2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Play-Back Buffers in Networked Control Systems: Evaluation and Design

Graham Alldredge, Michael S. Branicky, and Vincenzo Liberatore
Electrical Engineering and Computer Science Department
Case Western Reserve University, Cleveland, OH 44106
{gwa,mb,vl}@case.edu

Abstract— **The networked environment presents many new challenges for the design of feedback control systems. A specific problem is that networked control systems (NCSs) are subject to the performance-degrading effects of time-varying, random loop delays. Model predictive controllers are a well-known method for delay compensation, but often exhibit poor performance under these types of loop delay. This paper investigates the use of play-back buffers to remove the uncertainty in the delay. First we study the value of removing all delay uncertainty with comparisons to unbuffered PID control using a bounded-interval delay distribution, and then we explore optimal play-back buffer design for bounded-interval and heavy-tailed delay distributions.**

I. Introduction

Communication networks offer considerable capabilities for the implementation of feedback control systems, but only if the control system can be designed to effectively compensate for the impact of network nondeterminism. Control systems in which information from the sensors and controllers is sent over an electronic communication network are called *networked control systems* (NCSs) [9], [14], [21], [23]; see Fig. 1. Some basic benefits of using networks to close the feedback loop include reduced cost, relatively simple implementation, and greatly increased flexibility. Of course, an NCS must also tolerate the performance-degrading aspects of the network, including time-varying random transmission delays.

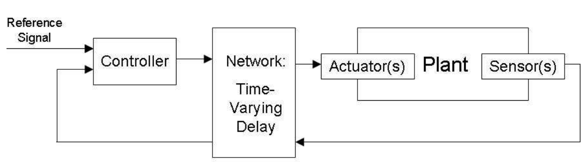

Fig. 1. A block diagram of a networked control system.

Control with loop delay is hardly a new issue, and much analysis has been devoted to time-delay systems and the problem of delay compensation [7], [12], [20], [22]. Perhaps the most popular method of delay compensation is *model predictive control* (MPC). MPC uses a model of the plant to be controlled to generate control signals so that the system can be controlled as if there were no delay in the loop. The Smith predictor is an example of such a controller [12]. However, MPC can be very sensitive to errors in the model, including specific sensitivity to errors (or "mismatches") in the model of the delay, which is usually constant [2]. Because NCSs are subject to random, time-varying delays, this presents a significant obstacle.

One proposed method for implementing MPC for delay compensation in a networked environment is the use of play-back buffers [11]. Play-back buffers can be used to reduce the variability in the loop delay, but only by effectively increasing the delay. This paper aims to initiate the discussion of the control-related aspects of using play-back buffers in systems with time-varying loop delays such as those found in NCSs. First, we describe the value of removing uncertainty in the delay, showing the types of situations where the value of removing uncertainty in the delay is greater than the decrease in performance due to the effective increase in loop delay. Then we address some design issues for the implementation of the Smith predictor using a play-back buffer.

In Section II, we introduce the concept of play-back buffers and their adaptation for networked control. In Section III, we consider the situation when all uncertainty in the delay can be removed (using a bounded-interval delay distribution) and compare the performance using a play-back buffer with a Smith predictor to the use of unbuffered PID control. In Section IV, we explore two basic problems in the design of a networked control system using play-back buffers: how to choose the optimal play-back delay with average-case analysis using knowledge of the delay distribution and how to design the internal controller for a Smith predictor. Finally, Section V concludes the paper and discusses possibilities for future work.

II. Play-Back Buffers

Play-back buffers were originally designed for multimedia play-back [16]. In [11], Liberatore proposed an algorithm to integrate a play-back buffer with networked control for the control algorithm, actuator, and sensor. The main feature is a buffer located at the actuator which delays the application of a control signal until a specified *play-back time* is reached. The play-back time is determined at the controller and is paired with the appropriate control signal in a single packet. Control signals which arrive after the play-back time are applied immediately. For an example of the effects of a play-back buffer on an arbitrary signal, see Fig. 2. The *play-back delay*, the difference between the time when the plant is sampled and when the control signal calculated from that output is applied, is chosen to remove much of the variability in the loop delay (or round-trip time), and therefore the application of the control signal is more predictable. This is particularly advantageous for model predictive controllers.

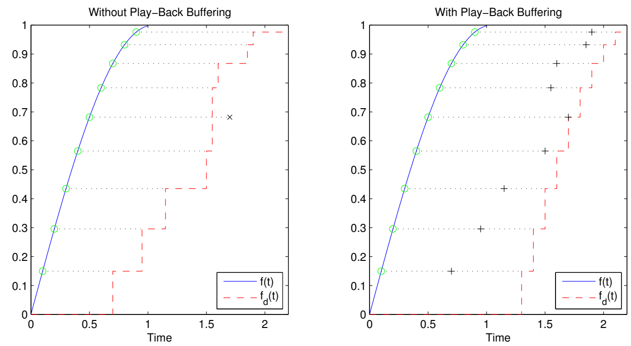

Fig. 2. An example demonstrating the effects of a play-back buffer in the time domain. The continuous-time signal to be sampled, $f(t)$, is solid blue. The left plot shows the delayed signal, $f_d(t)$, without play-back buffering, where one sample (marked by an "x") is dropped due to out of order arrival. The plot on the right shows the delayed signal using a play-back buffer, removing the uncertainty in the delay using a play-back delay of 1.2 s (thus the output samples are again equally spaced with $h = 0.1$). Instantaneous delays before buffering are denoted with "+".

