Quasi-Decentralized Networked Process Control Using an Adaptive Communication Policy

Yulei Sun and Nael H. El-Farra†
Department of Chemical Engineering & Materials Science
University of California, Davis, CA 95616 USA

Abstract—This work presents a model-based quasi-decentralized networked control structure with a state-dependent communication policy for plants with interconnected units that exchange measurements over a shared, resource-constrained communication network. The objective is to find a strategy for establishing and terminating communication between the local control systems in a way that minimizes network resource utilization without jeopardizing closed-loop stability. To this end, a Lyapunov-based controller that enforces closed-loop stability in the absence of communication suspensions is initially designed. A set of dynamic models are included within each local control system to provide estimates of the states of the neighboring units when measurements are not transmitted through the network. To determine when communication must be re-established, the evolution of each Lyapunov function is monitored locally within each unit such that if it begins to breach a certain stability threshold at any time, the sensor suites of the neighboring units are prompted to send their data over the network to update their corresponding models. Communication is then suspended for as long as the Lyapunov function continues to decay. The underlying idea is to use the Lyapunov stability constraint for each unit as the basis for switching on or off the communication between a given unit and its neighbors. This formulation, which leads to a state-dependent time-varying communication rate, allows the plant to respond adaptively to changes in operating conditions. Finally, the results are illustrated through an application to a chemical plant example.

I. INTRODUCTION

Chemical plants are large-scale dynamical systems that consist of a large number of distributed units which are tightly interconnected through mass, energy and information flows and recycle. Traditionally, the controller synthesis problem for such plants has been addressed within either the centralized or decentralized control frameworks. Both approaches have been the subject of numerous research studies aimed at understanding their advantages and limitations, as well as the development of strategies to overcome some of those limitations (e.g., see [1], [2], [3], [4]). Other important contributions on this problem include the development of plant-wide control strategies based on passivity theory and concepts from thermodynamics ([5]), the development of agent-based systems to control spatially-distributed reactor networks ([6]), and the analysis and control of integrated process networks using time-scale decomposition and singular perturbations ([7]).

An approach that provides a compromise between the complexity of traditional centralized control schemes, on the one hand, and the performance limitations of decentralized control approaches, on the other, is quasi-decentralized control, which refers to a distributed control strategy in which most signals used for control are collected and processed locally, while certain signals are transferred between the local units and controllers to adequately account for the interactions and minimize the propagation of process upsets. A key consideration in this strategy is to enforce the desired closed-loop stability and performance objectives of the plant with minimal cross communication between the component subsystems. This is an appealing objective particularly when the communication medium is a resource-constrained wireless sensor network (WSN) where conserving network resources is key to prolonging the service life of the network and minimizing frequent battery replacements in harsh plant environments. The integration of WSNs into process control systems, and the opportunities and challenges that this creates, have been the focus of significant attention in both the industrial and academic circles (e.g., see [8], [9], [10]; for additional results and references on the design of networked control systems, the reader may refer to [11], [12], [13], [14], [15]).

An effort to address this problem was initiated in [16] where a quasi-decentralized networked control architecture that keeps the rate of communication between the plant units to a minimum without jeopardizing closed-loop stability was developed. The main idea – inspired by the results in [17] on model-based networked control – was to embed in the local control system of each unit a set of dynamic models that provide estimates of the states of the neighboring units when state information is not transmitted over the network. The problem of incomplete state measurements was subsequently addressed in [18], and a quasi-decentralized networked control architecture for nonlinear plants was developed in [19]. A key feature of the communication logic used in both cases is that it is static in the sense that the communication rate is constant and can be computed offline prior to plant operation. A constant communication rate, however, may not always be the best choice, especially in cases when plant operations are subject to unpredictable and time-varying external disturbances which, if unaccounted for, can degrade the networked closed-loop performance and may even lead to instability.

One approach to deal with this problem is to use ro-
bust control techniques to obtain an upper bound on the minimum stabilizing communication rate if an upper bound on the size of the disturbances is available. However, the resulting bound is typically conservative and may lead to an unnecessary increase in the utilization of network resources, especially when the external disturbances are not persistent. Furthermore, in cases of unpredictable disturbances, information about the size of the disturbances may not be available or easily obtained.

