Networked Dual-Mode Adaptive Horizon MPC for Constrained Nonlinear Systems

Pengfei Li, Yu Kang, *Senior Member, IEEE*, Yun-Bo Zhao, *Senior Member, IEEE*, and Tao Wang

***Abstract*—This article investigates the predictive control scheme and related stability issue for a class of discrete-time perturbed nonlinear system with state and input constraints. First, we propose a novel control framework, i.e., networked dual-mode adaptive horizon model predictive control (MPC), which consists of a local controller, a remote controller that is subject to packet losses, and a judger coordinating the switchings between them. The optimization procedure of MPC with variable prediction horizon is implemented in the remote controller while a simple state-feedback control law is in the local one. Second, to establish the stability condition, we propose a new Lyapunov function. By specifying the relation between the Lyapunov function and the optimal MPC value function, the input-to-state practical stability is established. Finally, simulation results show the effectiveness of our proposed control scheme.**

***Index Terms*—Adaptive horizon model predictive control (MPC), bounded disturbances, dual-mode MPC, packet losses.**

I. Introduction

NETWORKED control systems (NCSs) have attracted considerable attention in the past decades [1]. Although the usage of the networks to convey the sensing and control data brings many advantages, it also gives rise to some new challenges due to the unreliable transmission, e.g., delays and packet losses. These imperfections deteriorate the control performance or even cause instability if without further treatment. Therefore, significant effort ranging from control schemes [2], scheduling strategies [3], sampling mechanisms [4], etc., has been devoted to overcome these problems. The related works can be found in [5]–[7] and the references therein.

To date, the results for NCSs mainly focus on the linear systems, but for the constrained nonlinear NCSs, fairly limited results have been reported. The difficulties mainly stem from the coupling of nonlinearity and network effects as well as the state and input constraints. A powerful tool in the study of the constrained nonlinear system is model predictive control (MPC), which optimizes the control cost while explicitly taking the constraints into consideration, and hence has potential applications in many research fields (see [8]–[11]). However, the MPC is required to solve a finite horizon optimal control problem (FHOCP) online within a specified time interval, which is a computation-intensive process for the problems with large prediction horizon and high dimensionality [12]. This requirement is often a limitation as the demand of computing resources may not be affordable by a simple controller or the large computing delay will corrupt the control performance. Therefore, when adopting the MPC-based scheme to study the nonlinear NCSs, a more practical the MPC algorithm is desirable and is able to further counteract the adverse network effects.

Fortunately, the problems of network effects and computing resource can be effectively handled by networked MPC. On the one hand, the networked configuration lessens the computing resource demand from the local controller by offloading the optimization procedure onto a remote controller which usually has strong computing power, e.g., the cloud computing system or the server. On the other hand, the adverse network effects can be actively compensated by the packet-based transmission, which is also a strength brought by networks [13]. Therefore, the networked MPC has received much attention. Associated works are reviewed as follows. In [14], the MPC strategy is designed for unconstrained nonlinear NCSs. The sensor-to-controller (S-C) packet loss and controller-to-actuator (C-A) packet loss are both considered and stability is guaranteed. For the constrained nonlinear NCSs, Quevedo and Nešić [15] proved the input-to-state stability (ISS) based on the packet-based scheme when considering the C-A packet loss. This result has been extended to the two-channel packet losses case in [16] and the input-to-state practical stability (ISpS) is established therein. Besides, the results for the random packet losses [17], [18] and the network delay [19], [20] have also been reported.

Throughout the above work, the proposed methods focus on the compensation for delays or packet losses. However, there exist two issues need to be further addressed: 1) the computing time may be large as the scale of the optimization

Manuscript received November 4, 2019; accepted January 17, 2020. Date of publication February 24, 2020; date of current version November 18, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61725304, Grant 61673361, and Grant 61673350, in part by the Scientific Research Staring Foundation for the Returned Overseas Chinese Scholars and Ministry of Education of China, in part by the Youth Innovation Promotion Association, Chinese Academy of Sciences, and in part by the Youth Yangtze River Scholar. This article was recommended by Associate Editor M. Chen. *(Corresponding author: Yu Kang.)*

Pengfei Li, Yu Kang, and Tao Wang are with the Department of Automation, University of Science and Technology of China, Hefei 230026, China (e-mail: puffylee@mail.ustc.edu.cn; kangduyu@ustc.edu.cn; wanghtao@mail.ustc.edu.cn).

Yun-Bo Zhao is with the College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: ybzhao@ieee.org).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2020.2971241.

Digital Object Identifier 10.1109/TSMC.2020.2971241

problem remains unchanged and 2) the frequent occupation of the unreliable networks will cause the degradation of the control performance and, meanwhile, waste the communication resource.

For the first challenge, the large computing load partly stems from the long prediction horizon. Therefore, adaptive horizon MPC (AHMPC) is proposed to reduce the computing load by adaptively adjusting the prediction horizon. As too long prediction horizon results in the large computing load while too short causes infeasibility, a vital question arises: how to determine a desirable prediction horizon online. A straightforward method is to treat the horizon as a decision variable [21], leading to a mixed-integer programming that may be intractable, especially, for nonlinear systems. One may notice that the closer the state is to the terminal set, the fewer steps are needed to steer the state into the terminal set by the admissible control sequence. With this observation, a shrinking horizon MPC is adopted in [22], where the prediction horizon is determined by a novel robust constraint tightening scheme in MPC formulation. The shrinking horizon has strict requirement on the disturbance, which may be too conservative when considering the packet losses. Hence, the more general time-varying horizon should be considered where an expanding horizon is also allowable (see the cyclic horizon [23], the heuristic horizon [24], and the sensitivity-based horizon [25]). Notice that both the methods in [23] and [24] are applied for the disturbance-free system, and cannot be directly extended to the perturbed case. The sensitivity-based horizon method requires the extra sensitivity calculations of the associated nonlinear programming which may be not simple in practice. Moreover, the network effects are not considered, which could influence the estimation of the prediction horizon and complicate the stability analysis.

With the packet-based control scheme, a new control signal will be provided for the actuator when C-A packet loss occurs and the stability is thereby guaranteed. But this control signal is in general not optimal, leading to the degradation of the control performance. To deal with this problem, an intuitive approach is to reduce the use of the unreliable network by employing a simple local controller (e.g., a microchip) which is hard wired to the sensor and actuator (without packet losses) and able to perform some moderate tasks. Consequently, the control tasks of the NCSs with dual-controller structure will be completed by the coordination between the two controllers. Taking the flight mission of an unmanned aerial vehicle (UAV) for example [26], the remote controller is the ground-control center that performs complex task, e.g., mission planning or path planning, while the local controller implemented in an integrated chip on the UAV executes some moderate task, e.g., cruising and collision avoidance. The remaining problems include the design of the MPC algorithm with two controllers and the design of the coordination mechanism such that the remote controller will be used only if necessary.

With the above inspirations, we propose a networked dual-mode AHMPC scheme, which consists of an AHMPC-based controller implemented at the remote side, a local controller with simple computing tasks and a judger coordinating them (shown in Fig. 1). The stability of the resultant NCSs is also analyzed. The main contributions are summarized as follows.

1) We develop a novel networked MPC framework which has high computational efficiency and good control performance. Compared with the conventional networked MPC in [15]–[17], the main advantages are threefold: a) an efficient AHMPC scheme with simple prediction horizon estimation algorithm is implemented; b) the deployed judger reduces the number of transmissions and thus reduces the energy consumption, especially, for the battery-powered sensor; and c) better control performance is obtained as part of the control signals generated by the local controller do not subject to packet losses.
2) The stability conditions are established by generalizing the method proposed in [15]. Due to the two-channel packet losses and the time-varying horizon, the optimal value function cannot serve as the Lyapunov function. We construct a new Lyapunov function dependent on the prediction horizon, actual state, and the estimated state-related optimal control sequence, then based on which the stability is obtained.

The remainder of this article is organized as follows. Section II provides the problem formulation. The control framework is elaborated in Section III. The stability analysis is performed in Section IV. Section V shows the effectiveness of our scheme by a numerical example. Section VI concludes this article.

Notations: Throughout this article, $\mathbb{R}(\mathbb{R}_{\geq 0})$ and $\mathbb{Z}_0$ represent the sets of reals (non-negative reals) and non-negative integers, respectively. $\mathbb{R}^n$ represents the n-dimensional Euclidean space. 0_n and I_n stand for the $n \times n$-dimensional zero matrix and identity matrix, respectively. For a vector x, x^{T} is the transpose of x and $\|x\|$ means the Euclidean norm of x. $\emptyset$ is the empty set. Given two sets $A, B \in \mathbb{R}^n$, the Pontryagin difference set is defined as $A \ominus B \triangleq \{x : x+y \in A \ \forall y \in B\}$. Let $\mathcal{B}^n(r)$ is a closed ball in $\mathcal{R}^n$ centered in origin of radius r, i.e., $\mathcal{B}^n(r) \triangleq \{x \in \mathcal{R}^n : \|x\| \leq r\}$. $\alpha_1 \circ \alpha_2$ is the composition of two functions α_1 and α_2. A continuous function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to be of *class*-$\mathcal{K}$ function, if it is strictly increasing, and $\alpha(0) = 0$. It is said to be of *class* $\mathcal{K}_\infty$, if it is of class $\mathcal{K}$ and $\alpha(s) \to \infty$ as $s \to \infty$. A continuous function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is of *class*-$\mathcal{KL}$ function if $\beta(s, k)$ is of class $\mathcal{K}$ for fixed k, and is strictly decreasing in k with $\beta(s, k) \to 0$ as $k \to \infty$ (see [27] for details).

II. Problem Statement

Consider the NCS with control structure illustrated in Fig. 1, which consists of a plant (including sensor and actuator), local and remote controllers, communication networks, and a judger.

The plant is described by the discrete-time nonlinear system

$$x(k+1) = f(x(k), u(k), w(k)), \ \ k \in \mathbb{Z}_0 \tag{1}$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^r$, and $w(k) \in \mathbb{R}^m$ are the plant state, control input, and uncertain external disturbance, respectively. Assume that $x(0) = x_0$ and $f(0, 0, 0) = 0$. The plant states

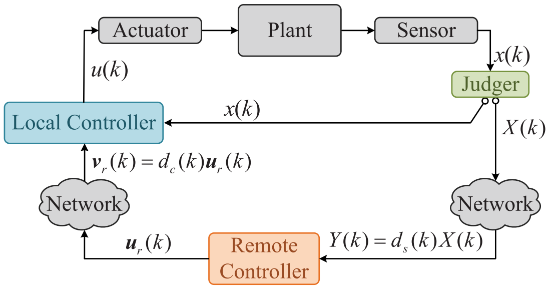


Fig. 1. Overview of NCSs with dual-controller structure.

and inputs are constrained as

$$x(k) \in \mathbb{X}, \quad u(k) \in \mathbb{U} \quad \forall k \in \mathbb{Z}_0$$

where $\mathbb{X}$ and $\mathbb{U}$ are compact set and contain the origin as an interior point. Also, $w(k) \in \mathbb{W}$ with $\mathbb{W}$ being a compact set and $0 \in \mathbb{W}$. It can be observed that $\|w(k)\| \leq \|\mathbb{W}\| < \infty\ \forall k \in \mathbb{Z}_0$, where $\|\mathbb{W}\| \triangleq \max_{w\in\mathbb{W}}(\|w\|)$. We also impose a continuity assumption for system (1) to guarantee that (1) has a unique solution.

Assumption 1: There exist class $\mathcal{K}$ functions φ_x and φ_w such that the following inequality:

$$\|f(x,u,w) - f(y,u,0)\| \leq \varphi_x(\|x-y\|) + \varphi_w(\|w\|) \tag{2}$$

holds for any $x, y \in \mathbb{X}$, $u \in \mathbb{U}$, and $w \in \mathbb{W}$.

The definitions about the invariant set and stability will be used in the following analysis.

Definition 1 (Control Invariant Set, [28]): A set $X \subseteq \mathbb{R}^n$ is control invariant for the system $x_{k+1} = f(x_k, u_k)$, if for all $x_k \in X$, there exists a $u_k \in \mathbb{U}$ such that $f(x_k, u_k) \in X$.

Definition 2 (RPI, [28]): A set $\Omega \subseteq \mathbb{R}^n$ is robust positive invariant (RPI) for the system $x_{k+1} = f(x_k, w_k)$ and the constraint set $(\mathbb{X}, \mathbb{W})$, if $\Omega \subseteq \mathbb{X}$ and $f(x_k, w_k) \in \Omega$ for all $x_k \in \Omega$ and $w_k \in \mathbb{W}$.

Definition 3 (ISpS, [29]): A nonlinear system $x(k+1) = f(x(k), w(k))$ is input-to-state practical stable (ISpS) if there exist functions $\gamma \in \mathcal{K}$, $\beta \in \mathcal{KL}$ and a constant $c \geq 0$ such that

$$\|x(k)\| \leq \beta(\|x_0\|, k) + \gamma(\|\mathbb{W}\|) + c \quad \forall k \in \mathbb{Z}_0. \tag{3}$$

A. Network Model

The data exchange of the remote controller with the judger and the local controller is through communication networks. We here consider the data-like networks, the data packets of which commonly contain fairly large data field. For example, IEEE 802.11 provides a data field of 2296 bytes at most [30], and Ethernet IEEE 802.3 frame provides a data field ranging between 46 and 1500 bytes in length [31]. The large data field can meet the requirements of the state packet and control packet for most practical control systems.

