Network Architecture and Communication Modules for Guaranteeing Acceptable Control and Communication Performance for Networked Multi-Agent Systems

Feng-Li Lian, *Member, IEEE*, John K. Yook, Dawn M. Tilbury, *Member, IEEE*, and James Moyne, *Member, IEEE*

***Abstract*—When sensory and actuation devices in a control system are exchanging data through one common communication medium, the sharing of communication bandwidth will induce unavoidable data latency and might degrade the control performance. Hence, the utilization of communication resource and the requirement of control specification should be analyzed and properly designed when implementing a control system over a network architecture. In this paper, we analyze the performance of information sharing of multiple cooperative agents over one communication network, and propose design methodologies of guaranteeing acceptable control and communication performance in a networked control system. In particular, we study the relationship between the sampling rates of a control system, and the transmission rates of a communication network, and then utilize an integrated networked control design chart to help select design parameters and visualize overall system performance at different sampling and transmission rates. Based on the design parameters selected, the communication modules by utilizing deadband control and state estimation are presented for guaranteeing both control and communication performance. Simulation studies are conducted in a network-and-control simulation tool that is developed on the Matlab/Simulink platform and is used to demonstrate the proposed design methodologies. Both the analysis and simulation results illustrate the characteristics of designing mechanisms between control and communication performance and show the improvement of implementing the proposed communication modules.**

***Index Terms*—Communication rate, deadband control, networked control systems, sampling rate, state estimation.**

I. INTRODUCTION

THE trend of modern industrial and commercial systems is to integrate computing, communication, and control into different levels of machine/factory operations and information processes. The introduction of common-bus network architectures can improve the efficiency, flexibility and reliability of these integrated applications, and reduce installation, reconfiguration, and maintenance time and costs. When dealing with this class of large-scale control applications, functional agents such as sensors, actuators, and controllers are usually spatially distributed. In order to achieve the overall goal of all tasks performed, it is necessary for all the agents to exchange their own information through communication media adaptively or maybe intelligently. Hence, the mechanism of communicating information plays an important role on the stability and performance of the control systems implemented over communication networks. In addition to their data acquisition capability, sensors communicating over a network need well-developed methodology to accurately report their data and adjust communication mechanism adaptively. In this paper, we discuss one network architecture and the design of communication modules, designed for a class of networked multi-agent systems for guaranteeing both control and communication performance.

In the past, traditional control systems had a single centralized control unit, which controlled all other processes and devices (sensors and/or actuators) via generally short point-to-point connections. Nowadays, however, various industrial plants cover large areas and have sophisticated control systems. These systems control a great number of devices and associate them by means of computationally complicate algorithms. Having these algorithms in a single centralized processor or controller can induce several problems due to the hardware and software constraints. These include single point of failure, poor reliability, poor performance, and inability to support advanced distributed control scheme.

Examples of modern complex systems include industrial automation, building automation, office and home automation, intelligent vehicle systems, and advanced aircraft and spacecraft [1]–[6]. The common features of these systems are a large number of devices interconnected together to perform the desired operations, and a large physical area of coverage. Hence, the processing load of a centralized control unit can be large if all demands and computations are handled by this unit. A large physical area also requires a large amount of wiring among devices. Therefore, it is unrealistic to implement the traditional point-to-point connections to a simple centralized control unit in these modern complex systems.

Manuscript received May 4, 2005; revised July 20, 2005. This work was supported in part by the NSF Engineering Research Center for Reconfigurable Machining Systems under Grant EEC95-92125, the DARPA-IXO Mixed Initiative Control of Automa-teams Program, and the National Science Council, Taiwan, R.O.C., under Grants NSC 92-2213-E-002-053 and 93-2213-E-002-049.

F.-L. Lian is with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C. (e-mail: fengli@ntu.edu.tw).

J. K. Yook is with Raytheon Electronic Systems, El Segundo, CA 90245 USA (e-mail: john_k_yook@raytheon.com).

D. M. Tilbury and J. Moyne are with The University of Michigan, Ann Arbor, MI 48109 USA (e-mail: tilbury@umich.edu; moyne@umich.edu).

Digital Object Identifier 10.1109/TII.2005.857611

1551-3203/$20.00 © 2006 IEEE

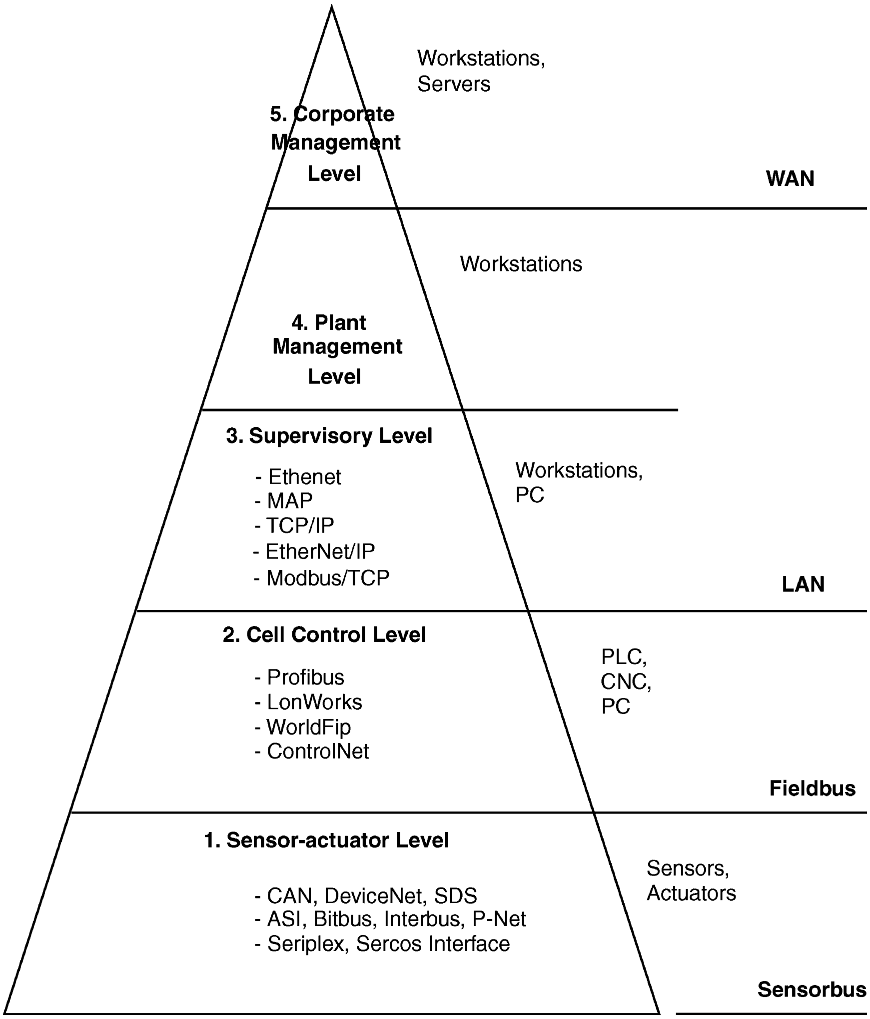


Fig. 1. General network hierarchy model.

The solution currently adopted to address modern control problems is to distribute the processing functions of these systems over several physical nodes, each dedicated to a part of the control process and to a group of sensors/actuators [7]–[9]. These nodes cooperate with each other, communicating through a shared physical channel which generally has a bus topology. These common-bus systems require less complex wiring than point-to-point systems, thereby reducing the setup and maintenance costs. At the same time, they also reduce the possibility of a single fault affecting the whole system. Physically, sensor bus systems can be divided into several modular subsystems that connect to the main system directly. This modularity can result in improved speed and convenience of diagnostics and maintenance [2], [3], [10], [11].

In the light of these advantages, it appears that the common-bus systems represent an attractive alternative to point-to-point solutions. However, the common-bus systems also introduce a number of issues that, if not dealt with properly, can greatly reduce control system effectiveness. For example, two major problems for an integrated communication and control system are addressing and timing. For a simple bus network, because all the devices are interconnected by a single bus, every data communication should be augmented by a header and/or a trailer to specify the source and the destination. Furthermore, because of the bus topology, devices may have to wait for some amount of time before they can send out a message. This mechanism increases the guaranteed response time (when compared to point-to-point solutions) of data transmission. These time delays influence system performance dramatically, especially for controller design [12]–[15].

Generally speaking, there are two types of data needed for control applications, namely, *states* and *events*. Typical state data are position, velocity, temperature, or pressure signals which are the raw information for any control applications. Sensor design for the state-based systems is simple and straightforward. That is, the processing functionality at sensors and actuators is simply the conversion between electrical and mechanical worlds. However, due to the design simplicity, more communications are needed among sensors and actuators over the network. In a small-scale and centralized system, it should not be a big problem. However, in a large-scale or distributed system, the required communication load might degrade the overall performance or destabilize the system if without any design consideration.

