Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

Wireless Medium Access Control in Networked Control Systems

Xiangheng Liu and Andrea Goldsmith
Dept. of Electrical Engineering, Stanford University, Stanford, CA 94305-9515, USA
Email: {liuxh,andrea}@wsl.stanford.edu

***Abstract*— We present a framework to study the performance of networked control systems where the feedback control loops are closed over a shared wireless network. In particular, we study the effects of the wireless network medium access control (MAC) protocols on the performance of the networked control systems. We consider a joint performance index of all the systems sharing the wireless channel. We compare several wireless network MAC protocols with a numerical example of two inverted pendulum systems. We show that the control performance degrades due to distributed medium access control in addition to the performance degradation due to communication faults including limited data rates, random delay and packet losses.**

I. Introduction

Networked control has become an enabling technology for many military, commercial and industrial applications. Information among distributed sensors, controllers and actuators must be exchanged over a communication network. Wireless communication is playing an increasingly important role in such distributed systems. Transmitting sensor measurements and control commands over wireless links allows rapid deployment, flexible installation, fully mobile operation and prevents the cable wear and tear problem in industrial automation.

Building a networked control system over wireless is a challenging task. The scarce spectrum imposes a fundamental limit on the performance of the wireless channel. Random delays and packet losses are inevitable. Even though this is true for any communication network, it is much more pronounced in wireless networks due to limited spectrum and power, time-varying channel gains and interference. Besides the design challenges in the link layer, multiple transmitter and receiver pairs need to share the channel and an efficient channel access mechanism needs to be designed. As we shall see, random access without centralized control can waste a great deal of the scarce resources in the network.

There are numerous different research directions that are worth pursuing in networked control system designs. A very imminent need, in the authors' opinion, is to design controllers that are robust to the communication faults including random delays and packet losses. However, to the best of the authors' knowledge, there is little theory to design such controllers. Our research focuses on the design of the wireless communication network such that the performance degradation due to imperfect communication is minimized. In our previous work [5], we studied the link layer trade-offs in the communication designs and showed that the choice of data rate, error correction coding and the maximum number of retransmissions can greatly affect the performance of a networked control system. We also found that the optimal selection of the control parameters, such as the sample period, should depend on the choice of communication parameters. In this prior work, we only considered the control system whose sensor measurements are transmitted to the controller over wireless links. In our follow-up paper [6], we extended the framework to include the wireless link from the controller to the actuator such that all the information exchange is done wirelessly. We compared the performance of a single control system when different MAC protocols are adopted by different transmitter/receiver pairs sharing the channel. In this paper, we further generalize the framework to include multiple control systems sharing a wireless network as shown in Figure 1. Different sensors can be at different locations and their measurements need to be encoded and transmitted separately over different wireless links to the controller. The information exchange between the controller and the actuator also requires a wireless link when they are not co-located. Multiple closed-loop control systems can coexist to share the channel. Thus, multiple transmissions may be initiated simultaneously and a MAC protocol is needed to determine which transmission shall take place.

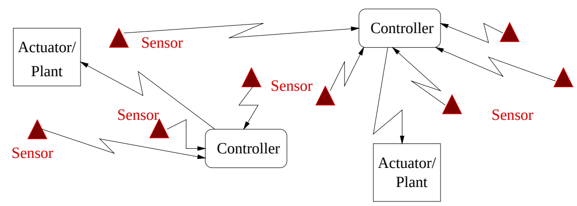

Fig. 1. Networked Control Systems

Other interesting work that addresses similar scenarios includes [12] where the author considered an ALOHA MAC protocol for multiple control systems sharing the channel and gave sufficient and necessary conditions for all the systems to stay stable simultaneously. Previous work that has a similar design principle includes [11] where data rate tradeoffs among multiple sensor measurements are studied. Some network architecture design considerations for distributed control were discussed in [4]. Specifically, a working range of sample periods was determined to achieve acceptable control performance based on the trade-offs between sample periods and network induced delays.

This work is supported under NSF grant CMS-0120912.