Motivated by these considerations, we present in this work a model-based quasi-decentralized networked control structure with a state-dependent communication policy for plants with interconnected units that exchange information over a shared, resource-constrained communication network. The objective is to find a strategy for establishing and terminating communication between the local control systems in a way that minimizes the utilization of network resources without jeopardizing closed-loop stability. The rest of the paper is organized as follows: Following some preliminaries in Section II, we initially synthesize in Section III for each unit a Lyapunov-based controller that enforces closed-loop stability in the absence of communication suspensions. To reduce network resource utilization, we include within each local control system a set of dynamic models that provide estimates of the states of its neighbors in the plant when communication is suspended and measurements are not transmitted through the network. An adaptive communication policy in which a Lyapunov stability constraint is used as the basis for switching on or off the communication between a given unit and its neighbors is then devised. Finally, the results are demonstrated through an application to a chemical plant example.

II. PRELIMINARIES AND PROBLEM FORMULATION

We consider a large-scale distributed plant composed of \( n \) interconnected processing units, each of which is described by a continuous-time uncertain nonlinear system, and represented by the following state-space description:

\[
\begin{align*}
\dot{x}_1 &= f_1(x) + G_1(x)u_1 + W_1(x)\theta_1(t) \\
\dot{x}_2 &= f_2(x) + G_2(x)u_2 + W_2(x)\theta_2(t) \\
&\vdots \\
\dot{x}_n &= f_n(x) + G_n(x)u_n + W_n(x)\theta_n(t)
\end{align*}
\]

where \( x_i := [x_i^{(1)} \ x_i^{(2)} \ \ldots \ x_i^{(\rho_i)}]^{T} \in \mathbb{R}^{\rho_i} \) denotes the vector of process state variables associated with the \( i \)-th processing unit, \( x' \) denotes the transpose of a vector \( x \), \( x := [x_1' \ x_2' \ \ldots \ x_n']^{T} \in \mathbb{R}^{\rho} \) denotes the vector of manipulated inputs associated with the \( i \)-th processing unit, \( \theta_i := [\theta_i^{(1)} \ \theta_i^{(2)} \ \ldots \ \theta_i^{(\rho_i)}]^{T} \in \mathbb{R}^{\rho_i} \) denotes the vector of uncertain (possibly time-varying), but bounded, variables which takes values in a nonempty compact convex subset of \( \mathbb{R}^{\rho_i} \), and satisfies \( \| \theta_i \| \leq \theta_{bi} \), for \( i = 1, \ldots, n \), where \( \theta_{bi} \) is a positive real number and \( \| \cdot \| \) denotes the standard Euclidean norm. The uncertain variables may describe time-varying parametric uncertainty and/or exogenous disturbances. The functions \( f_i(\cdot) \), \( G_i(\cdot) \), and \( W_i(\cdot) \) are sufficiently smooth nonlinear functions. Without loss of generality, it is assumed that the origin is an equilibrium point of the nominal uncontrolled plant (i.e., \( f_i(0) = 0 \) for \( i = 1, \ldots, n \)). Note from Eq.1 that each processing unit can in general be connected to all the other units in the plant. The overall objective is to design a distributed, networked control strategy that robustly stabilizes the individual units (and the overall plant) at or near the origin, and accounts simultaneously for the constrained resources of the plant-wide communication network.

III. ROBUST QUASI-DECENTRALIZED NETWORKED CONTROL STRUCTURE

A. Robust feedback controller synthesis

To realize the desired robust quasi-decentralized networked control structure, the first step is to synthesize for each unit a feedback controller that enforces robust closed-loop stability and an arbitrary degree of asymptotic attenuation of the effect of the uncertainty on the closed-loop system in the absence of communication suspension (i.e., when the sensors of each unit transmit their data continuously to the control systems of the other plant units). Depending on the particular structure of the plant, a number of nonlinear controller synthesis techniques can be used to design the desired controllers. Examples include Lyapunov-based control methods, geometric control approaches as well as optimization-based control methods. In this work, we consider a Lyapunov-based controller synthesis method as an example, but in general other controller design methods can be used. Specifically, using a robust control Lyapunov function \( V_i(x_i) \) for the \( i \)-th unit, the following robust nonlinear controller can be designed suing the results in [20] (see also [21], [22]):

\[
u_i = k_i(x, \theta_{bi}, \rho_i, \chi_i, \phi_i), \quad i = 1, 2, \ldots, n\]

\[
\begin{align*}
L_{f_i}V_i + \frac{\sqrt{(L_{f_i}V_i)^{2} + \| (L_{G_i}V_i)^{\prime} \|^{4}}}{\| (L_{G_i}V_i)^{\prime} \|^{2}} (L_{G_i}V_i)^{\prime}(x)
\end{align*}
\]

