Design of Sampling and Transmission Rates for Achieving Control and Communication Performance in Networked Agent Systems[1]

Feng-Li Lian^a, John Yook^b, Paul Otanez^c, Dawn Tilbury^c, James Moyne^c
(a) National Taiwan University, (b) Raytheon Electronic Systems, (c) The University of Michigan

Abstract

When sensory and actuation agents 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. In this paper, we analyze the performance of control actions over distributed networks, and propose a design methodology of achieving control and communication performance in a networked agent 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, several estimation algorithms are also implemented to achieve both control and communication performance. Simulation studies are conducted in a network-and-control simulator to demonstrate the networked control systems characteristics investigated.

1 Introduction

When dealing with 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. Hence, the mechanism of communicating information plays an important role on the stability and performance of the control systems implemented over communication networks. Typical examples of networked agent systems include Intelligent Transportation Systems (ITS), Integrated Manufacturing Systems (IMS), Battlefield Management Systems (BMS), etc.

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. Agent design for the state-based systems is simple and straightforward. For example, 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 agents. In a small-scaled and centralized system, it should be not a big problem. However, in a large-scaled or distributed system, the required communication load might degrade the overall performance or destabilize the system if without further design consideration.

On the other hand, events could be the abstract information of detailed control actions. The line-change maneuver of one vehicle in ITS, the processing schedule of one product in IMS, the formation change of unmanned vehicles in BMS are examples of events. Based on the event information received, each agent needs to first recover the original content of the abstract information, and then let local state-based controllers react upon it. The system performance mainly focuses on the logical correctness of events, and relies on the well-designed local controllers. In this case, the communication amount needed might be decreased, but the performance of a time-critical control applications 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 systems with time-critical applications. For satisfying both control and communication performance, several design methodologies are adopted to generate proper control actions and save communication bandwidth. This paper consists six sections, including the Introduction section. Section 2 describes problem formulation. Section 3 discusses the design methodology, including deadband control and state estimation. Section 4 outlines a network and control simulator that is used to simulate communication networks as well as control actions. Section 5 presents an illustrative example of a networked two-agent sys-

[1] This research was supported in part by the NSF Engineering Research Center for Reconfigurable Machining Systems under the grant EEC95-92125, the DARPA-IXO Mixed Initiative Control of Automa-teams Program, and the National Science Council, Taiwan, ROC, under the grant NSC 91-2213-E-002-133. Corresponding Author: Feng-Li Lian, Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan.

0-7803-7896-2/03/$17.00 ©2003 IEEE

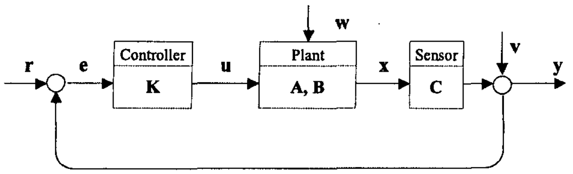


Figure 1: A standard MIMO closed-loop system.

tem. Summary and future work are provided in Section 6.

2 Problem Formulation

2.1 A standard MIMO system

In this paper, we 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. 1 and described as follows:

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

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. $\mathbf{w}(\cdot)$ and $\mathbf{v}(\cdot)$ are system disturbance and measurement noise, respectively, and are assumed to be bounded, i.e., $\mathbf{w}(\cdot) \leq \mathbf{b}_w$ and $\mathbf{v}(\cdot) \leq \mathbf{b}_v$ For ease of presentation, we assume that $\mathbf{C} = \mathbf{I}$, that is, all the states are measurable. The state feedback controller for system (1) can then be designed by any standard MIMO control design technique as follows:

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

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

2.2 A networked MIMO system

Next, we consider a distributed control architecture where sensors, actuators, and controllers are physically distributed and exchanging data through communication networks as shown in Fig. 2. 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. Each sampled data has an inevitable waiting time, i.e., time delay, because of the sharing of communication media. 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.

