncepu.edu.cn (X. Liu), liuyuqi@sia.cn (Y. Liu), shuyou@jlu.edu.cn (S. Yu).

https://doi.org/10.1016/j.automatica.2025.112574
0005-1098/

be notably increased in Lucia, Rumschinski, Krener, and Findeisen (2015). The early method in Yu, Chen, Böhm, and Allgöwer (2009) utilized a polytopic linear differential inclusions description to capture nonlinear dynamics, yielding a parameter-dependent local control law that provides more freedom in choosing terminal ingredients compared to time-invariant linear state feedback control laws. Recently, Lazar and Tetteroo (2018) proposed a terminal region characterization approach using both linear and nonlinear controllers for terminal region characterization. They suggested using a second-order Taylor series approximation of system dynamics to enlarge the terminal region.

In the previous literature, the stable local control is predefined, and the corresponding terminal region is calculated. Thus, there is no guarantee that the consequent terminal region is the maximal one in different control laws. Yafeng Wang et al. introduced a novel approach that progressively approximates the maximal terminal state region without a predefined feedback controller (Wang, Sun, Liu, & Yang, 2012). The shortcoming is that a predefined terminal cost is required, also imposing limitations on terminal region enlargement. Therefore, this paper aims to enlarge the terminal region without a predetermined feedback controller or terminal cost.

Reinforcement learning (RL) is designed to iteratively derive the optimal control policy for infinite-horizon objective functions through the Bellman equation. Currently, a prevalent approach for integrating RL with MPC involves modifying the MPC terminal cost to enhance closed-loop performance and approximate infinite-horizon optimality. Moreno et al. utilized a value-based learning method to obtain the terminal cost (Moreno-Mora, Beckenbach, & Streif, 2023). Similarly, Min Lin et al. also employed the learned value function as the terminal cost in MPC, treating MPC as a policy generator whose performance is evaluated through RL techniques (Lin, Sun, Xia, & Zhang, 2023). Both studies ensure closed-loop stability by selecting a sufficiently long prediction horizon to steer the terminal state into a predefined terminal region rather than enforcing explicit terminal constraints. However, this would require a larger prediction horizon to guarantee the asymptotic stability of the closed loop, which will result in a greater online computing burden.

To the best of the authors' knowledge, no existing literature explicitly employs RL techniques to directly construct the terminal ingredients for enlarging the terminal region without reliance on predetermined elements. In this study, RL is integrated into an MPC framework to iteratively reshape the terminal cost within a constructed terminal set sequence, thus eliminating the need for any predetermined feedback controller or terminal cost. The key contributions of this work are summarized as:

(1) A promising approach to characterizing terminal ingredients for the MPC optimization problem is introduced using a value-based RL technique. The deep neural network (DNN) is harnessed to learn the parametric state value function, serving as the terminal cost for stable MPC. The local controller adopts a one-step MPC instead of a predefined linear or nonlinear feedback controller. Subsequently, a terminal set sequence is iteratively constructed based on this terminal cost through the one-step set expansion.

(2) In this RL learning process, this work introduces a novel method for learning and exploration. Information from both model knowledge and MPC computations is leveraged to calculate target values, enhancing the overall learning efficiency. Besides, the actor used for exploration is explicitly chosen as a one-step MPC controller instead of a randomly selected policy, which may avoid the need for an exhaustive action space search.

(3) Comprehensive convergence and stability analyses are provided theoretically, and the proposed approach demonstrates its effectiveness through simulation studies.

2. Preliminaries

2.1. System description

In this work, we consider the following discrete-time, time-invariant nonlinear system

$$x_{k+1} = f(x_k, u_k), \forall k \in \mathbb{N}_0, \tag{1}$$

with state $x \in \mathbb{X} \subseteq \mathbb{R}^{n_x}$ and control $u \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$ for some $n_x, n_u \in \mathbb{N}$, whose dynamic $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ is assumed to be Lipschitz continuous in both arguments, with $f(0, 0) = 0$. $\mathbb{U}$ is a compact set, $\mathbb{X}$ is a closed and bounded set, satisfying $0 \in \mathbb{U}$ and $0 \in \mathbb{X}$.

2.2. Model predictive control

The finite horizon optimal control problem of MPC to be solved online for system (1) at each time $k \in \mathbb{N}_0$ may read as

$$V_N^*(x_k) := \min_{u_i} \sum_{i=0}^{N-1} \gamma^i l(x_i, u_i) + \gamma^N V_f(x_N) \tag{2a}$$

$$\text{s.t. } x_0 = x_k \tag{2b}$$

$$x_{i+1} = f(x_i, u_i), i \in \mathbb{I}_0^{N-1} \tag{2c}$$

$$u_i \in \mathbb{U}, i \in \mathbb{I}_0^{N-1} \tag{2d}$$

$$x_i \in \mathbb{X}, i \in \mathbb{I}_1^N \tag{2e}$$

$$x_N \in X_f \tag{2f}$$

$$i = 0, \ldots, N-1 \tag{2g}$$

where N is the prediction horizon, and $0 < \gamma < 1$ is the discount factor. The stage cost function $l(x, u)$ is positive-definite, like the set-point tracking function. The set X_f is the terminal region and V_f is the terminal cost. Compared to the infinite-horizon optimal control, the solution space of the MPC optimization problem is changed to be finite dimensional vector space and can be solved by using a numerical optimization solver such as the interior point method. Define the optimal solution of MPC optimization problem (2) as $\boldsymbol{u}_N^*(x_k) = \{u_0^*(x_k), u_1^*(x_k), \ldots, u_{N-1}^*(x_k)\}$, the corresponding state trajectory as $\boldsymbol{x}_N^*(x_k) = \{x_0, x_1^*(x_k), \ldots, x_{N-1}^*(x_k), x_N^*(x_k)\}$. Let $V_N^*(x_k)$ be the minimum of (2). According to the receding horizon mechanism, the control law of MPC can be denoted as

$$\boldsymbol{u}_{RH}(x_k) = u_0^*(x_k), k = 0, 1, 2, \ldots. \tag{3}$$

The stability of this controller will be taken into consideration since optimality does not imply stability. The terminal cost $V_f(\cdot)$ and terminal region X_f should be pre-designed which are associated with the local feedback law $u = \kappa_f(x)$. The terminal ingredients should consider the following assumptions (Mayne et al., 2000).

