

Networked Control System Design over a Wireless LAN

Michael Drew, Xiangheng Liu, Andrea Goldsmith, and Karl Hedrick

Abstract—We analyze and implement a networked control system where the communication between the sensors, the controller, and the actuator takes place over a wireless LAN (802.11b) ad hoc system. The wireless LAN system introduces random delays and packet loss in the feedback loop. We present an extended version of the separation principle of Linear Quadratic Gaussian (LQG) control with both random delays and packet loss in the feedback loop. This paper is motivated by many of the implementation constraints encountered during the setup of an experimental NCS. We see from our experimental data that the optimal LQG controller that takes the delays and loss into account indeed performs better than traditional control methods.

I. INTRODUCTION

Control over wireless networks has many compelling applications, including automated highway systems, automated factories, smart homes and appliances. The deployment of wireless networks allows fully mobile operation, flexible installation and rapid deployment, while reducing maintenance costs. However, building a distributed control system supported by a wireless network is a challenging task because a wireless network inevitably introduces random delays and packet loss in the feedback loop. These communication faults are generally not accounted for in the controller design, because the control analysis and modeling has traditionally assumed that the communication among the plant, sensors, controller, and actuators is reliable and instantaneous, or with a small constant delay. This is not a valid assumption when the information exchanges take place over a wireless network.

A. Networked Control Systems

Control systems that rely on a network for some or all of its feedback loops fall into the category of Networked Control Systems (NCS). With the increasing performance and lowering costs of digital computation and communication, NCS is a growing field of interest in industry and research [?], [?], [?], [?], [?], [?]. The performance characteristics of all NCSs are largely dictated by the network medium they utilize.

B. Wireless Networks

Most communication networks introduce delays and packet loss that negatively affect control performance. These communication faults are a lot more pronounced in wireless networks, due to their time-varying channels, limited spectrum, and interference. The radio signal is always subject to physical and environmental conditions like interference, signal-attenuation and multi-path effects; all of which can cause latency and packet loss. This situation is further complicated by the low-level hardware construction and MAC layer design of many wireless radios that utilize packet retransmission when the signal strength is poor. Retransmissions can result in a blocking process preventing software from executing at desired intervals. Thus, poor signal strength not only leads to packet loss or latency, but also can affect the sampling process itself by inducing variance in the sampling period (jitter).

Despite all the problems that a control system faces when its feedback loop is closed over a wireless network, the convenience and flexibility offered by wireless networks is virtually limitless. This is the motivation for wireless NCS research.

C. Related Work

NCS design involves both control system and communication system design. Designing reliable networks or improving existing network protocols to improve the control performance is studied in [?], [?], [?], [?], [?]. Some authors have focused on wireless network designs to best support control applications [?], [?], [?], [?].

In this paper, we are interested in designing the controller for existing network communication technologies while taking the issues of packet delay and loss into account. A good overview of these issues and the approaches taken to address them in control can be found in [?], [?]. In [?], [?] Nilsson et al. present methods of analysis and control to handle random network feedback delays.

The issue of packet loss is really separate from that of delay since it involves loss of information versus delayed information. Control with packet loss is considered in [?], [?], [?]. MJLS methods are particularly useful in analyzing LTI systems subject to packet loss. One of the most famous control results for jump linear systems is Chizeck and Ji's optimal Jump Linear Quadratic Gaussian (JLQG) solution found in [?]. By assuming the system is observable in all conditions they prove that a separation property exists which allows the separate design of the optimal JLQ controller and state estimator. The separation principle has been recently extended to networked control systems in [?], [?], [?], [?] under various assumptions. Estimation with packet loss has been addressed by [?], [?], [?].

To design an NCS, the impact of delays and packet loss should not be considered separately. This is because there is a design trade-off between delays and packet loss from the communication design point of view. Very few authors consider the delays and packet loss jointly. An example of such work can be found in [?], where the LQG optimal controller is proved to satisfy the separation principle even in the presence of random delays and packet loss, if there is a reliable and instantaneous acknowledgment (ACK) from the actuator to the controller.

The work in this paper is largely motivated by our experimental NCS implementation. We have a non-deterministic wireless network subject to delay and packet loss. We favor performance over data accuracy, so we use UDP packets. Also, since packet retransmissions can cause excessive delay and block process execution, we assume ACK packets are not in use. Thus, each time a packet is sent, the sender does not immediately know if it has been successfully received. Finally, we use a "passive" controller strategy, meaning that the controller and estimator are triggered by the receipt of sensor packets rather than acting periodically¹.