(2)

when \( \| (L_{G_i}V_i)^{\prime} \| \neq 0 \), and \( u_i = 0 \) when \( \| (L_{G_i}V_i)^{\prime} \| = 0 \), where

\[
L_{f_i}V_i = L_{f_i}V_i + \rho_i \| x_i \| + \chi_i \| (L_{W_i}V_i)^{\prime} \| \theta_{bi}
\]

(3)

\[
L_{f_i}V_i = L_{f_i}V_i + (L_{f_i}V_i - L_{f_i}V_i) \left( \frac{\| x_i \|}{\| x_i \| + \phi_i} \right)
\]

(4)

and \( L_{f_i}V_i = (\partial V_i/\partial x_i)f_i(x) \), \( L_{G_i}V_i = [L_{g_{i,1}}V_i \ \ldots \ L_{g_{i,q}}V_i] \), \( L_{g_{i,j}}V_i = (\partial V_i/\partial x_i)g_{i,j}(x) \), \( g_{i,j}(x) \) is the \( j \)-th column of \( G_i(x) \), \( L_{W_i}V_i = [L_{w_{i,1}}V_i \ \ldots \ L_{w_{i,q}}V_i] \), \( L_{w_{i,j}}V_i = (\partial V_i/\partial x_i)w_{i,j}(x) \), \( w_{i,j}(x) \) is the \( j \)-th column of \( W_i(x) \), and \( \rho_i, \chi_i, \phi_i \) are tunable parameters that satisfy \( \rho_i > 0, \chi_i > 1 \) and \( \phi_i > 0 \).

Consider now the \( i \)-th subsystem of the nonlinear plant of Eq.1 under the control law of Eqs.2-4. Evaluating the
time-derivative of the Lyapunov function along the closed-loop trajectories, it can be verified after some algebraic manipulations that $\dot{V}_i$ satisfies the following bound:

$$\dot{V}_i \leq -\sqrt{(L_{ij}^T V_j)^2 + \| (L_G, V_j)^T \|^4 - \rho_1 \| x_i \|^2} - \| W_i, V_i \| \theta_b \left( \frac{\| x_i \|}{\| x_i \| + \phi_i} - 1 \right)$$

(5)

From the above inequality and the fact that $\chi_i > 1$ and $\rho_1 > 0$, it is clear that whenever $\| x_i \| \geq \delta_i := \phi_i (x_i - 1)^{-1}$, the right-hand side is strictly negative. Specifically, we have:

$$\dot{V}_i \leq -\rho_1 \frac{\| x_i \|^2}{\| x_i \| + \phi_i} < 0 \forall \| x_i \| \geq \delta_i, \quad i = 1, 2, \ldots, n$$

(6)

which implies that the closed-loop state of the $i$-th unit remains bounded and converges in finite-time to a terminal neighborhood of the origin whose size can be made arbitrarily small by appropriate selection of the controller tuning parameters $\phi_i$ and $\chi_i$.

B. Quasi-decentralized controller implementation

The implementation of each control law in Eqs.2-4 on the nonlinear plant requires the availability of state measurements from both the local subsystem being controlled and the units that are connected to it. Unlike the local measurements which are available continuously through a dedicated network, the measurements from the neighboring units are available only through the shared plant-wide network whose resources are to be conserved. To reduce the transfer of information between the local control systems as much as possible without sacrificing stability, a set of dynamic models of the interconnected plant units is embedded in the local control system of each unit to provide it with an estimate of the evolution of the states of its neighboring units when measurements are not sent over the network. The use of models allows the sensors of the neighboring units to send their data at discrete time instants since the model can provide an approximation of the plant’s dynamics. “Feedforward” from one unit to another is performed by updating the state of each model using the actual states of the corresponding unit provided by its sensors at discrete time instances.