In this paper, we focus on the control-related performance and design issues regarding the basic idea of removing uncertainty in the delay by adding a play-back delay. Clearly, adding any delay to a closed-loop system generally degrades performance. Therefore, we must investigate the design of a controller which takes advantage of an effectively more deterministic loop delay and evaluate the overall performance of the resulting system.

In addition to the control signal and play-back time, the integrated algorithm proposed in [11] includes mechanisms that we do not implement here. For example, it proposes *contingency control*: a conservative control signal which has a longer play-back delay and is intended to prevent extended application of an aggressive control signal when a new control signal is not received on time. The integrated algorithm also assumes the actuator only has enough local memory to be able to hold a single packet, and thus discards, or *quashes*, a control signal waiting to be applied if a newer control signal is received. To limit the number of quashed signals, the algorithm uses variable sampling periods and an updated guess of the minimum round-trip time (RTT). However, when using a constant sampling rate and a play-back delay significantly beyond the body of the delay distribution, a high percentage of control signals may be quashed because several packets are usually received before the play-back time. This leads to significantly degraded performance, so we always assume the actuator has enough memory so that no signals are quashed, and therefore we can independently study the effects of longer play-back delays.

III. Initial Evaluation

In general, we divide controllers for systems with time-varying loop delays into two categories: buffered and unbuffered. First, a buffered controller (i.e., one using a play-back buffer) can take advantage of an effectively more deterministic loop delay by using MPC for delay compensation. As a consequence, the controller can be designed very aggressively if a good plant model is available. The drawback of this method is the effective increase in loop delay due to the play-back buffer. Here, we use the Smith predictor for MPC and abbreviate its combination with a play-back buffer "SP-PB".

On the other hand, an unbuffered controller applies the control signal as soon as it is received. An example of a controller which can be implemented effectively without buffering is a PID controller (because the "D" term serves as a crude predictor that is sufficiently robust to the time-varying nature of the delay). Intuitively, we know that for small delays, a PID controller can perform very well. However, the drawback is that for longer delays, the gains of the PID controller become conservative and performance is significantly degraded. Certainly, gain scheduling of PID control could be done in this case, as Nilsson did for LQR in [14], but this would require knowledge of the current round-trip time (RTT), including a prediction of the next controller-to-actuator delay. Implementation of this type of unbuffered controller is beyond the scope of this paper, and, indeed, the SP-PB does not require such information. (Also, an unbuffered Smith predictor could be used to compensate for the delay, but when the delay is time-varying, the Smith predictor's sensitivity to these changes in delay leads to prohibitively poor performance.)

Therefore, in this section we compare the performance of a buffered controller to an unbuffered controller in order to more precisely characterize the situations in which the buffered controller shows superior performance. To simplify the analysis, we will use a bounded-interval delay distribution and set the play-back delay equal to the maximum possible delay, thereby removing all variability in the delay. We start by describing how time-varying, random loop delays were generated for our comparison experiments.

A. Generating Delays for Experiments

While network delays are usually modeled by a semi-infinite, heavy-tailed distribution defined on $[\tau_{min}, +\infty)$ [13] (and we will also use a similar model later in Section IV. C.), we begin by studying the case when all uncertainty in the loop delay can be removed. Therefore, we generate random delays using a bounded-interval distribution defined on $[\tau_{min}, \tau_{max}]$. We use the beta distribution, whose *probability density function* (PDF) on the interval $[\tau_{min}, \tau_{min}]$ is given by

$$f(x) = \frac{\left(\frac{x-\tau_{min}}{\tau_{max}-\tau_{min}}\right)^{\alpha-1}\left(1-\left(\frac{x-\tau_{min}}{\tau_{max}-\tau_{min}}\right)\right)^{\beta-1}}{(\tau_{max}-\tau_{min})\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\,du}. \quad (1)$$

It should be noted that this distribution was not chosen because it specifically matched any real network data. However, we chose this distribution for qualitative reasons: in most real network delay distributions, most delays will be close to the minimum delay and the system will be subject to less frequent, long delay spikes [4]. If we set $\alpha = 1$, as β increases from 1 to ∞, the beta distribution shifts from uniform to an impulse at the minimum value; see Fig. 3. Therefore, as β increases and the play-back delay stays at τ_{max}, we can study the importance of how close the play-back delay is to the body of the delay distribution.