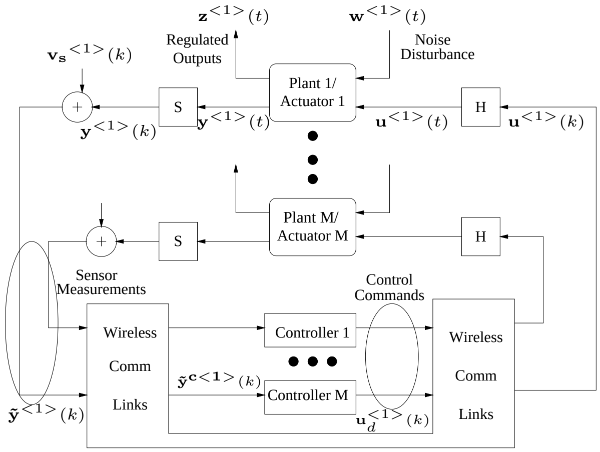

Fig. 2. System Model

II. System Model and Assumptions

Our system model is outlined in Figure 2. We assume all the plants are continuous-time linear time-invariant (LTI) systems while all the controllers are discrete-time. Sensor measurements $\tilde{\mathbf{y}}^{<\mathrm{n}>}$ and controller commands $\mathbf{u}_d^{<\mathrm{n}>}$ need to be sent over a shared wireless network. The superscript $<\mathrm{n}>$ refers to the signals of the n^{th} system. The time index t and k are for continuous-time signals and discrete-time signals, respectively. In this section, we describe our model of the control systems, the communication link design and the MAC protocols that determine the channel sharing.

A. Control System Model

All the plants are continuous-time LTI systems and we can represent the n^{th} system with the following state space equations:

$$\begin{cases} \dot{\mathbf{x}}^{<\mathrm{n}>}(t) = \mathrm{A}^{<\mathrm{n}>}\mathbf{x}^{<\mathrm{n}>}(t) + \mathrm{B}_1^{<\mathrm{n}>}\mathbf{w}(t) + \mathrm{B}_2^{<\mathrm{n}>}\mathbf{u}(t), \\ \mathbf{z}^{<\mathrm{n}>}(t) = \mathrm{C}_1^{<\mathrm{n}>}\mathbf{x}^{<\mathrm{n}>}(t) + \mathrm{D}_{12}^{<\mathrm{n}>}\mathbf{u}^{<\mathrm{n}>}(t), \\ \mathbf{y}^{<\mathrm{n}>}(t) = \mathrm{C}_2^{<\mathrm{n}>}\mathbf{x}^{<\mathrm{n}>}(t). \end{cases}$$

Here $\mathbf{x}^{<\mathrm{n}>}(t)$ is the system state, $\mathbf{w}^{<\mathrm{n}>}(t)$ is the disturbance acting on the plant, $\mathbf{u}^{<\mathrm{n}>}(t)$ is the control force, $\mathbf{y}^{<\mathrm{n}>}(t)$ is the measured output and $\mathbf{z}^{<\mathrm{n}>}(t)$ is the regulated output. The regulated output $\mathbf{z}^{<\mathrm{n}>}(t)$ usually depends on the performance measure. Note that all boldface variables are vectors. Since all the systems have the same state-space representations, we drop the superscript $<\mathrm{n}>$ except when needed for clarification.