The network is prone to packet losses because of the noise and interference, and the control performance is corrupted as a result. We denote the packet loss processes of measurement network (from sensor to controller) and control network (from controller to actuator) by $d_s(k)$ and $d_c(k)$, $k \in \mathbb{Z}_0$, respectively. That is

$$d_s(k) = \begin{cases} 1, & \text{if state packet is successfully delivered} \\ 0, & \text{if packet loss occurs} \end{cases} \tag{4}$$

$$d_c(k) = \begin{cases} 1, & \text{if control packet is successfully delivered} \\ 0, & \text{if packet loss occurs.} \end{cases} \tag{5}$$

The following assumption imposes the upper bounds of the consecutive packet losses, which reflects the reliability of the networks and can be obtained by experiments.

Assumption 2: The packet losses of two networks occur randomly, and the maximum lengths of the consecutive packet losses are bounded by N_s and N_c, respectively. In addition, we define a useful constant $\bar{N}_d \triangleq N_c + N_s$.

B. Fixed-Horizon MPC

In general, the conventional MPC algorithm in many existing literatures is fixed-horizon. At each time step, we need to solve the following FHOCP with initial state x and prediction horizon N, obtain the optimal predictive control sequence $\boldsymbol{v}^* = \{v_0^*, v_1^*, \ldots, v_{N-1}^*\}$, and apply the first element of $\boldsymbol{v}^*$ to the actuator. Such procedures are then repeated at the next sampling time. The FHOCP is defined as

$$\begin{aligned} V_N^0(x) := \min_{\boldsymbol{v}} & \sum_{i=0}^{N-1} l(z_i, v_i) + F(z_N) \\ \text{s.t. } & z_{i+1} = f(z_i, v_i, 0) \\ & z_0 = x \\ & z_i \in \mathbb{X} \\ & z_N \in \mathbb{X}_f \\ & v_i \in \mathbb{U}, \ i = 0, \ldots, N-1 \end{aligned} \tag{6}$$

where $\mathbb{X}_f$ is the terminal constraint set, $l(z, v)$ and $F(z)$ are the stage cost and terminal cost, respectively.

The following assumption on the terminal set $\mathbb{X}_f$ is important in establishing the stability condition and also provides a way to design the local controller.

Assumption 3: There exists an auxiliary control law $\kappa(x) : \mathbb{X}_f \to \mathbb{U}$ such that, for all $x \in \mathbb{X}_f$

$$F(f(x, \kappa(x), 0)) - F(x) \leq -l(x, \kappa(x)) \tag{7}$$

and $f(x, \kappa(x), 0) \in \mathbb{X}_f$.

The above assumption indicates that the terminal cost $F(x)$ is a local control Lyapunov function and $\mathbb{X}_f$ is control invariant for the disturbance-free system. Hence, this assumption is often adopted to establish the stability in many MPC [28]. This assumption also provides a guideline for designing the local state-feedback control law $\kappa(x)$. Indeed, many methods can be used to design this control law, among which we prefer the LQR controller based on the linearized system because of the better control performance in terminal set $\mathbb{X}_f$ [25].

C. Research Objective

In the following, we present our main research objective, i.e., design an efficient control scheme that meets the control and state constraints, in order that the state is driven to/near to the operation point. The core challenges of this objective can be summarized as the following two problems.

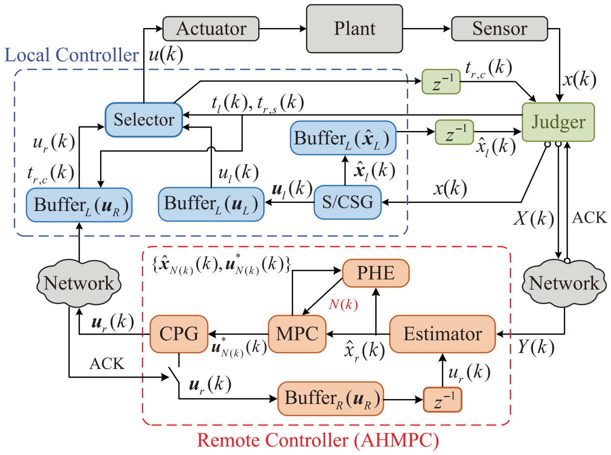


Fig. 2. Configuration of networked dual-mode MPC.

1) How to implement the local and remote controllers as well as the coordination mechanism between them, and how to compensate for the possible packet losses?
2) How to analyze the stability for the resultant nonlinear NCS with two controllers?

The solutions of both questions are discussed in detail in the following two sections, respectively.

III. Networked Dual-Mode AHMPC

The schematic block diagram of the networked dual-mode AHMPC is illustrated in Fig. 2, where the AHMPC is implemented in the remote controller (surrounded by a red box) to generate a predictive control packet, and the role of the local controller (surrounded by a blue box) includes calculating and storing control sequence and selecting a proper control signal for the actuator. The judger decides which controller will be used. The specific functions of these components constitute the main contents of this section.

A. Local Controller

The local controller has limited computing power and, as a result, simple tasks, such as logical judgment and arithmetic operations, can be carried out. So all components, including a state/control sequence generator (S/CSG), three buffers and a control signal selector deployed in the local controller, have simple computing and storing tasks. Next, we discuss the specific functions of each component. To begin with, we define some necessary notions in Fig. 2 that are important in design the buffering and selection rule.

Let $t_l(k)$, $t_{r,s}(k)$, and $t_{r,c}(k)$ represent the time, before or at time k, when the local controller is last adopted, the state packet is last delivered successfully, and the control packet is last accepted by Buffer $_L(\mathbf{u}_R)$, respectively. Based on these definitions, we have

$$t_l(k) = \begin{cases} k & \text{if local controller is selected} \\ t_l(k-1) & \text{otherwise} \end{cases} \tag{8}$$

$$t_{r,s}(k) = \begin{cases} k & \text{if } d_s(k) = 1 \\ t_{r,s}(k-1) & \text{otherwise} \end{cases} \tag{9}$$

where $t_l(0) = t_{r,s}(0) = 0$. Define an indicator function $d_a(k)$, if $t_l(k) < t_{r,s}(k)$, $d_a(k) = 1$; otherwise, $d_a(k) = 0$. Then, the time $t_{r,c}(k)$ with $t_{r,c}(0) = 0$ is determined by

$$t_{r,c}(k) = \begin{cases} k & \text{if } d_a(k) = 1,\ d_c(k) = 1 \\ t_{r,c}(k-1) & \text{otherwise.} \end{cases} \tag{10}$$

Note that the remote control packet can be accepted only when new state has been received by the remote controller after $t_l(k)$ ($d_a(k) = 1$) and the control packet has been successfully transmitted. Indeed, if $d_a(k) = 0$, the remote control sequence, compared with the local one, is obtained by using an older system state, and thus should be discarded.

S/CSG: The role of S/CSG is to compute both the predictive control sequence and state sequence. The specific process is

$$\begin{aligned} \nu_i(k) &= \kappa\big(\hat{x}_i(k)\big) \\ \hat{x}_{i+1}(k) &= f\big(\hat{x}_i(k), \nu_i(k), 0\big),\ \ 0 \le i \le \bar{N}_d - 1 \end{aligned} \tag{11}$$

where $\hat{x}_0(k) = x(k)$, $\kappa(.)$ is an auxiliary control law that can be designed based on Assumption 3. The resultant predictive control sequence $\boldsymbol{u}_l(k) = \{\nu_0(k), \ldots, \nu_{\bar{N}_d-1}(k)\}$ is stored in Buffer$_L(\mathbf{u}_L)$ and the corresponding predicted state sequence $\hat{\mathbf{x}}_l(k) = \{\hat{x}_1(k), \ldots, \hat{x}_{\bar{N}_d}(k)\}$ is stored in Buffer$_L(\hat{\mathbf{x}}_L)$.

Three Buffers and a Selector: Two buffers storing the control sequence provide two alternative control signals, namely, either $u_l(k)$ or $u_r(k)$. Then a proper control signal is selected based on a certain selection rule.

Denote the content of Buffer$_L(\mathbf{u}_R)$ at time k by $\boldsymbol{b}_{lr}(k)$, then the update mechanism can be formulated by

$$\begin{aligned} \boldsymbol{b}_{lr}(k) &= (1 - d_c(k)d_a(k))S\boldsymbol{b}_{lr}(k-1) + d_c(k)d_a(k)\boldsymbol{u}_r(k) \\ u_r(k) &= e^{\mathrm{T}}\boldsymbol{b}_{lr}(k) \end{aligned} \tag{12}$$

where $\boldsymbol{u}_r(k)$ is the remote predictive control packet, $u_r(k)$ is the first element of $\boldsymbol{b}_{lr}(k)$, the matrices S and e have the following structures with compatible dimensions:

$$S \triangleq \begin{bmatrix} 0 & I & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & I & 0 \\ 0 & \ldots & \ldots & 0 & I \\ 0 & \ldots & \ldots & \ldots & 0 \end{bmatrix}, \quad e \triangleq \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Similarly, denote the content of Buffer$_L(\mathbf{u}_L)$ by $\boldsymbol{b}_{ll}(k)$, then

$$\begin{aligned} \boldsymbol{b}_{ll}(k) &= (1 - d_l(k))S\boldsymbol{b}_{ll}(k-1) + d_l(k)\boldsymbol{u}_l(k) \\ u_l(k) &= e^{\mathrm{T}}\boldsymbol{b}_{ll}(k) \end{aligned} \tag{13}$$

where $d_l(k) = 1$ if the local controller is selected at time k; $d_l(k) = 0$, otherwise.

The content of Buffer$_L(\hat{\mathbf{x}}_L)$ is denoted by $\boldsymbol{s}_{ll}(k)$, we have

$$\begin{aligned} \boldsymbol{s}_{ll}(k) &= (1 - d_l(k))S\boldsymbol{s}_{ll}(k-1) + d_l(k)\hat{\boldsymbol{x}}_l(k) \\ \hat{x}_l(k+1) &= e^{\mathrm{T}}\boldsymbol{s}_{ll}(k) \end{aligned} \tag{14}$$

where $\hat{x}_l(k+1)$ is one step prediction of state and will be provided to the judger at the next time step $k+1$.

Finally, we design the following selection rule:

$$u(k) = \begin{cases} u_r(k) & \text{if } t_{r,c}(k) \ge t_l(k) \\ u_l(k) & \text{otherwise.} \end{cases} \tag{15}$$

Remark 1: One may notice the following.

1) In general, $t_l(k)$ and $t_{r,s}(k)$ are available to the judger and remote controller, respectively. Moreover, $t_{r,s}(k)$ is also available to the judger due to the acknowledgment (ACK) signal. Buffer$_L(\mathbf{u}_R)$ determines the time $t_{r,c}(k)$ on the basis of $t_l(k)$ and $t_{r,s}(k)$ that delivered from the judger. Similarly, the judger also knows the time $t_{r,c}(k-1)$ (see Fig. 2).
2) It should be emphasized that the remote controller cannot distinguish whether the packet loss occurs or the local controller is selected when the state packet is not received. Hence, the remote controller will operate normally although the local controller is selected. That is to say, Buffer$_L(\mathbf{u}_R)$ may receive control packet which is computed based on a somewhat outdated estimated state and should be discarded. The packet acceptance condition $t_l(k) < t_{r,s}(k)$ in (10) implies that the remote controller is selected currently and new state packet has been received by remote controller after time $t_l(k)$, then if the control packet is successfully delivered, it will be accepted.
3) The purpose of deploying Buffer$_L(\hat{\mathbf{x}}_L)$ is to provide the actual control input for the state estimator in remote controller when the actual input is selected from Buffer$_L(\mathbf{u}_R)$. This is because $u_l(k)$ can be directly derived from the local estimated state $\hat{x}_l(k)$, i.e., $u_l(k) = \kappa(\hat{x}_l(k))$. Based on the actual input, AHMPC will make a more precise state estimation and further produce more proper control sequence (see Section III-C for details).

B. Judger

The judger coordinates the switching of the two controllers based on a predefined selection rule. Motivated by the dual-model MPC [32], where the auxiliary control law is applied replacing the MPC law when the state enters into the terminal constraint set. Hence, we design the following switching rule.

Rule 1: If $x(k) \in \mathbb{X}_f$, the local controller is selected; otherwise, the remote one is selected.

One may notice that the terminal set $\mathbb{X}_f$ in this article is a control invariant set for the nominal system of (1) (see Assumption 3). Therefore, because of the disturbance and the consecutive packet losses of two networks, the state $x(k)$ may leave from $\mathbb{X}_f$, then the activated controller can be switched from the local one to the remote counterpart.

Another important role of the judger is to lump the necessary data into a packet and transmit it if the remote controller is selected. In Fig. 2, $X(k)$ represents the transmitted packet from the judger and $Y(k)$ the corresponding received packet.

We adjust the packing scheme of $X(k)$ for the purpose that the remote estimator is able to make a more precise estimation. To be specific

$$X(k) = \begin{cases} \{x(k), \hat{x}_l(k)\} & \text{if } t_{r,c}(k-1) < t_l(k) \\ \{x(k)\} & \text{otherwise} \end{cases} \tag{16}$$

where $t_{r,c}(k-1) < t_l(k)$ means that no new control packet has been accepted by Buffer$_L(\mathbf{u}_R)$ during time interval $(t_l(k), k)$, and consequently, the local control input will be selected as the actual input. However, the local control input is unavailable to the remote controller, the local estimated state $\hat{x}_l(k)$, which is closely related to the local input $u_l(k)$ according to (11), is added into $X(k)$ to overcome such difficulty.