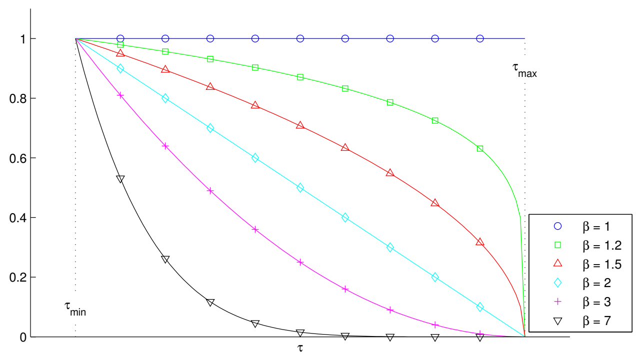

Fig. 3. Normalized PDFs for beta distribution (i.e., the numerator of Eq. (1)) for several values of β and $\alpha = 1$.

Here, we study varying values of β, τ_{min}, and τ_{range}, where $\tau_{range} = \tau_{max} - \tau_{min}$, and we always use $\alpha = 1$. We start with $\beta = 1$, which is the case of uniformly distributed delays, and increase β to 7. In [11], the default value of the minimum delay was 50 ms, so we consider values of the minimum delay in that neighborhood. Specifically, we consider $\tau_{min} \in [10, 150]$ ms and $\tau_{range} \in [0, 150]$ ms.

When generating independent random values for delays, the policy of dropping out-of-order packets can lead to the loss of a high percentage of packets. This is especially true for systems where the probability of longer delays is low, as these infrequent delay spikes will usually be dropped due to out-of-order arrival because the less likely long delays are most likely to be followed by a short delay. As a consequence, the system effectively suffers from a dropped packet instead of a delay spike. However, our purpose in this paper is to specifically study the effects of time-varying delays, not packet loss. (For analysis of systems with packet drops, see [3], [8], [19].) Additionally, in practice, many packets often arrive almost simultaneously after a delay spike. To simulate this behavior, the randomly generated delays were modified so that no packets would be dropped due to out-of-order arrival. Instead, the arrival of any packet that would arrive out-of-order (i.e., before a preceding packet) is moved until slightly after the older packet arrives. For example, if a packet sent at $t = 1$ s has a delay of 5 s and is followed by a packet sent at $t = 2$ s which has a delay of 3 s, the first packet would arrive at $t = 6$ s, after the second packet arrives at $t = 5$ s. In this case, we would increase the delay of the second packet to $4 + \delta$ s, where δ is an arbitrary, small number. Here, we used $\delta = 1$ ms. A similar method was used in [18].

B. Performance Comparison Methodology

Our purpose is to characterize the situations where a controller with a play-back buffer outperforms an unbuffered controller, and vice versa. These two controllers are described in detail below. It is important to test the two controllers on a plant whose dynamics are fast enough to be affected by the range of delay values which we will consider, so we use a first-order plant $G(s) = 1/(Ts + 1)$ with $T = 0.1$. We use a constant sampling time, $h = 0.01$.

We use step responses to test each controller, and the integral absolute error (IAE) metric is used to measure the performance. We compare their performance by simply subtracting the IAE of the SP-PB controller from the IAE of the unbuffered PID controller. Hence, because we want to minimize IAE, a positive cost difference will indicate superior performance for the SP-PB.

C. The Smith Predictor

The Smith predictor [2], [12] takes advantage of a plant model and a constant known loop delay (here, the same as the play-back delay) to control the system as if there were no loop delay. Thus, the internal controller should be designed for the delay-free system. Because we are controlling a first-order plant, we use a PI controller.

Because all the uncertainty in the delay will be removed by the play-back buffer, we can use the most aggressive PI gains possible for the delay-free design of the discrete controller. With $T = 0.1$, the gains $K_P = 10.508$, $K_I = 200$ bring the plant to the set point within one sampling period and hold it there until the set point changes. We can use these gains throughout.

With all uncertainty removed from the loop delay, the cost of a step response for the SP-PB can be trivially calculated without repeated simulation. It is always equal to the cost due to dead-time (which is equal to the dead-time, τ_{pb}, for a unit step response) plus the transient cost, which, again, is independent of the random loop delays. We also note that the component of the cost metric which is due to the dead-time (i.e., the play-back delay before the system responds) usually dominates the component of the cost metric which is due to the transient response. For example, for our range of parameters for the loop delay distribution, the dead-time cost will range from 0.010 to 0.300, whereas the transient cost is about 0.0049.

D. The Unbuffered PID Controller

The main drawback of using play-back buffers is the cost due to the intentionally added delay. Therefore, the SP-PB controller should be compared to a controller which instead acts immediately. Here, we will use a PID controller, but we must be careful to pick good gains so that the comparison is fair. In this section we describe how gains were optimized for different parameterizations of the delay distribution.