Each measured output is sampled every sample period T and we denote these samples as $\mathbf{y}(k) = \mathbf{y}(kT)$. Different systems may have different sample periods. When this is the case, the traffic patterns in the network may no longer be periodic. Thus the associated delay distributions can be non-stationary. In this work, we assume all the systems sharing a common network have the same sample period for simplicity.[1] The measurement noise $\mathbf{v_s}(k)$ is additive Gaussian with covariance matrix Σ_s. Thus, the discrete-time samples $\tilde{\mathbf{y}}(k) = \mathbf{y}(k) + \mathbf{v_s}(k)$ are sent through the communication network. The controller calculates the desired control command $\mathbf{u}_d(k)$ as a function of the decoded sensor measurements and the delays associated with their transmissions. The transmission delays depend on the MAC protocol of the network, the network traffic patterns and the channel conditions. Note that $\mathbf{u}_d(k)$ depends on $\mathbf{y^c}(k-1)$ but not on $\mathbf{y^c}(k)$ due to causality. The desired controller output $\mathbf{u}_d(k)$ is also transmitted over a wireless link. The received control signal $\mathbf{u}(k)$ is then converted to a continuous time signal via a zero order hold. Thus $\mathbf{u}(t) = \mathbf{u}(k)$ for $kT \leq t < (k+1)T$ and $\mathbf{u}(t)$ directly acts on the plant. We assume the controller updates its control command $\mathbf{u}_d(k)$ right before it grabs the channel to transmit to the actuator. The control command is updated as if the next transmission to the actuator would be successful. We allow the control algorithms to depend on the time delay incurred in the control commands $\mathbf{u}_d(k)$, thus the controllers are time-varying. We assume the actuators are event driven: the actuator updates the control input to its plant upon the receipt of the control command from the controller provided that the control command is updated based on the full sensor measurements. Otherwise, the actuator updates at the end of the time slot with whatever control command is available.

B. Wireless Channel and Link Model

We consider a discrete time channel with stationary, ergodic, slowly time-varying gain $\sqrt{g_i(k)}$ and additive white Gaussian noise (AWGN) $n_i(k)$, where the subscript i refers to the i^{th} link and k refers to the k^{th} time instant. In this paper, our analysis will be based on static channel gains. This is justified by the assumption of very slow fading where the channel coherence time (the time over which the channel remains roughly constant) is long enough so that the control system converges to steady-state within a coherence time interval. We assume that the channel power gain $g_i(k)$ is independent of the channel input and the transmission power P_i does not change as the channel gain varies.

Different link layer design choices (coding, modulation, etc.) lead to different performance for the data transfer. We assume a simple class of communication link designs as shown in Figure 3. The figure shows the wireless link from a sensor to a controller. We assume the same link model for the wireless links including the links from controllers to the actuators with the input of $\mathbf{u}_d(k)$ and the output of $\mathbf{u}(k)$. At the transmitter, the data is first quantized and converted into a binary bit stream via a uniform quantizer [2][5]. The bit stream then goes through the channel coder that uses linear block codes for error correction and error detection. We assume BPSK modulation at the

[1] When the ratio of the sample period and time slot length are rational for all the systems, the delay distribution will be cyclostationary. We are currently extending this work to include the cyclostationary scenario.

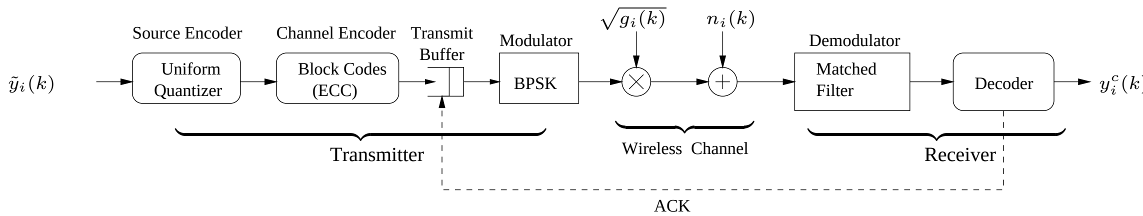

Fig. 3. Wireless Communication Link Model with Hard Decoding

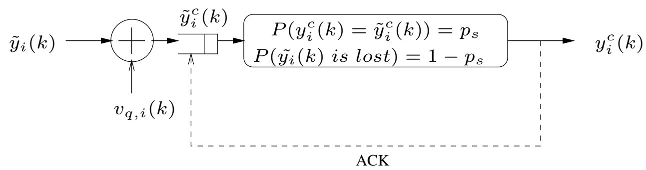

Fig. 4. Simplified Model for Hard Decoding Communication Link

transmitter. At the receiver, we assume matched filtering followed by a maximum likelihood detector. The probability of successful transmission p_s for each packet can be easily calculated given the link design, wireless channel gain and transmit power. We assume time is slotted and we allow retransmissions if there are extra time slots. An optional feedback channel from the receiver to the transmitter may exist. With such a feedback channel, the receiver sends an "ACK" to the transmitter upon a successful transmission. If the transmitter receives the ACK, it clears its transmit buffer and does not transmit until a new packet arrives. We assume the transmit buffer only has a capacity of one data packet. Thus a packet will be discarded[2] if it has not been successfully transmitted by the end of the sample period.