Algorithm 1: State Estimation

```
$\hat{x}_r(k)$=generateES($Y(k)$);
Input: the received packet $Y(k)$
Output: the estimated state $\hat{x}_r(k)$
if $Y(k) \neq \emptyset$ then
    $t_{r,s}(k) = k$, $s(k) = 0$;
    $\hat{x}_r(k) = x(k)$;
    record $Y(t_{r,s}(k))$;
    if $Y(k) = \{x(k), \hat{x}_l(k)\}$ then
        $u_l(k) = \kappa(\hat{x}_l(k))$;
    end
else
    $t_{r,s}(k) = t_{r,s}(k-1)$;
    $s(k) = s(k-1) + d_c(k-1)$;
    if $s(k) > 0$ or $Y(t_{r,s}(k)) = \{x(t_{r,s}(k))\}$ then
        $\hat{x}_r(k) = f(\hat{x}_r(k-1), u_r(k-1), 0)$;
    else
        $\hat{x}_r(k) = f(\hat{x}_r(k-1), u_l(k-1), 0)$;
        $\hat{x}_l(k) = f(\hat{x}_l(k-1), u_l(k-1), 0)$;
        $u_l(k) = \kappa(\hat{x}_l(k))$;
    end
end
return $\hat{x}_r(k)$;
```

C. Remote Controller: AHMPC

The AHMPC is implemented at the remote side (see Fig. 2), which has strong computing power and is capable to solve the FHOCP efficiently. The control structure of remote controller includes five components and is elaborated in the following.

Buffer: Similar to the buffers deployed in the local controller, we deploy a buffer at the remote side [Buffer$_R(\mathbf{u}_R)$ in Fig. 2] for storing the control sequence and providing the control input for the state estimator. Denote the content of this buffer by $\boldsymbol{b}_{rr}(k)$, then we have

$$\begin{aligned} \boldsymbol{b}_{rr}(k) &= d_c(k)S\boldsymbol{b}_{rr}(k-1) + (1-d_c(k))\boldsymbol{u}_r(k) \\ u_r(k) &= e^{\mathrm{T}}\boldsymbol{b}_{rr}(k) \end{aligned} \tag{17}$$

where $\boldsymbol{b}_{rr}(0) = \mathbf{0}$, $\boldsymbol{u}_r(k)$ is remote predictive control packet, $d_c(k)$ is defined in (5) and is available to remote controller through ACK signal.

State Estimator: The state estimator calculates an estimation of the current plant state if the packet loss occurs. The estimation algorithm is given by Algorithm 1.

The following three aspects about this algorithm are noteworthy.

1) The time instant $t_{r,s}(k)$ defined in (9) representing the time when the latest state packet is successfully delivered before or at time k, is also recorded in the state estimator.
2) The variable $s(k)$ can be rewritten as $s(k) = \sum_{i=t_{r,s}(k)}^{k-1} d_c(i)$, which reflects the transmissions over

Algorithm 2: Prediction Horizon Estimation

```
$N(k)$=estimatePH($\hat{x}_r(k)$);
Input: The estimated state $\hat{x}_r(k)$
Output: The prediction horizon $N(k)$
Initialization: Update the latest $M$ control and state
sequences $\boldsymbol{B}_M(k)$, and let the lower bound of the
prediction horizon be $\underline{N}(k) = N(t_M) - (k - t_M)$;
if $k = 0$ then
    $N(k) = N_{\max}$;
else
    for $i = 1 : M$ do
        find $H$ nearest points $\{\hat{x}_{l_1}(t_i), \dots, \hat{x}_{l_H}(t_i)\}$ for
        $\hat{x}_r(k)$ from $\hat{\boldsymbol{x}}_{N(t_i)}(t_i)$;
        for $j = 1 : H$ do
            if $\{v^*_{l_j}(t_i), \dots, v^*_{N(t_i)-1}(t_i)\}$ is a feasible control
            sequence for $\hat{x}_r(k)$ then
                $\hat{N}(i, j) = \max\{\underline{N}(k), N(t_i) - l_j\}$;
            else
                $\hat{N}(i, j) = N_{\max}$;
            end
        end
    end
    $N(k) = \min_{i,j}\{\hat{N}(i, j)\}$;
end
return $N(k)$;
```

control network. $s(k) = 0$ means no control packet has been successfully transmitted during $[t_{r,s}(k), k)$.

3) Note that although $s(k) > 0$, the local controller may be selected at time $t_l(k)$, where $t_{r,s}(k) < t_l(k) \le k$, then the remote control packet will be rejected by Buffer$_L(\mathbf{u}_R)$ even if it is transmitted successfully. Since no new state packet has been successfully transmitted during the interval $[t_l(k), k)$, the remote estimated state can be arbitrary.

MPC: The key component of the remote controller is MPC, which is employed to calculate the predictive control sequence by solving the FHOCP (6). At time k, the prediction horizon of the FHOCP is $N(k)$ and the initial condition is the estimated state $\hat{x}_r(k)$. The resultant control sequence is denoted by $\boldsymbol{u}^*_{N(k)}(k) = \{v^*_0(k), \dots, v^*_{N(k)-1}(k)\}$, the predicted state sequence is $\hat{\boldsymbol{x}}_{N(k)}(k) = \{\hat{x}_0(k), \dots, \hat{x}_{N(k)-1}(k)\}$ and the optimal MPC value function is defined as

$$V^0_{N(k)}\big(\hat{x}_r(k)\big) = \sum_{j=0}^{N(k)-1} l\Big(\hat{x}_j(k), v^*_j(k)\Big) + F\big(\hat{x}_{N(k)}(k)\big) \quad (18)$$

where $\hat{x}_0(k) = \hat{x}_r(k)$. $\boldsymbol{u}^*_{N(k)}(k)$ and $\hat{\boldsymbol{x}}_{N(k)}(k)$ will be used to estimate the prediction horizon. The prediction horizon can be variable and dependent on the estimated state at each time step, thus the overall process is also called AHMPC.

Prediction Horizon Estimator (PHE): Once the estimated state is obtained, we need to determine the prediction horizon $N(k)$ before solving the MPC. It is a fairly difficult problem in general as the relation between the initial state and resultant control sequence is hard to be explicitly specified.

Inspired by the fact that the historical predicted control sequences can be used to construct a feasible control sequence as long as the difference between the actual system state and predicted state is small [33], we here use such feasible control sequence to determine the prediction horizon by recognizing the time steps required to steer the state into the terminal constraint set. To realize this idea, we need the latest M state and control sequence pairs $\boldsymbol{B}_M(k) \triangleq \{B(t_1), \dots, B(t_M)\}$ where $B(t_i) = \{\hat{\boldsymbol{x}}_{N(t_i)}(t_i), \boldsymbol{u}^*_{N(t_i)}(t_i)\}$, and $t_1, \dots, t_M$ are the latest M times when the MPC is performed before k with t_M being the latest. We also set a lower bound of the prediction horizon $\underline{N}(k) = N(t_M) - (k - t_M)$ in order for easy stability analysis ($k + N(k) \ge t_M + N(t_M)$). Algorithm 2 gives the specific estimation process.

It should be indicated that:

1) the above algorithm exploits MH nearest points from the previous M control and state sequence pairs to increase the robustness against the disturbance and the possible packet losses;
2) if the number of available control and state sequence pairs is smaller than M, all these pairs will be considered. Similarly, if the length of the state sequence is less than H, all elements of the sequence will be checked;
3) the M control and state sequence pairs can be analyzed in a parallel manner to lower the computing time.

Control Packet Generator (CPG): In order to compensate for the consecutive packet losses, the control packet length is set to be $\bar{N}_d$. The role of the CPG is to generate the control packet on the basis of the optimal control sequence $\boldsymbol{u}^*_{N(k)}(k)$ and an auxiliary control law. Specifically, if $\bar{N}_d \le N(k)$, the first $\bar{N}_d$ elements of $\boldsymbol{u}^*_{N(k)}(k)$ is selected to constitute the control packet. Otherwise, in addition to the $\boldsymbol{u}^*_{N(k)}(k)$, the CPG will generate the remaining control signals with resorting to an auxiliary control law $\kappa(x)$. The existence of such control law is guaranteed by Assumption 3. The constructing process is formulated below

$$\begin{aligned} v_i(k) &= \kappa\big(\hat{x}_i(k)\big) \\ \hat{x}_{i+1}(k) &= f\big(\hat{x}_i(k), v_i(k), 0\big), \;\; N(k) \le i \le \bar{N}_d - 1. \end{aligned} \quad (19)$$

By this way, the generated control packet is denoted by

$$\boldsymbol{u}_r(k) = \begin{cases} \Big\{v^*_0(k), \dots, v^*_{\bar{N}_d-1}(k)\Big\}, & \text{if } N(k) \ge \bar{N}_d \\ \Big\{\boldsymbol{u}^*_{N(k)}(k), v_{N(k)}(k), \dots, v_{\bar{N}_d-1}(k)\Big\}, & \text{otherwise} \end{cases}$$

which will be transmitted to the local controller through the control network.

Work in an Efficient Manner: The control packet will be generated in a more efficient manner. To describe this manner, we denote the time when the control packet is last transmitted successfully before or at time k by $t_{cp}(k)$. The MPC, PHE, and CPG only work if there is further state information provided, i.e., if $t_{r,s}(k) > t_{cp}(k)$ [new sensing data has been received after $t_{cp}(k)$]. Note that Buffer$_R(\mathbf{u}_R)$ will update continuously. For more details, please refer to [16].

Example: We take the following scalar system for example to clarify the AHMPC algorithm. The system is given by

$$x(k+1) = 1.2x(k) + 0.08x(k)u(k) + 0.5u(k) + w(k)$$

with the constraints $|x(k)| < 4$, $|u(k)| < 1.2$, $|w(k)| < 0.03$. The stage cost and terminal cost are $l(x, u) = 0.1x^2 + u^2$ and $F(x) = 2.2x^2$, the terminal set is $\mathbb{X}_f = \{x|F(x) \le 0.1\}$. The initial prediction horizon is $N_{\max} = 6$.

A distinct feature of the AHMPC algorithm is the variable prediction horizon. Hence, in this example, the PHE algorithm will be explained in detail. For simplicity, we set $H = 1$ for Algorithm 2. Suppose that the MPC has been performed at $k = 3$ ($\hat{x}_r(3) = 1.100, N(3) = 5$) and at $k = 5$ ($\hat{x}_r(5) = 0.7236$, $N(5) = 3$). The related state and control sequence pairs are $B(3) = \{\hat{\boldsymbol{x}}_{N(3)}(3), \boldsymbol{u}^*_{N(3)}(3)\}$ and $B(5) = \{\hat{\boldsymbol{x}}_{N(5)}(5), \boldsymbol{u}^*_{N(5)}(5)\}$, where $\hat{\boldsymbol{x}}_{N(3)}(3) = \{1.100, 0.9444, 0.7259, 0.4662, 0.3083\}$

$$\boldsymbol{u}^*_{N(3)}(3) = \{-1.2, -1.2, -1.1158, -0.7277, -0.475\}$$
$$\hat{\boldsymbol{x}}_{N(5)}(5) = \{0.7236, 0.4649, 0.3077\}$$
$$\boldsymbol{u}^*_{N(5)}(5) = \{-1.1122, -0.7254, -0.4734\}.$$

Assume that the MPC is performed at $k = 6$. We set $M = 1$ for Algorithm 2, i.e., the latest state and control sequence pair $B(5)$ is utilized, then the lower bound of $N(6)$ is $\underline{N}(6) = 2$. Given that $\hat{x}_r(6) = 0.4642$, the nearest point is 0.4649 ($l_1 = 1$) and the control sequence $\{-0.7254, -0.4734\}$ is feasible, then we have $N(6) = 2$. However, if $\hat{x}_r(6) = 0.4655$, we can verify that $\{-0.7254, -0.4734\}$ is no longer feasible, which leads to $N(6) = N_{\max} = 6$. This is because of the disturbance and the fact that packet loss may occur at $k = 5$. Now, if we set $M = 2$ for Algorithm 2, then both $B(3)$ and $B(5)$ will be used. For $B(3)$, the nearest point is 0.4662 ($l_1 = 3$) and the control sequence $\{-0.7277, -0.475\}$ is feasible. Then, we have $N(6) = N(3) - l_1 = 2$. Therefore, we can observe that large M increases the robustness against the disturbance and packet losses. In fact, another parameter H plays similar role in Algorithm 2. The remaining tasks of the AHMPC include computing the remote control packet and transmitting it to the local controller, which are completed by the MPC and CPG.

D. Overall Control Algorithm

With the above configurations, the flowchart of the proposed networked dual-mode AHMPC scheme is shown in Fig. 3, and the overall principle is summarized as follows.

1) At each time k, the sensor measures the plant state. The state will be transmitted to the remote controller over the network if the condition $x(k) \in \mathbb{X}_f$ is violated. Otherwise, the state will be delivered to the local controller.
2) If the state is transmitted to the remote controller, the AHMPC scheme is adopted. First, the estimated state is yielded by the state estimator (Algorithm 1). Next, the prediction horizon $N(k)$ can be determined by the PHE (Algorithm 2). Then, solving the FHOCP (6) with prediction horizon $N(k)$ obtains the predictive control sequence $\boldsymbol{u}^*_{N(k)}(k)$. Finally, the CPG generates the remote control packet $\boldsymbol{u}_r(k)$ with packet length $\bar{N}_d$, and transmits it to the local controller.
3) If the state is delivered to the local controller, the deployed S/CSG is used to compute the local predictive control sequence $\boldsymbol{u}_l(k)$ and state sequence $\hat{\mathbf{x}}_l(k)$.