Optimal PID gains for systems with time-varying, random delays is a topic which has not been thoroughly studied (though some analysis was performed in [6]). For every plant and every delay distribution, PID gains could be found using optimization algorithms such as those used in [10] and [17], but this is hardly a feasible proposal when studying as many different delay distributions and parameterizations thereof as we are in this paper. Instead, we propose a method which tries to take advantage of the extensive work which has already been done for PID control of systems with a fixed loop delay. For first-order plants, a tuning rule can be used which takes a loop delay τ_d as a parameter so that gains become more conservative in a nearly optimal way as the loop delay increases. Herein, we use Wang's tuning rule which minimizes IAE for first-order plants with an ideal PID controller (see p. 162 in [15]). The tuning rule is given by

$$K_c = \frac{(0.7645 + \frac{0.6032}{\tau_d/T})(T + 0.5\tau_d)}{K(T + \tau_d)}, \quad (2)$$

$$T_i = T + 0.5\tau_d, \quad (3)$$

$$T_d = \frac{0.5T\tau_d}{T + 0.5\tau_d}, \quad (4)$$

where $K_P = K_c$, $K_I = K_c/T_i$, and $K_D = K_c T_d$. We propose using these gains for systems with random delays by finding an *effective delay*, τ_{eff}, for a given delay distribution. This effective delay will intuitively be between the minimum delay and maximum delay (if it exists) in the system; where it lies in between those values is determined by how the delays are distributed. This method takes advantage of the relationships between the gains given by the tuning rule so we only need to optimize over one parameter, τ_d, instead of optimizing over the full space of all three gains. This optimal τ_d is τ_{eff}.

For example, if the loop delays belong to the beta distribution with $\beta = 1$ (uniformly distributed delays) and $\tau_{min} = 50$ ms and $\tau_{max} = 100$ ms, we used simulation and found $\tau_{eff} = 86.7$ ms, which gives $K_P = 1.121$, $K_I = 7.820$, and $K_D = 0.0339$. If the delays are distributed more closely to the minimum delay, or equivalently, if β is higher, the gains can be more aggressive and consequently τ_{eff} is lower. For example, if β is increased to 4, $\tau_{eff} = 66.0$ ms.

The cost for unbuffered PID control of the system with random loop delays is estimated through simulation by averaging 24 successive unit step responses, alternating from zero to one, then from one to zero. The length of each step response is three seconds, which gives the system ample time to settle between changes in the reference signal.

E. Comparison Results

In Fig. 4, we show the cost difference for step tracking on a grid of values of τ_{min} and τ_{range} representing changes in the delay distribution. Here, $\beta = 3$, and the colormap is chosen specifically to distinguish the two regions where one controller outperforms the other. We pick $\beta = 3$ to show in Fig. 4 because for all lower values of β, the SP-PB always outperforms the unbuffered PID controller. Therefore, for $\beta < 3$, the play-back delay can always be set to τ_{max}, removing all uncertainty in the delay, because the unbuffered PID controller must be tuned too conservatively (relative to the SP-PB controller) due to the distribution of the delays between the minimum and maximum values.

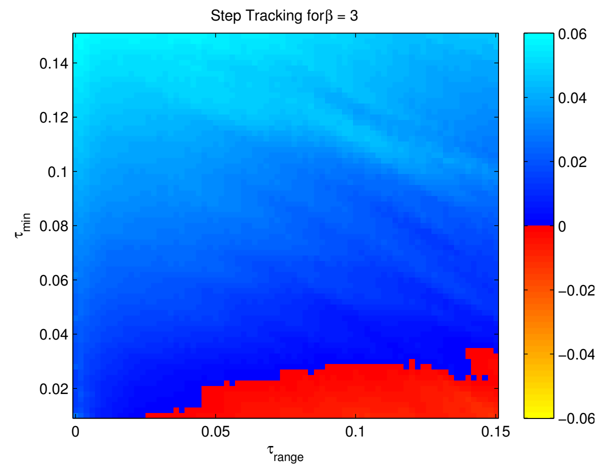

Fig. 4. Cost comparisons via subtraction for step tracking with $\beta = 3$. Positive values indicate SP-PB shows superior performance.

An important feature in Fig. 4 is the curve where the cost difference is zero that divides the regions for which each controller is best. Fig. 5 shows these isocurves for several values of β. As expected, the area below the curve (which is the region where the unbuffered PID controller outperforms the SP-PB) grows as β increases. The majority of the delay distribution is moving closer and closer to τ_{min}, so we expect the play-back delay of τ_{max} to eventually become too conservative.