Under this architecture, the local control law for each unit is implemented as follows:

$$u_i(t) = k_i(\tilde{x}_i, x_i(t)), \quad i = 1, 2, \ldots, n$$

$$\tilde{x}_j(t) = f_j(\tilde{x}_j, x_i(t)) + \tilde{G}_j(\tilde{x}_j, x_i(t))\tilde{u}_j(t)$$

$$\tilde{u}_j(t) = k_j(\tilde{x}_j, x_i(t)), \quad t \in (t_k, t_{k+1})$$

$$\tilde{x}_j(t_k) = x_j(t_k), \quad j = 1, \ldots, n, \quad j \neq i, \quad k = 0, 1, 2, \ldots$$

(7)

where $\tilde{x}_j$ is an estimate of $x_j$ used by the local control system of the $i$-th unit, $\tilde{x}_j$ is a vector containing the estimates of the states of all the plant units except the $i$-th unit, i.e., $\tilde{x}_j = [\tilde{x}_j^1, \ldots, \tilde{x}_{i-1}^j, \tilde{x}_{i+1}^j, \ldots, \tilde{x}_n^j]$, $f_j(\cdot)$ and $\tilde{G}_j(\cdot)$ are nonlinear functions that model the dynamics of the $j$-th unit. The notation $t_k^i$ is used to indicate the $k$-th time instance that the states of the models embedded in the $i$-th control system are updated using the state measurements transmitted from the rest of the plant.

C. A State-Dependent Communication Policy

A key parameter in the analysis of the control and update laws in Eq.7 is the update period for each unit, $h_k^i := t_{k+1}^i - t_k^i$, which determines the frequency at which the $i$-th control system receives measurements from the other units through the network to update the corresponding model estimates. The update period (the reciprocal of which is the communication rate) is an important measure of the extent of network resource utilization, with a larger $h_k^i$ indicating a larger reduction in resource utilization. In [9], we developed a static communication policy in which the update period was considered constant and the same for all the units (i.e., $t_k^i = t_{k+1}^i := h, \quad k = 1, 2, \ldots, n$) and thus could be calculated off-line prior to plant operation.

Our aim in this section is to devise a dynamic communication policy that allows the local control system to determine and adjust the necessary communication rate online (i.e., during plant operation) based on the state of the plant. The main idea is to use the Lyapunov stability condition derived in Section III-A as a guide for establishing and suspending communication. Specifically, consider the nonlinear plant of Eq.1 subject to the model-based network controller of Eq.7. Evaluating the time-derivative of the $i$-th Lyapunov function, $V_i$, along the trajectories of the $i$-th networked closed-loop subsystem for $t \in (t_k^i, t_{k+1})$ yields:

$$\dot{V}_i = L_{f_i} V_i(x) + L_{G_i} V_i(x)k_i(\tilde{x}_i, x_i) + W_i(x)\theta_i$$

$$\dot{V}_i = L_{f_i} V_i(x) + L_{G_i} V_i(x)k_i(\tilde{x}_i, x_i) + W_i(x)\theta_i$$

$$\dot{V}_i = L_{f_i} V_i(x) + L_{G_i} V_i(x)k_i(\tilde{x}_i, x_i) + L_{G_i} V_i(x)\left[k_i(\tilde{x}_i, x_i) - k_i(x)\right]$$

$$\forall \| x_i \| \geq \delta_i, \quad i = 1, 2, \ldots, n$$

(8)

where we have used the bound in Eq.6 to derive the last inequality in Eq.8. Examining this inequality and comparing it with the inequality of Eq.6 obtained in the case of the non-networked plant (i.e., under continuous communication) reveals explicitly the perturbation effect of suspending communication between the plant units on stability. Specifically, the discrepancy between $k_i(\tilde{x}_i, x_i)$ and $k_i(x)$, which arises due to the reliance of the local controller of the $i$-th unit on the states of the uncertain models during periods of communication suspension, alters the rate at which the Lyapunov function decays. As the model estimation error grows, the error in the implemented control action grows as well and may become large enough so as to dominate the stability margin (the negative term) thus causing growth of the Lyapunov function and rendering the closed-loop system potentially unstable. When this happens, communication with the rest of the plant must be re-established to allow updating the states of the models embedded in the local control system in a way such that the plant-model mismatch can be corrected in time to avert instability. This communication policy is formalized in the following theorem.