The goal of this paper is to combine some of the existing NCS delay and packet loss results using a practical approach which can easily be applied to real systems. This paper is arranged as follows: In Section II we present our NCS setup and model formulation. In Section III we derive a control law which considers delay and packet loss and we demonstrate that under the appropriate

¹Throughout this paper we will use this definition of "passive", referring only to our chosen method of control strategy.

assumptions separation between control and estimation exists. In Section IV we describe a test apparatus which was created to investigate these techniques, and we present some experimental data which validates the performance of our controller. Finally, in Section V we present our conclusions and describe the direction of future research in this area.

II. NETWORKED CONTROL SYSTEM FORMULATION

Our system model is shown in Fig. 1. We consider a linear, time-invariant, continuous-time plant described by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + v(t) \\ y(t) &= Cx(t) + w(t) \end{aligned} \quad (1)$$

sampled with a constant period h and controlled by a remote controller with communication networks on *both* sides. Here we have represented the packet transport delay from the sensor to controller as τ_k^{sc} , and the delay from the controller back to the actuator as τ_k^{ca} . The computational delay of the controller is denoted as τ_k^c , but since it is small and nearly constant we can neglect it or include it as part of τ_k^{ca} . Thus, the total round-trip packet time is $\tau_k^{rt} = \tau_k^{sc} + \tau_k^{ca}$. In our system setup, we have $\tau_k^{rt} \leq h \forall k$ since we declare a packet is lost if it is not received by the end of the sample period.

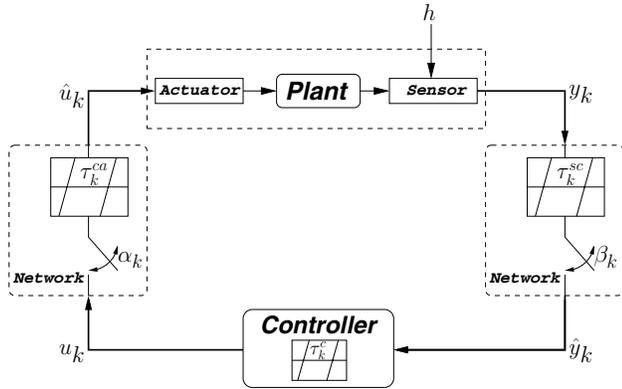


Fig. 1. Wireless control with packet delay and loss on both sides of the feedback loop

We assume that both the controller and the actuator are event-driven. That is, the controller does not compute new control commands unless it receives the sensor measurements and the actuator continues to use the old control command until a new one is received. In this paper, we assume all the sensor measurements (possibly from different sensors) are encoded and sent in a single packet for each sample period. This is just for the simple implementation with our experimental setup. Our analytical results can be generalized to the case where different sensors encode and send their sensor measurements in separate packets.

Discretizing the continuous-time dynamics of (1) and considering the effects of delay and packet loss gives us

$$x_{k+1} = \Phi x_k + \Gamma_0(\tau_k^{sc}, \tau_k^{ca}) \hat{u}_k + \Gamma_1(\tau_k^{sc}, \tau_k^{ca}) \hat{u}_{k-1} + v_k \quad (2a)$$

$$\hat{y}_k = \beta_k (Cx_k + w_k). \quad (2b)$$

where $\Phi = e^{Ah}$, $\Gamma_0(\tau_k^{sc}, \tau_k^{ca}) = \int_0^{(h-\tau_k^{sc}-\tau_k^{ca})} e^{As} ds B$, $\Gamma_1(\tau_k^{sc}, \tau_k^{ca}) = \int_{(h-\tau_k^{sc}-\tau_k^{ca})}^h e^{As} ds B$, and $\alpha_k, \beta_k \in \{0, 1\}$. At time step k , we have $\beta_k = 0$ if a sensor packet is lost and $\alpha_k = 0$ if a control command is lost.

Since we assume the actuator uses the previous control command in the event of an actuation packet loss, and since we also assume

that the controller is passive, only supplying actuation when a packet is received from the sensor, we have

$$\hat{u}_k = \alpha_k \beta_k u_k + (1 - \alpha_k \beta_k) \hat{u}_{k-1}. \quad (3)$$

Note that if a packet is lost the value of τ_k^{rt} is irrelevant since $\hat{u}_k = \hat{u}_{k-1}$.

III. NCS CONTROLLER DEVELOPMENT

Now we will consider a method of NCS control which considers both delay and packet loss. Let I_k be the information set available to the controller at the k^{th} time step. The maximum possible set of I_k is

$$I_k^{MAX} = \{\hat{y}_0, \dots, \hat{y}_k; \hat{u}_1, \dots, \hat{u}_{k-1}; \alpha_0, \dots, \alpha_{k-1}; \beta_0, \dots, \beta_k; \tau_0^{sc}, \dots, \tau_k^{sc}; \tau_0^{ca}, \dots, \tau_k^{ca}\}$$

Note that we have included β_k, τ_k^{sc} and τ_k^{ca} in the information set. Knowledge of τ_k^{ca} is possible if the NCS features a deterministic network with known delays and β_k can be known if the controller expects periodic sensor packets. α_k is not included in I_k^{MAX} since it can be known only after control is computed *and* if the network provides an ACK packet upon successful delivery of the controller packet to the plant. However, α_k does not affect the behavior of the controller at time k because if the k^{th} control packet is lost any change in control at time k is irrelevant.