From the control system point of view, the relevant parameters from the wireless links are data rate, time delay and probability of packet loss.[3] For this purpose, we can simplify the link model as in Figure 4. This simplified model is sufficient to calculate all the communication parameters that may affect the control performance. The data rate is implicit in the covariance of the quantization noise $v_{q,i}$. Both the time delay distribution and the probability of packet loss are determined by the MAC protocols, total number of time slots and probability of successful transmission p_s. We focus on the delay due to retransmissions and assume the processing time at the transmitter and the receiver is negligible. Since a packet is dropped if not decoded correctly by the end of the sample period, the time delay is always bounded by one sample period.

[2]In a control system, a new measurement is always more valuable than old measurements. In a single hop network, each transmitter only needs to send the newest data available.

[3]Undecoded errors can occur and the erroneous data will be used as if it were the correct data. The impact of the undecoded errors are not studied in this paper. We assume heavy error detection codes are used so that the probability of undetected errors is sufficiently small to be negligible.

C. MAC Protocols

Wireless channels are broadcast channels in nature. When multiple transmitter/receiver pairs share a single channel, we need a MAC protocol that determines which user gets to use the channel. Time is slotted and we assume no spatial reuse. Thus a collision occurs when more than one user transmits over the channel in a given time slot.

Many MAC protocols are known. A traditional way of allocating a single channel among competing users is "Polling". This can be realized in a token ring network where all the users are connected in a ring architecture. A token is circulated among all the users who have packets to transmit and a user can only transmit if it seizes the token. Polling guarantees no collision and the channel is in use as long as there are packets that need to be sent. TDMA (Time Division Multiple Access) is also a collision-free multiple access scheme. In TDMA, time slots are assigned in advance and never changed. In this paper, we consider fixed TDMA and we assume time slots are divided evenly among all the transmitter/receiver pairs. Since the time slots are pre-assigned, a time slot can be wasted if the pre-assigned transmitter no longer has a packet to transmit.

Both Polling and TDMA need some built-in centralized coordination in the system. A hot trend in sensor networks is to form ad hoc networks where such coordination may not be possible. Random access protocols are usually adopted in ad hoc networks. We consider a simple form of random access (RA) where each transmitter attempts to grab the channel independently with a probability of p at any given time slot. If we have n users, the probability that only one user transmits in any time slot is $np(1-p)^{n-1}$. This probability is maximized at $p = 1/n$. This random access scheme works with or without acknowledgment (ACK) from the receiver upon successful transmission. With ACK, the transmitter does not send redundant packets for the information that is already successfully decoded. Hence, the amount of traffic in the system is reduced and the probability of collision is smaller at the cost of added complexity and additional bandwidth to transmit the "ACK". Another class of random access based MAC protocols uses carrier sensing. Basically, a user that wishes to transmit senses the channel before it attempts to send. Carrier sensing reduces the collisions by avoiding collisions with ongoing transmissions. The medium access control sub-layer of the wireless LAN 802.11 standards uses CSMA/CA (Carrier

Sensing Multiple Access/Collision Avoidance). In case of collision, each transmitter will back off for a random period of time before its next attempt. In particular, we consider an exponential back-off algorithm [10].

In this paper, we study the performance of a networked control system with a wireless network adopting one of the MAC protocols discussed above. Given a MAC protocol, we can find the probability distribution of time delay and packet losses as a function of p_s, the probability of success on each transmission, and the number of time slots in a sample period. The performance of different MAC protocols are compared in terms of the control performance index we choose.