**Theorem 1:** Consider the nonlinear plant of Eq.1, for which
the Lyapunov functions \( V_i, i = 1, \cdots, n \), satisfy Eq.6 when state measurements are exchanged continuously between the plant units. Consider also the \( i \)-th plant unit subject to the model-based networked controller of Eq.7. If at any time \( t_k^i \) such that \( \| x_i(t_k^i) \| > \delta_i := \phi_i(x_i) - 1 \) the following condition holds:
\[
\dot{V}_i(t_k^i) \geq 0
\]
where \( x_i(t_k^i) = \lim_{t \rightarrow t_k^i} x_i(t) \), then the update law given by \( \ddot{x}_i(t_k^i) = x_i(t_k^i) \) ensures that \( V_i(x_i(t_k^i)) < 0 \).

**Proof:** The proof of this result can be obtained by noting that at any time the states of the models embedded in the \( i \)-th control system are re-set such that \( \ddot{x}_j(t_k^i) = x_j(t_k^i) \), for \( j \neq i \), we have \( k_i(\ddot{x}_j(t_k^i), x_i(t_k^i)) = k_i(x(t_k^i)) = 0 \) which when substituted into Eq.8 yields \( V_i(x_i(t_k^i)) < 0 \) for all \( \| x_i(t_k^i) \| > \delta_i \).

**Remark 1:** The communication policy described in Theorem 1 is dynamic in the sense that the update times (and hence the update periods) for each unit are not determined a priori, but are instead defined implicitly by Eq.9 and are thus state-dependent. The implementation of this policy requires that each local control system monitor the evolution of the corresponding Lyapunov function to determine when the models’ states must be updated and communication re-established. Specifically, if \( V_i \) begins to increase at any time, the sensor suites of the neighboring units are prompted to send their data over the network to update their corresponding models embedded in the \( i \)-th unit. Communication from the rest of the plant to the \( i \)-th unit is then suspended for as long as the Lyapunov function \( V_i \) continues to decay. In this way, only units that require attention (i.e., those on the verge of instability) receive measurement updates over the network, while the rest do not. This targeted update strategy helps reduce overall network utilization further.

**Remark 2:** An advantage of this dynamic communication policy is that it is more robust to unpredictable disturbances (when compared with a static policy) and allows the plant to respond quickly in an adaptive fashion to a unit that requires immediate attention. Another key advantage of this approach is that it ultimately leads to a more efficient utilization of network resources since the communication rate is increased only when necessary to maintain closed-loop stability.

**Remark 3:** The update law given in Theorem 1 applies when the monitored local state \( x_i \) has not yet converged to the terminal set. By ensuring that the time-derivative of \( V_i \) along the trajectories of the \( i \)-th networked closed-loop subsystem remains negative-definite for all times that \( x_i \) is outside the terminal set, this law acts to enforce stability and ultimate boundedness and guarantees that the state of this subsystem also converges in finite-time to the terminal set. Once the closed-loop state enters the terminal set, however, a different criterion for terminating and establishing communication need to be employed since the time-derivative of \( V_i \) (even for the non-networked plant) is no longer expected to remain negative inside the terminal set. Specifically, transmission of measurements to the \( i \)-th unit can be suspended for as long as \( x_i \) remains confined within the terminal set. As soon as \( x_i \) starts to escape this set, however, the \( i \)-th unit prompts its neighboring units to send their measurements to update the corresponding models within the local control system and keep \( x_i \) confined within the terminal set.