Assumption 2.1. (1) Let the set X_f satisfy $0 \in X_f \subseteq \mathbb{X}$. (2) For all $x \in X_f$, $\kappa_f(x) \in \mathbb{U}$. (3) The set X_f is positively invariant under $\kappa_f(x)$, i.e., for all $x \in X_f$, there is $f(x, \kappa_f(x)) \in X_f$. (4) The value function $V_f(\cdot)$ with $V_f(0) = 0$ is continuous and positive definite, and there has $\gamma V_f(f(x, \kappa_f(x))) - V_f(x) \leq -l(x, \kappa_f(x)), \forall x \in X_f$.

The closed-loop stability of the controlled system is shown in the following Lemma.

Lemma 2.1. *For any $x_0 \in \mathbb{X}$, if (2) has a feasible solution under Assumption 2.1, it is guaranteed that x_0 will be steered to the origin by using the control law of MPC.*

In satisfying these conditions in Assumption 2.1, there is much freedom in the choice of $\kappa_f(\cdot)$, X_f and $V_f(\cdot)$. It is common to choose $\kappa(x) = kx$ with a fixed k and $V_f = x^T Px$ to be a Lyapunov function related to the linear model of (1) at the origin. X_f is chosen to be the level set $X_P = \{x : x^T Px \leq \alpha\}$, and α is chosen small so that V_f remains a Lyapunov function in region X_P.

Remark 2.1. In the proposed MPC framework (2), the discount factor γ is introduced to ensure that the value iteration update operator, defined subsequently in Section 3, satisfies the contraction mapping condition (Szepesvári, 2010). This contraction property is essential to guarantee the convergence of the approximate value iteration process toward the optimal value function. It is worth noting that standard MPC formulations without a value iteration step typically do not require a discount factor.

2.3. Reinforcement learning

RL is the direct adaptive optimal control aiming to approximate the optimal value function or optimal policy based on the following Bellman equation

$$V^*(x_k) = \min_{\kappa(\cdot)}\{l(x_k, \kappa(x_k)) + \gamma V^*(x_{k+1})\}, \tag{4}$$

$$u_k^* = \kappa^*(x_k) = \arg\min_{\kappa(\cdot)}\{l(x_k, \kappa(x_k)) + \gamma V^*(x_{k+1})\}, \tag{5}$$

for all $x_k \in \mathbb{X}$. The optimal value function $V^*(x)$ is a Lyapunov function for the system (1) with u_k^* as the corresponding optimal policy. The value-based RL is essentially value iteration which means learning the optimal value function by recursively applying Eq. (4), and the value function is typically approximated by a DNN.

2.4. Problem formulation

For stable MPC as described in (2), it is essential to carefully design the terminal controller $\kappa_f(\cdot)$, terminal region X_f, and terminal cost $V_f(\cdot)$, as a larger terminal region directly leads to an expanded region of attraction for the closed-loop system. Consequently, developing methods that provide sufficient degrees of freedom for constructing the terminal cost and terminal region within MPC becomes particularly important. Moreover, Assumption 2.1 must be satisfied by the terminal ingredients to guarantee closed-loop stability. Value-based reinforcement learning inherently yields an approximation of the optimal value function, facilitating effective characterization of the terminal region. Utilizing parameterized value functions further enhances the degrees of freedom available for designing these terminal ingredients.

In this work, a novel approach for designing the terminal cost and terminal region in the MPC optimization problem (2) is proposed by leveraging value-based RL. Firstly, the characterization of the terminal region is established, wherein a sequence of terminal subsets is iteratively constructed through one-step set expansions. The resulting set sequence provides an approximation of the maximal terminal region. Secondly, the optimal value function $V_f(x)$, which serves as the terminal cost, is learned within each subset using value-based RL. An update operator is designed to iteratively steer an arbitrarily initialized value function to the optimal one. Subsequently, a learning framework employing a double-network setting is presented. Finally, a rigorous stability analysis of the proposed approach is provided.

3. Proposed approach

3.1. Characterization of terminal region

To guarantee stability, the design of the terminal region X_f and terminal cost $V_f(x)$ should satisfy Assumption 2.1. Moreover, to enlarge the terminal region, the one-step MPC is chosen as the local controller, for $\forall x \in X_f$. The local control law is the solution of the following optimization problem,

$$Q_f^*(x) = \min_{u\in\mathbb{U}} Q_f(x, u) = l(x, u) + \gamma V_f(f(x, u)) \tag{6a}$$

$$\text{s.t. } f(x, u) \in X_f \tag{6b}$$

Define the optimal local action as $\pi(x) = u^*$ where u^* is the solution of the optimization problem (6). Then, the terminal region X_f can be expressed as

$$X_f := \{x \in \mathbb{X} | V_f(x) \geq Q_f^*(x)\} \tag{7}$$

It can be seen from the expression of X_f that the terminal region is essentially a positively invariant set using the optimal control resulting from the optimization problem (6) with terminal cost $V_f(\cdot)$. However, from this expression, it cannot be determined whether a state point belongs to X_f since the X_f acts as the constraint in the optimization problem (6). To solve this problem, the method of using a one-step set expansion iteratively is adopted. The iterative process begins with an initial positively invariant set estimation X_f^0. With this initial estimation X_f^0, the corresponding V_f^0 is the optimal value function that can be obtained using the value-based RL, and it will be elaborated later on.