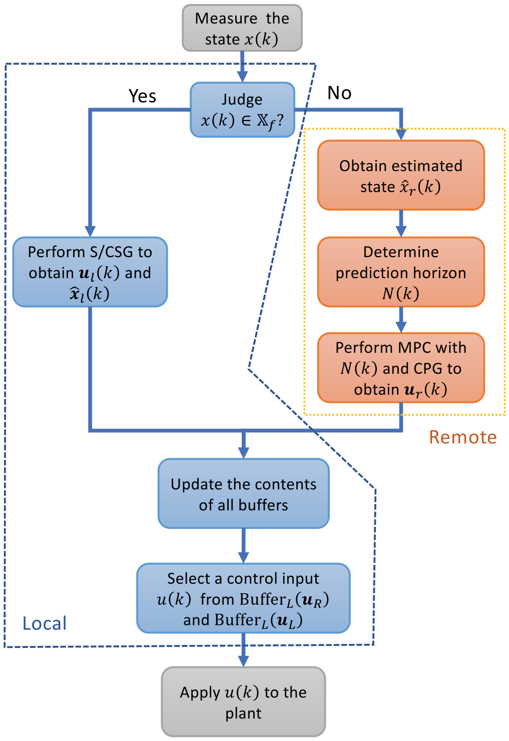

Measure the state $x(k)$
Yes
Judge $x(k) \in \mathbb{X}_f$?
No
Obtain estimated state $\hat{x}_r(k)$
Determine prediction horizon $N(k)$
Perform MPC with $N(k)$ and CPG to obtain $\boldsymbol{u}_r(k)$
Remote
Perform S/CSG to obtain $\boldsymbol{u}_l(k)$ and $\hat{\boldsymbol{x}}_l(k)$
Update the contents of all buffers
Select a control input $u(k)$ from Buffer$_L(\boldsymbol{u}_R)$ and Buffer$_L(\boldsymbol{u}_L)$
Local
Apply $u(k)$ to the plant

Fig. 3. Flowchart of the networked dual-mode AHMPC scheme.

4) Finally, all the buffers [Buffer$_L(\mathbf{u}_L)$, Buffer$_L(\mathbf{u}_R)$, and Buffer$_L(\hat{\mathbf{x}}_L)$] in the local controller update their contents based on the corresponding rules. The selector chooses a proper control input $u(k)$ for the actuator according to the rule (15).

The overall procedures of the networked dual-mode AHMPC scheme are demonstrated in Algorithm 3. We assume the initial state is available to the remote controller and the predictive control sequence is also available to the Buffer$_L(\boldsymbol{u}_R)$. Such assumption is not restrictive because the initial control sequence can be computed offline.

Moreover, the advantages of using the proposed scheme are also noteworthy.

1) Compared with the non-networked MPC [28], the networked configuration reduces the computing burden of the local controller, and thus has the potential for broad applications.
2) The deployment of the AHMPC with the CPG can actively compensate for the packet losses, and meanwhile, in contrast to the conventional networked MPC [16], has superior computational efficiency.
3) The separate installation of the auxiliary control law $\kappa(x)$ (local side) and the AHMPC scheme (remote side) reduces the usage of the unreliable networks and gains better control performance in the terminal set $\mathbb{X}_f$ as the local controller is not subjected to the packet losses.

E. Resultant Control Law

The control law that affected by the possible packet losses, buffering procedures, judge mechanism, and local controller can be rewritten in a compact form. Denote three time sequences by $\{t_l(k)\}_{k=0}$, $\{t_{r,s}(k)\}_{k=0}$, and $\{t_{r,c}(k)\}_{k=0}$ with the

Algorithm 3: Overall System Evolution

```
Input: Initial state $x_0$
Output: State evolution $x(k)$ and control input $u(k)$
Initialization: Set $\hat{x}_r(0) = x(0) = x_0$,
$t_l(0) = t_{r,s}(0) = t_{r,c}(0) = 0$, $\boldsymbol{b}_{ll}(0) = \mathbf{0}$, $\boldsymbol{s}_{ll}(0) = \mathbf{0}$,
performing AHMPC yields control sequence $\boldsymbol{u}_r(0)$.
Let $\boldsymbol{b}_{lr}(0) = \boldsymbol{b}_{rr}(0) = \boldsymbol{u}_r(0)$, and $u(0)$ is then obtained;
for $k = 1, 2, \ldots$ do
    measure the current state: obtain $x(k)$;
    Buffer$_L(\hat{\mathbf{x}}_L)$ and Buffer$_L(\boldsymbol{u}_R)$ provides $\hat{x}_l(k)$ and
    $t_{r,c}(k-1)$;
    if $x(k) \in \mathbb{X}_f$ then
        $t_l(k) = k$;
        the S/CSG obtains $\boldsymbol{u}_l(k)$ and $\hat{\boldsymbol{x}}_l(k)$;
        set $\boldsymbol{b}_{ll}(k) = \boldsymbol{u}_l(k)$, $\boldsymbol{s}_{ll}(k) = \hat{\boldsymbol{x}}_l(k)$;
        $u(k) = u_l(k)$, generate $x(k+1)$;
    else
        $t_l(k) = t_l(k-1)$;
        determine data packet $X(k)$ based on (16);
        if $t_{r,s}(k) > t_{cp}(k)$ then
            update $t_{r,s}(k)$, $\hat{x}_r(k)$=generateES($Y(k)$);
            estimate $N(k)$ = estimatePH($\hat{x}_r(k)$);
            perform MPC and CPG to obtain $\boldsymbol{u}_r(k)$;
        end
        update $\boldsymbol{b}_{rr}(k)$, $\boldsymbol{b}_{lr}(k)$, $\boldsymbol{b}_{ll}(k)$, $\boldsymbol{s}_{ll}(k)$ and $t_{r,c}(k)$
        based on (17), (12), (13), (14) and (10);
        select control input $u(k)$ according to (15), and
        generate $x(k+1)$;
    end
end
```

meaning being mentioned before. Define another sequence $\{t_s(k)\}_{k=0} \triangleq \{\max\{t_{r,s}(k), t_l(k)\}\}_{k=0}$ to represent the latest time when the state is used by the local or remote controller. We iteratively define the sequence $\{q_i\}_{i=0}$ by letting $q_{i+1} = \inf_k\{t_l(k)|t_l(k) > t_l(q_i)\}$ with $q_0 = 0$, which records the time that the local controller is selected. We also define $\{p_i\}_{i=0}$ to denote the time sequence that the received control packet is accepted by Buffer$_L(\mathbf{u}_R)$ via $p_{i+1} = \inf_k\{t_{r,c}(k)|t_{r,c}(k) > t_{r,c}(p_i)\}$ with $p_0 = 0$, for any k, there exist i and j such that $q_i \leq k < q_{i+1}$ and $p_j \leq k < p_{j+1}$. Let $\underline{m}_{ij} = \max\{q_i, p_j\}$, $\bar{m}_{ij} = \min\{q_{i+1}, p_{j+1}\}$, then the control law is written via

$$u(k) = \boldsymbol{\phi}_{t_s(k)}\Big(k - \underline{m}_{ij}\Big), \quad \underline{m}_{ij} \leq k < \bar{m}_{ij} \tag{20}$$

where $\boldsymbol{\phi}_{t_s(k)} \triangleq \text{col}\{\phi_{t_s(k)}(0), \phi_{t_s(k)}(1), \ldots, \phi_{t_s(k)}(\bar{N}_d - 1)\}$ that computed based on the actual state $x(t_s(k))$, and

$$\phi_{t_s(k)}\Big(k - \underline{m}_{ij}\Big) = \begin{cases} v_{k-q_i}(q_i|t_s(k)), & q_i = \underline{m}_{ij} \\ v_{k-p_j}\big(p_j|t_s(k)\big), & p_j = \underline{m}_{ij} \end{cases}$$

with $v_{k-q_i}(k|t_s(k))$ representing the $k - q_i + 1$th element of control sequence $\boldsymbol{u}_r(q_i)$ and $v_{k-p_j}(k|t_s(k))$ the $k - p_j + 1$th element of sequence $\boldsymbol{u}_l(p_j)$.

IV. Stability Analysis

In this section, we introduce some basic assumptions, and then based on which prove the robust stability.

We first iteratively define a mapping of state evolution as $f^j(x(k), \boldsymbol{\phi}_{t_s(k)}([0 : j-1]), \boldsymbol{w}([k : k+j-1])) \triangleq f(f^{j-1}(x(k), \boldsymbol{\phi}_{t_s(k)}([0 : j-2]), \boldsymbol{w}([k : k+j-2])), \phi_{t_s(k)}(j-1), w(k+j-1))$, $\forall j = 1, 2, \ldots, \bar{m}_{ij} - \underline{m}_{ij}$, where $\boldsymbol{w}([k : k+j-1])) \in \mathbb{W}^j$ and $f^0(x, \boldsymbol{\phi}_{t_s(k)}([0 : \ -1]), w) = x$. Further, we iteratively define a $\mathcal{K}$ function γ_i via $\gamma_{i+1} = \varphi_x \circ \gamma_i + \varphi_w$ with $\gamma_1 = \varphi_w$, then we obtain $\|f^j(x(k), \boldsymbol{\phi}_{t_s(k)}([0 : j-1]), \boldsymbol{w}([k : k+j-1])) - f^j(x(k), \boldsymbol{\phi}_{t_s(k)}([0 : j-1]), 0)\| \leq \gamma_j(\|w\|)$.

Assumption 4: The compact set $\mathcal{X}^{\text{MPC}}$ satisfying $\{0\} \subset \mathcal{X}^{\text{MPC}} \subseteq \mathbb{X}$ and $\mathcal{X}^{\text{MPC}} \subseteq \mathbb{X}_{N_{\max}} \ominus \mathcal{B}^n(\gamma_{\bar{N}_d}(\|w\|))$ is an RPI set for the mapping $f^j(x(k), \boldsymbol{\phi}_{t_s(k)}([0 : j-1]), \boldsymbol{w}([k : k+j-1]))$ $\forall j = 1, 2, \ldots, \bar{N}_d$, where $\mathbb{X}_{N_{\max}}$ is a set of feasible states, i.e., the state from which can be steered into $\mathbb{X}_f$ within $N_{\max}$ steps by using an admissible control sequence.

Remark 2: The feasibility of the proposed MPC algorithm is guaranteed by this assumption that is also seen in [16]. Due to the possible consecutive packet losses (with maximum number $\bar{N}_d$), the set $\mathcal{X}^{\text{MPC}}$ should be the RPI set for all mappings $f^j(.), j = 1, \ldots, \bar{N}_d$. In fact, the existence of such compact set is guaranteed if the disturbance small enough.

Assumption 5: There exist class $\mathcal{K}$ functions α_l, α_F, φ_l, φ_F, and $\varphi_V^{[N]}$ dependent on prediction horizon N such that the following inequalities hold for all $x, y \in \mathbb{X}$, $u \in \mathbb{U}$:

$$l(x, u) \geq \alpha_l(\|x\|), \ \ l(0, 0) = 0 \tag{21}$$

$$|l(x, u) - l(y, u)| \leq \varphi_l(\|x - y\|) \tag{22}$$

$$\left|V_N^0(x) - V_N^0(y)\right| \leq \varphi_V^{[N]}(\|x - y\|), \ \ V_N^0(0) = 0 \tag{23}$$

and the following inequalities hold for all $x, y \in \mathbb{X}_f$:

$$F(x) \geq \alpha_F(\|x\|), \ \ F(0) = 0 \tag{24}$$

$$|F(x) - F(y)| \leq \varphi_F(\|x - y\|). \tag{25}$$

The local $\mathcal{K}$-continuous assumptions on stage cost (22) and terminal costs (25) are often employed to derive the stability results (see [28], [34]), while the continuous assumption of the optimal value function (23) is often used to bound the effects of the disturbance on the optimal value function [15]. Since we have set the maximum prediction horizon $N_{\max}$, (23) can be rewritten as

$$\left|V_N^0(x) - V_N^0(y)\right| \leq \varphi_V(\|x - y\|) \tag{26}$$

where φ_V is the pointwise maximum for all possible prediction horizon N, i.e., $\varphi_V \triangleq \max\{\varphi_V^{[1]}, \ldots, \varphi_V^{[N_{\max}]}\}$. It can be easily verified that φ_V is also a $\mathcal{K}$ function.

By introducing these assumptions, we derive the stability result in the remaining part. Before proceeding, we introduce some notions and lemmas for ease of presentation.