III. Performance Analysis and Optimization with Communication Faults

There are many control performance measures that can be considered and the impact of imperfect communication for different measures can be different. We consider the linear quadratic cost function as our performance measure. Specifically, we want to minimize

$$\mathrm{J}_{LQG} = \sum_{n=1}^{M} \lim_{t\to\infty} \mathrm{E}\ \mathbf{x}'^{<\mathrm{n}>}(t)\mathrm{Q}^{<\mathrm{n}>}\mathbf{x}^{<\mathrm{n}>}(t)$$

$$+\mathbf{u}'^{<\mathrm{n}>}(t)\mathrm{R}^{<\mathrm{n}>}\mathbf{u}^{<\mathrm{n}>}(t),$$

where the weight matrix $\mathrm{Q}^{<\mathrm{n}>}$ is positive semi-definite and $\mathrm{R}^{<\mathrm{n}>}$ is positive definite. We can tune the system performance by choosing different $\mathrm{Q}^{<\mathrm{n}>}$ and $\mathrm{R}^{<\mathrm{n}>}$. For each system, we perform a Cholesky factorization $[\mathrm{C}_1\ \mathrm{D}_{12}]'\,[\mathrm{C}_1\ \mathrm{D}_{12}] = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}$ and define $\mathbf{z}(t) = \mathrm{C}_1\mathbf{x}(t) + \mathrm{D}_{12}\mathbf{u}(t)$. Then our objective can be rewritten as $J_{LQG} = \lim_{t\to\infty} \mathrm{E}\ \mathbf{z}'(t)\mathbf{z}(t)$. As a standard normalization, the system input $\mathbf{w}(t)$ is unit white Gaussian noise in continuous time and $\mathbf{v}(k)$ is unit white Gaussian in discrete-time. Therefore, our performance measure for a single system is the trace of the steady-state covariance matrix of the regulated output $\mathbf{z}(t)$ in Figure 2 when the noise input is unit white Gaussian. Note that the square root of this performance measure is equivalent to the H_2 norm for continuous (or discrete) time systems with perfect feedback. It is also equivalent to the generalized H_2 norm of the sampled-data system when the plant is continuous time and the controller is discrete. We thus refer to our performance measure as H_2 norm due to this equivalence.

Different systems sharing the wireless channel correlate with each other only through the delays and packet losses caused by the wireless network. If the delay distribution and packet loss probabilities are known, the analysis for the joint performance measure can be completely decoupled. In the following sections, we discuss the controller design and the performance evaluation for a single system.

A. Controller Design

To the best of the authors' knowledge, the optimal LQG controller with packet losses in sensor feedbacks and/or control commands is not yet known. We use the controller in [8]. This controller is LQG optimal when the delay is bounded by one sample period and there is no packet loss. Since we discard old packets after one sample period, the delay is bounded by one sample period. However, we do have packet losses, so we need to do some approximations in order to use this controller design.

The controller has two cascaded parts: the Kalman filter and the state feedback controller. The Kalman filter calculates the minimum mean square error state estimate based on received sensor measurements. When all the sensor measurements are received, the classical steady state Kalman filter is used. When none of the sensor measurements are received, we can have the Kalman filter run one step forward open loop and this also gives the optimal state estimate. When only part of the sensor measurements are received, it is possible to compute the optimal state estimate [7] but we do not yet know if a steady state solution exists. For simplicity, we treat partial observation loss as complete observation loss in this paper.[4] The state feedback controller is a function of the total time delay in the feedback loop. Thus, it is time varying. The total time delay is from the time when measurements are taken to the time when the actuator updates with the received control command. We assume the controller calculates its control command right before its turn to transmit to the actuator. Therefore, the controller knows the time delay of the control command if the next transmission is successful. Upon receiving a control command, the actuator updates immediately if the control command is calculated based on full observation. Otherwise, if the received control command is computed based on incomplete observation, the actuator holds the control command and waits for another control command that is computed based on full observation. If no control command based on full observation is received by the end of the sample period, the actuator updates with the control command that is computed based on incomplete observation. This means the total time delay is equal to one sample period. The last scenario is that the actuator does not receive any new control command within one sample period. When this occurs, the actuator continues to use the last control command it has received and we assume that the total time delay is equal to one sample period.