Based on this X_f^0 and corresponding V_f^0, a more accurate estimation of X_f^1 is achieved expressed as

$$X_f^1 := \{x \in \mathbb{X} | V_f^0(x) \geq Q_f^{0*}(x)\} \tag{8}$$

where $Q_f^{0*}(x)$ is the minimum of

$$Q_f^{0*}(x) := \min_{u\in\mathbb{U}} Q_f^0(x, u) = l(x, u) + \gamma V_f^0(f(x, u)) \tag{9a}$$

$$\text{s.t. } f(x, u) \in X_f^0 \tag{9b}$$

As shown, the construction of X_f^1 means there exists $u \in \mathbb{U}$ satisfying $f(x, u) \in X_f^0$ for any $x \in X_f^1$. Besides, the minimum of optimization (9) satisfies $Q_f^{0*}(x) \leq V_f^0(x)$. With this terminal region X_f^1, the corresponding optimal value function V_f^1 can also be obtained using the value-based RL technique. Repeatedly, a more accurate estimation X_f^2 can be constructed using the information of X_f^1 and V_f^1. Written in iterative form, the terminal region $X_f^j, j = 2, 3, \ldots, \infty$ can be constructed as

$$X_f^j := \{x \in \mathbb{X} | V_f^{j-1}(x) \geq Q_f^{j-1*}\} \tag{10}$$

where Q_f^{j-1*} is the minimum of

$$\begin{aligned} Q_f^{j-1*}(x) :&= \min_{u\in\mathbb{U}} Q_f^{j-1}(x, u) \\ &= l(x, u) + \gamma V_f^{j-1}(f(x, u)) \end{aligned} \tag{11a}$$

$$\text{s.t. } f(x, u) \in X_f^{j-1} \tag{11b}$$

As shown in Fig. 1, by employing this one-step set expansion for constructing X_f^j, a subsets sequence of the terminal region, denoted by $\{X_f^j, j = 0, 1, 2, \ldots, \infty\}$, can be achieved. The corresponding optimal value function V_f^j can also be acquired within each subset.

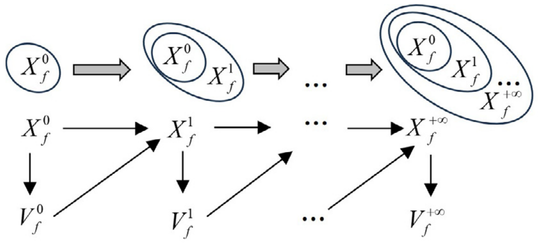


Fig. 1. A simplified schematic for the construction of terminal ingredients.

3.2. Characterization of terminal cost

During the iterations of the terminal subsets sequence, the corresponding optimal value function $V_f^j(x)$ is gained by value-based RL. Value iteration is to learn the optimal value function by recursively applying the Bellman optimality equation. In a value iteration step, an update operator should be constructed to steer an arbitrarily initialized value function to the optimal one. The iteration j will be omitted in the subsequent content for the sake of brevity. In this learning framework, for $\forall x_k \in X_f$, the one-step MPC is integrated into the construction of the update operator used for value iteration. The operator for all $x_k \in X_f$ based on this one-step MPC is defined

$$\begin{aligned} V_{f,i+1}(x_k) &= T^v V_{f,i}(x_k) \\ &= l(x_k, \pi(x_k)) + \gamma Q_{f,i}(x_{k+1}, \pi(x_{k+1})) \end{aligned} \tag{12}$$

where T^v is the update operator and i is the update step. The term $Q_{f,i}(x_{k+1}, \pi(x_{k+1}))$ is obtained from the optimization (6) with value function $V_{f,i}$ at x_{k+1}.

To demonstrate the convergence of this update operator (12), i.e., it can steer an initial value function to the optimal value function, the operator (12) should be explained as a contraction mapping first, and the optimal value function is the unique fixed point (Szepesvári, 2010).

Assumption 3.1. For all $x_k \in X_f$, the initialized value function $V_{f,0}$ satisfies

$$V_{f,0}(x_k) \geq \min_{u_k \in \mathbb{U}, x_{k+1} \in X_f} \{l(x_k, u_k) + \gamma V_{f,0}(x_{k+1})\} \tag{13}$$

Lemma 3.1. *Suppose Assumption 3.1 holds, the operator T^v defined in (12) is a contraction mapping, and the optimal value function is the unique fixed point of this operator. Then, the proposed update defined in (12) can steer the value function $V_{f,i}(x)$ to the optimal one $V_f(x)$ for all $x_k \in X_f$ with the following inequality satisfied.*

$$V_f(x_k) \geq \min_{u_k \in \mathbb{U}, x_{k+1} \in X_f} \{l(x_k, u_k) + \gamma V_f(x_{k+1})\} \tag{14}$$

Proof. First, define two other operators T and T' for all $x_k \in \mathbb{X}_f$,

$$TV_{f,i}(x_k) = \min_{u_k \in \mathbb{U}, x_{k+1} \in X_f} \left\{ l(x_k, u_k) + \gamma V_{f,i}(x_{k+1}) \right\} \tag{15}$$

$$\begin{aligned} T'V_{f,i}(x_k) = &\min_{\substack{u_k, u_{k+1} \in \mathbb{U}, \\ x_{k+1}, x_{k+2} \in X_f}} \{l(x_k, u_k) \\ &+ \gamma l(x_{k+1}, u_{k+1}) + \gamma^2 V_{f,i}(x_{k+2})\} \end{aligned} \tag{16}$$