**Remark 4:** In addition to stability considerations, performance specifications can also be incorporated into the proposed communication policy by appropriate modification of the update law. For example, an update law of the form:
\[
\ddot{x}_i(t_k^i) = x_i(t_k^i), \quad \text{where} \quad \dot{V}_i(t_k^i) \geq -(1 - \alpha)\rho_i \| x_i(t_k^i) \|^2 / \| x_i(t_k^i) \| + \phi_i, \quad \| x_i(t_k^i) \| > \delta_i
\]
where \( \alpha \in (0, 1) \), ensures not only that \( V_i \) decays monotonically along the trajectories of the \( i \)-th networked closed-loop subsystem, but also that it does so at a certain minimum rate (which is a fraction of the rate prescribed for the non-networked plant per Eq.6). By examining Eq.8, it can be seen that an update law of the form of Eq.10 with \( \alpha \neq 1 \) imposes a stronger restriction on the growth of the model estimation error than the stability-based logic of Theorem 1, in that it limits the extent to which model estimation errors (resulting from communication suspensions) can slow down the non-networked closed-loop response. This in turn implies that accommodating the additional performance requirements may come at the expense of an increase in the rate at which the \( i \)-th control system needs to receive measurement updates from the rest of the plant.

**Remark 5:** The choice to use \( V_i \) to monitor and assess the local stability properties of the \( i \)-th unit is a natural one given that the same function is used in the design of the local controller for the non-networked plant. However, since \( V_i \) in general is not a control Lyapunov function for the \( i \)-th networked closed-loop subsystem, an increase in its value at some time may not necessarily mean that the unit will become unstable. Therefore by requesting that \( V_i \) decrease monotonically in the presence of the network, the update law of Theorem 1 may sometimes act aggressively and prompt the neighboring units to send state updates prematurely thus leading to an increase in network utilization. To increase the flexibility of the communication logic in dealing with this problem, one approach is to consider several Lyapunov function candidates for the same unit. The idea here would be to monitor those functions simultaneously and request an update only if all the functions begin to increase at some time. As long as at least one of the Lyapunov functions continues to decrease, no updates from the rest of the plant are necessary.

IV. Simulation Study: Application to Chemical Reactors with Recycle

We consider a plant composed of two cascaded non-isothermal continuous stirred-tank reactors (CSTRs) with recycle (see the plant model in [16]). The output of CSTR 2 is passed through a separator that removes the products
and recycles the unreacted material to CSTR 1. The reactant species \( A \) is consumed in each reactor by three parallel irreversible exothermic reactions; and a jacket is used to remove/provide heat to each reactor. The control objective is to stabilize the plant at the (open-loop) unstable steady-state with the rates of heat input, denoted by \( Q_1 \) and \( Q_2 \), chosen as the manipulated variables for the two reactors. The control objective is to be achieved with minimal data exchange between the local control systems of the reactors over a shared communication network. Following the methodology presented in Section III, the plant is cast in the form of Eq.1 with \( n = 2 \), where \( x_i \) and \( u_i \) are the (dimensionless) state and manipulated input vectors for the \( i \)-th unit, respectively, and \( \theta_i \) represents the vector of parametric uncertainties in the enthalpies of the three reactions

A Lyapunov-based controller of the form of Eqs.2-4 with

\[
V_i = x_i^T P_i x_i, \quad i = 1, 2, \quad P_1 = P_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix},
\]

is then designed for each reactor to enforce robust stability and uncertainty attenuation in the absence of communication suspensions. The controller design parameters were chosen to ensure that the closed-loop state of each unit converges in finite time to a small neighborhood of the desired steady-state. It was verified that, when state measurements are communicated continuously between the two units, the feedback controllers successfully stabilize the closed-loop state of the plant near the desired steady state.