On the other hand, events could be the abstract information of detailed control actions. Based on the event information re-

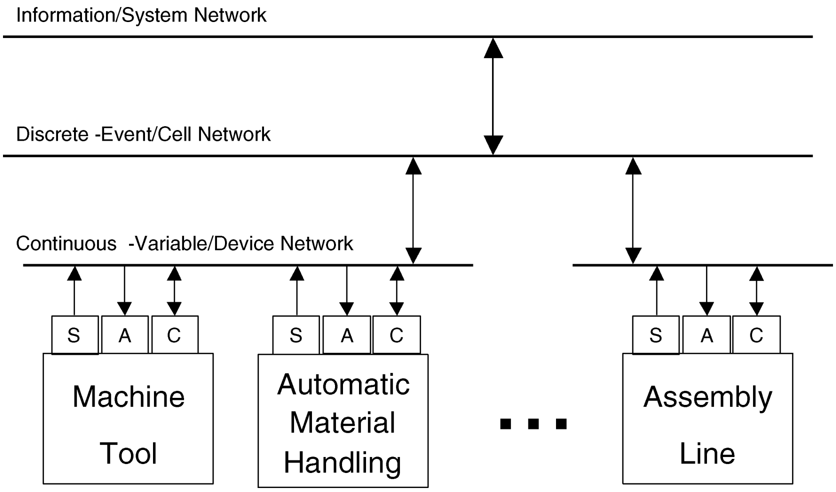


Fig. 2. Network architecture in a modern manufacturing system.

ceived, each device needs to first recover the original content of the abstract information, and then advise local state-based controllers to react upon it. The system performance mainly focuses on the logical correctness of events, and may rely on the well-designed local controllers. In this case, the communication amount needed might be decreased, but the performance of a time-critical control application might not be acceptable due to too many processing components between states and events.

In this paper, we focus on the analysis and design methodologies of state-based networked agents with time-critical applications. For satisfying both control and communication performance, several design methodologies for networked agents are adopted to generate proper control actions and utilize communication bandwidth optimally. The design methodology of choosing sampling and communication rates is presented in [34]. This paper consists six sections, including the Introduction section. Section II discusses a general network architecture for sensing and control applications, and related design issues for networked devices. Section III describes the standard and networked dynamical models for communication module design. Section IV addresses the design methodologies of communication modules, including deadband control and state estimation. Section V presents an illustrative example of a networked multi-agent system with communication modules, and outlines a network and control simulation tool that is used to simulate communication networks as well as sensing and control actions. Conclusion and future work are provided in Section VI.

II. NETWORK ARCHITECTURE AND NETWORKED AGENTS

A network architecture allows sensors, and other agents such as actuators and controllers to be interconnected together, using less wiring, and requiring less maintenance than a point-to-point architecture. It also makes it possible to distribute processing functions and computing loads into several small units. Moreover, distributing control between multiple processors can make the system more robust and fault-tolerant whereas centralized control suffers from the drawback of a single point of failure. Interest in computer networks has increased significantly in the last decade due to networks being considered as a primary mechanism to simplify the transfer of information. Fig. 1 illustrates a general network hierarchy model [5], [8]. This model consists of five levels, each one having different goals and also different communication capabilities, protocols and complexity. In modern manufacturing systems, for example, level one is the device or sensor-actuator level which is used to interconnect controllers, sensors or actuators. Level two is the cell control level and is designed to be used with cell controllers such as at milling, lathe and control workstations in manufacturing plants. Generally speaking, levels one and two are called "sensor bus" and "field bus," respectively. Level three is the supervisory level and is used to interconnect machine cells which perform different manufacturing processes. Level four is the plant management level and is used to coordinate various tasks executed inside a plant such as manufacturing engineering, production management, resource allocation. Level five is the corporate management level and may interconnect workstations located in different cities or countries.

Since sensing and action agents are interconnected on the network at the first or second level, we further consider a typical network architecture such as in modern manufacturing system as shown in Fig. 2. This network architecture has three different levels, namely, the Information/System (IS) network, the Discrete-Event/Cell (DEC) network, and the Continuous-Variable/Device (CVD) network. This classification is based on the functionality as well as signal characteristics in typical industrial applications.

The top level IS network is used to carry nontime-critical information such as daily or hourly production data, and to communicate with factory-wide databases. Messages on an IS network typically have a large data size but low frequency. Research topics in IS networks include throughput analysis, flow control, information management, database merging, distributed query optimization, and security [16], [17].

The middle level is the DEC network, which carries commands or updates working configurations for different cells or subsystems. Generally speaking, the messages in a DEC network are discrete and event-based. The DEC network messages

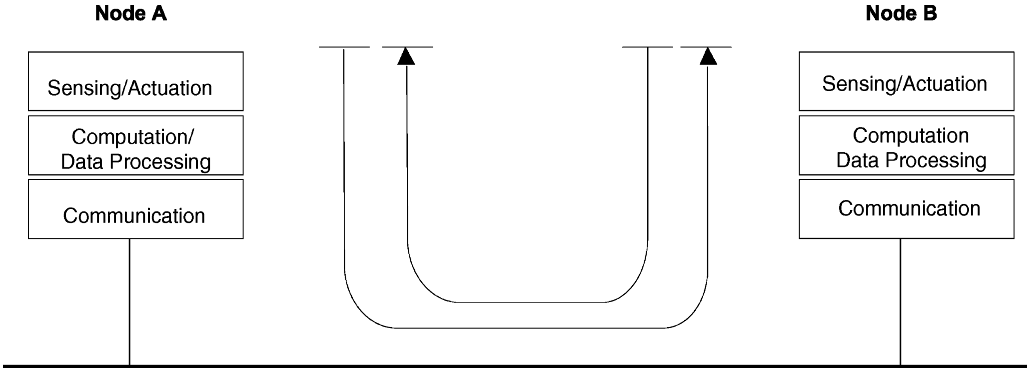


Fig. 3. Key functionalities of networked agents.

may be periodic, sporadic, or time-critical. When timing is critical, large time delays and lost data at this level may cause coordination problems between different subsystems. The analysis and control of DEC network systems such as manufacturing systems and multitask robotic systems has been studied using discrete-event system techniques such as finite state machines and Petri nets, with or without timing parameters [10], [18], [19]. The research focuses on the correctness and safety issue of system operation in a reasonable and optimal logical order.

The bottom level is the CVD network, which communicates physical signals such as position, velocity, and temperature by the means of network coding and messaging. Sensors, actuators, and controllers are the types of devices interconnected by the CVD networks. Messages are transmitted periodically and in real-time; data sizes are small, but message transmission frequency may be high. Time delays and lost data at this level may degrade the system performance and even cause system instability [20]–[23].

The three levels of networks are separated because they require different information characteristics and functionalities, although they may be connected by gateways or bridges. If the same network is used for multiple levels, the large-size data packets transmitted at the IS-network level could degrade network efficiency in both the CVD and DEC networks, and the high-frequency data packets at the CVD-network level might further delay the message transmission in either the DEC or IS network.

Typical agents in the CVD networks include smart sensors, smart actuators and networked controllers. Sensing or actuation, data processing, and communication capabilities are the three key features of a networked agent. The schematic diagram of the message transmission among networked agents is shown in Fig. 3. Specifically, smart sensors have three major features: data acquisition, intelligence, and communication ability [2], [24], [25]. Smart sensors are sensors that acquire proper physical data such as temperature and motion data from the industrial environment and have a network-capable application processor which is the interface between the sensor and network. Intelligence gives smart sensors the ability to function independently and flexibly; they may have capabilities such as self-calibration and self-diagnostics and the ability to function as a component of a distributed control system. Because of the different information levels in the physical environment and network medium, the sensor must be able to properly encode the information before sending it out on the network.

Smart actuators, similar to smart sensors, have the features of actuation, intelligence and communication [26]. The actuator should be able to decode the information from the network medium and transmit it to the physical devices. Besides network-capable application processors, the major functionalities of networked controllers are to analyze the sensor data, make decisions, and give commands to actuation devices. The control algorithms should handle decentralized information analysis as well as the traditional centralized analysis. That is, controllers should be able to handle the situation of multiple time delays induced from different networked devices and cooperate with other networked controllers. Networked controllers may also provide a human-machine interface to operators or higher-level managers.

In order to guarantee the interoperability and interchangeability of devices, however, it is necessary to perform different types of conformance tests and performance evaluation on the networked devices. The main conformance tests include the protocol test, the physical layer test, and the interoperability test. The protocol test is to verify the application behavior of a networked agent for conformance to the protocol specification and to validate the features of the networked agent provided to communicate with other agents. During the test, the testing software verifies the correctness of response messages from the agent that are requested by the software. The physical layer test is used to verify the electrical characteristics such as bit time and voltage level of the networked agent. The interoperability test checks the operation of the networked agent with other agents under different operating conditions. Different examples of conformance tests for ControlNet, DeviceNet, EtherNet/IP, and Modbus/TCP at the Sensor Bus Laboratory, the University of Michigan can be found at [27].