When Assumption 3.1 holds for $V_{f,i}(x_k)$, the following equations hold for $x_k \in X_f$,

$$\begin{aligned} TV_{f,i}(x_k) &= \min_{u_k \in \mathbb{U}} \{l(x_k, u_k) + \gamma \min_{x_{k+1} \in X_f} V_{f,i}(x_{k+1})\} \\ &\geq \min_{u_k \in \mathbb{U}} \{l(x_k, u_k) + \\ &\quad \min_{\substack{u_{k+1} \in \mathbb{U}, \\ x_{k+1}, x_{k+2} \in X_f}} \{\gamma l(x_{k+1}, u_{k+1}) + \gamma^2 V_{f,i}(x_{k+2})\}\} \\ &= \min_{\substack{u_k \in \mathbb{U}, \\ x_{k+1} \in X_f}} \{l(x_k, u_k) + \gamma TV_{f,i}(x_{k+1})\}. \end{aligned} \tag{17}$$

Similarly, for operator T', we have

$$T'V_{f,i}(x_k) \geq \min_{\substack{u_k \in \mathbb{U}, \\ x_{k+1} \in X_f}} \{l(x_k, u_k) + \gamma T'V_{f,i}(x_{k+1})\}. \tag{18}$$

Since $\pi_i(x_k) := \arg\min_{u_k} Q_{f,i}(x_k, u_k)$, expand $TV_{f,i}(x_k)$,

$$\begin{aligned} TV_{f,i}(x_k) &= l(x_k, \pi_i(x_k)) + \gamma \min_{x_{k+1} \in X_f} V_{f,i}(x_{k+1}) \\ &\geq l(x_k, \pi_i(x_k)) \\ &+ \min_{\substack{u_{k+1} \in \mathbb{U}, x_{k+1}, \\ x_{k+2} \in X_f}} \{\gamma l(x_{k+1}, u_{k+1}) + \gamma^2 V_{f,i}(x_{k+2})\} \\ &= T^v V_{f,i}(x_k) \end{aligned} \tag{19}$$

for $x_k \in X_f$. Expanding $T^v V_{f,i}(x_k)$, we have

$$\begin{aligned} &T^v V_{f,i}(x_k) = l(x_k, \pi_i(x_k)) + \\ &\min_{\substack{u_{k+1} \in \mathbb{U}, \\ x_{k+1}, x_{k+2} \in X_f}} \{\gamma l(x_{k+1}, u_{k+1}) + \gamma^2 V_{f,i}(x_{k+2})\} \\ &\geq \min_{\substack{u_k, u_{k+1} \in \mathbb{U}, \\ x_{k+1}, x_{k+2} \in X_f}} \{l(x_k, u_k) + \gamma l(x_{k+1}, u_{k+1}) + \gamma^2 V_{f,i}(x_{k+2})\} \\ &= T'V_{f,i}(x_k) \end{aligned} \tag{20}$$

for $x_k \in X_f$. Based on (19) and (20),

$$TV_{f,i}(x_k) \geq T^v V_{f,i}(x_k) \geq T'V_{f,i}(x_k) \tag{21}$$

From Szepesvári (2010), we know that operators T and T' are two contraction mappings, and the fixed point is optimal value function $V_f(x)$. Then, the $V_f(x)$ is also a fixed point for operator T^v based on (21). To prove uniqueness, we need to prove that for all fixed points $V_f^v(x)$ of mapping T^v, they are the same as $V_f(x)$. For every $V_f^v(x)$ satisfying $T^v V_f^v(x_k) = V_f^v(x_k)$, we have

$$\begin{aligned} &T^v V_f^v(x_k) = l(x_k, u_k) + \gamma T^v V_f^v(x_{k+1}) \\ &\geq \min_{u_k \in \mathbb{U}, x_{k+1} \in X_f} \{l(x_k, u_k) + \gamma T^v V_f^v(x_{k+1})\}. \end{aligned} \tag{22}$$

for all $x_k \in X_f$. Then, we get $V_f^v(x_k) \geq TV_f^v(x_k)$, combined with (21),

$$TV_f^v = V_f^v \tag{23}$$

So every fixed point of T^v, i.e., $V_f^v(x)$, is also the unique fixed point of operator T, i.e., $V_f(x)$. In summary, the $V_f(x)$ is the unique fixed point of operator T^v. □

Based on the operator T^v defined in (12), the optimal value function $V_f(x)$ can be achieved. However, this operation is performed for all states in the region X_f. In other words, an infinite-dimensional vector will be operated using T^v to reach the optimal value function $V_f(x)$. Value-based RL is an executable route that parameterizes the value function by DNN, trained to learn the optimal value function $V_f(x)$. Deep-Q-network (DQN), as an effective value-based RL method, tends to learn the action-value function with discrete action settings by enumerating a finite number of all possible candidate actions. To prevent the selection of the

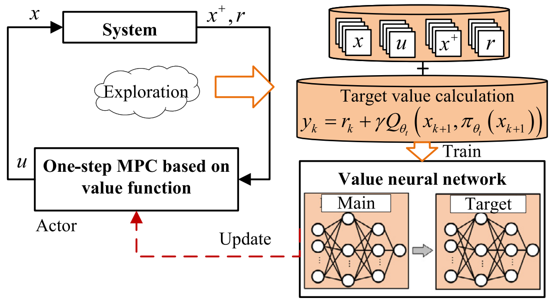

x
System
x^+, r
Exploration
u
One-step MPC based on value function
Actor
Update
x
u
x^+
r
Target value calculation
$y_k = r_k + \gamma Q_{\theta_t}\left(x_{k+1}, \pi_{\theta_t}\left(x_{k+1}\right)\right)$
Train
Value neural network
Main
Target

Fig. 2. The learning framework of GDDVN.

overestimated value, DQN uses two DNNs to learn the action-value functions, called target network with parameter θ_t and main network with parameter θ_m. The target network is utilized to evaluate the target values, while the main network is used to generate the actions for exploration.