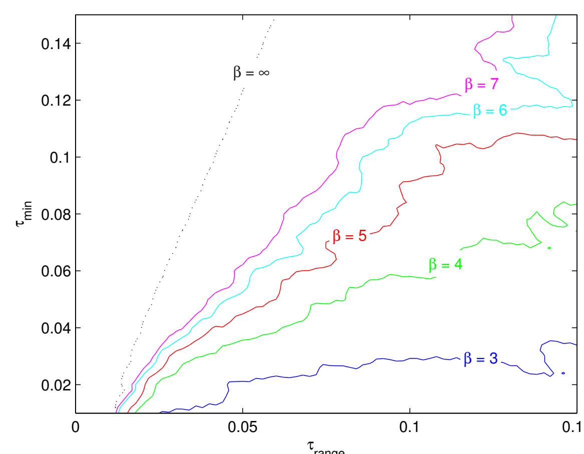

Fig. 5. Curves of equal cost for the unbuffered PID controller and the SP-PB. For all values of β, the area below and to the right of the curves is where the unbuffered PID controller has superior performance.

Interestingly, the curves in Fig. 5 seem to approach a limit. As β increases, we know that τ_{eff} increases less with τ_{range} because more and more delays are closer to τ_{min}. When $\beta = \infty$, all the delays are equal to τ_{min}, and therefore $\tau_{eff} = \tau_{min}$ regardless of the value of τ_{range}. As β increases, the cost of the unbuffered PID controller should approach this

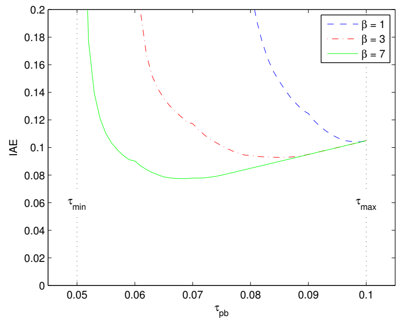

Fig. 6. Cost as a function of the play-back delay.

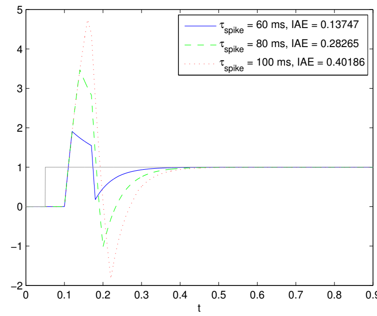

Fig. 7. Step responses for three delay spikes. Here, $T = 0.1$ and $\tau_{pb} = 50$ ms.

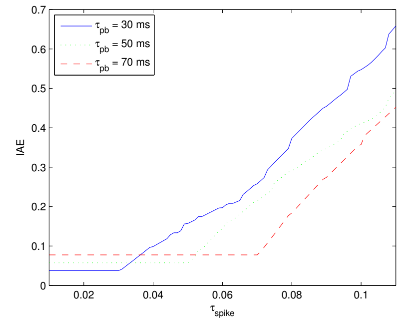

Fig. 8. Delay-spike cost as a function of the delay for three values of τ_{pb}.

case, and consequently, the cost comparison should approach the comparison with this constant delay case. The dividing curve for the $\beta = \infty$ case is shown with a dotted line in Fig. 5, and it appears to be the line which the other curves are approaching. This curve also shows that even when there is no randomness in the delay, as τ_{min} increases, the margin by which the SP-PB outperforms the PID controller increases as well.

Another way to interpret the results is to recognize the importance of choosing the play-back delay close to the body of the delay distribution. As β increases, the body of this delay distribution moves away from the play-back delay because we are naively always choosing the play-back delay equal to the maximum delay. Therefore, in these cases, choosing $\tau_{pb} < \tau_{max}$ is likely advantageous.

IV. DESIGN

In the previous section, we simply considered the case when all uncertainty in the loop delay could be removed and characterized situations when this improved performance. However, if the distribution of the delays is concentrated near the minimum delay, performance of the SP-PB controller can be improved by setting the play-back delay below τ_{max}. For example, Fig. 6 shows the IAE as a function of τ_{pb} when the delays are given by the beta distribution. We see that as β increases, the optimal play-back delay moves closer towards τ_{min}. Also, the maximum delay may be unknown or infinite if the delay is modeled by a semi-infinite interval delay distribution. In this section, we explore the optimization of the play-back delay using an approximation method (which uses analytical cost calculations) and then use simulation to verify the results. We also discuss the design of the controller's gains.

A. Choosing the Play-Back Delay: An Analytical Approximation Method

While complete simulation and optimization is the most straightforward way to determine the play-back delay, it can certainly be computationally expensive. Here, we propose a method to determine the optimal play-back delay for any arbitrary delay distribution using a matrix of costs determined by the step response of the plant in certain deterministic situations. Specifically, each cost entry in the matrix is associated with a possible loop delay and a candidate play-back delay. Then, for every candidate play-back delay, we find the weighted average of the cost over all possible loop delays, with weights being given by the probability distribution of the delay. This weighted average gives an "expected cost" for each candidate play-back delay, and the play-back delay with the minimum such expected cost should be near optimal.