On the other hand, performance evaluation is used to identify the response time or processing time of networked agents. The timing identification is an important factor not only for network performance, but also for control performance. Especially for deterministic network protocols, the variance of the device processing time affects the designated network performance significantly. The outcome of network performance also influences the end-to-end time delay of control signals, and, moreover, changes the control system performance. Therefore,

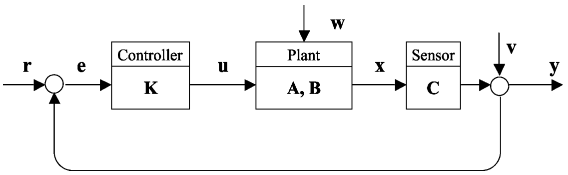


Fig. 4. Standard MIMO closed-loop system.

the response-time modeling of networked devices is crucial to a networked control designer to determine the network and control performance, and should be identified properly. Detailed performance evaluation of networks and devices can be found in [29] and [30].

III. Dynamical Model of Communication Module

In this section, we discuss the dynamical model of designing communication modules in networked agents. The difference between standard and networked system models will be first analyzed and the key features of designing network-type sensing agents will be then discussed.

A. Standard MIMO System

For the objective of control system design, we first consider a standard discrete-time, linear-time-invariant (LTI), multi-input and multi-output (MIMO) system with n states, m inputs, and r outputs, shown in Fig. 4 and described as follows:

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k)\end{aligned} \tag{1}$$

where k is the time index associated with the sampling time T in discrete time domain, $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, and $\mathbf{C} \in \mathbb{R}^{r\times n}$ are system, input, and output matrices, respectively, and $\mathbf{w}(\cdot)$ and $\mathbf{v}(\cdot)$ are system disturbance and measurement noise, respectively. $\mathbf{w}(\cdot) = [w_1(\cdot), \ldots, w_n(\cdot)]^T$ and $\mathbf{v}(\cdot) = [v_1(\cdot), \ldots, v_r(\cdot)]^T$ are assumed to be bounded, i.e., $|w_i(\cdot)| \le b_{iw}$, $i = 1, \ldots, n$ and $|v_j(\cdot)| \le b_{jv}$, $j = 1, \ldots, r$, where b_{iw}'s and b_{jv}'s are some known positive constants. For simplicity, we assume that $\mathbf{C} = \mathbf{I}$, that is, all the states are assumed measurable. The state feedback controller for system (1) can then be designed by any standard MIMO control design technique as follows:

$$\begin{aligned}\mathbf{u}(k) &= \mathbf{K}\mathbf{e}(k) \\ &= \mathbf{K}\left[\mathbf{r}(k) - (\mathbf{x}(k) + \mathbf{v}(k))\right].\end{aligned} \tag{2}$$

That is, we assume the MIMO control system shown in Fig. 4 is well-designed. Hence, the system stability and performance of system (1) could be guaranteed by properly choosing the sampling time T in (1), and designing the state feedback gain K in (2). In this paper, we consider system dynamics (1) and controller design (2) as the baseline design framework.

B. Networked MIMO System

We next consider a distributed control architecture where sensors, actuators, and controllers are physically distributed and exchanging data through one communication network as shown in Fig. 5. The advantages of using the distributed architecture include reducing significant wiring, sharing information, easily monitoring and diagnosing system health, etc. However, there are two key drawbacks with respect to time-critical control applications. Because of the sharing of communication media, each sampled data has an inevitable waiting time, i.e., time delay. On the other hand, the frequency of information needed for a state-based control system might consume too much network bandwidth or even saturate network traffic load. The situation of high bandwidth utilization might induce additional time delays, and further degrade the control performance.

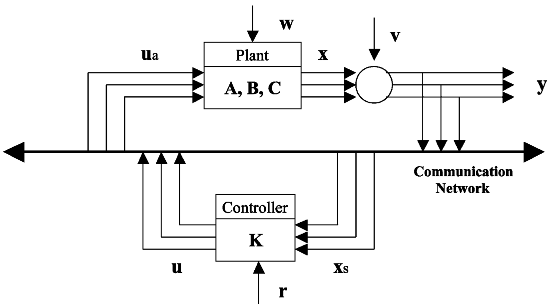


Fig. 5. MIMO control system with communication of control signals over distributed networks.

If the data are sampled asynchronously and have transmission time delays, the system and controller dynamics at the controller sampling instants should be modified as follows:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}_a(k) + \mathbf{w}(k) \tag{3}$$

$$\mathbf{u}(k) = \mathbf{K}\left[\mathbf{r}(k) - (\mathbf{x}_s(k) + \mathbf{v}(k))\right] \tag{4}$$

where $\mathbf{x}_s(\cdot)$ and $\mathbf{u}_a(\cdot)$ are the delayed version of $\mathbf{x}(\cdot)$ and $\mathbf{u}(\cdot)$, respectively. That is, for the ith element of $\mathbf{x}(\cdot)$, $x_s^i(k) = x^i(k - s^i)$, and, for the jth element of $\mathbf{u}(\cdot)$, $u_a^j(k) = u^j(k - a^j)$, where s^i and a^j are the sum of the transmission delays and mismatched sampling instants of the ith sensory and jth actuation data, respectively. A detailed discussion of the system analysis and controller design for the system with asynchronous and mismatched sampling instants can be found in [33].

In this paper, we will focus on the performance analysis of networked agents with a standard MIMO controller in a networked control architecture and study the interaction between control and communication mechanism. Hence, the feedback controller and state estimator discussed in next section are assumed to be designed based on the system dynamics (1), but evaluated based on the architecture of Fig. 5 and the framework of (3) and (4).

IV. Communication Module Design

When implementing a distributed MIMO control system over a network, one should study the network protocols, evaluate the network performance, and understand the impact of data latency on sensor action as well as control performance. In this section, we characterize the key features of designing a networked control system in general. In particular, we discuss the network design considerations of sensor parameters in terms of networked control design chart. To achieve both the control and network

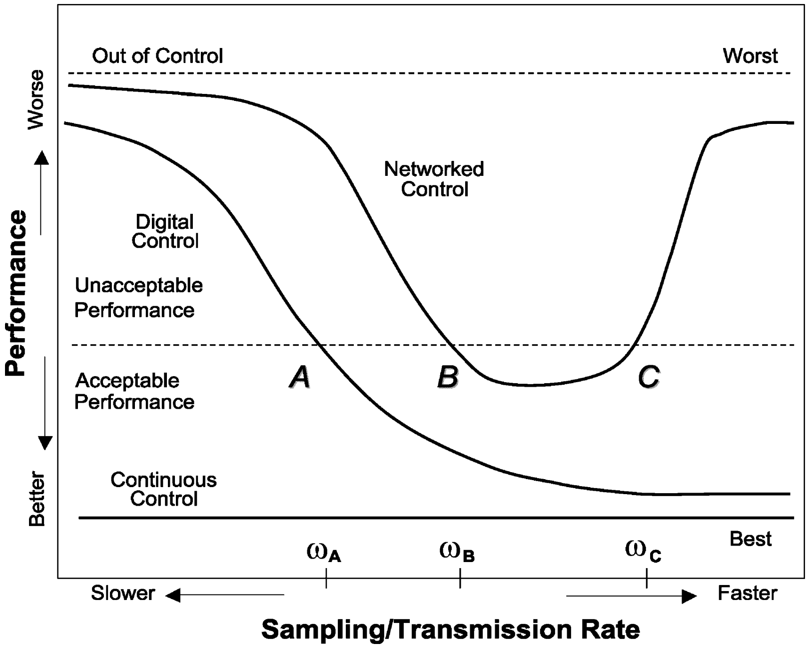


Fig. 6. Performance comparison of continuous control, digital control, and networked control cases.

performance under the limited communication bandwidth, we utilize deadband control and state estimation approaches to actively adjust the communication rate of a sensor agent and guarantee designated control performance. In the end, an integrated network and control design chart will be used to visualize the overall system performance and dynamically choose communication parameters of a networked agent.

A. Network and Control Parameters

When selecting a communication network for control applications, two key questions related to control performance are: how much time does one message need from the source agent to the destination agent, and how reliable the message transmission is. These two questions address the issue of control performance degradation by data latency and timing uncertainty, respectively. The total time delay of one message transmission is a function of the preprocessing time and waiting time at the source device (sensor or controller), the transmission time and propagation time on the network medium, and the postprocessing time at the destination device (controller or actuator). The pre- and postprocessing times are those needed to convert signals between physical environment and network data format and mainly depend on the characteristics of agent software and hardware. On the other hand, the waiting and transmission times depend on the network protocol implemented and the real-time network traffic load. Different network protocols support different applications and provide different quality of service. The network traffic load is a function of the number of transmitting agents and available bandwidth. Hence, a control designer should first understand the magnitude and characteristics of data transmission and analyze the expected control performance before implementing a control system with networked agents.