To handle continuous state space and obtain the expression of the optimal value function, we propose a generalized algorithm of value-based RL by using one-step MPC as an actor. Besides, the operator T^v is employed for the evaluation of the target value which integrates the model and the minimum value of one-step MPC into the RL framework.

3.3. Generalized value-based RL for obtaining the optimal value function

The parameterized terminal cost $V_{f_\theta}(\cdot)$ is employed to learn $V_f(\cdot)$ via DNNs, and a generalized value-based RL will be introduced in this section. The learning procedure is similar to that of DQN, the difference is that the action used for exploration is from the actor obtained by the optimization problem (6) rather than an arbitrary one. Besides, another difference is that the calculation of the target value used for updating the parameter θ is based on the minimum of the optimization problem (6). Double network settings are applied to this generalized value-based RL, named generalized double deep-V-network (GDDVN). Double network for value functions are denoted as main network $V_{f_{\theta_m}}$ and target network $V_{f_{\theta_t}}$. For arbitrary $x \in X_f$, one-step MPC is regarded as an actor for exploration to generate data, and the optimization problem read as,

$$\min_{u_k \in \mathbb{U}} Q_{f_{\theta_m}}(x_k, u_k) = l(x_k, u_k) + \gamma V_{f_{\theta_m}}(f(x_k, u_k)) \tag{24a}$$

$$\text{s.t. } f(x_k, u_k) \in X_f \tag{24b}$$

Similar to DQN, the actor of GDDVN is represented as an optimization problem. The difference is that one-step MPC is constructed with the model predicting the state transition. The use of the model helps to stably compute the input and dramatically reduces the required amount of data to improve the control policy. Besides, the state constraints can be explicitly imposed to obtain the current action. The learning framework of GDDVN is shown in Fig. 2.

Another difference is performing the optimization of the target network to obtain the target value whereas the DQN obtains the optimal solution by enumeration. For collecting data, $\forall x \in X_f$, after $\pi_{\theta_m}(x)$, i.e., the solution of the optimization (24), is taken, we get the reward $r_k = l(x, \pi_{\theta_m}(x))$ and the next state $x^+ = f(x, \pi_{\theta_m}(x))$. The GDDVN evaluates the target value y_k based on the T^v operator, shown as

$$y_k = r_k + \gamma Q_{\theta_t}(x_{k+1}, \pi_{\theta_t}(x_{k+1})) \tag{25}$$

where $Q_{\theta_t}(x_{k+1}, \pi_{\theta_t}(x_{k+1}))$ is the minimum of optimization problem (24) with terminal cost $V_{f_{\theta_t}}$. Let B be the batch data set. Then, the update of the main network is performed by applying the one-step gradient descent step with the appropriate step size α,

$$\theta_m^+ = \theta_m + \alpha \nabla_{\theta_m} \frac{1}{|B|} \sum_B [V_{f_{\theta_m}}(x_k) - y_k]^2 \tag{26}$$

The parameter of the target network is updated by

$$\theta_t^+ = (1 - \rho)\theta_t + \rho\theta_m \tag{27}$$

where $\rho \in [0, 1]$ is the update coefficient.

The algorithm framework of this proposed approach is summarized in Algorithm 1.

Algorithm 1 Terminal ingredients designed based on GDDVN

```
Input: Initialization:$X_f^0 \subseteq \mathbb{X}$ is a positively invariant set
    containing the origin, $j \leftarrow 0$
repeat
    Initialization:$V_{f_{\theta_m}}^j$, $V_{f_{\theta_t}}^j$
    for Each episode $n$ do
        Generate the initial state in region $X_f^j$
        for Each time step $k$ do
            Compute the action $u_k = \pi_{\theta_m}(x_k)$ by solving (24)
            Observe the next state $x_{k+1}$ and reward $r_k = l(x_k, u_k)$
            Add the data tuple $(x_k, u_k, r_k, x_{k+1})$ into the replay buffer
        end for
    end for
    if $n$ reaches the training period then
        Select the batch data set $B$ from the replay buffer
        Calculate the target value by equation (25) for data tuple in $B$
        Update the main network by (26)
    end if
    Return the trained $V_{f_{\theta_m}}^j$
    if $n$ reaches the updating period then
        Update the target network by (27)
    end if
    Return the trained $V_{f_{\theta_t}}^j$
    Define $X_f^{j+1}$ using trained $V_f^j$ by equation (10)
    Obtain the explicit expression of the region $X_f^{j+1}$ using SVM
    $j \leftarrow j + 1$
until $X_f^j \rightarrow X_{f,max}$
Output: $X_f$, $V_f$
```

4. Stability analysis

The terminal ingredients are obtained via two iterative processes. After choosing the one-step MPC as the local controller, the terminal region X_f is constructed by a subsets sequence defined in (10), while the terminal cost and corresponding local controller can be learned via value-based RL using the contraction operator (12). Here, we want to show the effectiveness of these proposed terminal ingredients with an enlarged terminal region.

Lemma 4.1. *If Assumption 3.1 is satisfied, and X_f^0 is a positively invariant set, there is X_f^1 such that $X_f^0 \subset X_f^1$, and X_f^1 is a positively invariant set.*

Proof. If Assumption 3.1 is satisfied, it follows that $V_f^0(x) \geq Q_f^{0*}(x)$. From the construction of X_f^1, we have $x \in X_f^1$, namely, $X_f^0 \subset X_f^1$. □

In our approach, the one-step MPC is chosen, and its corresponding maximal terminal region $X_{f,max}$ can be defined as follows.

Definition 4.1. For the terminal region X_f, the following two conditions are satisfied: (1) For any $x \in X_f$, there exists the inequality satisfied.

$$V_f(x) \geq \min_{u \in \mathbb{U}} \{l(x,u) + \gamma V_f(f(x,u))\}. \tag{28}$$

where V_f is the terminal cost. (2) The set $X_f \subseteq \mathbb{X}$ is a control invariant set under one-step MPC with terminal cost $V_f(x)$. Then, the largest terminal region that satisfies the above two conditions is defined as the maximal terminal region $X_{f,max}$.