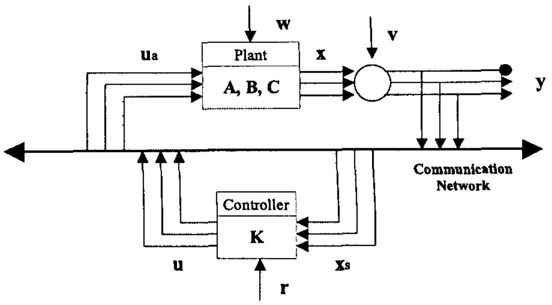


Figure 2: A MIMO control system with communication of control signals over distributed networks.

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

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

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

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 mis-matched sampling instants of the sensory and 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 [3, 5].

In this paper, we focus on the performance analysis of 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. 2 and the framework of Eqs. (3) and (4).

3 Design Methodology

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 control performance. In this section, we summarize the key characteristics of designing a networked control system in general. In particular, we discuss the network design considerations of network and control parameters in terms of networked control design chart. To achieve

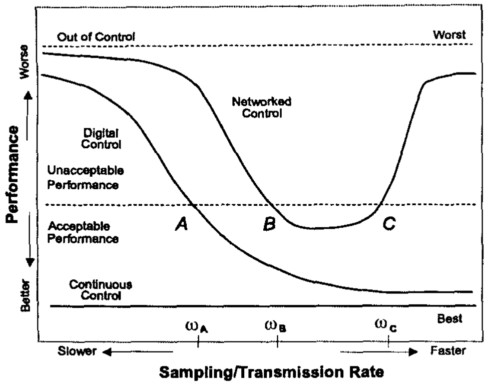


Figure 3: Performance comparison of continuous control, digital control, and networked control cases.

both the control and network performance under the limited communication bandwidth, we utilize state estimator approaches to actively adjust the communication rates and guarantee designated control performance. In the end, an integrated network-and-control design chart will be used to help visualize the overall system performance and select design parameters.

3.1 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. An in-depth performance evaluation of several popular control networks can be found in [2] and a detailed analysis of these timing components can be found in [4]. Here, we summarize the results presented in the two references in terms of network and control parameters.

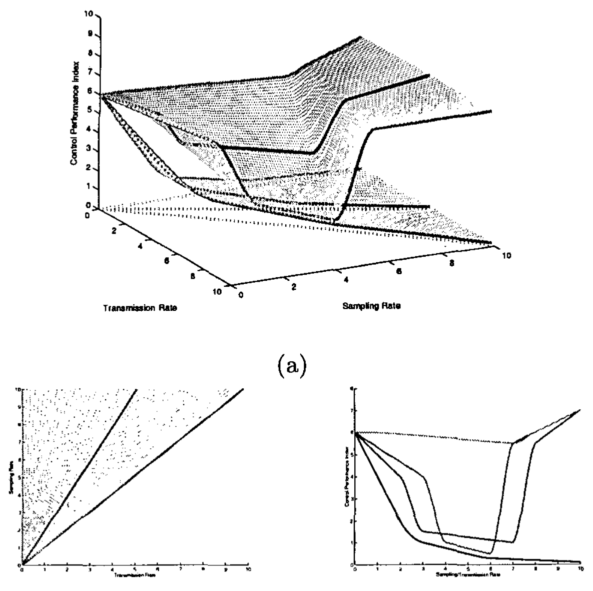
There are three important control parameters: the sampling rate, control system bandwidth, and phase margin that characterize control performance closely and have a strong influence on the network parameters. Since a networked agent system is essentially a discrete time system, choosing a proper sampling 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. 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.

3.2 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. 3 [4]. This design chart provides a clear way to choose the proper sampling or transmission rates for a networked agent system. Fig. 3 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. 3 could be chosen to reflect a subset of these metrics.

Given a set of network and control parameters, several fundamental formulae are discussed in [4] to characterize the design chart. To guarantee the best control performance, all the agents need the newly updated data from other agents. Hence, all the scenarios considered in [4] are assumed to have an identical sampling and transmission rate, that is, to transmit every sampled data to the destination agent. 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 network-and-control design chart of Fig. 3 that decouples the sampling and transmission rates.