The calculation of the time varying state feedback controller requires the knowledge of the delay distribution. We assume the time is slotted and there are L time slots in each sample period T. Note that L depends on the link design choices of the modulation scheme and the frame size. The delay distribution is discrete and the delay only takes a finite number of values of $\frac{i}{L}T$ for $i = 1, \cdots, L$. Given a MAC protocol and a link design, this delay distribution can be found. Let D denote the random delay of the control command. Note that $\Pr(D = T)$ is the sum of

[4] This can be a pessimistic design and we plan to further investigate this issue in our future research.

three probabilities: the probability that the control command based on full observation is received for the first time in the L^{th} time slot, the probability that the controller receives a control command based on incomplete observation by the end of the time slot but not a control command based on full observation, and the probability that the actuator does not receive any control command by the end of the sample period. Even though these three scenarios all lead to $D = T$ in the controller calculation, each of them has a different impact on the system performance. This will be made clear in the next subsection.

The closed-loop system is a sampled-data system since it involves both continuous-time and discrete-time dynamics. It was shown in [1], [8] that we can find an equivalent discretized system and design the optimal controller based on the discrete-time MJLS (Markovian Jump Linear System). The closed loop performance can be evaluated based on the MJLS model with a proper Markovian state space.

B. Performance Evaluation

We evaluate the system performance by choosing the right Markovian state and model the closed loop system as a MJLS. Define the augmented system state vector, $\hat{\mathbf{x}}(k) = [\tilde{\mathbf{x}}(k); \hat{\mathbf{x}}(k|k-1); \mathbf{y}^{\mathbf{c}}(k-1); \mathbf{u}(k-1)]$ and the joint noise vector $\hat{\mathbf{w}}(\mathrm{k}) = [\tilde{\mathbf{w}}(\mathrm{k}); \mathbf{v}(\mathrm{k})]$. Note $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{w}}$ are the discretized state and disturbance and $\hat{\mathbf{x}}(\mathrm{k}|\mathrm{k}-1)$ is the Kalman filter state estimate. We choose the Markovian state $r = (D, s)$ where D is the time delay in the control command and s indicates the sensor measurement loss and/or the control command loss:

$$s = \begin{cases} 0 & \text{no packet losses,} \\ 1 & \text{sensor measurement losses only,} \\ 2 & \text{control command is lost.} \end{cases}$$

Note that for all $D < T$, where T is the sample period, we always have $s = 0$ while when $D = T$, we can have $s = 0, 1, 2$. Therefore we have $L + 2$ Markovian states and we can write the system in the form of a MJLS as $\hat{\mathbf{x}}(k+1) = F_r\hat{\mathbf{x}}(k) + G_r\hat{\mathbf{w}}(k)$ for $r = 1, 2, \cdots, L+2$. The system matrices F_r, G_r can be easily derived. Let $P(k) = \mathrm{E}\ \hat{\mathbf{x}}(k)\hat{\mathbf{x}}(k)'$, then

$$\mathrm{P}(k+1) = \sum_{r=1}^{L+2} q_r F_r P(k) F_r' + \sum_{r=1}^{L+2} q_r G_r G_r',$$

where q_r is the probability that the MJLS is in state r. As $\mathrm{k} \to \infty$, it can be shown [8] that a unique steady-state covariance matrix $\mathrm{P} = \lim_{k\to\infty} P(k)$ exists when the recursion is stable. We can now evaluate the linear quadratic cost function since $\mathrm{J}_{LQG} = \mathrm{Trace}\left(\left[\begin{array}{cccc} Q & 0 & 0 & 0 \end{array}\right] P\right) + \mathrm{Trace}\left(\left[\begin{array}{cccc} 0 & 0 & 0 & R \end{array}\right] P\right)$.