Next, we discuss three important control parameters: the sampling rate, control system bandwidth, and phase margin that characterize control performance closely and have a strong relationship to the above-mentioned network parameters. Since a networked control system is essentially a discrete time system, choosing a proper sampling rate of sensing and actuation data is as important as that in the digital controller design. The sampling rate should depend on the control system bandwidth which is defined as the maximum frequency at which the output a system will track an input sinusoid in a satisfactory manner. In order to achieve a reasonable control performance, the “rule of thumb” for selecting a sampling rate in digital control is that the desired sampling multiple, i.e., the ratio of sampling rate w_s and control system bandwidth w_{bw}, should fulfill the following relation [28]:

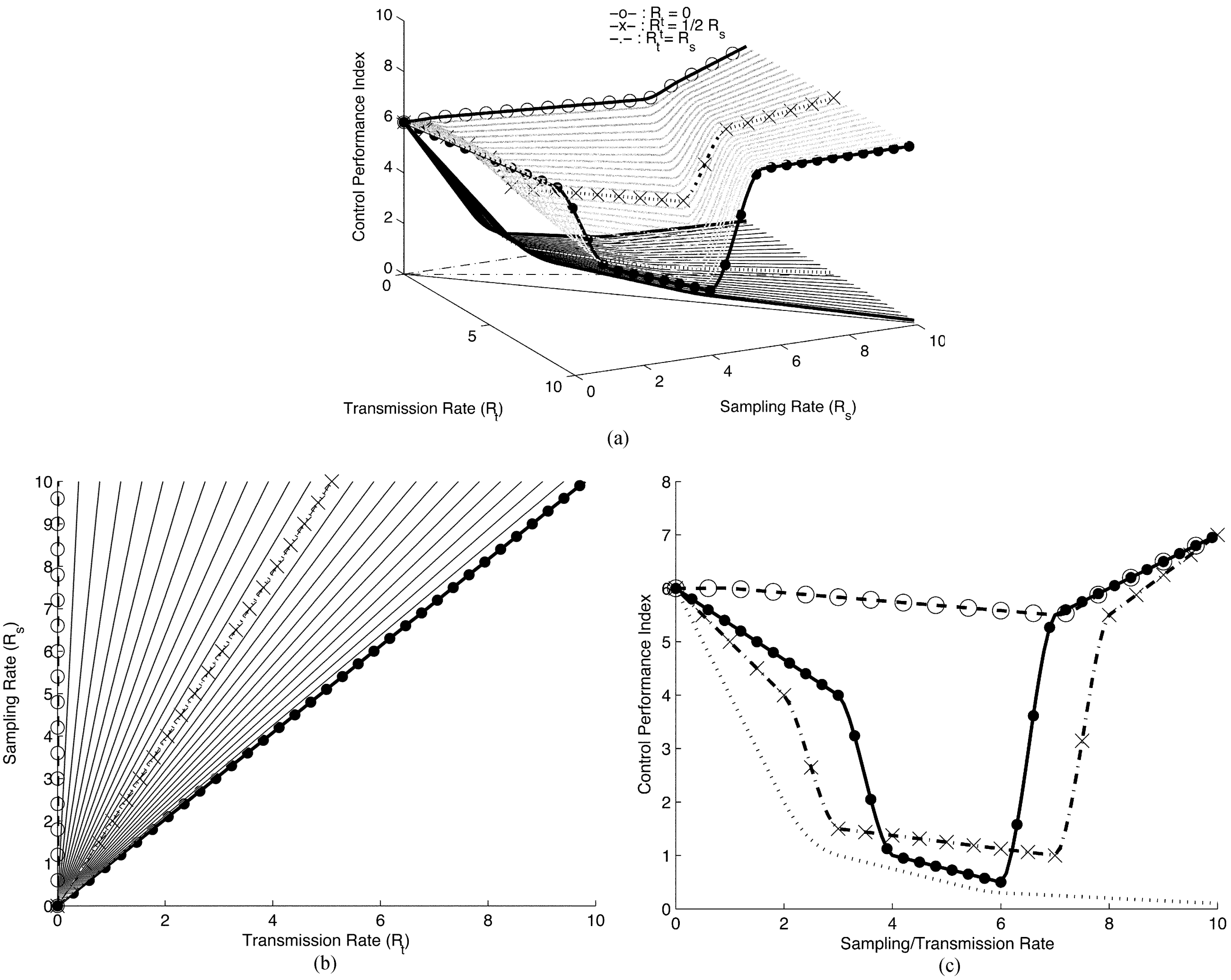
$$20 < \frac{w_s}{w_{bw}} < 40. \tag{5}$$

The phase margin of a dynamical system is the amount by which the phase of an open-loop system exceeds -180° when the magnitude equals one. The phase margin can then be used to characterizes the degree of tolerance on the total time delay by the sampling mechanism and data transmission. The phase lag due to discretization $(\Delta\phi_s)$ and additional time delay $(\Delta\phi_d)$ are summarized as follows [28]:

$$\Delta\phi_s = wT_s \quad \text{and} \quad \Delta\phi_d = wT_d \tag{6}$$

where w is the frequency variable, T_s is the sampling time, and T_d is the additional time delay.

Due to the integral link between the network and control parameters, the selection of the best sampling rate is a compromise. In next section, we will discuss the construction of a networked control design chart which can be used to visualize the interaction of network and control systems and help select proper design parameters.



(a)

(b)

(c)

Fig. 7. (a) Schematic diagram of a revised networked control design chart. (b) Top view: the ratio of the sampling and transmission rates. (c) Side view: the reduction of transmission rate and control performance degradation.

B. Networked Control Design Chart

During the implementation of one MIMO controller over a communication network, a design chart can be derived as shown in Fig. 6 [30]. This design chart provides a clear way to choose the proper sampling or transmission rates for a networked control system. Fig. 6 is the comparison of control performance versus sampling rate for continuous control, digital control, and networked control. The worst, unacceptable, acceptable, and best regions can be defined based on required control system specifications such as overshoot, steady state error, and/or phase margin. The performance axis in Fig. 6 could be chosen to reflect a subset of these metrics.

Since the performance of continuous control is not a function of sampling rate, the performance index is constant for a fixed control law. For the digital control case, the performance only depends on the sampling rate assuming no other uncertainties. The performance degradation point A in digital control could be estimated based on the relationship between control system bandwidth and sampling rate. That is, w_A is the practical minimum sampling rate where the control performance degrades due to sampling effect as discussed in previous section.

For the networked control case, point B can be determined by further investigating the characteristics and statistics of network-induced delays and device processing time delays. This is due to the additional phase lag associated with time delays, smaller sampling periods may be needed to guarantee a certain level of control performance. Hence, w_B should be larger than w_A. As the sampling rate gets faster, the network traffic load becomes heavier, the possibility of more contention time or data loss increases in a bandwidth-limited network, and longer time delays result. Point C is the situation when the network is becoming saturated. At this point, larger time delays are expected due to missing data or longer waiting times due to message contention. For a fixed controller law, the best system performance is the continuous case and the worst is when the system is out of control due to missed sensor or actuator data.

In [30], given a set of network and control parameters, we have provided several fundamental formulas to determine these points. To guarantee the best control performance, all the networked devices need the newly updated data from other devices. Hence, all the scenarios considered in [30] are assumed to have an identical sampling and transmission rate, that is, to transmit every sampled data to the destination device. However, a faster sampling rate for guaranteeing good control performance might potentially saturate the network traffic load, and eventually increase the total transmission time and further degrade the control performance. Two adaptive approaches of decoupling the sampling and transmission rates will be discussed in the next section. In the following, we first describe a revised networked control design of Fig. 6 that decouples the sampling and transmission rates.

The schematic view of a revised integrated network and control design chart is shown in Fig. 7. The two independent variables are sampling and transmission rates, and the control performance index could be chosen to reflect a subset of control system specifications as described previously. The sampling rate is similarly selected based on the required control performance. However, the transmission rate is determined by the redesigned communication module at each networked device. Based on a designated controller algorithm, the communication module decides whether it needs to broadcast the newly sampled data to other devices or not. In Fig. 7(a), the line with "•" is the case where the sampling rate (R_s) equals the transmission rate (R_t), the line with "x" is the case where the transmission rate is only a half of the sampling rate, and the line with "o" is the case of no transmission. The schematic view of the ratio of the sampling and transmission rates can be seen in Fig. 7(b). The lower-right region, i.e., $\text{transmission rate} > \text{sampling rate}$, is undefined because it will waste communication bandwidth if the number of message transmission is more than that of the sampled data. From Fig. 7(c), i.e., the side view of Fig. 7(a), it can be easily seen that, as the transmission rate decreases, the control performance degrades, but, the operating range of sampling rate becomes wider. Therefore, by analyzing the required control specification, we will study two communication module design methodologies to dynamically adjust the sampling and transmission rates in Sections IV-C and IV-D. The stability and performance analysis of these two approaches is discussed in Section IV-E.