A. Our NCS Arrangement

We take a practical approach to the communication and control strategy used in our NCS. We assume that our network is non-deterministic and that the round-trip delay τ_k^{rt} is random. Without relying on ACK packets and using our passive control strategy, sensor packets are sent according to the sampling period h and include the following information:

$$\text{Sensor Packet} : \{y_k, t_k, k, s, u_s, u_{s-1}, \tau_s^{rt}\} \quad (4)$$

Here, k is the packet i.d. corresponding to the current time step and s is the time step of the last packet received by the actuator from the controller. t_k is the time the packet is sent from the sensor (time stamp). Since the packet size of most communication protocols is more of a function of overhead rather than actual data, this additional information does not impose much of a burden to most systems. By including this data with each sensor packet, when $\beta_k = 1$, we have

$$I_{k|\beta_k=1} = \{\hat{y}_0, \dots, \hat{y}_k; \hat{u}_0, \dots, \hat{u}_{k-1}; \alpha_0, \dots, \alpha_{k-1}; \beta_0, \dots, \beta_k; \tau_0^{sc}, \dots, \tau_k^{sc}; \tau_0^{ca}, \dots, \tau_{k-1}^{ca}\}. \quad (5)$$

Note that $I_{k|\beta_k=1}$ includes τ_k^{sc} which is determined from t_k and the controller's clock which we assume is synchronized with the plant/sensor clock. τ_k^{ca} , however, is not included in (5) because of our NCS assumptions. Obviously, for $\beta_k = 0$, the information set is reduced, but for our NCS this is not an issue since control is only calculated when $\beta_k = 1$. Nevertheless, we must update our state estimate appropriately when packets are dropped.

B. Estimator Design

Since packets can be lost between the sensor and controller, our Kalman state estimator has the following form:

$$\begin{aligned} \hat{x}_{k|k} &= (I - K_k C) \left[\Phi \hat{x}_{k-1|k-1} \right. \\ &\quad \left. + \Gamma_0(\tau_{k-1}^{rt}) \hat{u}_{k-1} + \Gamma_1(\tau_{k-1}^{rt}) \hat{u}_{k-2} \right] + K_k y_k. \end{aligned} \quad (6)$$

Here $K_k = P_{k|k-1} C^T (C P_{k|k-1} C^T + R)^{-1}$ is the time-varying Kalman filter gain matrix and $P_{k|k-1}$ is the error covariance which

is a function of packet loss between the sensor and controller. $P_{k|k-1}$ is updated according to

$$P_{k+1|k} = \Phi P_{k|k-1} \Phi^T + Q - \beta_{k+1} \Phi P_{k|k-1} C^T \left(C P_{k|k-1} C^T + R \right)^{-1} C P_{k|k-1} \Phi^T. \quad (7)$$

If one or more packets are dropped on either side of the feedback loop, then the next time a packet successfully arrives at the controller the estimator must be run in open loop up to time step $k-1$ before using (6).

For example, say the last time a control packet successfully arrived at the plant was at time step s , and say the last estimate of x is $\hat{x}_{s|s}$. If \hat{u}_s , \hat{u}_{s-1} , and $\tau_s^{rt} = \tau_s^{sc} + \tau_s^{ca}$ are known, we can run the estimator open loop using the iteration

$$\hat{x}_{s+1|s+1} = \hat{x}_{s+1|s} = \Phi \hat{x}_{s|s} + \Gamma_0(\tau_s^{rt}) \hat{u}_s + \Gamma_1(\tau_s^{rt}) \hat{u}_{s-1} \quad (8a)$$

$$\hat{u}_{s+1} = \hat{u}_s \quad (8b)$$

$$P_{s+1|s} = \Phi P_{s|s-1} \Phi^T + Q \quad (8c)$$

until we have $\hat{x}_{k-1|k-1}$. Notice that (8b) is implied by our NCS control strategy from (3) and thus the value of τ_{s+1}^{rt} is irrelevant. Also, consider that we may already have a proper estimate of $x_{s+1|s+1}$ if $\beta_{s+1} = 1$ and $\alpha_{s+1} = 0$. In this case the iteration is begun by calculating $\hat{x}_{s+2|s+2}$ according to $\hat{u}_{s+1} = \hat{u}_s$.

Note that without ACK packets an accurate *periodic* state estimate is not possible because the estimator/controller can not know whether previous control actuation packets have successfully reached the plant. This is the justification for the passive control strategy and the inclusion of extra data within each sensor packet.