Let $\hat{x}(k|k')$ represent the estimated state $\hat{x}_r(k)$ that is computed based on the latest actual state $x(k')$. Note that $k' = t_{s,r}(k)$ and $\hat{x}(k|k) = x(k)$. Similarly, we use $\hat{x}_j(k|k')$ to represent the predicted state of the MPC algorithm with initial state $\hat{x}(k|k')$, i.e., $\hat{x}_j(k|k') = f^j(\hat{x}(k|k'), \boldsymbol{u}_R([0{:}j-1]), 0)$, $0 \leq j \leq N_{\max}$. Let $K = \{k_i\}_{i=0} \subseteq \mathbb{N}_{\geq 0}$ be the time sequence when either the local controller is adopted or new control packet is accepted by the Buffer$_L(\mathbf{u}_R)$. We define the time sequence $\{k_i^-\}_{i=0} \triangleq \{t_{s,r}(k_i)\}_{i=0}$ that will be used in the following analysis. We assume that $k_0 = k_0^- = 0$. Define $\Delta_i \triangleq k_{i+1} - k_i$

and $\delta_i \triangleq k_i - k_i^-$. We also define the following cost function dependent on the actual state, optimal control sequence and the time-varying prediction horizon

$$J_{N(k)}(x(k)) = \sum_{j=0}^{N(k)-1} l\Big(\hat{x}_j(k|k), v_j^*(k)\Big) + F\Big(\hat{x}_{N(k)}(k|k)\Big) \quad (27)$$

where $J_{N(k)}(x(k)) = F(x(k))$ if $N(k) = 0$, and $\hat{x}_j(k|k) = f(\hat{x}_{j-1}(k|k), v_{j-1}^*(k), 0)$ with $\hat{x}_0(k|k) = x(k)$.

The following lemma explores the upper bound of the estimation error and specifies the discrepancy between the cost function (27) and the optimal value function (18).

Lemma 1: If Assumptions 1 and 5 hold, then we have

$$\left\|\hat{x}_j(k|k) - \hat{x}_j(k|k')\right\| \le \varphi_x^j \circ \gamma_{k-k'}(\|\mathbb{W}\|) \quad (28)$$

with $\varphi_x^j = \varphi_x(\varphi_x^{j-1})$, $\varphi_x^0(b) = b$, and for any prediction horizon $1 \le N \le N_{\max}$, we have

$$\left|J_N(x(k)) - V_N^0\big(\hat{x}(k|k')\big)\right| \le \varphi_J \circ \gamma_{k-k'}(\|\mathbb{W}\|) \quad (29)$$

where $\varphi_J = \max\{\varphi_J^{[1]}, \ldots, \varphi_J^{[N_{\max}]}\}$ is a class $\mathcal{K}$ function and $\varphi_J^{[N]} = \sum_{j=0}^{N-1} \varphi_l \circ \varphi_x^j + \varphi_F$.

Proof: 1) Inequality (28) can be directly obtained by recursively using to Assumption 1. The proof can also be found in [15, Lemma 1].

2) According to the definitions, we obtain

$$\begin{aligned}
&\left|J_N(x(k)) - V_N^0\big(\hat{x}(k|k')\big)\right| \\
&\le \sum_{j=0}^{N-1} \left| l\Big(\hat{x}_j(k|k), v_j^*(k)\Big) - l\Big(\hat{x}_j(k|k'), v_j^*(k)\Big)\right| \\
&\quad + \left|F\big(\hat{x}_N(k|k)\big) - F\big(\hat{x}_N(k|k')\big)\right| \\
&\le \sum_{j=0}^{N-1} \varphi_l\Big(\left\|\hat{x}_j(k|k) - \hat{x}_j(k|k')\right\|\Big) \\
&\quad + \varphi_F\Big(\left\|\hat{x}_N(k|k) - \hat{x}_N(k|k')\right\|\Big) \\
&\le \varphi_J\Big(\left\|x(k) - \hat{x}(k|k')\right\|\Big) \le \varphi_J \circ \gamma_{k-k'}(\|\mathbb{W}\|). \quad (30)
\end{aligned}$$

These complete the proof. ■

Theorem 1: Consider the system in (1) with the designed control scheme. Suppose that Assumptions 1–5 hold, and if the initial state satisfies $x_0 \in \mathcal{X}^{\text{MPC}}$, then the overall system is ISpS, i.e., there exists a $\mathcal{KL}$ function β, a $\mathcal{K}$ function γ, and a constant $c \ge 0$ such that

$$\|x(k)\| \le \beta(\|x_0\|, k) + \gamma(\|\mathbb{W}\|) + c \quad \forall k \ge 0. \quad (31)$$

Proof: Take the cost function $J_{N(k)}(x(k))$ defined in (27) as a Lyapunov function candidate. In the following, we first discuss the difference between the Lyapunov functions at two consecutive control sequence update instants $k_i, k_{i+1} \in K$, i.e., the time when $x(k) \in \mathbb{X}_f$ or when a new remote control packet is accepted by $\text{Buffer}_L(\mathbf{u}_R)$, which is a routine in the Lyapunov-based method. Four scenarios are shown in Fig. 4, including: (a) $x(k_i) \notin \mathbb{X}_f$ and $x(k_{i+1}) \notin \mathbb{X}_f$; (b) $x(k_i) \in \mathbb{X}_f$ and $x(k_{i+1}) \notin \mathbb{X}_f$; (c) $x(k_i) \notin \mathbb{X}_f$ and $x(k_{i+1}) \in \mathbb{X}_f$; and (d) $x(k_i) \in \mathbb{X}_f$ and $x(k_{i+1}) \in \mathbb{X}_f$. All these scenarios reflecting the switchings between the two controller are likely to occur, and thus should be considered. Note that when $x(k) \in \mathbb{X}_f$, $N(k) = 0$; otherwise, $N(k) > 0$.

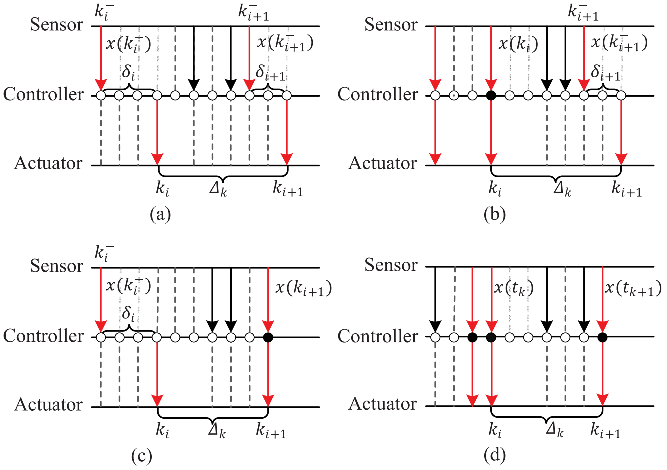


Fig. 4. Four scenarios of the state at the two consecutive control sequence update instants. (a) $x(k_i) \notin \mathbb{X}_f$ and $x(k_{i+1}) \notin \mathbb{X}_f$. (b) $x(k_i) \in \mathbb{X}_f$ and $x(k_{i+1}) \notin \mathbb{X}_f$. (c) $x(k_i) \notin \mathbb{X}_f$ and $x(k_{i+1}) \in \mathbb{X}_f$. (d) $x(k_i) \in \mathbb{X}_f$ and $x(k_{i+1}) \in \mathbb{X}_f$. If $x(k) \in \mathbb{X}_f$, the local controller (representing by "•") is used; otherwise, the remote one (representing by "○") is used.

For scenario Fig. 4(a), the remote controller is adopted during time interval $[k_i, k_{i+1}]$, then we have

$$\begin{aligned}
&J_{N(k_{i+1})}(x(k_{i+1})) - J_{N(k_i)}(x(k_i)) \\
&\le J_{N(k_{i+1})}(x(k_{i+1})) - V^0_{N(k_{i+1})}\big(\hat{x}_{\Delta_i}\big(k_i|k_i^-\big)\big) \\
&\quad + V^0_{N(k_{i+1})}\big(\hat{x}_{\Delta_i}\big(k_i|k_i^-\big)\big) - V^0_{N(k_i)}\big(\hat{x}\big(k_i|k_i^-\big)\big) \\
&\quad + V^0_{N(k_i)}\big(\hat{x}\big(k_i|k_i^-\big)\big) - J_{N(k_i)}(x(k_i)). \quad (32)
\end{aligned}$$

First, we analyze $V^0_{N(k_{i+1})}(\hat{x}_{\Delta_i}(k_i|k_i^-)) - V^0_{N(k_i)}(\hat{x}(k_i|k_i^-))$ by constructing a feasible control sequence for predicted state $\hat{x}_{\Delta_i}(k_i|k_i^-)$. If $N(k_{i+1}) = N(k_i) - \Delta_i$, we consider the control sequence $\boldsymbol{u}^{\#}(k_{i+1}) = \{v^*_{\Delta_i}(k_i), \ldots, v^*_{N(k_i)-1}(k_i)\}$ and if $N(k_{i+1}) > N(k_i) - \Delta_i$, we set $\boldsymbol{u}^{\#}(k_{i+1}) = \{v^*_{\Delta_i}(k_i), \ldots, v^*_{N(k_i)-1}(k_i), v^{\#}_{N(k_i)}, \ldots, v^{\#}_{\Delta_i+N(k_{i+1})-1}\}$, where $v_j^{\#} = \kappa(\hat{x}_j(k_i|k_i^-))$, $\hat{x}_{j+1}(k_i|k_i^-) = f(\hat{x}_j(k_i|k_i^-), \kappa(\hat{x}_j(k_i|k_i^-)), 0)$ $\forall j = N(k_i), \ldots, N(k_{i+1}) + \Delta_i - 1$. By applying Assumption 3, we have

$$\begin{aligned}
&V^0_{N(k_{i+1})}\big(\hat{x}_{\Delta_i}\big(k_i|k_i^-\big)\big) - V^0_{N(k_i)}\big(\hat{x}\big(k_i|k_i^-\big)\big) \\
&\le \sum_{j=N(k_i)}^{\Delta_i+N(k_{i+1})-1} \Big(l\Big(\hat{x}_j\big(k_i|k_i^-\big), v_j^{\#}(k_i)\Big) - F\big(\hat{x}_j\big(k_i|k_i^-\big)\big) \\
&\qquad + F\big(\hat{x}_{j+1}\big(k_i|k_i^-\big)\big)\Big) - \sum_{j=0}^{\Delta_i-1} l\Big(\hat{x}_j\big(k_i|k_i^-\big), v_j^*(k)\Big) \\
&\le -l\big(x(k_i), v_0^*(k_i)\big) + l\big(x(k_i), v_0^*(k_i)\big) - l\big(\hat{x}\big(k_i|k_i^-\big), v_0^*(k_i)\big) \\
&\le -\alpha_l(\|x(k_i)\|) + \varphi_l \circ \gamma_{\delta_i}(\|\mathbb{W}\|). \quad (33)
\end{aligned}$$

Second, the discrepancy between $V^0_{N(k_i)}(\hat{x}(k_i|k_i^-))$ and $J_{N(k_i)}(x(k_i))$ has been studied in Lemma 1, i.e.,

$$\left|J_{N(k_i)}(x(k_i)) - V^0_{N(k_i)}\big(\hat{x}\big(k_i|k_i^-\big)\big)\right| \le \varphi_J \circ \gamma_{\delta_i}(\|\mathbb{W}\|). \quad (34)$$

Third, we study $J_{N(k_{i+1})}(x(k_{i+1})) - V^0_{N(k_{i+1})}(\hat{x}_{\Delta_i}(k_i|k_i^-))$. However, the control sequences in $J_{N(k_{i+1})}(x(k_{i+1}))$ and in

$V^0_{N(k_{i+1})}(\hat{x}_{\Delta_i}(k_i|k_i^-))$ are inconsistent, then Lemma 1 is not directly applicable. By utilizing Assumption 5, Lemma 1 and (30), we have

$$\begin{aligned}
&\left|J_{N(k_{i+1})}(x(k_{i+1})) - V^0_{N(k_{i+1})}\big(\hat{x}_{\Delta_i}\big(k_i|k_i^-\big)\big)\right| \\
&\quad\le \left|J_{N(k_{i+1})}(x(k_{i+1})) - V^0_{N(k_{i+1})}\big(\hat{x}\big(k_{i+1}|k_{i+1}^-\big)\big)\right| \\
&\qquad+\left|V^0_{N(k_{i+1})}\big(\hat{x}\big(k_{i+1}|k_{i+1}^-\big)\big) - V^0_{N(k_{i+1})}\big(\hat{x}_{\Delta_i}\big(k_i|k_i^-\big)\big)\right| \\
&\quad\le \alpha_{a1}(\|\mathbb{W}\|)
\end{aligned} \tag{35}$$

where $\alpha_{a1} = \varphi_J \circ \gamma_{\delta_{i+1}} + \varphi_V \circ \varphi_x^{\delta_{i+1}} \circ \gamma_{\delta_i+\Delta_i-\delta_{i+1}}$.

Incorporating (33), (30), and (35) yields

$$J_{N(k_{i+1})}(x(k_{i+1})) - J_{N(k_i)}(x(k_i)) \le -\alpha_l(\|x(k_i)\|) + \alpha_a(\|\mathbb{W}\|) \tag{36}$$

where $\alpha_a = \varphi_J \circ \gamma_{\delta_i} + \alpha_{a1} + \varphi_l \circ \gamma_{\delta_i}$.