The schematic view of a revised integrated network-and-control design chart is shown in Fig. 4. 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 agent. Based on a designated controller algorithm, the communication module decides whether it needs to broadcast the newly sampled data to other agents or not. In Fig. 4(a), the red line is the case where the sampling rate equals the transmission rate, the blue line is the case where the transmission rate is only a half of the sampling rate, and the green line is the case of no transmission. The schematic view of the ratio of the sampling and transmission rates can be seen in Fig. 4(b). The lower-right region, i.e., transmission rate > 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. 4(c), the side view of Fig. 4(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.



(a)

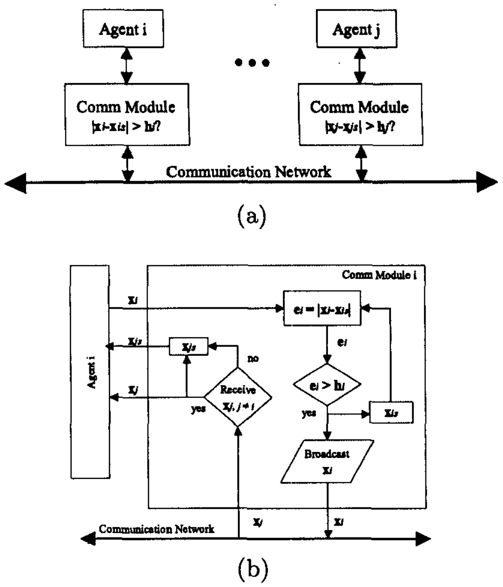
(b) (c)

Figure 4: (a) A schematic diagram of a revised network-and-control design chart. (b) the top view: the ratio of the sampling and transmission rates. (c) the side view: the reduction of transmission rate and control performance degradation.

Therefore, by analyzing the required control specification, we will study two communication control algorithms to separately, adaptively adjust the sampling and transmission rates in Sections 3.3 and 3.4.

3.3 Using deadband control to decouple sampling and transmission rates

In this section, we discuss the implementation of a deadband controller framework, studied in [6] and adopted here for the proposed networked agent system. As shown in Fig. 5, the deadband control is implemented at each individual agent such as sensor, actuator, and controller. An 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, agents 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 trade-off between control and communication performance. Therefore, by properly selecting the deadband thresholds of all networked agents, an optimal performance of control and communication can be achieved.



(a)

(b)

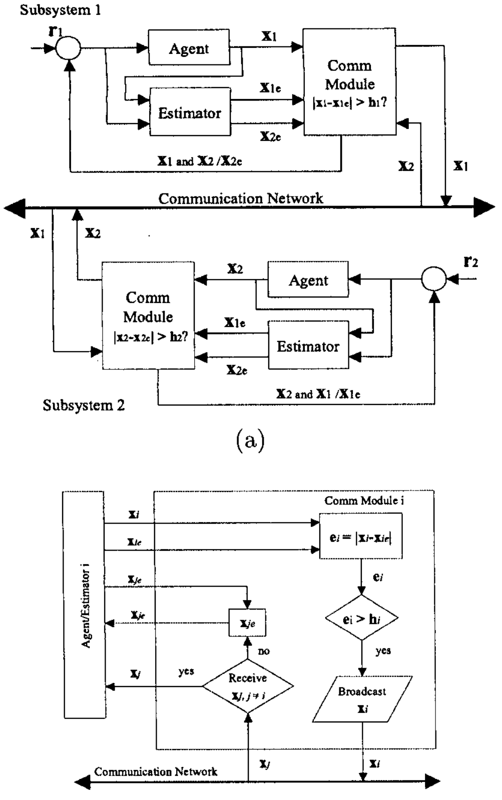
Figure 5: (a) A networked agent system with deadband-control communication module. (b) the schematic diagram of the communication module.