Of course, there is a limit to how many sensor packets we can loose before the estimator loses stability. The sequence $\{\beta_k\}_{0}^{\infty}$ is assumed to be a Bernoulli random sequence and thus, so is the $P_{k+1|k}$ iteration. Defining the probability of successful sensor packet transmission as $\Pr[\beta_k = 1] = \rho^{sc}$ there is a critical value ρ_{cest}^{sc} above which the mean state covariance $E[P_{k|k-1}]$ is bounded for all initial conditions. For $\rho^{sc} \leq \rho_{cest}^{sc}$ the mean state covariance diverges for some initial condition. In [?] Sinopoli et al. show the existence of this critical value and find bounds on it such that $\rho^{sc} \leq \rho_{cest}^{sc} \leq \overline{\rho^{sc}}$. There is a closed-form solution for the lower bound: $\underline{\rho^{sc}} = 1 - \frac{1}{\phi^2}$ where ϕ is the spectral radius of Φ . The authors show that the upper bound is the solution of a linear matrix inequality.

C. Controller Design

Given the system dynamics of (2), we desire to minimize the cost function

$$J_N = E \left[\sum_{k=0}^{N-1} \left\{ x_k^T Q_x x_k + \hat{u}_k^T Q_u \hat{u}_k \right\} + x_N^T Q_N x_N \right] \quad (9)$$

Here Q_x is positive semi-definite and Q_u is positive definite. We previously defined the probability of successful packet transmission from the sensor to the controller as $\rho^{sc} = \Pr[\beta_k = 1]$. Likewise, we define $\rho^{ca} = \Pr[\alpha_k = 1]$. Therefore, the probability of the actuator receiving a packet at the k^{th} time step with $\tau_k^{rt} \leq h$ is defined $\rho^{rt} = \Pr[\tau_k^{rt} \leq h] = \rho^{sc} \cdot \rho^{ca}$. Because we are using a passive NCS control strategy, the event of losing a packet either on the sensor side or on the actuation side due to packet corruption, interference, excessive delay ($\tau_k^{rt} > h$), etc. results in maintaining the previous control input for at least one more time step. Thus, we seek a state feedback control law based on the state variable

$z_k = [x_k^T \quad \hat{u}_{k-1}^T]^T$. From (3) we have

$$\hat{u}_k = \alpha_k \beta_k \left\{ -L_{\rho^{rt}}(\tau_k^{sc}) \begin{bmatrix} x_k \\ \hat{u}_{k-1} \end{bmatrix} \right\} + (1 - \alpha_k \beta_k) \left\{ \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ \hat{u}_{k-1} \end{bmatrix} \right\}. \quad (10)$$

Defining additional terms to handle the case of packet loss allows us to extend Nilsson's optimal delay-compensating controller and leads to the following theorem:

Theorem 3.1: *When the probability of receiving a control command at the actuator with $\tau_k^{rt} \leq h$ is ρ^{rt} (thus, loss or delayed with a probability of $1 - \rho^{rt}$), the optimal state feedback controller is given by:*

$$u_k^*(\tau_k^{sc}) = -L_{\rho^{rt}}(\tau_k^{sc}) \begin{bmatrix} x_k \\ \hat{u}_{k-1} \end{bmatrix}, \quad (11)$$

for $0 < \tau_k^{rt} \leq h$, where h is the sample period and

$$L_{\rho^{rt}}(\tau_k^{sc}) = (Q_u + \tilde{S}_{k+1}^{22})^{-1} \begin{bmatrix} \tilde{S}_{k+1}^{21} & \tilde{S}_{k+1}^{23} \end{bmatrix},$$

$$\tilde{S}_{k+1}(\tau_k^{sc}) = E_{\tau_k^{ca}} \left\{ G^T(\tau_k^{sc}, \tau_k^{ca}) S_{k+1} G(\tau_k^{sc}, \tau_k^{ca}) \mid \tau_k^{sc} \right\}, \quad (12)$$

$$\tilde{S}_{k+1}^{pl} = G^T(0, h) S_{k+1} G(0, h), \quad (13)$$

$$G(\tau_k^{sc}, \tau_k^{ca}) = \begin{bmatrix} \Phi & \Gamma_0(\tau_k^{sc}, \tau_k^{ca}) & \Gamma_1(\tau_k^{sc}, \tau_k^{ca}) \\ 0 & I & 0 \end{bmatrix},$$