For scenario Fig. 4(b), the local controller is used at time k_i but then the plant state leaves $\mathbb{X}_f$. During the time interval $[k_i, k_{i+1})$, the control inputs are provided by $\text{Buffer}_L(\mathbf{u}_L)$. When $x \in \mathbb{X}_f$, the Lyapunov function becomes $F(x)$. Then, analyzing $J_{N(k_{i+1})}(x(k_{i+1})) - F(x(k_i))$ obtains

$$\begin{aligned}
&J_{N(k_{i+1})}(x(k_{i+1})) - F(x(k_i)) \\
&\quad\le J_{N(k_{i+1})}(x(k_{i+1})) \\
&\qquad+ \sum_{j=0}^{N(k_{i+1})+\Delta_i-1} \Big(l\Big(\hat{x}_j(k_i|k_i), \nu_j^{\#}(k_i)\Big) - F\big(\hat{x}_j(k_i|k_i)\big) \\
&\qquad\qquad +F\big(\hat{x}_{j+1}(k_i|k_i)\big)\Big) \\
&\qquad- \sum_{j=0}^{\Delta_i-1} l\Big(\hat{x}_j(k_i|k_i), \nu_j^{\#}(k_i)\Big) - F\Big(\hat{x}_{N(k_{i+1})+\Delta_i}(k_i|k_i)\Big) \\
&\qquad- \sum_{j=0}^{N(k_{i+1})-1} l\Big(\hat{x}_{j+\Delta_i}(k_i|k_i), \nu_{j+\Delta_i}^{\#}(k_i)\Big) \\
&\quad\le J_{N(k_{i+1})}(x(k_{i+1})) - V^0_{N(k_{i+1})}\big(\hat{x}_{\Delta_i}(k_i|k_i)\big) \\
&\qquad- \sum_{j=0}^{\Delta_i-1} l\Big(\hat{x}_j(k_i|k_i), \nu_j^{\#}(k_i)\Big) \\
&\quad\le - \sum_{j=0}^{\Delta_i-1} l\Big(\hat{x}_j(k_i|k_i), \nu_j^{\#}(k_i)\Big) + \alpha_b(\|\mathbb{W}\|) \\
&\quad\le -\alpha_l(\|x(k_i)\|) + \alpha_b(\|\mathbb{W}\|)
\end{aligned} \tag{37}$$

where $\alpha_b = \varphi_J \circ \gamma_{\delta_{i+1}} + \varphi_V \circ \varphi_x^{\delta_{i+1}} \circ \gamma_{\Delta_i-\delta_{i+1}}$. The feasible control input is $\nu_j^{\#}(k_i) = \kappa(\hat{x}_j(k_i|k_i))$, and $\hat{x}_j(k_i|k_i) = f(\hat{x}_{j-1}(k_i|k_i), \nu_{j-1}^{\#}(k_i), 0)$ with $\hat{x}_0(k_i|k_i) = x(k_i)$, $j = 0, 1, \ldots, N(k_{i+1}) + \Delta_i - 1$.

For scenario Fig. 4(c), the adopted controller is switched from the remote controller to the local one at time k_{i+1}. During the time interval $[k_i, k_{i+1})$, $\text{Buffer}_L(\mathbf{u}_R)$ provides the control signal. In contrast to scenario Fig. 4(b), we mainly study $F(x(k_{i+1})) - J_{N(k_i)}(x(k_i))$. Two cases, $\Delta_i \ge N(k_i)$ and $\Delta_i < N(k_i)$, are both considered. For $\Delta_i \ge N(k_i)$, we have

$$\begin{aligned}
&F(x(k_{i+1})) - J_{N(k_i)}(x(k_i)) \\
&\quad\le F(x(k_{i+1})) - V^0_{N(k_i)}\big(\hat{x}\big(k_i|k_i^-\big)\big) \\
&\qquad+V^0_{N(k_i)}\big(\hat{x}\big(k_i|k_i^-\big)\big) - J_{N(k_i)}(x(k_i)) \\
&\quad\le F(x(k_{i+1})) - V^0_{\Delta_i}\big(x\big(k_i|k_i^-\big)\big) \\
&\qquad+ \sum_{j=N(k_i)}^{\Delta_i-1} \Big(F\big(\hat{x}_{j+1}\big(k_i|k_i^-\big)\big) + l\Big(\hat{x}_j\big(k_i|k_i^-\big), v_j^{\#}(k_i)\Big) \\
&\qquad\qquad -F\big(\hat{x}_j\big(k_i|k_i^-\big)\big)\Big) + \varphi_j \circ \gamma_{\delta_i}(\|\mathbb{W}\|) \\
&\quad\le - \sum_{j=0}^{\Delta_i-1} l\Big(\hat{x}_j\big(k_i|k_i^-\big), v_j^{*}(k_i)\Big) + \alpha_{c1}(\|\mathbb{W}\|) \\
&\quad\le -\alpha_l(\|x(k_i)\|) + \alpha_c(\|\mathbb{W}\|)
\end{aligned} \tag{38}$$

where $\alpha_{c1} = \varphi_j \circ \gamma_{\delta_i} + \varphi_F \circ \gamma_{\Delta_i+\delta_i}$ and $\alpha_c = \alpha_{c1} + \varphi_l \circ \gamma_{\delta_i}$. For $\Delta_i < N(k_i)$, we directly obtain

$$F(x(k_{i+1})) - J_{N(k_i)}(x(k_i)) \le -\alpha_l(\|x(k_i)\|) + \epsilon \tag{39}$$

where $\epsilon \triangleq \max_{x\in\mathbb{X}_f} F(x)$. Combining both cases yields

$$\begin{aligned}
&F(x(k_{i+1})) - J_{N(k_i)}(x(k_i)) \\
&\quad\le -\alpha_l(\|x(k_i)\|) + \alpha_c(\|\mathbb{W}\|) + \epsilon.
\end{aligned} \tag{40}$$

For scenario Fig. 4(d), local controller is activated at time instants k_i and k_{i+1}. By similar method in scenario Fig. 4(b), we have

$$\begin{aligned}
F(x(k_{i+1})) - F(x(k_i)) &\le - \sum_{j=0}^{\Delta_i-1} l\big(\hat{x}_j(k_i|k_i), \kappa\big(\hat{x}_j(k_i|k_i)\big)\big) \\
&\quad - F\big(\hat{x}_{\Delta_i}(k_i|k_i)\big) + F(x(k_{i+1})) \\
&\le -\alpha_l(\|x(k_i)\|) + \varphi_F \circ \gamma_{\delta_i}(\|\mathbb{W}\|).
\end{aligned} \tag{41}$$

In fact, if without disturbance, the terminal set $\mathbb{X}_f$ is positive invariant for system (1) with $w(k) \equiv 0$ and $u(k) = \kappa(x(k))$, thus $k_{i+1} = k_i + 1$ as no packet loss occurs.

Incorporating the results of four scenarios, we obtains

$$\begin{aligned}
&J_{N(k_{i+1})}(x(k_{i+1})) - J_{N(k_i)}(x(k_i)) \\
&\quad\le -\alpha_l(\|x(k_i)\|) + \bar{\gamma}(\|\mathbb{W}\|) + \epsilon
\end{aligned} \tag{42}$$

where $\bar{\gamma} = \max\{\alpha_a, \alpha_b, \alpha_c, \varphi_F \circ \gamma_{\delta_i}\}$.

In order to obtain the ISpS result, we follow a similar line as in [15]. First, we prove that there exist two $\mathcal{K}_\infty$ functions α_1 and α_2 such that

$$\alpha_1(\|x\|) \le J_N(x) \le \alpha_2(\|x\|) \tag{43}$$

holds for all $x \in \mathcal{X}^{\text{MPC}}$ and $0 \le N \le N_{\max}$. We can directly let $\alpha_1 = \min\{\alpha_l, \alpha_F\}$ from Assumption 5, then we only need to find α_2. Notice that when $x \in \mathbb{X}_f$, $N = 0$ and $J_N(x) = F(x) \le \varphi_F(\|x\|)$. To obtain the upper bound of $J_N(x)$ in $\mathcal{X}^{\text{MPC}}$, we use the arguments stated in [35, Lemma 4]. There exists $\bar{J}$ such that $J_N(x) \le \bar{J}$ for all $x \in \mathcal{X}^{\text{MPC}}$ due of the compactness of $\mathcal{X}^{\text{MPC}}$ and $\mathbb{U}$. Define $\mathcal{B}_r = \{x \in \mathbb{R}^n : \|x\| \le r\}$ as a ball such that $\mathcal{B}_r \subset \mathbb{X}_f$. Let $\xi = \max\{1, \bar{J}/\varphi_F(r)\}$. Observed that if $x \notin \mathbb{X}_f$, we have $\varphi_F(\|x\|) > \varphi_F(r)$ and

$$J_N(x) \le \bar{J} \le \frac{\bar{J}}{\varphi_F(r)}\varphi_F(\|x\|).$$

That is, we can set $\alpha_2(\|x\|) = \xi\varphi_F(\|x\|)$ for $x \in \mathcal{X}^{\text{MPC}}$.

Next, it should be noted that Assumption 4 guarantees that $x(k_i) \in \mathcal{X}^{\text{MPC}}$ for all k_i. Incorporating (42) and (43), and based on [29, Th. 1], we obtain that there exists a $\mathcal{KL}$ function $\hat{\beta}$ and $\mathcal{K}$ functions $\hat{\gamma}$ and $\hat{\rho}$ which gives for all $k_i \in K$

$$\|x(k_i)\| \le \hat{\beta}(\|x(k_0)\|, k_i - k_0) + \hat{\gamma}(\|\mathbb{W}\|) + \hat{\rho}(\epsilon).$$

According to the inequality (43), we can claim that there exist $\tilde{\beta} \in \mathcal{KL}$, $\tilde{\gamma} \in \mathcal{K}$, and $\tilde{\rho} \in \mathcal{K}$ such that

$$J_{N(k_i)}(x(k_i)) \le \tilde{\beta}\big(J_{N(k_0)}(x(k_0)), k_i - k_0\big) + \tilde{\gamma}(\|\mathbb{W}\|) + \tilde{\rho}(\epsilon).$$

In the remaining part of the proof, we explore the upper bound of $\|x(k)\|$ for $k > k_0$ and $k \notin K$. By employing (43), we obtain the lower bound of the optimal value function

$$\begin{aligned} J_{N_{\max}}(x(k_i)) &\ge \sum_{j=0}^{N_{\max}-1} \alpha_l\big(\|\hat{x}_j(k_i|k_i)\|\big) \\ &\ge \alpha_l\left(\frac{1}{N_{\max}} \sum_{j=0}^{N_{\max}-1} \|\hat{x}_j(k_i|k_i)\|\right) \qquad (44) \end{aligned}$$

where the second inequality holds because of the inequality introduced and proved in [36]: $(1/n)(\alpha_l(a_1) + \cdots + \alpha_l(a_n)) \le \alpha_l(a_1 + \cdots + a_n) \le \alpha_l(na_1) + \cdots + \alpha_l(na_n)$ with $\alpha_l \in \mathcal{K}$.

By applying (44) and Assumption 5, we obtain

$$\begin{aligned} N_{\max}\alpha_l^{-1}\big(J_{N_{\max}}(x(k_i))\big) &\ge \|x(k_i)\| + \sum_{j=1}^{N_{\max}-1} \|\hat{x}_j(k_i|k_i)\| \\ &\ge \|x(k_i)\| + \sum_{j=1}^{N_{\max}-1} \big(\|x(k_i+j)\| - \gamma_j(\|\mathbb{W}\|)\big) \\ &\ge -\sum_{j=0}^{N_{\max}-1} \gamma_j(\|\mathbb{W}\|) + \sum_{j=0}^{\Delta_i-1} \|x(k_i+j)\| \end{aligned} \qquad (45)$$

where $x(k_i + j) = f^j(x(k_i), \boldsymbol{\phi}_{t_s(k_i)}([0 : j-1]), \boldsymbol{w}([k_i : k_i + j - 1]))$ is the actual state, and $\boldsymbol{\phi}_{t_s(k_i)}$ is defined in (20). The last inequality holds because of $k_{i+1} - k_i \le \bar{N}_d \le N_{\max}$.