C. Using Deadband Control to Decouple Sampling and Transmission Rates

In this section, we discuss the implementation of a deadband control framework, studied in [31] and adopted here to dynamically adjust the transmission rate of a networked agent as shown in Fig. 8. The agent with deadband control first compares the most recent state, says x_i, to the last state x_{is} sent to the network. If the absolute value of the difference between x_i and x_{is} is within a deadband threshold, says h_i, then no data is sent to the network. Hence, sensors with the deadband control communication module can reduce transmission rates while maintaining acceptable control performance. Furthermore, it can be predicted that as the deadband threshold h_i increases, the transmission rate decreases further and the control performance degrades as well. That is, for one networked control application, there might exist a tradeoff between control and communication performance. Therefore, by properly selecting the deadband thresholds of all networked sensors, an optimal performance of control and communication can be achieved.

Since a networked agent with deadband control adjusts its transmission rate based on its own state, the deadband control framework is only suitable for the system with slowly-varying states such as manufacturing systems, chemical processing plants. For highly dynamical systems and with strong performance requirements, the actual transmission rate might be as much as one in the networked MIMO case, and the deadband control framework dose not have much improvement on communication performance. In next section, we will further implement an estimator of the states of all networked agents and a modified communication module that adjusts the state transmission based on the actual and estimated states.

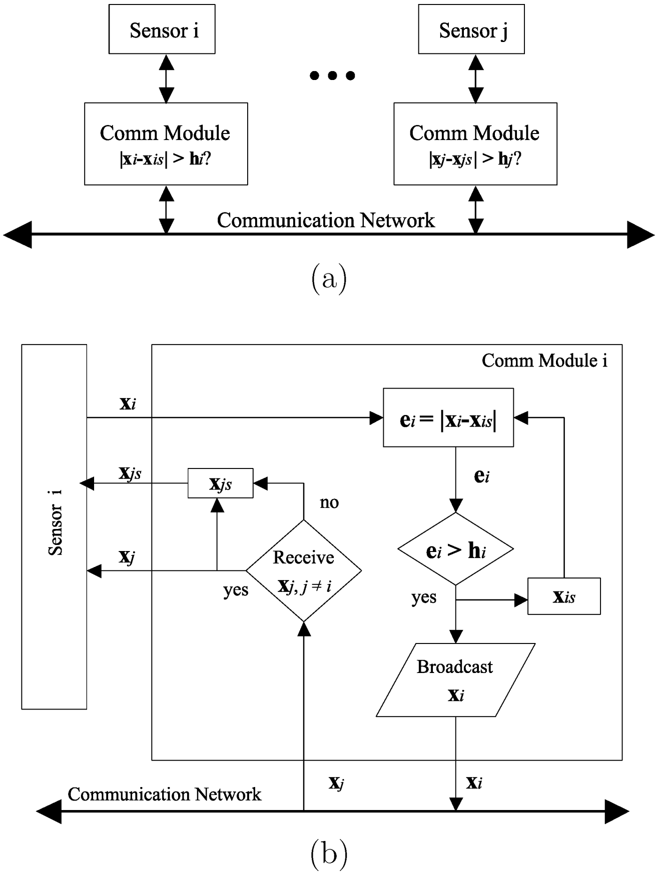


Fig. 8. (a) Networked agent system with deadband-control communication module. (b) The schematic diagram of the communication module.

D. Using State Estimator to Decouple Sampling and Transmission Rates

In this section, we discuss a state estimator framework studied in [32] and modify it for the proposed networked agent system. An example of a networked two-agent system with state estimator and communication module is shown in Fig. 9(a), and the schematic diagram of the Control/Comm module is depicted in Fig. 9(b). The basic idea is to let one agent use estimated states for control actions and broadcast its current states to other agents if estimation is not acceptable. The framework of state estimator and communication module is discussed in the following and the stability and performance will be analyzed in Section IV-E.

Next to the agent is one estimator which computes the states of the agent and the other agents, based on any well-designed estimation algorithm. The main functionality of the Control/Comm module is to compute the difference of the true and estimated states of the agent, control the communication frequency,

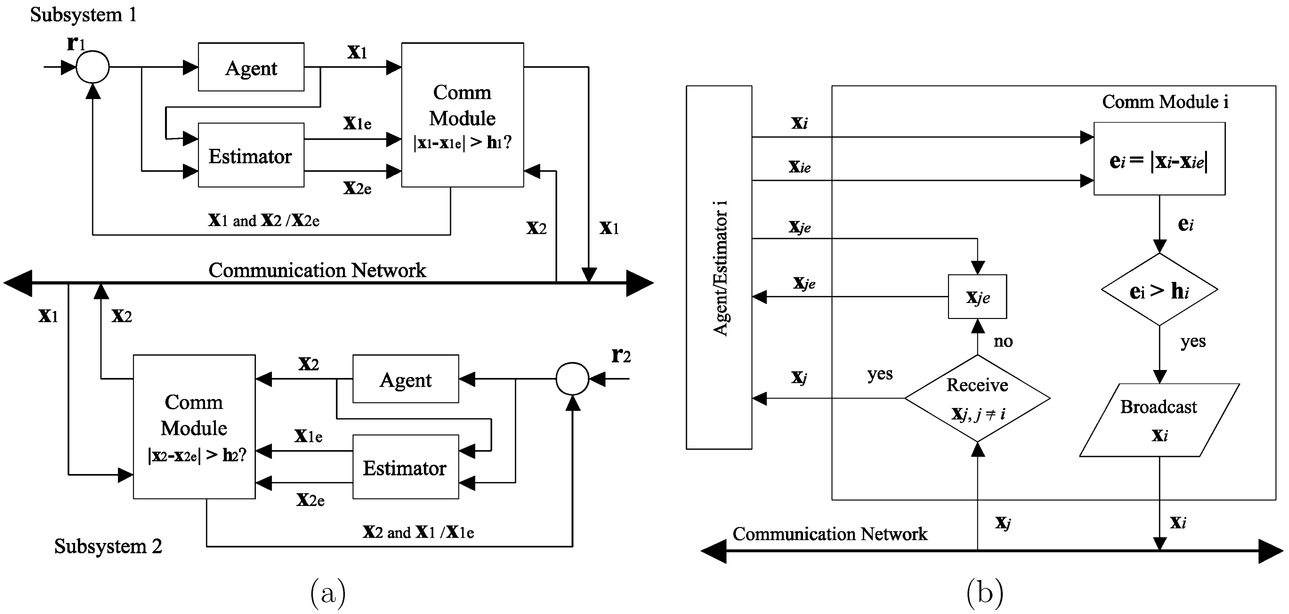


Fig. 9. (a) Networked two-agent system with state estimator and communication module. (b) Schematic diagram of the communication module.

and update the estimated states by the true states of other agents. For example, in Fig. 9, Estimator 1 computes x_{1e} and x_{2e} and Estimator 2 computes x_{1e} and x_{2e} as well. At a normal scenario, i.e., no communication required, Agent 1 is operating based on its own state x_1 and the estimated state of Agent 2, x_{2e}. When Control/Comm module 1 receives new x_2, it informs Agent 1 to use the newly arrived x_2 instead of the estimated state x_{2e}. In addition, Control/Comm module 1 broadcasts x_1 to Agent 2 if $|x_1 - x_{1e}|$ is larger than a predefined threshold, say h_1. Similar estimation and communication mechanisms are designed at Agent 2 and other agents.

For an n-agent system, there are n estimators of n states and n communication modules. Hence, the additional computational complexity is $n \times n + n$ compared with the networked MIMO system. However, utilizing the locally estimated states can save certain amount of communication cost/bandwidth and also achieve good control performance. In next section, we analyze the overall system stability and characterize the control and communication performance.

E. Stability and Performance Analysis

In this section, we first analyze the stability of the networked control systems with the deadband control and state estimator frameworks, and then characterize the control and communication performance.

We consider the system (1) with the controller (2) as the baseline system, and assume that the controller is well-designed and, therefore, the closed-loop system is exponentially stable. If we look at the controller, the implementing control law is as follows:

$$\hat{\mathbf{u}}(k) = \mathbf{K}\left[\mathbf{r}(k) - \hat{\mathbf{x}}(k) - \bar{\mathbf{v}}(k)\right] \tag{7}$$

where $\hat{\mathbf{x}}$ is the sent state $\mathbf{x}_s$ for the deadband-control case, or the estimated state $\mathbf{x}_e$ for the state-estimator case. Therefore, the closed-loop system can be described as follows:

$$\begin{aligned}\bar{\mathbf{x}}(k+1) &= \mathbf{A}\bar{\mathbf{x}}(k) + \mathbf{B}\hat{\mathbf{u}}(k) + \bar{\mathbf{w}}(k)\\ &= \mathbf{A}\bar{\mathbf{x}}(k) + \mathbf{BK}\left[\mathbf{r}(k) - \hat{\mathbf{x}}(k) - \bar{\mathbf{v}}(k)\right] + \bar{\mathbf{w}}(k)\\ &= \mathbf{A}\bar{\mathbf{x}}(k) - \mathbf{BK}\hat{\mathbf{x}}(k) + \mathbf{BKr}(k)\\ &\quad - \mathbf{BK}\bar{\mathbf{v}}(k) + \bar{\mathbf{w}}(k)\\ &= [\mathbf{A} - \mathbf{BK}]\bar{\mathbf{x}}(k) + \mathbf{BK}\left[\bar{\mathbf{x}}(k) - \hat{\mathbf{x}}(k)\right]\\ &\quad + \mathbf{BKr}(k) - \mathbf{BK}\bar{\mathbf{v}}(k) + \bar{\mathbf{w}}(k),\end{aligned} \tag{8}$$

where $\bar{\mathbf{x}}(\cdot)$ is the system state when an old or estimated state is used to compute the control input. Based on the design algorithms discussed in Sections IV-C and IV-D, the difference $|x_i(k) - \hat{x}_i(k)|$ should be bounded, i.e., $|x_i(\cdot) - x_{is}(\cdot)| \leq h_i$ or $|x_i(\cdot) - x_{ie}(\cdot)| \leq h_i$ as specified in the communication modules, respectively. Furthermore, the disturbance and noise are assumed to be bounded, i.e., $|\bar{v}_i(k)| \leq b_{iv}$, $|\bar{w}_i(k)| \leq b_{iw}$, and $\mathbf{r}(\cdot)$ is a well-defined reference trajectory. Therefore, since $\mathbf{x}(k+1) = [\mathbf{A} - \mathbf{BK}]\mathbf{x}(k)$ is exponentially stable, the closed-loop system is stable by the Bounded-Input-and-Bounded-Output stability property.