$$S_k = \rho^{rt} \cdot E_{\tau_k^{sc}} \left\{ F_1^T(\tau_k^{sc}) \begin{bmatrix} Q_x & 0 \\ 0 & Q_u \end{bmatrix} F_1(\tau_k^{sc}) + F_2^T(\tau_k^{sc}) \tilde{S}_{k+1}(\tau_k^{sc}) F_2(\tau_k^{sc}) \right\} \Big|_{\tau_k^{rt} \leq h} + (1 - \rho^{rt}) \left\{ F_1^{pl T} \begin{bmatrix} Q_x & 0 \\ 0 & Q_u \end{bmatrix} F_1^{pl} + F_2^{pl T} \tilde{S}_{k+1}^{pl} F_2^{pl} \right\}, \quad (14)$$

$$F_1(\tau_k^{sc}) = \begin{bmatrix} I & 0 \\ -L_{\rho^{rt}}(\tau_k^{sc}) & I \end{bmatrix}, \quad F_2(\tau_k^{sc}) = \begin{bmatrix} I & 0 \\ -L_{\rho^{rt}}(\tau_k^{sc}) & I \end{bmatrix},$$

$$F_1^{pl} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}, \quad F_2^{pl} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix},$$

$$S_N = \begin{bmatrix} Q_N & 0 \\ 0 & 0 \end{bmatrix}.$$

This is a modification of the work of Liu et al. found in [?]. The proof follows that of Theorem 5.1 from [?] and will not be shown here. The definitions of F_1^{pl} and F_2^{pl} come from (10). Combined, the above is a stochastic Riccati equation evolving backwards in time from S_{k+1} to S_k . The iteration involves expectation calculations with respect to τ^{sc} and τ^{ca} and is performed until a stationary value S_{∞} of S_k is found. As expected, if the probability of packet loss is too high (ρ^{rt} too low) the iteration will not converge. A lower bound on ρ^{rt} can be found using MJLS methods. See [?]. Assuming the iteration converges, $L_{\rho^{rt}}(\tau_k^{sc})$ is calculated according to

$$L_{\rho^{rt}}(\tau_k^{sc}) = L_{\rho^{rt}}(\tau_k^{sc}, S_{\infty}) = \left(Q_u + \tilde{S}_{\infty}^{22}(\tau_k^{sc}) \right)^{-1} \begin{bmatrix} \tilde{S}_{\infty}^{21}(\tau_k^{sc}) & \tilde{S}_{\infty}^{23}(\tau_k^{sc}) \end{bmatrix}. \quad (15)$$

Note that S_{∞} is independent of τ_k^{sc} though \tilde{S}_{∞} is not; it is calculated from (12). In practice, a table for $L_{\rho^{rt}}(\tau_k^{sc}, S_{\infty})$ is

calculated off-line which is interpolated on-line to get a control law in the form of (11).

The above iteration can easily be modified for different control strategies. For instance, as noted above, if the actuator applies zero control input when a packet is not received by some time-out period τ^{to} then the $[0 \ I]$ row of F_1^{pl} and F_2^{pl} is replaced by $[0 \ 0]$ and equation (13) is calculated with τ^{to} instead of h .

Next we show that separation between estimation and control still holds even in the presence of packet loss.

Theorem 3.2: *For the output feedback control problem, the optimal controller that generates a control command to be updated at time $kh + \tau_k^{rt}$ according to this implementation and with the information set I_k of (5) is*

$$u_k(\tau_k^{sc}, I_k) = -L_{\rho^{rt}}(\tau_k^{sc}) \begin{bmatrix} E[x_k | I_k] \\ \hat{u}_{k-1} \end{bmatrix}. \quad (16)$$

where $L_{\rho^{rt}}(\tau_k^{sc})$ is calculated according to Theorem 3.1 and $E[x_k | I_k]$ is the state estimate given the information set I_k and the estimation technique discussed above.

The proof of this theorem may be found in [?]. This theorem extends the separation principle to the case of packet loss on both the sensor side and the actuation side of the feedback loop without relying on the use of ACK packets. It does, however, rely on the passive control strategy and the extra data included with each sensor packet from (4).

IV. EXPERIMENTAL NCS TEST PLATFORM

A. Description

An experimental wireless NCS is created to investigate NCS control techniques. For our plant, we choose the classical control problem of the inverted pendulum since it is unstable, non-minimum phase and easily linearizable. Similar systems have been created for the purpose of wireless control research. See [?], [?] for example. Our pendulum is stabilized by moving the base cart along a linear path. The actual pendulum and its schematic form are shown in Fig. 2. Here the cart and pendulum mass and inertia are M , m , and I respectively. The cart position and pendulum rotation are x and θ .