Since $V^0_{N_{\max}}(x(k_i|k_i^-)) \le V^0_{N(k_i)}(x(k_i|k_i^-))$ [28], we have

$$\begin{aligned} J_{N_{\max}}(x(k_i)) - J_{N(k_i)}(x(k_i)) &\le J_{N_{\max}}(x(k_i)) - V^0_{N_{\max}}\big(x\big(k_i|k_i^-\big)\big) \\ &\quad + V^0_{N(k_i)}\big(x\big(k_i|k_i^-\big)\big) - J_{N(k_i)}(x(k_i)) \\ &\le 2\varphi_J \circ \gamma_{\delta_i}(\|\mathbb{W}\|). \qquad (46) \end{aligned}$$

Combining the above results gives

$$\begin{aligned} \|x(k)\| &\le \sum_{j=0}^{\Delta_i-1} \|x(k_i + j)\| \\ &\le N_{\max}\alpha_l^{-1}\big(J_{N_{\max}}(x(k_i))\big) + \sum_{j=0}^{N_{\max}-1} \gamma_j(\|\mathbb{W}\|) \\ &\le N_{\max}\alpha_l^{-1}\big(J_{N(k_i)}(x(k_i)) + 2\varphi_J \circ \gamma_{\delta_i}(\|\mathbb{W}\|)\big) \\ &\quad + \sum_{j=0}^{N_{\max}-1} \gamma_j(\|\mathbb{W}\|) \\ &\le N_{\max}\alpha_l^{-1}\big(2J_{N(k_i)}(x(k_i))\big) + \check{\gamma}(\|\mathbb{W}\|) \\ &\le N_{\max}\alpha_l^{-1}\Big(2\Big(\tilde{\beta}\big(J_{N(k_0)}(x(k_0)), k_i - k_0\big) + \tilde{\gamma}(\|\mathbb{W}\|) \\ &\qquad + \tilde{\rho}(\epsilon)\Big)\Big) + \check{\gamma}(\|\mathbb{W}\|) \\ &\le \beta(\|x(k_0)\|, k - k_0) + \gamma(\|\mathbb{W}\|) + c \qquad (47) \end{aligned}$$

for all $k \in \{k_i, \ldots, k_{i+1} - 1\}$, where

$$\begin{aligned} \check{\gamma}(\|\mathbb{W}\|) &= N_{\max}\alpha_l^{-1}\big(4\varphi_J \circ \gamma_{\delta_i}(\|\mathbb{W}\|)\big) \\ \gamma(\|\mathbb{W}\|) &= \check{\gamma}(\|\mathbb{W}\|) + N_{\max}\alpha_l^{-1}(6\tilde{\gamma}(\|\mathbb{W}\|)) \\ \beta(\|x(k_0)\|, k - k_0) &= N_{\max}\alpha_l^{-1}\Big(6\tilde{\beta}(\alpha_2(\|x(k_0)\|), k_i - k_0)\Big) \\ c &= N_{\max}\alpha_l^{-1}(6\tilde{\rho}(\epsilon)). \end{aligned}$$

Since $k_0 = 0$, the proof is then completed. ■

Notice that the constant c appears in (31) because the case of $\Delta_i < N(k_i)$ may occur in scenario Fig. 4(c). In fact, such case can be avoided by making use of the knowledge of the prediction horizon $N(t_{r,c}(k))$. Let $N_j(k)$ be the estimated prediction horizon at the judger side, then the update of $N_j(k)$ is $N_j(k) \triangleq \max\{0, N(t_{r,c}(k)) - (k - t_{r,c}(k))\}$. Based on $N_j(k)$, the selection rule of judger becomes the following rule.

Rule 2: If $x(k) \in \mathbb{X}_f$ and $N_j(k) = 0$, the local controller is selected; otherwise, the remote one is selected.

Then, for scenario Fig. 4(c), we have $t_{r,c}(k) = k_i$ and $N(k_i)$ can be obtained from the received control sequence. If $N_j(k) > 0$ $(N(k_i) > k - k_i)$, the local controller will never be selected, which avoids the case of $N(k_i) > k_{i+1} - k_i$, where k_{i+1} here refers to the time when the local controller is selected.

Corollary 1: Consider the system in (1) with the designed control scheme. If Rule 2 is adopted, Assumptions 1–5 hold, and $x_0 \in \mathcal{X}^{\text{MPC}}$, then the overall system is ISS, i.e., there exists a $\mathcal{KL}$ function β and a $\mathcal{K}$ function γ such that

$$\|x(k)\| \le \beta(\|x_0\|, k) + \gamma(\|\mathbb{W}\|) \quad \forall k \ge 0. \qquad (48)$$

Proof: Since the case of $\Delta_i < N(k_i)$ in scenario Fig. 4(c) is avoided, Expression (42) becomes

$$J_{N(k_{i+1})}(x(k_{i+1})) - J_{N(k_i)}(x(k_i)) \le -\alpha_l(\|x(k_i)\|) + \bar{\gamma}(\|\mathbb{W}\|). \qquad (49)$$

Then, following the similar lines of Theorem 1, the ISS of the system can be established. ■

Remark 3: When considering the disturbance-free system, i.e., $w(k) \equiv 0$ in system (1), the term $\gamma(\|\mathbb{W}\|)$ in (31) vanishes while the constant c remains. It seems that the asymptotical stability result cannot be established accordingly. Indeed, this result can be derived from another aspect. For any initial state $x_0 \in \mathbb{X}_{N_{\max}}$ with $\mathbb{X}_{N_{\max}}$ being defined in Assumption 4, if $x_0 \in \mathbb{X}_f$, then the asymptotical stability can be easily obtained on the basis of Assumption 3. For $x_0 \in \mathbb{X}_{N_{\max}}/\mathbb{X}_f$, we only need to prove that the state can be steered into $\mathbb{X}_f$ in finite steps. It can be proved by contradiction. Suppose that the state will not enter $\mathbb{X}_f$ forever, then based on (36), there exists a positive integer j such that $J_{N(k_j)}(x(k_j)) < 0$, which contradicts with the positive definiteness of $J_{N(k)}(x(k))$.

V. Numerical Example

Consider a networked cascaded two-tanks system with the overall configuration being depicted in Fig. 5. The operations of all these components (including the judger, local controller,

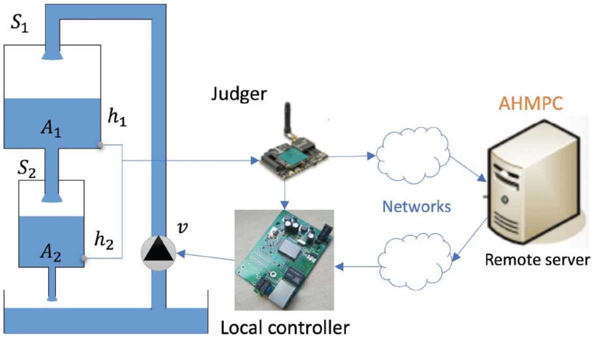


Fig. 5. Configuration of the networked dual-mode AHMPC scheme for a cascaded two-tanks system.

TABLE I
PARAMETERS AND NOMINAL OPERATING CONDITIONS

Symbols	Description	Value
S_1	The section of tank 1	$2500\,\mathrm{cm}^2$
S_2	The section of tank 2	$1600\,\mathrm{cm}^2$
A_1	The section of outflow orifice in tank 1	$9\,\mathrm{cm}^2$
A_2	The section of outflow orifice in tank 2	$4\,\mathrm{cm}^2$
g	Gravity constant	$980\,\mathrm{cm}\,/s^2$
ρ	Water density	$0.001\,\mathrm{Kg}\,/\,\mathrm{cm}^3$

and remote controller) as well as the data exchange among them follow the rules introduced in Section III. The plant is modeled by the following nonlinear system [37]:

$$\dot{h}_1 = -\frac{A_1}{S_1}\sqrt{2gh_1} + \frac{1}{\rho S_1}v + w_1$$
$$\dot{h}_2 = \frac{A_1}{S_2}\sqrt{2gh_1} - \frac{A_2}{S_2}\sqrt{2gh_2} + w_2 \tag{50}$$

where h_1 and h_2 are the states representing the water levels of both tanks, v is the input denoting the water flow supplied to tank 1, w_1 and w_2 are the external disturbance satisfying $|w_i| \le 0.05$, $i = 1, 2$, and the other related symbols are given in Table I. Our aim is to design the control signal such that the water level of tank 2 reaches to its set-point $h_2^e = 105$ cm and meanwhile the control and state constraints are satisfied

$$1 \le h_1 \le 55,\ \ 10 \le h_2 \le 200,\ \ 0 \le v \le 4.$$

Note that the stationary operation conditions of the system are $h_1^e = 20.7407$, $h_2^e = 105$, and $v^e = 1.8146$, then the discrete-time nonlinear system (1) is obtained by letting $x_1 = h_1 - h_1^e$, $x_2 = h_2 - h_2^e$, $u = v - v^e$, and then using the forward-Euler discretized method with sampling interval $T_s = 5$ s.

For the MPC settings, the stage cost and terminal cost are chosen as $l(x, u) = x^{\mathrm{T}}Qx + u^{\mathrm{T}}Ru$ and $F(x) = x^{\mathrm{T}}Px$, where $Q = \mathrm{diag}\{0.3, 1.0\}$ and $R = 0.1$. By adopting the method proposed in [38], the auxiliary control law (local controller) is designed based on LQR

$$\kappa(x) = \begin{bmatrix}-0.5218 & -0.6551\end{bmatrix}x$$

the positive matrix P is determined as

$$P = \begin{bmatrix}0.6328 & 0.7765\\ 0.7765 & 6.5545\end{bmatrix}$$

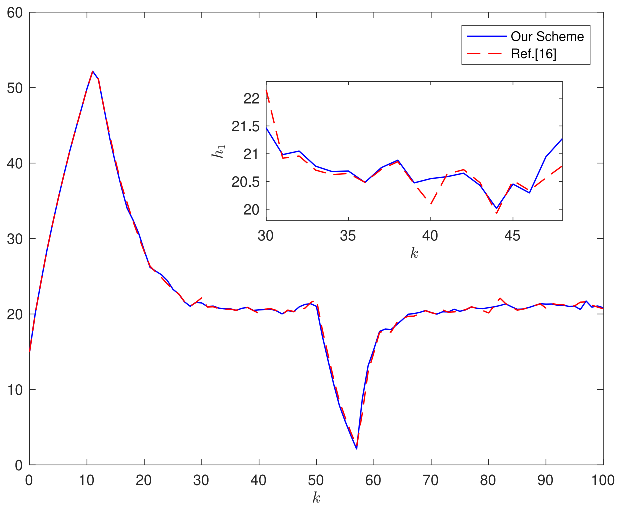


Fig. 6. Height of tank 1.

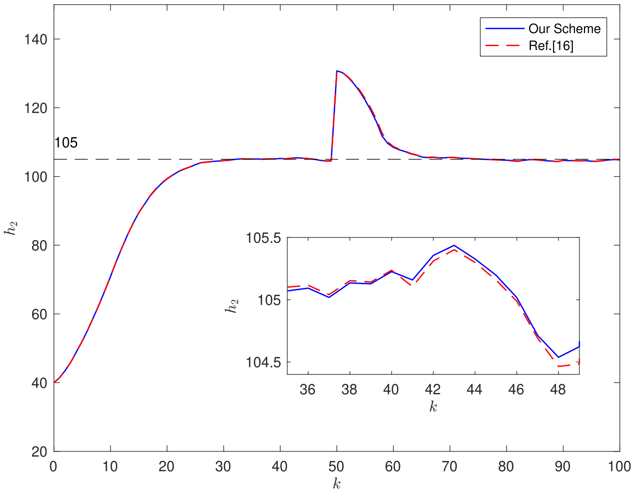


Fig. 7. Height of tank 2.

and the terminal set $\mathbb{X}_f$ is chosen as

$$\mathbb{X}_f = \left\{x : x^{\mathrm{T}}Px \le 1.5\right\}.$$

The maximum prediction horizon is $N_{\max} = 30$, which is also the prediction horizon of the fixed-horizon MPC. Besides, we set $H = 2$ and $M = 5$ for the PHE (Algorithm 2).

The initial water levels of the two tanks are $h_1(0) = 15$ and $h_2(0) = 20$. In order to show the robustness of the proposed scheme, an abrupt change of water level of tank 2 occurs at $k = 50$. We assume the maximum length of the consecutive packet losses of two networks is $N_s = 3$ and $N_c = 4$, then the control packet length and the lengths of the associated buffers are $\bar{N}_d = 7$. In the simulation, the FHOCP is solved by using the MATLAB function `fmincon` on an Intel i5-6500 3.2-GHz CPU.

The efficiency of the proposed control scheme is verified by comparing with the networked fixed-horizon MPC in [16]. The simulation results are shown in Figs. 6–10. Specifically, the height levels of the two tanks are shown in Figs. 6 and 7, respectively, and the input flow of tank 1 is depicted in Fig. 8. It can be observed that the control objective can be achieved

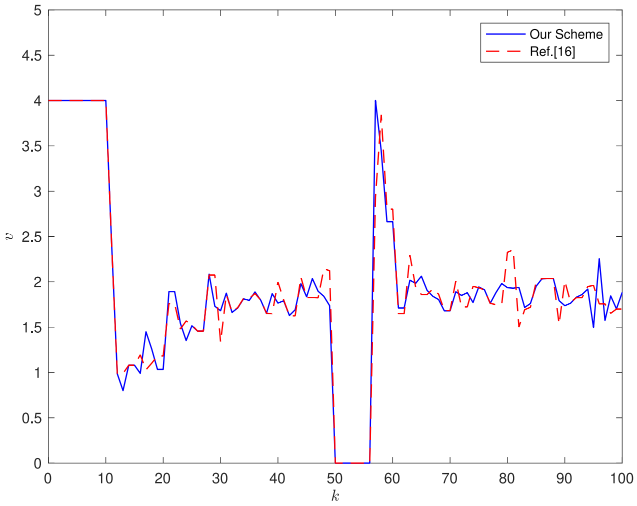


Fig. 8. Control input.

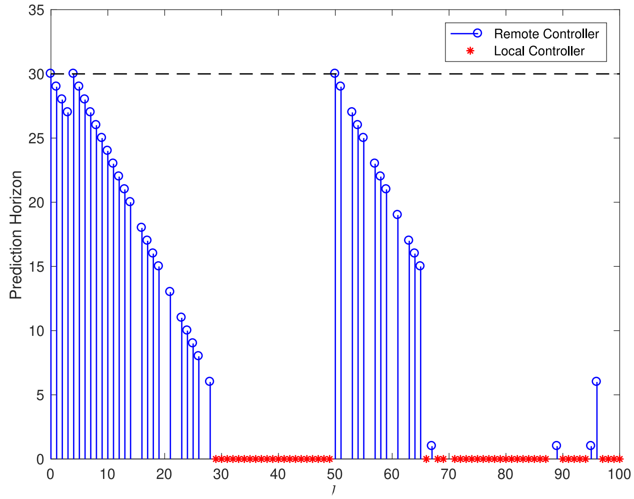


Fig. 9. Time-varying prediction horizon.