Assumption 4.1. There exists a set $X_f^{+\infty}$ to which the subsets sequence $\{X_f^j, j = 1, 2, \dots, +\infty\}$, constructed according to (10) and (11), will converge as $j \to +\infty$.

Theorem 1. *If Assumptions 3.1 and 4.1 are satisfied, then for X_f^j with initialized $X_f^0 \subset \mathbb{X}$ being a positively invariant set and containing the origin,*

(1) the subsets sequence $\{X_f^j, j = 1, 2, \dots, +\infty\}$ will converge to $X_{f,max}$ when j goes to infinity.

(2) there exists optimal terminal cost V_f^j obtained by operator (12). Then the equilibrium $x = 0$ is asymptotically stable with X_f^j and V_f^j serving as the terminal ingredients for MPC (2).

Proof. (1) Similar to Lemma 4.1, any subset in the sequence is positively invariant and any two neighboring subsets satisfy $X_f^{j-1} \subset X_f^j$ in terms of the construction of X_f^j and Lemma 3.1. Subsequently, we will prove by contradiction, based on Assumption 4.1, that the region $X_f^{+\infty}$ is equivalent to the largest terminal region $X_{f,max}$.

When $X_{f,max} \subset X_f^{+\infty}$, for $\forall x \in X_f^{+\infty}$, there exists corresponding terminal cost $V_f^{+\infty}$ satisfying $V_f^{+\infty}(x) \geq \min_{u \in \mathbb{U}, f(x,u) \in X_f^{+\infty}} \{l(x,u) + \gamma V_f^{+\infty}(f(x,u))\}$ according to Lemma 3.1. This is contradicted with that $X_{f,max}$ is the largest terminal region under Definition 4.1.

When $X_f^{+\infty} \subset X_{f,max}$, for $\forall x \in X_{f,max} \backslash X_f^{+\infty}$, there exists no such a $u \in \mathbb{U}$ satisfying $V_f^{+\infty}(x) \geq \min_{u \in \mathbb{U}, f(x,u) \in X_f^{+\infty}} \{l(x,u) + \gamma V_f^{+\infty}(f(x,u))\}$. However, from Definition 4.1, for $\forall x \in X_{f,max}$, there exists $f(x,u) \in X_{f,max} \backslash X_f^{+\infty}$ and the inequality satisfied (28). Besides, it is obvious that $0 \notin X_{f,max} \backslash X_f^{+\infty}$ which is contradicted with that $V_f(x)$ is regarded as the local Lyapunov function.

(2) From the construction of X_f^j, Assumption 2.1 is satisfied with X_f^j and V_f^j serving as the terminal ingredients for MPC. Then, the equilibrium $x = 0$ is asymptotically stable with this MPC strategy. □

Remark 4.1. If the number of iteration steps is finite, the resulting terminal set may only approximate a large positively invariant subset within $X_{f,max}$. Although this approximation does not compromise the stability of the controlled system, it does result in a smaller region of attraction compared to the one associated with $X_{f,max}$. Consequently, selecting the number of terminal set expansion iterations inherently involves a trade-off between offline computational complexity and the size of the achievable region of attraction.

Remark 4.2. The terminal region can be constructed through the implicit expression (10). However, it cannot serve as the terminal constraint in the optimization problem (11) directly. One approach to the characterization of this terminal region explicitly is needed. In this work, SVM learning is exploited to obtain the explicit expression of each terminal region, and a conservative approximation method is leveraged from Ong et al. (2006). It takes the form of a scalar function, $O^j(x)$ such that $\tilde{X}_f^j := \{x \in \mathbb{X} | O^j(x) \geq 0\}$ closely approximates X_f^j. For every $x \in X_f^j$ $(x \in \mathbb{X} \backslash X_f^j)$, an additional label variable $y_s = +1 (y_s = -1)$ is applied. This SVM is allowed to classify safe points (the points belong to X_f^j) as unsafe (the points do not belong to X_f^j) but not the other way. Finally, the formulation of SVM finds a separating hyperplane, expressed as $O^j(x) = w^j \phi(x) + b^j = 0$, between X_f^j and $X \backslash X_f^j$.

Remark 4.3. The proposed approach inherently exhibits favorable scalability properties to higher-dimensional systems, primarily because the exploration process leverages a structured one-step MPC approach rather than random exploration, thus effectively mitigating computational complexity. Moreover, although SVM-based terminal region approximations face challenges due to dimensionality, the conservative nature of our approximations ensures the safety and stability of the resulting terminal regions.

5. Simulation results and discussion

Consider a system described by the following ODEs (Chen & Allgöwer, 1998):

$$\dot{x}_1 = x_2 + u(\mu + (1-\mu)x_1) \tag{29a}$$

$$\dot{x}_2 = x_1 + u(\mu - 4(1-\mu)x_2) \tag{29b}$$

where u and x have to satisfy the following constraints:

$$\mathbb{U} = \{u \in \mathbb{R}^1 | -2 \leq u \leq 2\} \tag{30}$$

$$\mathbb{X} = \{x \in \mathbb{R}^2 | [-4 \ \ -4]^T \leq x \leq [4 \ 4]^T\} \tag{31}$$

Assume $\mu = 0.5$ for the moment. To guarantee the stability of MPC, terminal cost, terminal region, and local controller need to be designed. In Chen and Allgöwer (1998), the quadratic terminal function and the corresponding terminal region Ω_α can be obtained based on a linear locally stabilizing state feedback controller. The optimal control problem of MPC is solved in discrete time with a sampling time of $\delta = 0.1$ and a prediction horizon of $T_p = 1.5$.