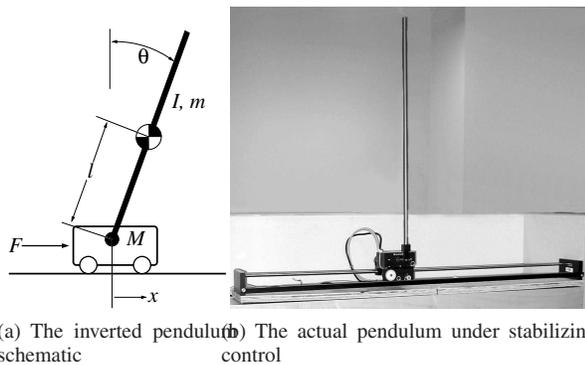


Fig. 2. The inverted pendulum

Both pendulum angle and cart position are measured with rotational encoders provided with the pendulum, and the cart is driven by a DC electric motor by the accompanying power amplifier. Actuation and sensing of the pendulum is accomplished using a Sensoray 626 data acquisition card installed in an AMD AthlonXP 2200 desktop PC. The Controller component runs on a Dell laptop PC with a 2 GHz Intel processor. Sensor and actuation data packets

are communicated to and from the controller using UDP sockets via Orinoco Silver PCMCIA 802.11b wireless cards running in ad hoc mode. Both the laptop (controller) and the desktop (plant) PCs use these cards. We use Gentoo Linux for the operating systems of both the controller and the plant. For more details refer to [?].

B. Results

Figure 3 shows the typical results of an experiment using a 2 inch pendulum rod. The upper plot shows the dynamics of the pendulum. After synchronizing the clocks, (which takes about 35 seconds for this run) from 35 to 40 seconds the controller inverts the pendulum. At 40 seconds it is stabilized in the inverted position and from 42 to 62.5 seconds the pendulum cart is forced to track a square wave while keeping the pendulum rod upright. At 62.5 seconds the pendulum loses stability from too much packet loss and must be re-inverted. Tracking begins again at about 72 seconds until the experiment is complete.

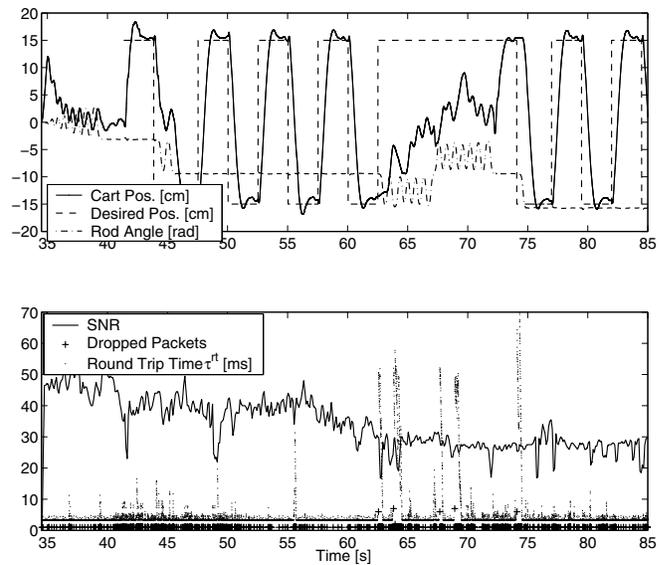


Fig. 3. 2 inch pendulum dynamics. Sampling period = 4 ms. Wireless bit rate = 1 Mb/s fixed. Pendulum-to-controller distance increasing: 1 → 20 meters.

In this example, the laptop computer (controller) is carried out of the pendulum room and down to the end of a hallway where it stops moving at about the 60th second of the experiment. The final distance is about 20 m from the pendulum. The lower plot demonstrates how SNR decreases as the distance increases. Note how the round-trip delay and packet loss increases as SNR decreases. Most of the packet loss in this experiment is due to excessive delay. In fact, of the 2407 dropped packets, only 33 were actually lost—the others were simply delayed beyond the 4 ms sampling period. Although a majority (77%) of the round-trip packet times (τ^{rt}) are well under 3.5 ms, periods of poor signal strength cause sudden jumps in packet latency, thus, the average round-trip packet time for this experiment is 5.2 ms.

C. Controller Performance Results

Our delay and loss-compensating estimator and controller improves the performance of our NCS when subjected to the delay and loss inherent to the wireless network. A detailed description of these results (and those that follow) can be found in [?]. Unfortunately, the bursty behavior and fast nonlinear dynamics of the 2 inch pendulum

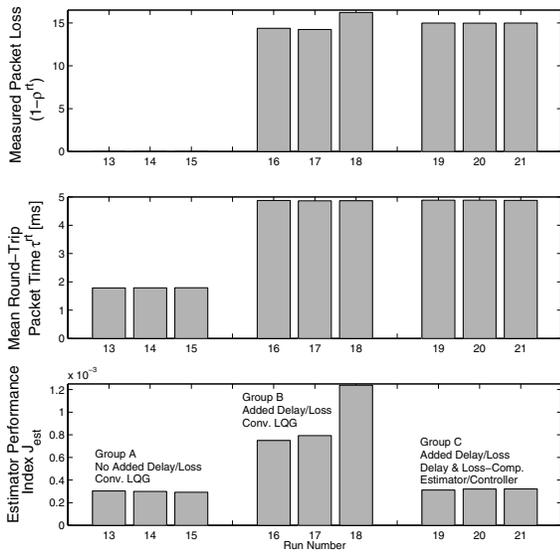


Fig. 4. Estimator results for the 8 inch pendulum. Sampling period = 20 ms. Artificial delay and packet loss used for Groups B and C.