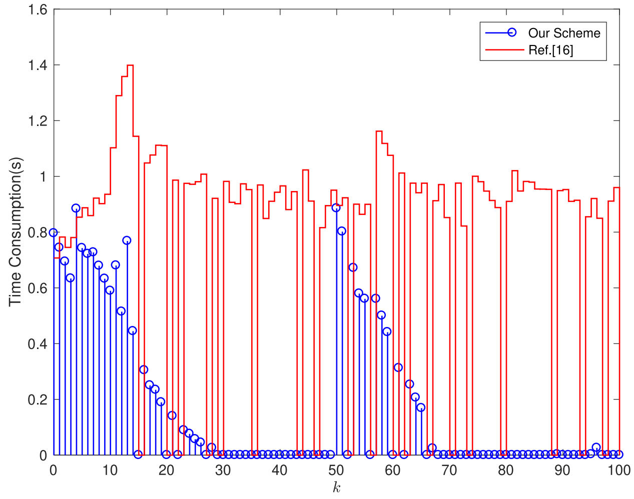


Fig. 10. Computation time of FHOPC.

for both control schemes while the state and control constraints are satisfied. Fig. 9 illustrates the time-varying prediction horizon obtained by the PHE algorithm (Algorithm 2), and Fig. 10 presents the computing time at each time step to evaluate the computational complexity for both two control schemes. Table II shows the average computing time and control cost (defined as $J \triangleq \sum_{k=0}^{T-1} x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k$, T is the operating time) for the non-networked MPC (without packet losses) [28], networked MPC [16], and our control scheme, respectively. For the networked MPC, a lower average computing time and a higher control cost are obtained when compared with the non-networked MPC. This is because of the packet losses, the MPC is not performed at each step and new control packet may be lost. For our control scheme, i.e., the networked dual-mode AHMPC, the prediction horizon tends to decrease as the plant state approaches the terminal set and equals to 0 when the state is in the terminal set. Therefore, the average computing time is significantly reduced (about a 77.76% decrease). Due to the packet-based control compensation scheme, the degradation of control performance is not serious compared with the non-networked MPC [28]. Besides, the control cost is comparable to that of the networked MPC [16] because the time-varying prediction horizon degrades the control performance on the one hand while the local controller without packet losses may reduce the control cost on the other.

TABLE II
AVERAGE COMPUTATIONAL TIME AND CONTROL COST

	Average solving time	Control cost
NMPC without dropouts [28]	1.0488s	3.9352×10^4
Networked MPC [16]	0.7860s	3.9858×10^4
Our scheme	0.1748s	3.9761×10^4

VI. CONCLUSION

We have studied the stabilization issue for a class of perturbed discrete-time nonlinear system with MPC-based control scheme. To mitigate the demand of computation resource and reduce the time for solving the FHOCP, a networked dual-mode AHMPC scheme has been proposed. Specifically, the control architecture includes a local state-feedback controller, an AHMPC-based remote controller that suffers from packet losses, and a judger determining the switchings between the two controllers. To prove the stability of the proposed control scheme, we have constructed a new Lyapunov function. By analyzing the decrease of the Lyapunov function, and specifying the difference between the Lyapunov function and the optimal MPC value function, the ISpS of the overall system has been established. Finally, the effectiveness of our proposed control scheme has been verified by a numerical example.

REFERENCES

[1] Y.-B. Zhao, G.-P. Liu, Y. Kang, and L. Yu, *Packet-Based Control for Networked Control Systems*. Singapore: Springer, 2018.

[2] P. Li, Y. Kang, Y.-B. Zhao, J. Qin, and W. Song, "Discretization-based stabilization for a class of switched linear systems with communication delays," *ISA Trans.*, vol. 80, pp. 1–11, Sep. 2018.

[3] K. Liu, E. Fridman, and K. H. Johansson, "Networked control with stochastic scheduling," *IEEE Trans. Autom. Control*, vol. 60, no. 11, pp. 3071–3076, Nov. 2015.

[4] S. Al-Areqi, D. Görges, and S. Liu, "Event-based networked control and scheduling codesign with guaranteed performance," *Automatica*, vol. 57, pp. 128–134, Jul. 2015.

[5] P. Park, S. C. Ergen, C. Fischione, C. Lu, and K. H. Johansson, "Wireless network design for control systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 978–1013, 2nd Quart., 2018.

[6] L. Zhang, H. Gao, and O. Kaynak, "Network-induced constraints in networked control systems—A survey," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 403–416, Feb. 2013.

[7] X.-M. Zhang, Q.-L. Han, and X. Yu, "Survey on recent advances in networked control systems," *IEEE Trans. Ind. Informat.*, vol. 12, no. 5, pp. 1740–1752, Oct. 2016.

[8] R. Zhang and F. Gao, "Two-dimensional iterative learning model predictive control for batch processes: A new state space model compensation approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, doi: 10.1109/TSMC.2018.2883754.

[9] F. Ke, Z. Li, H. Xiao, and X. Zhang, "Visual servoing of constrained mobile robots based on model predictive control," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1428–1438, Jul. 2016.

[10] B.-L. Ye *et al.*, "A survey of model predictive control methods for traffic signal control," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 3, pp. 623–640, May 2019.

[11] M. Yue, C. An, and Z. Li, "Constrained adaptive robust trajectory tracking for WIP vehicles using model predictive control and extended state observer," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 5, pp. 733–742, May 2018.

[12] M. B. Saltık, L. Özkan, J. H. A. Ludlage, S. Weiland, and P. M. J. Van den Hof, "An outlook on robust model predictive control algorithms: Reflections on performance and computational aspects," *J. Process Control*, vol. 61, pp. 77–102, Jan. 2018.

[13] T. Wang, J. Qiu, H. Gao, and C. Wang, "Network-based fuzzy control for nonlinear industrial processes with predictive compensation strategy," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 2137–2147, Aug. 2017.

[14] D. M. de la Peña and P. D. Christofides, "Lyapunov-based model predictive control of nonlinear systems subject to data losses," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2076–2089, Oct. 2008.

[15] D. E. Quevedo and D. Nešić, "Input-to-state stability of packetized predictive control over unreliable networks affected by packet-dropouts," *IEEE Trans. Autom. Control*, vol. 56, no. 2, pp. 370–375, Feb. 2011.

[16] H. Li and Y. Shi, "Network-based predictive control for constrained nonlinear systems with two-channel packet dropouts," *IEEE Trans. Ind. Electron.*, vol. 61, no. 3, pp. 1574–1582, Mar. 2014.

[17] P. Li, Y. Kang, Y.-B. Zhao, and Z. Yuan, "Packet-based model predictive control for networked control systems with random packet losses," in *Proc. 57th IEEE Conf. Decis. Control*, Miami Beach, FL, USA, 2018, pp. 3457–3462.

[18] D. E. Quevedo and D. Nešić, "Robust stability of packetized predictive control of nonlinear systems with disturbances and Markovian packet losses," *Automatica*, vol. 48, no. 8, pp. 1803–1811, 2012.

[19] G. Pin and T. Parisini, "Networked predictive control of uncertain constrained nonlinear systems: Recursive feasibility and input-to-state stability analysis," *IEEE Trans. Autom. Control*, vol. 56, no. 1, pp. 72–87, Jan. 2011.

[20] H. Li and S. Yang, "Networked min–max model predictive control of constrained nonlinear systems with delays and packet dropouts," *Int. J. Control*, vol. 86, no. 4, pp. 610–624, Oct. 2013.

[21] P. O. M. Scokaert and D. Q. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Trans. Autom. Control*, vol. 43, no. 8, pp. 1136–1142, Aug. 1998.

[22] Z. Sun, L. Dai, K. Liu, D. V. Dimarogonas, and Y. Xia, "Robust self-triggered MPC with adaptive prediction horizon for perturbed nonlinear systems," *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4780–4787, Nov. 2019, doi: 10.1109/TAC.2019.2905223.

[23] M. Kögel and R. Findeisen, "Stability of NMPC with cyclic horizons," in *Proc. 9th IFAC Symp. Nonlinear Control Syst.*, Toulouse, France, 2013, pp. 809–814.

[24] A. J. Krener, "Adaptive horizon model predictive control," in *Proc. 2nd IFAC Conf. Model. Identification Control Nonlinear Syst.*, Guadalajara, Mexico, 2018, pp. 31–36.

[25] D. W. Griffith, L. T. Biegler, and S. C. Patwardhan, "Robustly stable adaptive horizon nonlinear model predictive control," *J. Process Control*, vol. 70, pp. 109–122, Oct. 2018.

[26] X. Liang and J. Xu, "Control for networked control systems with remote and local controllers over unreliable communication channel," *Automatica*, vol. 98, pp. 86–94, Dec. 2018.

[27] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.

[28] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Madison, WI, USA: Nob Hill, 2017.

[29] D. Limón, T. Alamo, F. Salas, and E. F. Camacho, "Input to state stability of min–max MPC controllers for nonlinear systems with bounded uncertainties," *Automatica*, vol. 42, no. 5, pp. 797–803, 2006.

[30] M. Junaid, M. Mufti, and M. U. Ilyas, "Vulnerabilities of IEEE 802.11i wireless LAN CCMP protocol," *Trans. Eng. Comput. Technol.*, vol. 11, pp. 228–233, Feb. 2006.

[31] C. Mucci *et al.*, "Implementation of parallel LFSR-based applications on an adaptive DSP featuring a pipelined configurable gate array," in *Proc. ACM Conf. Design Automa. Test Europe*, 2008, pp. 1444–1449.

[32] H. Michalska and D. Q. Mayne, "Robust receding horizon control of constrained nonlinear systems," *IEEE Trans. Autom. Control*, vol. 38, no. 11, pp. 1623–1633, Nov. 1993.

[33] D. L. Marruedo, T. Alamo, and E. F. Camacho, "Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties," in *Proc. IEEE Conf. Decis. Control*, Las Vegas, NV, USA, 2002, pp. 4619–4624.

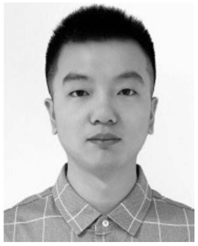
[34] L. Magni, D. M. Raimondo, and R. Scattolini, "Regional input-to-state stability for nonlinear model predictive control," *IEEE Trans. Autom. Control*, vol. 51, no. 9, pp. 1548–1553, Sep. 2006.

[35] M. Rubagotti, D. M. Raimondo, A. Ferrara, and L. Magni, "Robust model predictive control with integral sliding mode in continuous-time sampled-data nonlinear systems," *IEEE Trans. Autom. Control*, vol. 56, no. 3, pp. 556–570, Mar. 2011.

[36] J. B. Rawlings and L. Ji, "Optimization-based state estimation: Current status and some new results," *J. Process Control*, vol. 22, no. 8, pp. 1439–1444, 2012.

[37] A. Casavola, D. Famularo, and G. Franze, "Norm-bounded robust MPC strategies for constrained control of nonlinear systems," *IEE Proc. Control Theory Appl.*, vol. 152, no. 3, pp. 285–295, May 2005.

[38] C. Rajhans, S. C. Patwardhan, and H. Pillai, "Discrete time formulation of quasi infinite horizon nonlinear model predictive control scheme with guaranteed stability," *IFAC PapersOnLine*, vol. 50, no. 1, pp. 7181–7186, 2017.



Pengfei Li received the B.S. degree in automatic control from Hangzhou Dianzi University, Hangzhou, China, in 2014. He is currently pursuing the Ph.D. degree with the Department of Automation, University of Science and Technology of China, Hefei, China.

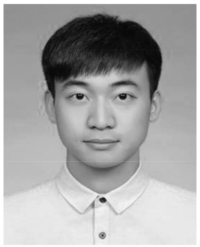
His research interests include networked control systems, model predictive control, and learning-based control.

Yu Kang (Senior Member, IEEE) received the Ph.D. degree in control theory and control engineering from the University of Science and Technology of China, Hefei, China, in 2005.

From 2005 to 2007, he was a Postdoctoral Fellow with the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China. He is currently a Professor with the State Key Laboratory of Fire Science, Department of Automation, Institute of Advanced Technology, University of Science and Technology of China, and the Key Laboratory of Technology in GeoSpatial Information Processing and Application System, Chinese Academy of Sciences, Beijing, China. His current research interests include monitoring of vehicle emissions, adaptive/robust control, variable structure control, mobile manipulator, and Markovian jump systems.

Yun-Bo Zhao (Senior Member, IEEE) received the B.Sc. degree in mathematics from Shandong University, Jinan, China, in 2003, the M.Sc. degree in systems sciences from the Key Laboratory of Systems and Control, Chinese Academy of Sciences, Beijing, China, in 2007, and the Ph.D. degree in control theory and engineering from the University of Glamorgan, Pontypridd, U.K., in 2008.

He is currently a Professor with the Zhejiang University of Technology, Hangzhou, China. He had worked with INRIA, Rocquencourt, France, the University of Glasgow, Glasgow, U.K., and Imperial College London, London, U.K., on various research projects. His main research interests include networked control systems, AI enabled control, human–machine integrated intelligence, and systems biology.



Tao Wang received the B.S. degree in automatic control from Harbin Engineering University, Harbin, China, in 2018. He is currently pursuing the M.S. degree with the Department of Automation, University of Science and Technology of China, Hefei, China.

His research interests include model predictive control and approximation-based control.