For the case when direct measurements from the neighboring reactor can be received only through the shared network, and in order to reduce utilization of network resources, instead of the actual state, an estimate is provided to the local controller of each unit by an embedded model (which is the same model used for controller synthesis). For example, in the control system of CSTR 1, an estimate of \( x_2 \) is provided by a model of the form \( \hat{x}_2 = \hat{f}_2(x_1, \hat{x}_2) + \hat{g}_2 u_2(x_1, \hat{x}_2) \), where, for simplicity, \( \hat{f}_1 = f_1 \) and \( \hat{g}_1 = g_1 \). Also, a model of the first reactor of the form \( \hat{x}_1 = \hat{f}_1(x_1, x_2) + \hat{g}_1 u_1(x_1, x_2) \) is embedded in the local control system of the second reactor. Using the state estimates, the control laws are implemented as in Eq.7 where the estimates are used by the local controller so long as no measurements from the neighboring unit are transmitted over the network, but are updated using the true measurements whenever they become available from the network. The solid profiles in Figs.1(a)-(b) depict the resulting evolution of the closed-loop state and manipulated input profiles when the plant is operated using the dynamic communication policy presented in Section III. In obtaining these plots, models with parametric uncertainty of \( \theta_i = [-0.2 - 0.2 - 0.2]' \), \( i = 1, 2 \), were used, and the following values were chosen for the controller tuning parameters: \( \chi_i = 1.1, \rho_i = 0.0001, \phi_i = 0.0001 \). In this case, the evolution of each Lyapunov function is monitored locally within each unit, and a measurement update is requested (and transmitted over the network) from the neighboring unit only when either (1) the Lyapunov function is on the verge of increasing while the state is outside the terminal set, or (2) the state is on the verge of escaping the terminal set while inside. It can be seen from figures that the closed-loop plant can be successfully stabilized near the desired steady-state.

Figs.1(c)-(d) show the time instances at which the models embedded in the local control systems of the two reactors are updated. The variable \( \text{Update}_i \) takes a value of 1 when the local control system for the \( i \)-th reactor requires a measurement from its neighbor to reset the state of the model embedded within it, and takes a value of zero when no updates are needed. It can be seen from the two plots that communication between the two local control systems is needed only initially and over a short period of time (only the first unit receives measurements, while the second unit does not). As the closed-loop plant state settles close to the desired operating point, no further communication between the two controllers is required, which implies that network resources can be further saved during this time.

![Fig. 1. Plots (a)-(b): Closed-loop state and manipulated input profiles under dynamic and static communication policies between the reactors. Plots (c)-(d): Update times of the models embedded in the local control systems of the two reactors under dynamic communication policy.](image-url)

For comparison, we also implemented the static communication policy which was presented in [19]. In this case, the two reactors communicate with each other over the network periodically, and each unit transmits its measurements at a constant rate (the same for both units) to update the model embedded in the local controller of its neighboring unit. This policy assumes that the sensors of all the units are given access to the network and can successfully transmit their data simultaneously. The dashed and dotted profiles in Figs.1(a)-(b) depict the evolution of the closed-loop state and manipulated input profiles of the two reactors when a static communication policy is applied. Using the results in [19], it can be verified that the maximum allowable update period that guarantees closed-loop stability is \( h = 0.18 \) hr, and that the closed-loop plant becomes unstable.
when it is operated under a larger value of $h$. However, operating at the maximum allowable update period leads to poor performance (see dashed profiles), especially when compared with the performance obtained under the dynamic communication policy. In order to achieve an overall closed-loop response comparable to the one obtained under the dynamic communication policy, a smaller update period of $h = 0.1$ hr must be used (see dotted profiles) which requires further increase in communication.

In addition to closed-loop performance and network utilization considerations, we have also investigated the disturbance-handling capabilities of both the static and dynamic communication policies in order to compare their robustness with respect to unanticipated disturbances during plant operation. To this end, a 20% step disturbance was introduced in the flow rate of the fresh feed stream in CSTR 1, $F_{0i}$, at time $t = 2$ hr (i.e., after the plant has reached its steady-state), and this disturbance lasts for 0.2 hrs. Fig.2 panels (a)-(b) depict the resulting closed-loop temperature and manipulated input profiles subject to the unexpected external disturbance. It can be seen that while the control systems under the dynamic communication policy can successfully recover from the disturbance and force the plant to return to its steady-state (see the solid profiles), stability cannot be maintained under the static communication policy where the states move away from the desired steady-state and settle elsewhere (see the dashed profiles). Fig.2 panels (c)-(d) show the update times of the models embedded in the local control systems of the two reactors when the dynamic communication policy is implemented. These two plots highlight the adaptive nature of the dynamic communication policy which is the reason for its ability to overcome the influence of the disturbance on the closed-loop plant. Specifically, it can be seen that the dynamic policy responds to the external disturbance by increasing the frequency of communication between the two control systems following the onset of the disturbance. This in turn allows the plant states to return to the desired steady-state by the end of the disturbance following which no further communication is needed to maintain stability. By contrast, the static communication policy with a fixed update period cannot.

**REFERENCES**