makes running experiments on the threshold of stability (which is where the new controller shows the greatest improvement) very difficult since the pendulum constantly falls over, rendering the performance measurements meaningless. Longer pendulum rods are more stable, but the normal wireless range of delay and loss is not significant enough to reliably demonstrate correlated performance degradation with the slower dynamics.

Fortunately, through our software we are able to add uniform random delay to both τ^{sc} and τ^{ca} , and we can prescribe packet arrival probabilities ρ^{sc} and ρ^{ca} . We use an 8 inch pendulum rod with a sampling period of 20 ms. The wireless network bit rate is set to a fixed 11 Mb/s and the pendulum-to-controller distance is fixed at about 1 meter. Thus, the delay induced by the hardware is minimal, and we expect zero packet loss except for that which is artificially added.

Using normalized performance indices (refer to [?] for details) we compare three scenarios depicted in Figures 4 and 5 by Groups A - C. For Groups A and B we use conventional time-invariant Kalman estimation and LQG control, but for Group A we have zero packet loss and mean round-trip packet delay is small so the estimator and controller performance is good (indicated by low performance indices).

In Groups B and C we artificially add delay and packet loss. We set τ^{sc} and τ^{ca} uniformly distributed on the interval [.89, 4] ms and [.85, 4] ms respectively. (Since there is still real packet transmission delay, the lower bound of the delay distributions cannot be controlled.) These delay distributions remained very constant for runs 16 - 21. For the packet arrival rates, we set $\rho^{sc} = \rho^{ca} = .92$, thus $\rho^{rt} = .85$.

As expected, Group B shows the worst estimation and control performance since there is no compensation for the added delay and packet loss. For Group C we use the stochastic delay and loss properties to calculate our control law according to Theorem 3.1. Figure 5 demonstrates that the delay and loss-compensating estimator and controller performs the best. In fact, runs 19 and 21 from Group C actually performed slightly better than all the runs of Group A. This is not strictly a result of better estimation since the same cannot be said for J_{est} in Figure 4.

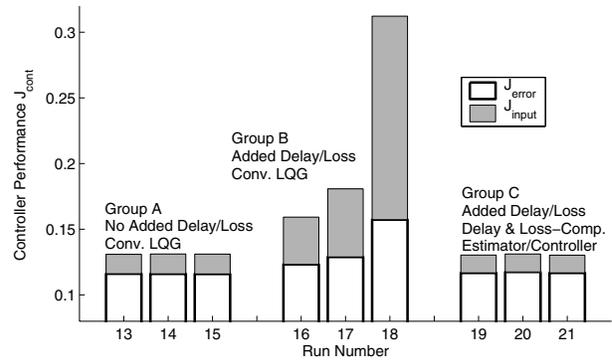


Fig. 5. Controller performance results. 8 inch pendulum, sampling period = 20 ms. Artificial delay and packet loss used for Groups B and C.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a practical NCS model which considers both random packet delay and loss on both the sensor and actuator sides of the feedback loop. This model can easily be applied to many NCS control and communication strategies. Motivated by our NCS implementation, we have presented an optimal control method which handles packet loss and delay without requiring the use of ACK packets in the network. By including some additional information in each sensor packet the controller's information set can be made sufficient to maintain the separation principle between controller and estimator.

Our NCS test platform allows us to verify these theoretical results and suggests future directions for this research. For example, we may be able to apply a more sophisticated packet loss and delay model to our control law derivation. A heuristic approach may also be applied where different control laws are used depending on which regime the channel statistics are in. As SNR drops, for instance, we may transition to a control law based on a higher loss probability.

One of the motivations behind NCS and distributed control is the assumed lack of processing power available at the actuator and/or plant. However, if we provide some computation power and intelligence to the actuator, we may implement different control strategies in the case of packet loss. Instead of maintaining the previous control input as dictated by (3), for instance, it might be better to apply zero input to the actuator, or perhaps a simplified control law based on an incomplete information set. These decisions depend on the plant dynamics and the overall NCS structure.

VI. ACKNOWLEDGMENTS

The National Science Foundation deserves recognition for their continued financial support of this research under award number 0120928.