Since an agent with deadband control adjusts its transmission rates based on its own state, the deadband control framework is only suitable for the system with slowly time-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 impact on the improvement of communication performance. In next section, we will further implement an estimator of the states of all networked agents and a modified communication module that adaptively adjusts the state transmission based on the actual and estimated states.

3.4 Using state estimator to decouple sampling and transmission rates

In this section, we discuss a state estimator framework studied in [7] 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. 6(a), and the schematic diagram of the Control/Comm module is depicted in Fig. 6(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.

Next to the agent is one estimator which computes the states of the agent and the other agents based on well-designed estimation models. The main functionality of



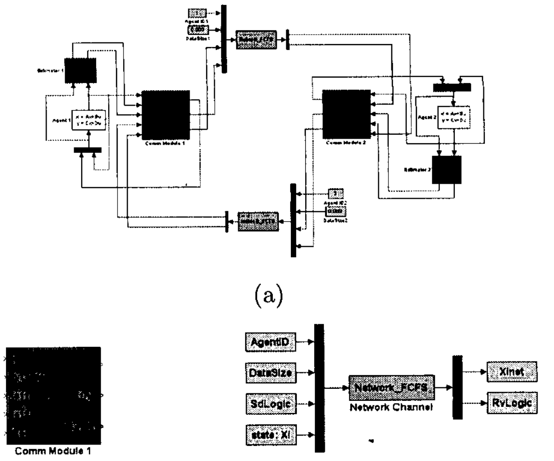
(a)

(b)

Figure 6: (a) A networked two-agent system with state estimator and communication module. (b) the schematic diagram of the communication module.

the Control/Comm module is to compute the difference of the true and estimated states of the agent, control the communication frequency, and update the estimated states by the true states of other agents. For example, Estimator 1 in Fig. 6 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 pre-defined threshold, say h_1. Similar estimation and communication mechanisms are designed at Agent 2 or 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 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-estimator cases are simply equal to that of the networked MIMO system. Hence, the implementation of the deadband control or state estimator locally can reduce the transmission rate, but guarantee a acceptable control performance. On the other hand, the reduction of transmission rate save the available communication bandwidth for other network usages and improve the overall performance of control and communication.



(a)

(b)

(c)

Figure 7: (a) An example of a networked two-agent system, (b) The communication module, and (c) The network channel.

4 Network and Control Simulator (NCsim)

In this section, we discuss the development of a network and control simulator, called NCsim, on the Matlab/Simulink platform. In addition to standard toolboxes for controller design in Simulink, two key elements in NCsim are network channel and communication module.

An example of a networked two-agent system with state estimators is shown in Fig. 7(a) and the detailed diagram of the communication module and network channel is depicted in Fig.7(b) and (c), respectively. The inputs from the agent to the communication module are the agent state (e.g., x_1), and the estimated states (e.g., x_1, x_2). The outputs of the communication module to the network channel are the agent state (x_{1net}) needed to send and the sending interrupt (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 and the current agent state forwarded from the communication module.