Here, we are about to redesign the terminal ingredients using RL. In this case, we employ the terminal region Ω_α from Chen and Allgöwer (1998) as an initial terminal region in this proposed approach. Value-network is incorporated into the MPC framework to shape the terminal cost which is regarded as the optimal cost. The learning rate for RL is $\eta = 0.0002$, and the discount factor is $\gamma = 0.9$. The conventional dense neural networks of the value-network have three layers where the number of nodes for each layer are 8, 4, and 1, respectively. We selected the number of nodes by trial and error, considering the trade-off between approximation ability and learning speed. The learning of the main network is performed every five time steps, and the update of the target network is performed after five times of updating the main network. The update coefficient is set as $\rho = 0.9$. Since the optimization problem of MPC is solved with a gradient-based numerical optimization solver, it is preferable to use a smooth neural network. In addition, a non-negative property is also preferred for the terminal cost. Therefore, we propose to use the following function as an activation function of the value-network (Oh, Park, Kim, & Lee, 2022),

$$A(x) = log(1 + x^T x), \tag{32}$$

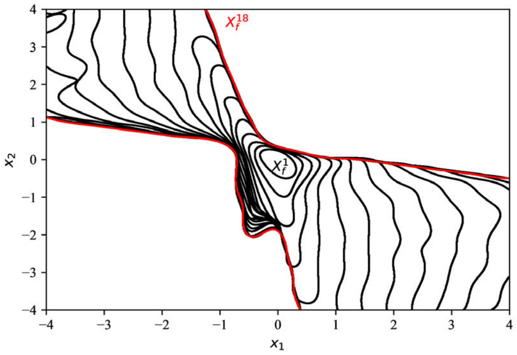


Fig. 3. The terminal region sequence.

Table 1
Comparison of the shortest possible prediction horizon (N) and the closed-loop performance (J) of the proposed method and paper (Chen & Allgöwer, 1998). None indicates that there is no feasible solution.

x_0	Paper (Chen & Allgöwer, 1998).			This paper.		
$(-4, 2)$	N = **14**	N = 15	N = 16	N = 1	N = 2	N = **5**
	None	93.13	85.48	98.62	80.66	**65.48**
$(-4, 4)$	N = **17**	N = 18	N = 20	N = 1	N = 2	N = **5**
	None	72.83	72.31	84.7	81.38	**71.8**
$(0.15, -4)$	N = **6**	N = 7	N = 8	N = 2	N = 3	N = **5**
	None	34.54	34.5	33.62	31.56	**31.52**
$(3.7, -0.5)$	N = **11**	N = 12	N = 13	N = 1	N = 2	N = **5**
	None	56.9	51.06	52.12	51.6	**47.9**

Based on this trained terminal cost, we can construct the terminal region through implicit expression (10). To execute the one-step set expansion iteratively, the characterization of this terminal region explicitly is achieved through SVM learning with the Gaussian kernel as the kernel function (Ong et al., 2006). To estimate each X_f^j, 5625 training points are generated in state admissible region $\mathbb{X}$. As observed from the simulation results, when j is iterated to 18, the terminal regions of adjacent iteration steps are nearly the same. The iterative process can be well illustrated in Fig. 3, and it stems from the initial set Ω_α. It can be seen that the terminal region designed via the proposed strategy in this work has been enlarged more than that in paper (Chen & Allgöwer, 1998; Ong et al., 2006).

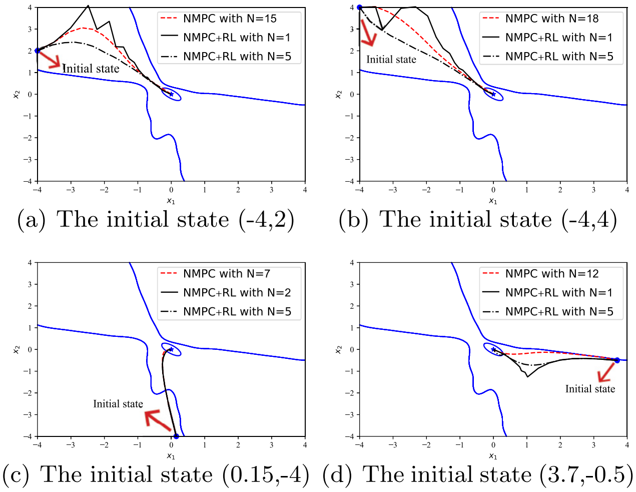
Fig. 4 and Table 1 depict the closed-loop state trajectory of some initial points, which are selected to implement the comparison between the proposed approach and the approach in Chen and Allgöwer (1998). The closed-loop performance can be evaluated with the following index

$$J = \sum_{k=1}^{Nsim} l(x_k, u_k), \tag{33}$$

where $Nsim$ is the simulation step. The smaller this value is, the better the closed-loop performance is. The closed-loop performance with these two approaches under different prediction horizons is revealed in Table 1, which shows that our proposed method has improved performance compared to Chen and Allgöwer (1998) in two important aspects:

(1) The proposed method can find a feasible solution with a smaller prediction horizon for the same initial state. As shown in Table 1, taking the initial point (−4,2) as an example, the traditional NMPC in Chen and Allgöwer (1998) has a feasible solution only when N is over 14, while the proposed NMPC has a feasible solution when $N = 1$. This improves the feasibility of solving optimization problems for NMPC and reduces the burden of online computing due to the small prediction horizon.

(2) According to the closed-loop performance index (33), our approach performs better as shown in Table 1. Taking the initial point (−4,2) as an example, the control performance is already significantly better when $N = 5$. The closed-loop state trajectory in Fig. 4 can also fully support the above conclusion.