Next, we discuss the control performance, that is, the difference between $\mathbf{x}(\cdot)$ and $\bar{\mathbf{x}}(\cdot)$. The two sets of closed-loop systems are rewritten as follows:

$$\begin{aligned}\mathbf{x}(k+1) &= [\mathbf{A} - \mathbf{BK}]\mathbf{x}(k) + \mathbf{BKr}(k) - \mathbf{BKv}(k) + \mathbf{w}(k)\\ \bar{\mathbf{x}}(k+1) &= [\mathbf{A} - \mathbf{BK}]\bar{\mathbf{x}}(k) + \mathbf{BK}\left[\bar{\mathbf{x}}(k) - \hat{\mathbf{x}}(k)\right]\\ &\quad + \mathbf{BKr}(k) - \mathbf{BK}\bar{\mathbf{v}}(k) + \bar{\mathbf{w}}(k).\end{aligned} \tag{9}$$

Therefore, the difference, defined as $\tilde{\mathbf{x}}(\cdot) = \mathbf{x}(\cdot) - \bar{\mathbf{x}}(\cdot)$, can be described as follows:

$$\tilde{\mathbf{x}}(k+1) = [\mathbf{A} - \mathbf{BK}]\tilde{\mathbf{x}}(k) - \mathbf{BK}\left[\bar{\mathbf{x}}(k) - \hat{\mathbf{x}}(k)\right]. \tag{10}$$

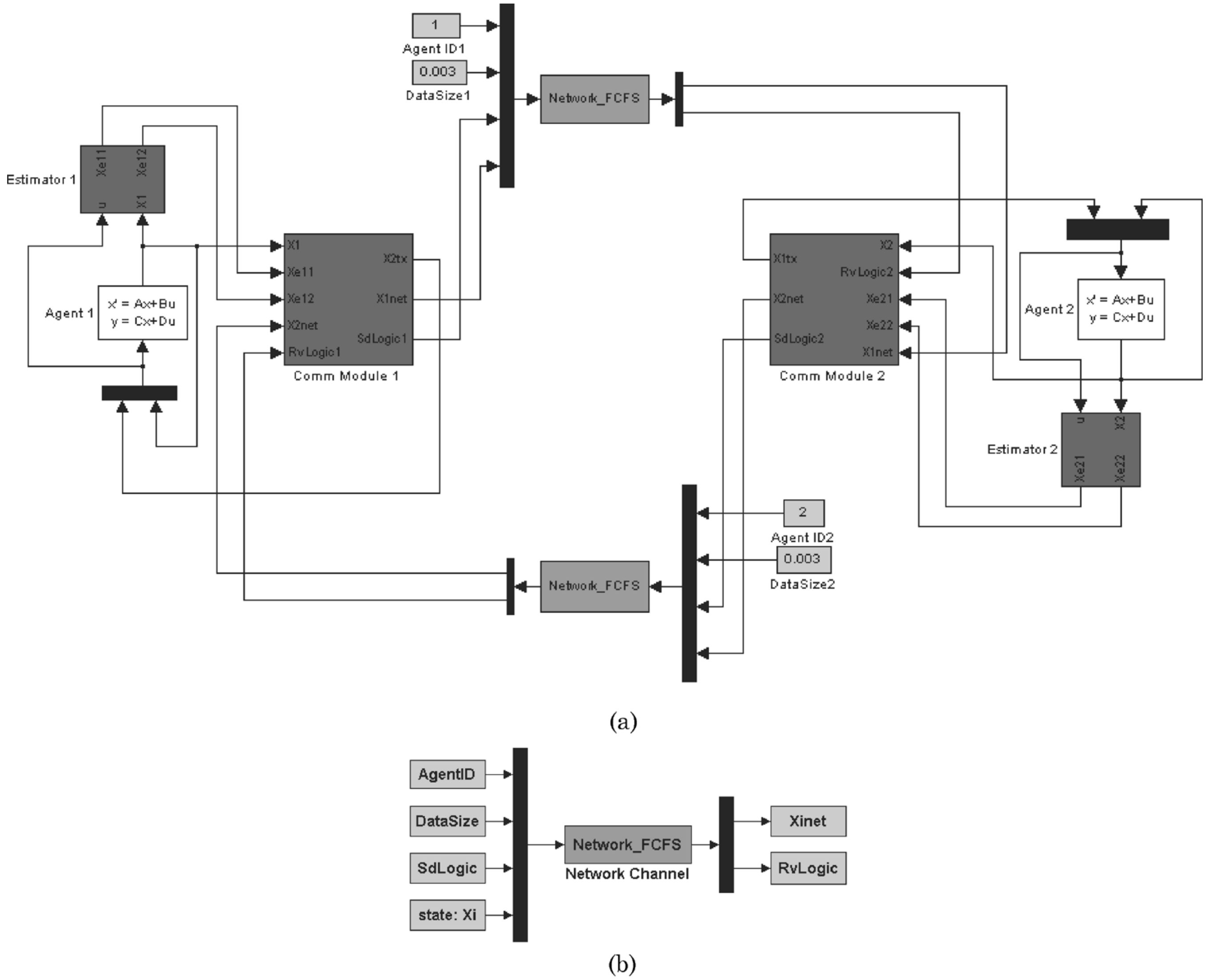


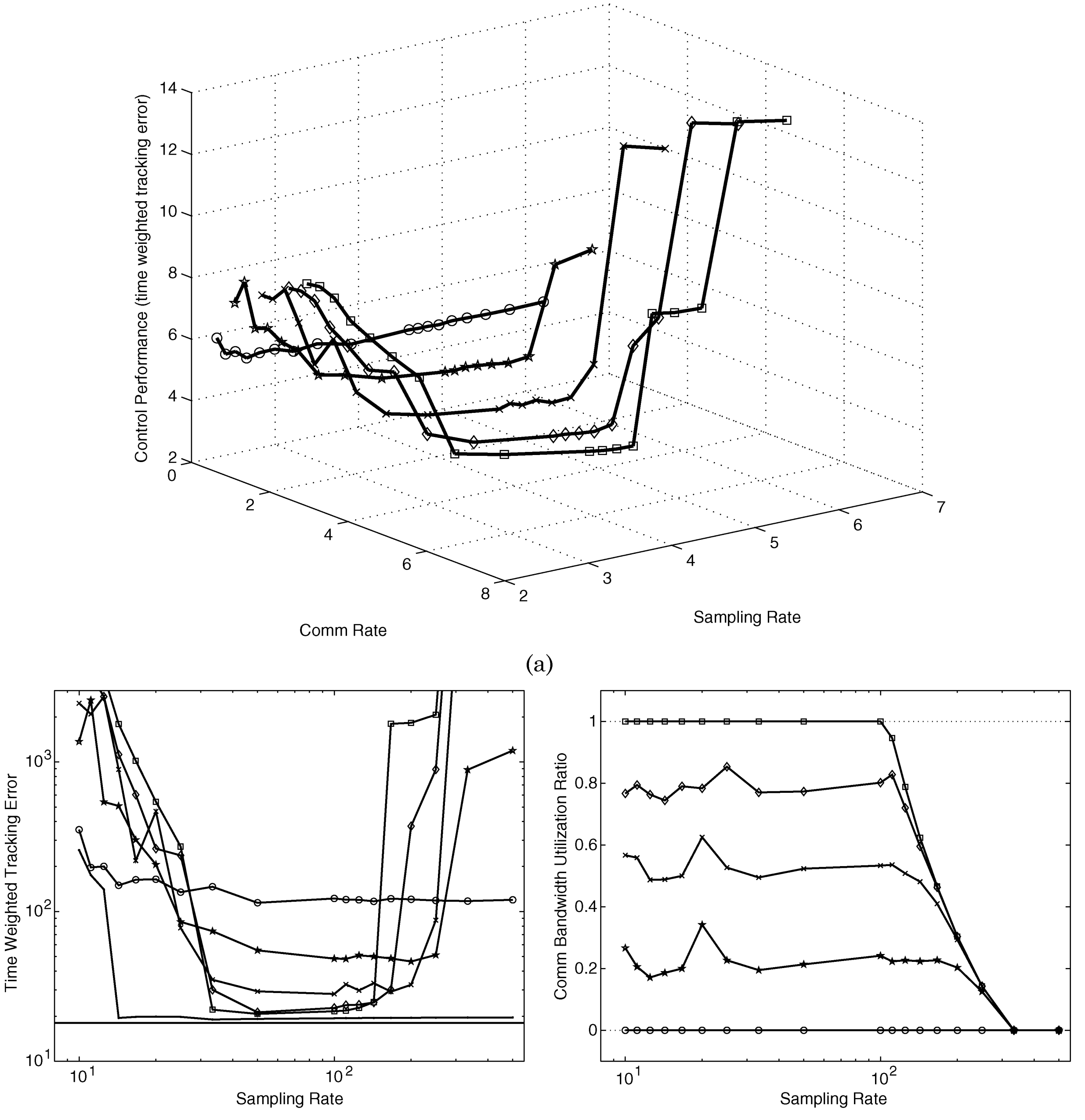
Fig. 10. (a) Example of a networked two-agent system and (b) the network channel.