IV. Numerical Results

A. Inverted Pendulum

The cart with an inverted pendulum, shown in Figure 5, is controlled with a force, F, to cancel the random disturbance w and maintain the pendulum in upright position. We

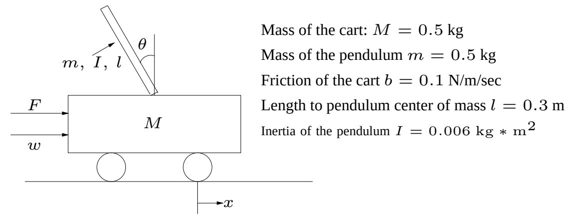

Fig. 5. Inverted Pendulum and Cart

use x to denote the cart position coordinate and θ is the pendulum angle from vertical. For this example, we assume two identical inverted pendulum plants with the parameter choices as listed in Figure 5. We determine the system dynamic equations by analyzing the force applied on the pendulum and the cart. The system dynamics are not linear in θ. We assume the pendulum does not move more than a few degrees away from the vertical and linearize the system dynamics about $\theta = 0$. We then get the standard linear model for the inverted pendulum (e.g., [13]). The state of the system is chosen as $\left[x(t), \dot{x}(t), \theta(t), \dot{\theta}(t)\right]$. The state space matrices are derived as in [13], [5]. We would like to minimize the linear quadratic cost function with

$$\mathrm{Q}^{<1>} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 10^6 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad R^{<1>} = 1,$$

and

$$\mathrm{Q}^{<2>} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 10^4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad R^{<2>} = 1.$$

These weight matrices $\mathrm{Q}^{<1>}, Q^{<2>}$ and $\mathrm{R}^{<1>}, R^{<2>}$ are chosen such that a desired performance is achieved. We choose to put a large weight on θ since the main goal is to keep θ small. When θ is small, our linear approximation model is accurate. The weight matrices can be carefully chosen to reflect priorities of different systems and different signals. The measurement noise $\mathbf{v_s}(k)$ is assumed to be Gaussian with zero mean and covariance matrix $\Sigma_s = [10^{-4}, 0; 0, 10^{-6}]$.

B. Communication Parameters

We assume that each time slot is $\frac{1}{6}$ milliseconds. Thus, for sample period $T = n$ milliseconds, there are $6n$ time slots in total. For simplicity, we assume all the channel gains on each link and the transmit power of each transmitter are identical. We assume $b_i = 5$ for $i = 1, \cdots, 6$, i.e., each sample is represented by 5 bits. For each collision-free transmission over any wireless link, we assume $p_s = 0.9$. For TDMA, the transmission order repeats as sensor measurement 1, sensor measurement 2, control command for systems 1 and then sensor measurement 1, sensor measurement 2, control command for system 2. For Polling,

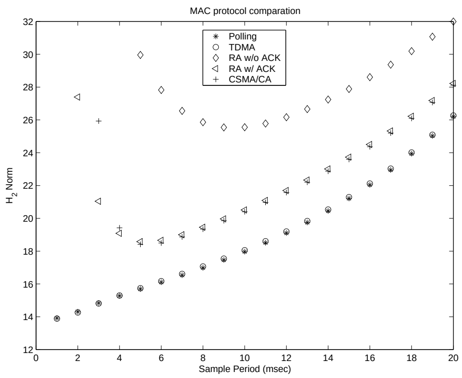

Fig. 6. Performance Comparison of Different Multiple Access Schemes

the time slots are allocated in the same order except Polling skips the transmitters with no packet to transmit in a given time slot. In random access, we assume each transmitter attempts to grab the channel with probability $p = \frac{1}{6}$ and this maximizes the probability of successful channel access at the beginning of each sample period. For CSMA/CA, we assume the minimum collision window $CW_{min} = 3$.

It is possible to derive all the probabilities analytically for the MAC protocols mentioned above by tedious book-keeping. In the numerical example shown below, the probabilities are obtained through simulation.