Formally, for each play-back delay, we define its associated cost to be the weighted average of the cost of each possible delay spike,

$$J_{pred}(\tau_{pb}) = \sum_{i=1}^{N} w(\tau_i) J_{spike}(\tau_i, \tau_{pb}), \tag{5}$$

where $J_{spike}(\tau_i, \tau_{pb})$ denotes the cost of a worst-case delay spike of value τ_i with the play-back delay τ_{pb}, and N is the number of discrete delay values used. The weights $w(\tau_i)$ come from the discretized probability distribution of the delays, where

$$w(\tau_i) = F(\tau_i) - F(\tau_{i-1}), \tag{6}$$

where $F(x)$ is the *cumulative distribution function* (CDF) of the delay distribution, i.e. $F(x) = \Pr(\tau < x)$. Using $F(\tau_i) - F(\tau_{i-1})$ indicates that delay values are always rounded up when discretized.

The first step is to specify a cost as a function of a possible loop delay value and a candidate play-back delay. Here, we define this cost to be the IAE from the "worst-case" delay spike of length τ in a step response with the play-back delay set to τ_{pb}. In a step response, the worst time for a delay spike is the sampling period immediately following a step change in the reference signal, because at this point, the controller has just sent an aggressive control signal to the plant to bring the plant output up to the reference signal quickly. The following control signal is a lower value which is designed to hold the plant output at the reference, but if this value is not received, the actuator holds the last, aggressive control signal until a new control signal is received. Thus, if a delay spike is above the play-back time, the system will typically suffer a high overshoot, and additionally, the Smith predictor

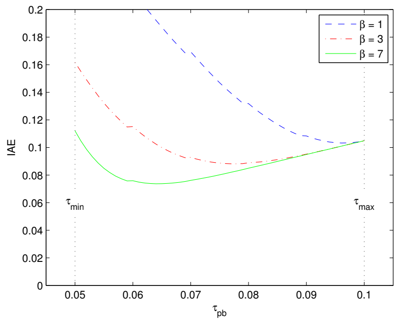

Fig. 9. Three examples of predicted play-back costs, $J_{pred}(\tau_{pb})$. Cf. Fig. 6.

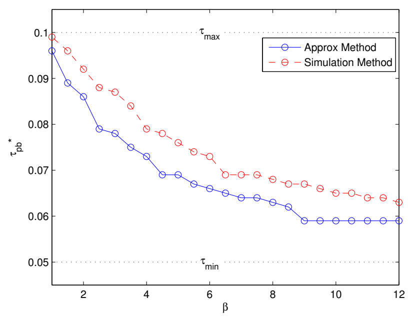

Fig. 10. Comparing τ_{pb}^* for both methods as β increases, with $\tau_{min} = 50$ ms and $\tau_{max} = 100$ ms.

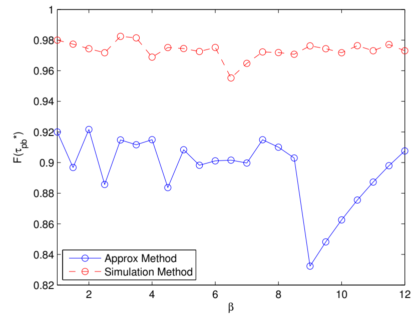

Fig. 11. Comparing $F(\tau_{pb}^*)$ for both methods as β increases, with $\tau_{min} = 50$ ms and $\tau_{max} = 100$ ms.

will also need a few sampling periods to re-synchronize with the plant. Fig. 7 shows three example step responses for such "worst-case" delay spikes above the play-back delay. In this section, we again use a PI controller with the aggressive gains $K_P = 10.508$ and $K_I = 200$.

For delay "spikes" below the play-back time (i.e., $\tau < \tau_{pb}$), the step response of the system will be unaffected, thus the cost associated with τ in this case will be equal to τ_{pb} (the dead-time cost for the step response) plus the delay-independent transient cost. For $\tau > \tau_{pb}$, the dead-time cost will be the same, but the cost from the transient response will now increase with the value of the delay spike (due to higher overshoot and longer settling time; see Fig. 7). The transient cost also depends on the play-back delay because for higher play-back delays, the controller will not be able to respond as quickly to bring the plant back to the reference signal. We used a grid of values from 10 to 150 ms for both τ and τ_{pb}, discretized to a 1 ms resolution. The cost of all delay spikes above τ_{pb} used in this paper are calculated analytically by using the solution of the first-order system. Fig. 8 shows the cost as a function of τ for a few values of τ_{pb}.