In the worst case, i.e., assuming $|\bar{x}_i(\cdot) - \hat{x}_i(\cdot)| = h_i$, $\tilde{\mathbf{x}}(\cdot)$ is the steady-state value, if exists, of an exponentially stable system with constant input $\mathbf{h}$. Hence, the maximum value of the state difference $\tilde{\mathbf{x}}(\cdot)$, i.e., the control performance degradation, can be characterized by $\mathbf{A}$, $\mathbf{B}$, $\mathbf{K}$, and $\mathbf{h}$.

The transmission rate also depends on the system and controller dynamics and the threshold $\mathbf{h}$ at the communication modules. For simplicity, we assume the transmission rate in the networked MIMO system is 100%. For the deadband-control case, the transmission rate mainly depends on the difference between the current state $x_i(k)$ and the past states $x_i(k-1)$, $x_i(k-2)$, etc., and the chosen threshold h_i. Hence, by computing the percentage of $|x_i - x_{is}| > h_i$ where x_{is} is one of the past states, the transmission rate can be characterized as a function of the system and controller dynamics, reference trajectory, and the chosen threshold.

For the state-estimator case, the transmission rate mainly depends on the difference between the actual state and the estimated state, i.e., $|x_i - x_{ie}|$, and the chosen threshold h_i. Hence, in addition to the system and controller dynamics, the dynamics of the estimator also play an important role in characterizing the transmission rate. For example, if the estimator can predict the other states perfectly, then no data transmission is required. However, due to the existence of disturbance and noise as well as modeling uncertainty, certain amount of data transmission is required.

In the worst case where the system is highly time-varying or the estimation perform poorly, the transmission rates of both the deadband-control and state-estimation cases are simply equal to that of the networked MIMO system. Hence, the implementation of the deadband control or state estimation locally can reduce the transmission rate, but guarantee an acceptable control performance. On the other hand, the reduction of transmission rate saves the available communication bandwidth for other network usages and improves the overall performance of control and communication. In next section, we discuss an illustrative example on a Matlab/Simulink-based network-and-control simulator that is used to simulate the timing response of control applications over network architectures, and to demonstrate the design methodology discussed in this section.



(a)

(b)

(c)

Fig. 11. (a) Simulation result, (b) control performance versus sampling rate, and (c) utilization of communication bandwidth.

V. ILLUSTRATIVE EXAMPLE

In this section, we demonstrate the proposed design methodologies by illustrating a system with two outputs (i.e., states) and one input. Two separate sets of estimators and communication modules are designed at these outputs and three networked agents of two sensors and one actuator are programmed to compete for the communication bandwidth based on a priority-based network protocol. A simple ramp-type curve is used as a reference trajectory and the sum of time-weighted error between the reference trajectory and the actual trajectory is used as a performance index. The simulation study is implemented in a network and control simulator on the Matlab/Simulink platform. In addition to standard toolboxes for controller design in Simulink, two key elements: network channel and communication module, are constructed in the simulation tool.

The simulation model of a networked two-agent system with state estimators is shown in Fig. 10(a) and the detailed diagram of the network channel is depicted in Fig. 10(b). The inputs from the agent to the communication module are the state (e.g., x_1), and the estimated states (e.g., x_{e11}, x_{e12}). The outputs of the communication module to the network channel are the agent state (x_{1net}) needed to send and the sending interrupt logic (SdLogic1) to tell the network channel to transmit the current state (x_{1net}). The inputs to the network channel are the agent identification number (AgentID), data size (DataSize), and the sending interrupt logic and the current agent state forwarded from the Communication Module.

Currently, there are two choices of network protocols available for the simulation of data transmission over the network: namely, the First Come First Serve (FCFS) and the priority-based Controller Area Network (CAN). Based on the medium access control mechanism of the network protocols, the network channel then outputs the transmitted state (x_{inet}) and the receiving interrupt logic (RvLogic) to notify the communication module of the receiving agent. The communication module then forward the transmitted state to the input of the receiving agent. Note that, although there are two separate blocks of network channel in Fig. 10, they actually compete with the same network medium if an identical network function (e.g., NetworkFCFS) is used. On the other hand, one can set up a networked multi-agent system with multiple network channels by selecting or programming different network functions such as "NetworkFCFS1" and "NetworkFCFS2."

The simulation result of the integrated design chart is shown in Fig. 11(a). These curves represent different threshold values; they are 0, 0.02, 0.05, 0.1, ∞ for $\Box$, $\diamond$, $\times$, $\star$, and $\bigcirc$. As the threshold value increases, the control performance decreases, but the operating range becomes wider as shown in Fig. 11(b), and the utilization of communication bandwidth reduces as shown in Fig. 11(c). The cases with thresholds 0 and ∞ are communicating every sampled data and of no communication, i.e., using estimated data, respectively. Hence, from the simulation result, it can be seen that under the proposed design framework, certain communication bandwidth can be saved and acceptable control performance can be guaranteed. Therefor, networked agents can dynamically adjust their signal transmission rates based on the network traffic condition as well as the designated control performance. The developed simulator can be further programmed for other network protocols and dynamical systems.

VI. CONCLUSIONS

This paper has discussed the network architecture of implementing networked agents as well as other networked agents for achieving the overall system performance of control and communication systems. Those networked agents are operating at the so-called sensor bus or field bus to execute time-critical tasks with real-time continuous system variables. The key components needed in a networked agent and the performance analysis of a network-type system are then addressed. Based on the standard control design framework, this paper also presented the standard and networked MIMO system models for adaptive communication module design. The two key features of the communication design are the integrated network and control design chart, and the utilization of deadband control and state estimation. By visualizing the operating range of a network and control design chart, deadband controller and state estimator can be used to decouple the sampling rate and transmission rate and to dynamically adjust the communication mechanism based on the required control and communication performance. Stability and performance analysis of the integrated control and communication system are also provided. Finally, this paper presented a network and control simulator developed on the Matlab/Simulink platform with commutation modules as well as control dynamics modules. The simulation tool was developed to help analyze and visualize the performance of proposed design schemes. Both the analysis and simulation results showed the tradeoff of designing the adjusting mechanism between control and communication performance.

The future work has two areas. One is to further analyze the tradeoff of selecting the threshold theoretically, that is, formulating the relationship between control system performance such as tracking errors, response time, and communication system performance such as network utilization, transmission delay, and data loss rate. The formulation will be helpful in designing intelligent networked agents to optimally utilize limited communication resource. One the other hard, based on the simulation tool developed, we will be working on designing a general software platform for simulating different network protocols as well as network architectures. The advance of the integrated control and communication simulation tool will be useful for analyzing and designing advanced methodology for distributed control systems with multiple networked agents.

REFERENCES

[1] S. Biegacki and D. VanGompel, "The application of DeviceNet in process control," *ISA Trans.*, vol. 35, no. 2, pp. 169–176, 1996.

[2] L. H. Eccles, "A smart sensor bus for data acquisition," *Sensors*, vol. 15, no. 3, pp. 28–36, 1998.

[3] L. D. Gibson, "Autonomous control with peer-to-peer I/O networks," *Sensors*, vol. 12, no. 9, pp. 83–90, Sep. 1995.

[4] A. Ray, "Introduction to networking for integrated control systems," *IEEE Control Syst. Mag.*, vol. 9, no. 1, pp. 76–79, Jan. 1989.

[5] G. Schickhuber and O. McCarthy, "Distributed fieldbus and control network systems," *Comput. Control Eng.*, vol. 8, no. 1, pp. 21–32, Feb. 1997.

[6] D. Song, T. Divoux, and F. Lepage, "Design of the distributed architecture of a machine-tool using FIP fieldbus," in *Proc. IEEE Int. Conf. on Application-Specific Systems, Architectures, Processors*, 1996, pp. 250–260.