C. Performance Comparisons

Figure 6 compares the joint linear quadratic cost function of the two identical inverted pendulum systems sharing a wireless network with different multiple access schemes: TDMA, Polling, Random Access without ACK, Random Access with ACK, and CSMA/CA. Both TDMA and Polling are collision free while the others are random access algorithms and collisions cannot be avoided since there is no centralized control. The figure shows that Polling does not give much performance gain over TDMA. This is partly because p_s is relatively high. For smaller p_s, the gain is bigger but still not significant. The figure shows all three decentralized access schemes lead to performance degradation compared to TDMA and Polling. Random Access without ACK has the worst performance. This is due to collisions. The probability of collision depends on the number of active transmitters sharing the link and the probability of access attempt p. Without ACK, all six transmitters will be active at any time slot. The probability that only one transmitter attempts to transmit is $p_a = 6*p*(1-p)^5$. The maximum of p_a is roughly 0.4 when $p = \frac{1}{6}$. Thus a maximum of 40% of the time slots are used. Both Random Access with ACK and CSMA/CA try to reduce the amount of collisions. This is why their performance is significantly better than Random Access without ACK. The performance of Random Access with ACK and CSMA/CA are comparable and the performance depends on the communication parameters chosen. For CSMA/CA, as we increase CW_{min}, which is the minimum collision window, the control performance first improves, then degrades. The performance first improves due to reduced collision. The performance then degrades when CW_{min} is too large because the transmitter tends to wait too long for the second attempt after a collision occurs. This figure also shows that the control system design should depend on the wireless network design. For example, the optimal sample period selection should be different based on what multiple access scheme is used by the wireless network. Faster sampling is not necessarily better. In fact, all three distributed MAC protocols lead to control instability for very small sample periods. This is because the probability that the actuator receives the control command based on full observation by the end of the sample period is small when the sample period is small.

V. Conclusions and Future Research

In this paper, we present a framework for studying the performance of closed-loop control systems closed over wireless links. The LQG control is considered. We analyze the system performance with the techniques of sampled-data systems and Markovian jump linear systems. We show that distributed random channel access schemes lead to significant performance degradation compared with TDMA and Polling. Therefore more efficient random access schemes need to be designed for these distributed control applications. We compare the performance of the distributed control system over a wireless network with several commonly used multiple access schemes. We also show that collision reduction and avoidance techniques help improve the overall control performance. Our future research includes the design of adaptive communication systems that improve the control performance and the design of control algorithms that are robust to the communication faults.

References

[1] T. Chen and B. Francis, *Optimal Sampled-Data Control Systems.* Springer 1995.
[2] G. F. Franklin, et. al., *Digital Control of Dynamic Systems.* Addison Wesley, 3rd edition, 1990.
[3] Andrea Goldsmith, *EE359 Course Reader,* Stanford University.
[4] F. Lian, et. al., *Network Design Consideration for Distributed Control Systems* IEEE Transactions on Control Systems Technology, March 2002.
[5] X. Liu and A. Goldsmith, *Wireless Communication Trade-offs in Distributed Control,* Conference on Decision and Control, 2003.
[6] X. Liu and A. Goldsmith, *Wireless Medium Access Control in Distributed Control,* Allerton Conference on communication, control and computing, 2003.
[7] X. Liu and A. Goldsmith, *Kalman Filtering with Partial Observation Losses,* Preprint, to be submitted to IEEE Transactions on Automatic Control.
[8] J. Nilsson, *Real-Time Control Systems with Delays.* Ph.D. Thesis, Lund Institute of Technology, 1998.
[9] J. G. Proakis and M Salehi, *Communication Systems Engineering.* Prentice Hall 1994
[10] A. S. Tanenbaum, *Computer Networks,* Third Edition, Prentice Hall, 1997.
[11] L. Xiao, et. al., *Joint optimization of communication rates and linear systems.* IEEE Transactions on Automatic Control, January 2003.
[12] W. Zhang, *Stabilization of Networked Control Systems Over a Sharing Link Using ALOHA, IEEE Conference on Decision and Control 2003*
[13] http://www.engin.umich.edu/group/ctm/examples/pend/invpen.html.