REFERENCES

- [1] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control Syst. Mag.*, pp. 84-99, February 2001.
- [2] F. Lian, J. Moyne, and D. Tilbury, "Performance evaluation of control networks," *IEEE Control Syst. Mag.*, pp. 66-83, Feb. 2001.
- [3] —, "Network design consideration for distributed control systems," *IEEE Trans. Contr. Syst. Technol.*
- [4] J. Nilsson, "Real-time control systems with delays," Ph.D. dissertation, Lund Institute of Technology, Lund, Sweden, May 1998.
- [5] H. Abdullah and C. Chatwin, "Distributed C³ environment for small to medium-sized enterprises," *Integrated Manufacturing Systems*, vol. 5, no. 3, pp. 20-28, 1994.

- [6] S. Jiang, "Wireless communications and priority access protocol for multiple mobile terminals in factory automation," *IEEE Trans. Robot. Automat.*
- [7] P. Otanez, J. Parrott, J. Moyne, and D. Tilbury, "The implications of ethernet as a control network," in *Global Powertrain Conference*, Ann Arbor, MI, Sept. 2002.
- [8] J. Wheelis, "Process control communications: Token Bus, CSMA/CD, or Token Ring?" *ISA Transactions*, vol. 32, no. 2, pp. 193–198, July 1993.
- [9] T. Skeie, S. Johannessen, and C. Brunner, "Ethernet in substation automation," *IEEE Control Syst. Mag.*
- [10] G. C. Walsh, H. Ye, and L. Bushnell, "Stability analysis of network control systems," in *Proceedings of the ACC*, San Diego, CA, June 1999, pp. 2876–2880.
- [11] X. Liu and A. Goldsmith, "Wireless network design for distributed control," in *Proc. IEEE CDC*, 2004.
- [12] X. Liu and A. J. Goldsmith, *Cross-layer Design of Distributed Control over Wireless Networks*, T. Basar, Ed. Birkhauser (In Press), 2005.
- [13] H. Ye, G. C. Walsh, and L. Bushnell, "Real-time mixed-traffic wireless networks," *IEEE Trans. Ind. Electron.*
- [14] Y. Lu, Y. Zhong, and B. Bhargava, "Packet loss in mobile ad hoc networks," Purdue University, West Lafayette, IN, 47904, Tech. Rep. CSD-TR 03-009.
- [15] B. Wittenmark, J. Nilsson, and M. Törngren, "Timing problems in real-time control systems," in *Proc. of the ACC*, Seattle, WA, June 1995, pp. 2000–2004.
- [16] 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, 1998.
- [17] B. Azimi-Sadjadi, "Stability of networked control systems in the presence of packet losses," in *Proc. of IEEE CDC*, December 2003.
- [18] O. C. Imer, S. Yüksel, and T. Başar, "Optimal control of dynamical systems over unreliable communication links," in *NOLCOS 2004*, Stuttgart, Germany, 2004.
- [19] P. Seiler and R. Sengupta, "Analysis of communication losses in vehicle control problems," in *Proc. of the ACC*, Arlington, VA, June 2001, pp. 1491–1496.
- [20] H. J. Chizeck and Y. Ji., "Optimal quadratic control of jump linear systems withgaussiann noise in discrete-time," in *Proc. of IEEE CDC*, December 1988, pp. 1989 – 1993.
- [21] V. Gupta, D. Spanos, B. Hassibi, and R. M. Murray, "Optimal LQG control across a packet-dropping link," *Submitted to IEEE Trans. Automat. Contr.* 2004.
- [22] N. Elia and J. Eisenbeis, "Limitations of linear remote control over packet drop networks," in *Proc. of IEEE CDC*, December 2004.
- [23] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, and S. Sastry, "LQG control with missing observation and control packets," *IFAC*, 2005.
- [24] B. Sinopoli et al., "Kalman filtering with intermittent observations," in *Proc. of the 42nd IEEE Conf. on Decision and Control*, vol. 1, Maui, Hawaii USA, Dec. 2003, pp. 701–708.
- [25] X. Liu and A. Goldsmith, "Kalman filtering with partial observation losses," in *Proc. of the 43d IEEE Conf. on Decision and Control*, Paradise Island, Bahamas, 2004.
- [26] Y. Mostofi and R. M. Murray, "On dropping noisy packets in kalman filtering over a wireless fading channel," in *Proceedings of the ACC*, Portland, OR, June 2005, pp. 4596 – 4600.
- [27] M. C. Drew, "Networked control system design over a wireless LAN," Ph.D. dissertation, U.C. Berkeley, 2005.
- [28] A. Hörjel, "Bluetooth in control," Master's thesis, Lund Institute of Technology, Lund, Sweden, 2001.
- [29] N. Ploplys and A. Alleyne, "UDP network communications for distributed wireless control," in *Proceedings of the ACC*, Denver, CO, June 2003, pp. 3335–3340.