(a) The initial state (-4,2) (b) The initial state (-4,4)

(c) The initial state (0.15,-4) (d) The initial state (3.7,-0.5)

Fig. 4. Comparison of the terminal regions and closed-loop trajectories. Terminal regions: non-ellipse for this proposed approach, ellipse for paper (Chen & Allgöwer, 1998).

6. Conclusions

This paper demonstrates the application of value-based RL in the design of terminal ingredients for MPC. The proposed construction provides significant degrees of freedom for expanding the terminal region, eliminating the need for predetermining a feedback controller or a terminal cost. As a result, the region of attraction under this proposed controller is substantially enlarged, and the online computational burden is reduced due to a shorter prediction horizon. Simulation results demonstrate the effectiveness and computational feasibility of this approach, showing improvement in the size of the terminal region and stability of the closed-loop system. Nevertheless, it remains critical to resolve numerical implementation issues to effectively extend the proposed approach to practical higher-dimensional systems. Future work will focus on enhancing data efficiency in value function training by adopting adaptive discretization strategies and reducing the complexity of explicit terminal region representations using compact alternatives such as maximal inscribed ellipsoid approximations, especially important for higher-dimensional systems.

Acknowledgments

Partially supported by the Natural Science Foundation of Jilin Province 20230101065JC, the National Natural Science Foundation of China under Grants 62573210, 62473171, 62473167, U24A20277 and 92467301.

References

Chen, Hong, & Allgöwer, Frank (1998). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, *34*(10), 1205–1217.

Lazar, Mircea, & Tetteroo, Martin (2018). Computation of terminal costs and sets for discrete-time nonlinear MPC. *IFAC-PapersOnLine*, *51*(20), 141–146.

Lin, Min, Sun, Zhongqi, Xia, Yuanqing, & Zhang, Jinhui (2023). Reinforcement learning-based model predictive control for discrete-time systems. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(3), 3312–3324.

Lucia, Sergio, Rumschinski, Philipp, Krener, Arthur J, & Findeisen, Rolf (2015). Improved design of nonlinear model predictive controllers. *IFAC-PapersOnLine*, *48*(23), 254–259.

Mayne, David Q, Rawlings, James B, Rao, Christopher V, & Scokaert, Pierre OM (2000). Constrained model predictive control: Stability and optimality. *Automatica*, *36*(6), 789–814.

Moreno-Mora, Francisco, Beckenbach, Lukas, & Streif, Stefan (2023). Predictive control with learning-based terminal costs using approximate value iteration. *IFAC-PapersOnLine*, *56*(2), 3874–3879.

Oh, Tae Hoon, Park, Hyun Min, Kim, Jong Woo, & Lee, Jong Min (2022). Integration of reinforcement learning and model predictive control to optimize semi-batch bioreactor. *AIChE Journal*, *68*(6), Article e17658.

Ong, Chong Jin, Sui, Dan, & Gilbert, Elmer G. (2006). Enlarging the terminal region of nonlinear model predictive control using the support vector machine method. *Automatica*, *42*(6), 1011–1016.

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

Rajhans, Chinmay, Patwardhan, Sachin C., & Pillai, Harish (2017). Discrete time formulation of quasi infinite horizon nonlinear model predictive control scheme with guaranteed stability. *IFAC-PapersOnLine*, *50*(1), 7181–7186.

Szepesvári, Csaba (2010). Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *4*(1), 1–103.

Wang, Yafeng, Sun, Fuchun, Liu, Huaping, & Yang, Dongfang (2012). Maximal terminal region approach for MPC using subsets sequence. *Frontiers of Electrical and Electronic Engineering*, *7*, 270–278.

Yu, Shuyou, Chen, Hong, Böhm, Christoph, & Allgöwer, Frank (2009). Enlarging the terminal region of NMPC with parameter-dependent terminal control law. *Nonlinear Model Predictive Control: Towards New Challenging Applications*, 69–78.

Jinghan Cui received the Ph.D. degree in control theory and control engineering from North China Electric Power University, Beijing, China, in 2019. From September 2017 to September 2018, she was a Visiting Scholar with the University of Alberta, Edmonton, AB, Canada. She was a Postdoctoral Researcher with the State Key Laboratory of Synthetical Automation for Process Industries, Northeast University, Shenyang, China, from 2019 to 2021. She is currently an Associate Professor with Jilin University, Changchun, China. Her research interests include optimization, economic model predictive control, and learning model predictive control of renewable energy generation.

Jinwu Gao received the B.Eng. degree from the Department of Automation Measurement and Control Engineering, and the Ph.D. degree from the Department of Control Science and Engineering, Harbin Institute of Technology, Harbin, China, in 2005 and 2012, respectively. From 2012 to 2014, he was an Assistant Professor with Sun Y at-sen University, Guangzhou, China. In July 2014, he held a Postdoctoral position with the Department of Engineering and Applied Science, Sophia University, Tokyo, Japan. From 2016 to 2020, he was an Associate Professor with Jilin University, Changchun, China, where he has been a Professor since September 2020. His research interests include control theory and application in automotive powertrain.

Xiangjie Liu received the Ph.D. degree in Automatic Control from the Research Center of Automation, Northeastern University, Shenyang, China, in 1997. He is now a Professor in North China Electric Power University, Beijing, China. His current research areas include model predictive control with its application in industrial processes.

Yuqi Liu received his B.Sc. and M.Sc. degrees from North China Electric Power University, Beijing, China, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree at the University of Chinese Academy of Sciences (UCAS), with a research focus on 3D scene perception, model predictive control, networked control systems, and reinforcement learning. He also serves as a Full Assistant Researcher at the Shenyang Institute of Automation, Chinese Academy of Sciences.

Shuyou Yu received the Ph.D. degree in engineering cybernetics from the University of Stuttgart, Germany, in 2011. He is currently a Professor with the Department of Control Science and Engineering, Jilin University. His current research interests include model predictive control, robust control, and its applications in mechatronic systems.