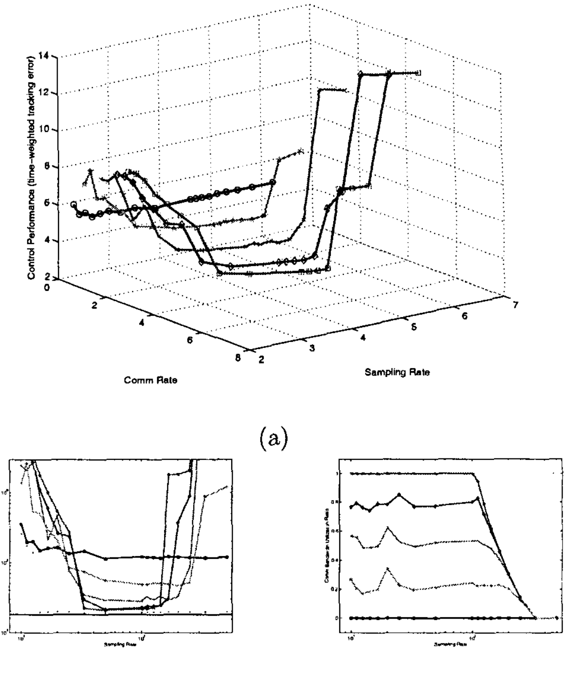
Currently, there are two choices of network protocols: namely, the first-come-first-serve (FCFS) and the priority-based Controller Area Network (CAN). Based on the medium access control mechanism of the network protocols, the network channel then outputs the transmitted state (x_{inet}) and the receiving interrupt (RvLogic) to notify the communication module of the receiving agent. The communication module then forwards the transmitted state to the input of the receiving agent. Note that, although there are two separate blocks of network channel in Fig. 7, they actually compete with the same network medium if one single network function (e.g., NetworkFCFS) is used. On the other hand, one can set up a networked agent system with multiple network channels by programming different network functions such as "NetworkFCFS1" and "NetworkFCFS2."

5 Illustrative Examples

In this section, we consider a system with two outputs (i.e., states) and one input. Two separate estimators and communication modules are designed at these outputs and three agents of two sensors and one actuator are programmed to compete for the communication bandwidth based on the priority-based network protocol. A simple ramp-type curve is used for 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 result of the integrated design chart is shown in Fig. 8(a). These curves represent different threshold values; they are $0, 0.02, 0.05, 0.1, \infty$ for red, blue, pink, green, and black curves. As the threshold value increases, the control performance decreases, but the operating range becomes wider and the utilization of communication bandwidth reduces as shown in Fig. 8(b) and (c). Hence, from the simulation result, it can be seen that under the proposed networked agent framework, certain communication bandwidth can be saved and acceptable control performance can be guaranteed.

6 Summary and future work

In this paper, we described the design methodologies for the networked agent systems to achieve acceptable control and communication performance. The decoupling of sampling and transmission mechanisms is done through deadband control and state estimation designed locally at each agent. A network and control simulator with commutation modules as well control dynamics modules was also developed to help analyze the performance of proposed design schemes. The simulation result showed the trade-off between control and communication performance. Hence, the future work includes characterizing their relationship precisely and designing adaptive threshold selector.



(a)

(b)

(c)

Figure 8: (a) Simulation result, (b) control performance v.s. sampling rate, and (c) utilization of communication bandwidth.

References

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

[2] F.-L. Lian, J.R. Moyne, and D.M. Tilbury, "Performance Evaluation of Control Networks: Ethernet, ControlNet, and DeviceNet," *IEEE Control Systems Magazine*, 21(1): 66-83, Feb. 2001.

[3] F.-L. Lian, J.R. Moyne, and D.M. Tilbury, "Analysis and Modeling of Networked Control Systems: MIMO Case with Multiple Time Delays," *2001 ACC*, Arlington, Virginia, June 25-27, 2001.

[4] F.-L. Lian, J.R. Moyne, and D.M. Tilbury, "Network Design Consideration for Distributed Control Systems," *IEEE Transaction on Control Systems Technology*, 10(2): 297-307, Mar. 2002.

[5] F.-L. Lian, J.R. Moyne, and D.M. Tilbury, "Optimal Controller Design and Evaluation for a Class of Networked Control Systems with Distributed Constant Delays," *2002 ACC*, Anchorage, Alaska, May 8-10, 2002.

[6] P.G. Otanez, J.R. Moyne, and D.M. Tilbury, "Using Deadbands to Reduce Communication in Networked Control Systems," *2002 ACC*, Anchorage, Alaska, May 8-10, 2002.

[7] J.K. Yook, D. Tilbury, H.S. Wong, and N.R. Soparkar, "Trading Computation for Bandwidth: State Estimators for Reduced Communication in Distributed Control Systems," *2000 Japan-USA Symposium on Flexible Automation*, July 23-26, 2000.