The next step is to find a predicted cost for each candidate play-back delay as prescribed by Eqs. (5) and (6). Fig. 9 shows the predicted play-back cost, $J_{pred}(\tau_{pb})$, for the beta distribution with $\tau_{min} = 50$ ms, $\tau_{max} = 100$ ms, and three values of β. We note that the play-back cost has two important expected features. First, the cost increases relatively quickly as τ_{pb} approaches τ_{min} from the right, because for these values of the play-back delay, the system is either unstable or has very poor performance as the play-back buffer is not removing enough uncertainty from the loop delay. Second, for higher values of β, the increase in cost approaches a linear function after the optimal play-back delay. We expect this because as the play-back delay increases past the most likely delays, the majority of the increase in cost will come from the dead-time cost, which is equal to the increase in play-back delay.

B. Optimal Play-Back Delays for the Beta Distribution

Once the two-dimensional grid of cost values is constructed, the optimal play-back delay can be found for an arbitrary delay distribution which predominantly lies within the set of delay spikes considered. In this section, we present the results of the approximation method for delays given by the β distribution. We compare these results to simulation results, which we find by averaging the costs from 70 simulated step responses. Fig. 10 compares the results as β increases. As expected, as β increases, the optimal play-back delay τ_{pb}^* decreases, or in other words, becomes more aggressive because the delay distribution moves towards τ_{min}. The results from the approximation and simulation methods are fairly similar, though the former is notably more aggressive.

We would also like to relate the optimal play-back delay to the delay distribution. For example, one might hypothesize that τ_{pb}^* is related to a constant value in the CDF of the delay distribution, i.e., $F(\tau_{pb}^*)$ should be constant even as the shape (here, β) or location (τ_{min} and τ_{max}) of the delay distribution changes. Specifically, one might expect τ_{pb}^* to always be above 95% of the possible delay values, i.e., $F(\tau_{pb}^*) \approx 0.95$ for all values of β, τ_{min}, and τ_{max}. Fig. 11 shows the value of the CDF evaluated at τ_{pb}^* as β increases. The simulation results show $F(\tau_{pb}^*)$ to be fairly constant around 0.98, but the approximation method does not show a similarly smooth relationship.

C. Optimal Play-Back Delays for a Heavy-Tailed Distribution

Now we consider the same distribution as Liberatore (see [11] and references therein) to model realistic network RTTs: a shifted gamma distribution for the body and a Pareto distribution for the tail. The PDF of the gamma distribution (before shifting) is defined on $[0, +\infty)$ and is given by

$$f(x; r, \lambda) = \frac{\lambda^r}{\Gamma(r)} x^{r-1} e^{\lambda x}, \qquad (7)$$

where

$$\Gamma(r) = \int_0^\infty s^{r-1} e^{-s} ds. \qquad (8)$$

The gamma distribution is then shifted so the minimum delay is τ_{min}. We will always use $\lambda = 1000$. Fig. 12 shows the gamma distribution for a few values of r. Delays are chosen

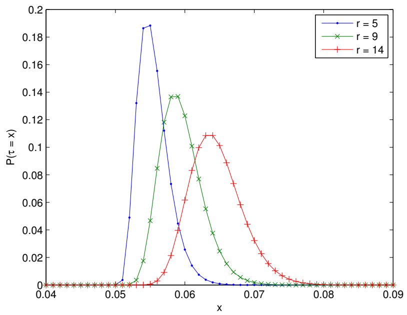

Fig. 12. The PDF of the gamma distribution discretized to a resolution of 1 ms. Here, $\tau_{min} =$ 50 ms.

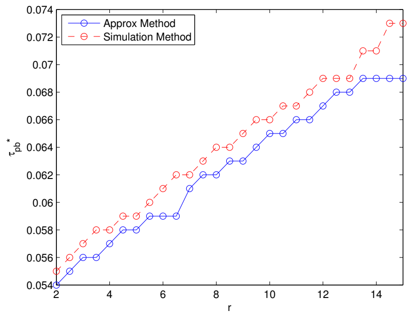

Fig. 13. Comparing τ_{pb}^* as a function of r for both methods, with $\tau_{min} = 50$ ms.

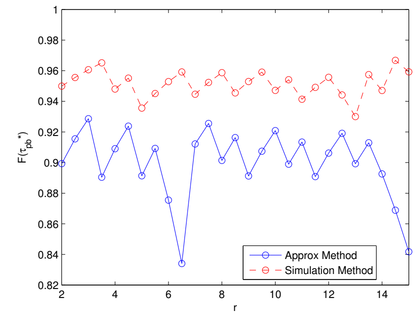

Fig. 14. Comparing $F(\tau_{pb}^*)$ as a function of r for both methods, with $\tau_{min} = 50$ ms.

from the body with a 99% probability. The PDF of the Pareto distribution is defined on $[K, +\infty)$ and is given by

$$f(x; K, \alpha) = \alpha \frac{K^\alpha}{x^{\alpha+1}}. \quad (9)$$

We will always use $\alpha = 1.5$ and $K = 2r/\lambda$, where K is chosen so that it is significantly beyond the mean of the gamma distribution. For $\alpha = 1.5$, the variance of the Pareto distribution is infinite, and therefore the distribution is said to be *heavy-tailed* [4].