[7] G. Cena, L. Durante, and A. Valenzano, "Standard field bus networks for industrial applications," *Comput. Stand. Interfaces*, vol. 17, pp. 155–167, Jan. 1995.

[8] W. Henning, "Bus systems," *Senso. Actuators A: Phys.*, vol. 25, no. 1–3, pp. 109–113, Oct.–Jan. 1990–1991.

[9] R. S. Raji, "Smart networks for control," *IEEE Spectrum*, vol. 31, no. 6, pp. 49–55, Jun. 1994.

[10] S. Cavalieri, A. D. Stefano, and O. Mirabella, "Impact of fieldbus on communication in robotic systems," *IEEE Trans. Robot. Automat.*, vol. 13, no. 1, pp. 30–48, Feb. 1997.

[11] T. Virvalo and P. Puusaari, "New solution of motion control problem in mechatronic system," *Comput. Elect. Eng.*, vol. 18, no. 1, pp. 41–50, Jan. 1992.

[12] F. Göktas, J. M. Smith, and R. Bajcsy, "μ-synthesis for distributed control systems with network-induced delays," in *Proc. 35th Conf Decision and Control*, Dec. 1996, pp. 813–814.

[13] R. Krtolica, Ü. Özgüner, H. Chan, H. Göktas, J. Winkelman, and M. Liubakka, "Stability of linear feedback systems with random communication delays," in *Proc. Amer. Control Conf.*, June 1991, pp. 2648–2653.

[14] P. Pleinevaux and J.-D. Decotignie, "Time critical communication networks: field buses," *IEEE Network*, vol. 2, no. 3, pp. 55–63, May 1988.

[15] B. Wittenmark, J. Nilsson, and M. Torngren, "Timing problems in real-time control systems," in *Proc. Amer. Control Conf.*, 1995, pp. 2000–2004.

[16] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[17] A. S. Tanenbaum, *Computer Networks*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[18] K. Feldmann, C. Schnur, and W. Colombo, "Modularised, distributed real-time control of flexible production cells, using Petri net," *Control Eng. Pract.*, vol. 4, no. 8, pp. 1067–1078, Aug. 1996.

[19] E. Park, D. M. Tilbury, and P. P. Khargonekar, "Modular logic controllers for machining systems: formal representation and performance analysis using Petri nets," *IEEE Trans. Robot. Automat.*, vol. 15, no. 6, pp. 1046–1061, Dec. 1999.

[20] H. Chan and Ü. Özgüner, "Closed-loop control of systems over a communications network with queues," *Int. J. Control*, vol. 62, no. 3, pp. 493–510, Sep. 1995.

[21] Y. Halevi and A. Ray, "Integrated communication and control systems: part I—analysis," *ASME J. Dynam. Syst., Meas. Control*, vol. 110, no. 4, pp. 367–373, Dec. 1988.

[22] J. Nilsson, B. Bernhardsson, and B. Wittenmark, "Stochastic analysis and control of real-time systems with random time delays," *Automatica*, vol. 34, no. 1, pp. 57–64, Jan. 1998.

[23] G. C. Walsh, H. Ye, and L. Bushnell, "Stability analysis of networked control systems," in *Proc. Amer. Control Conf.*, June 1999, pp. 2876–2889.

[24] N. Najafi and J. R. Moyne, "Smart sensors," in *Semiconductor Micromachining*, S. A. Campbell and H. J. Lewerenz, Eds. New York: Wiley, 1998.

[25] G. Y. Tian, Z. X. Zhao, and R. W. Baines, "A fieldbus-based intelligent sensor," *Mechatronics*, vol. 10, pp. 835–849, Dec. 2000.

[26] A. Kawamura, J. Hirai, and Y. Aoyama, "Autonomous decentralized manufacturing system using high-speed network with inductive transmission of data and power," in *Proc. IEEE 22nd Int. Conf. Industrial Electronics, Control, and Instrumentation*, Aug. 1996, vol. 2, pp. 940–945.

[27] Sensor Bus Conformance Testing Laboratory, Univ. Michigan. Ann Arbor, [Online] Available [Online]. Available: http://www.eecs.umich.edu/~sbus

[28] G. F. Franklin, J. D. Powell, and M. L. Workman*, Digital Control of Dynamic Systems*, 3rd ed. Reading, MA: Addison-Wesley, 1998.

[29] F.-L. Lian, J. R. Moyne, and D. M. Tilbury, "Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet," *IEEE Control Syst. Magazine*, vol. 21, no. 1, pp. 66–83, Feb. 2001.

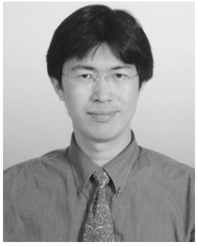
[30] ——, "Network design consideration for distributed control systems," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 2, pp. 297–307, Mar. 2002.

[31] P. G. Otanez, J. R. Moyne, and D. M. Tilbury, "Using deadbands to reduce communication in networked control systems," in *Proc. 2002 Amer. Control Conf.*, Anchorage, AK, May 8–10, 2002.

[32] J. K. Yook, D. Tilbury, and N. R. Soparkar, "Trading computation for bandwidth: reduced communication in distributed control systems using state estimators," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 4, pp. 503–518, Jul. 2002.

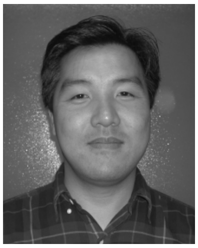
[33] F.-L. Lian, J. R. Moyne, and D. M. Tilbury, "Modeling and optimal controller design of networked control systems with multiple delays," *Int. J. Control*, vol. 76, no. 6, pp. 591–606, Apr. 2003.

[34] F.-L. Lian, J. K. Yook, P. Otanez, D. M. Tilbury, and J. R. Moyne, "Design of sampling and transmission rates for achieving control and communication performance in networked multi-agent systems," in *Proc. 2003 Amer. Control Conf.*, Denver, CO, Jun. 4–6, 2003.



Feng-Li Lian (M'02) received the B.S. and M.S. degrees from National Taiwan University (NTU), Taipei, Taiwan, R.O.C., in 1992 and 1994, respectively, and the Ph.D. degree from the University of Michigan, Ann Arbor, in 2001.

From 2001 to 2002, he was a postdoctoral scholar at the California Institute of Technology, Pasadena. Since 2002, he has been an Assistant Professor, Department of Electrical Engineering, NTU. His areas of research include distributed and networked control systems, multiple dynamical agent systems, optimization, and statistical process control.



John K. Yook was born in Jeon-Ju, Korea. He received the B.S. degree in mechanical engineering from the University of California, Berkeley, in 1992, and the Ph.D. degree in mechanical engineering from the University of Michigan, Ann Arbor, in 2000.

He was a Design/Production Engineer. Since 2000, he has worked in the Electronic System Department at Raytheon, El Segundo, CA. His research interests include distributed control of mechanical systems with network communication, reconfigurable machine tool control, electro-optical sensor, and satellite line-of-sight control.

Dawn M. Tilbury (M'95) received the B.S. degree in electrical engineering (summa cum laude) from the University of Minnesota, Minneapolis, in 1989, and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1992 and 1994, respectively.

In 1995, she joined the faculty of the Mechanical Engineering Department, University of Michigan, Ann Arbor, where she is currently an Associate Professor. She is coauthor (with Joseph Hellerstein, Yixin Diao, and Sujay Parekh) of the book *Feedback Control of Computing Systems* (Piscataway NJ: Wiley-IEEE Press, 2004). Her research interests include distributed control of mechanical systems with network communication, logic control of manufacturing systems, and performance management and control of computing systems.

Dr. Tilbury won the EDUCOM Medal (jointly with Professor William Messner of Carnegie Mellon University) in 1997 for her work on the web-based *Control Tutorials for Matlab*. An expanded version, *Control Tutorials for Matlab and Simulink*, was published by Addison-Wesley in 1999. She received an NSF CAREER award in 1999, and is the 2001 recipient of the Donald P. Eckman Award of the American Automatic Control Council. She is a member of ASME and SWE, and is a member of the 2004–2005 class of the Defense Science Study Group (DSSG).

James Moyne (M'90) received the B.S.E.E. and B.S.E.—Mathematics degrees, and the M.S.E.E. and Ph.D. degrees from the University of Michigan, Ann Arbor.

He is currently an Associate Research Scientist, Department of Mechanical Engineering and Computer Science, and Director of the Reconfigurable Factory Testbed within the Engineering Research Center for Reconfigurable Manufacturing Systems, University of Michigan. He is also Director of Automotive Technology at Brooks Automation. His research focus areas include industrial network systems, advanced process control, software control, and database technology. He is the author of a number of refereed publications in each of these areas. He holds the patent on a software control framework enabling technology, is coauthor of *Run-to-run Control in Semiconductor Manufacturing* (Boca Raton, FL: CRC Press), and is author of a number of semiconductor manufacturing standards in the areas of sensor bus, process control, and communications.