As in Section III, we also adjust the delays after independent generation to remove out-of-order drops, but instead of removing all drops, they are removed with a 99% probability as in [11], [18]. Again, this is to simulate the almost simultaneous arrival of several packets after a delay spike.

We again compare the results to those from simulation, where the simulated cost again averages the IAE of 70 step responses. As the parametrization of the gamma distribution changes, the change in τ_{pb}^* as predicted by the approximation method is similar to the change in τ_{pb}^* using simulation. Fig. 13 shows the increase in τ_{pb}^* with r for both methods. Both methods show a linear increase in τ_{pb}^* with r. In other experimentation, we also found a straightforward relationship between τ_{pb}^* and τ_{min}, i.e., $\tau_{pb}^*(r, \tau_{min}) \approx \tau_{pb}^*(r, 0) + \tau_{min}$.

Fig. 14 shows the value of the CDF for the values of τ_{pb}^* predicted by the analytical method is not constant, but cycles around 0.91. These larger cycles are probably due to the larger jumps in the discretized version of the CDF for values of τ closer to the body of the distribution.

D. Choosing PI Gains for the Predictor

Above, we only considered the design of the play-back delay, but in the entire problem, we have the freedom to choose a few other parameters. Because the Smith predictor uses a PI controller, the gains of this internal controller are obviously important parameters in the design of the overall controller. Our default choice of gains are those which maximize performance of the delay-free system, that is, the most aggressive gains for the given sampling time. So while there is no reason to investigate the use of even more aggressive gains, we might expect more conservative gains to improve performance for the following reason. From Section III, we know that the dead-time cost component of the IAE usually dominates the transient cost. Therefore, although more conservative gains will lead to a higher transient cost, we should be able to use a more aggressive play-back delay and thus decrease the dead-time cost. In this section, we study the optimal play-back time and overall performance of the system with changes in the aggressiveness of the PI gains.

Intuitively, we know K_P is the gain which is most closely related to the overall aggressiveness of the controller. However, to decrease the total aggressiveness of the controller, we must also try to appropriately decrease K_I as well. Therefore, we use an alternate parametrization of the PI controller (where $K_P = K_c$ and $K_I = K_c/T_i$) and change the aggressiveness of the controller by changing K_c but keeping T_i constant. Thus, K_I will decrease with K_c, reducing the aggressiveness of both gains simultaneously.

Fig. 15 shows performance as a function of play-back delay for decreasing values of K_c and the baseline plant dynamics ($T = 0.1$). In this figure, we improved the resolution on the play-back delays to 0.5 ms to illustrate more clearly the behavior of the cost function. The figure confirms that more conservative gains can lead to a more aggressive optimal play-back delay (τ_{pb}^* decreases for both $K_c = 9.708$ and $K_c = 9.308$), but the overall performance is not improved. We found similar results for all values of the parameters we considered for both the beta and gamma delay distributions.

V. Conclusions and Future Work

Networked control systems present unique challenges for traditional control system design. This paper specifically focused on the use of play-back buffers to eliminate the variability in the loop delay, thereby enabling more effective use of a Smith predictor for delay compensation. We studied the value of removing all uncertainty in the loop delay given by a bounded-interval distribution. Using a first-order linear plant, we showed that play-back buffering is most valuable when the delays are distributed more evenly between the minimum and maximum delay, and also more valuable when the minimum delay is higher. These results also show the

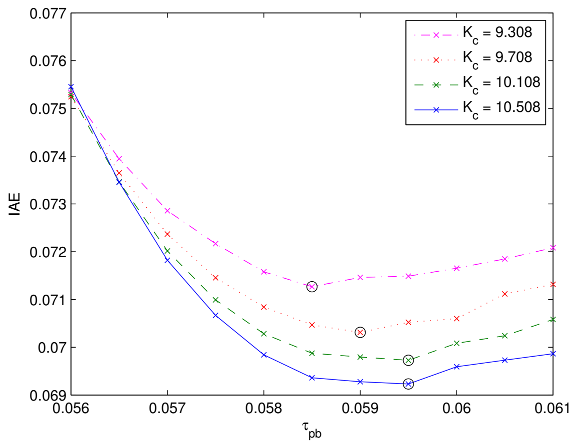

Fig. 15. Performance as a function of play-back delay for less aggressive gains. Here we use the combined gamma/Pareto distribution with $\tau_{min} = 50$ ms and $r = 5$.

importance of choosing a play-back delay close to the body of the delay distribution.

We also explored design issues for a Smith predictor with a play-back buffer controlling a first-order linear plant with loop delays given by both bounded-interval and heavy-tailed distributions. We presented an analytical approximation method for finding the optimal play-back delay. We also demonstrated a simple, intuitive relationship between the optimal play-back delay and the CDF of the delay distribution. We considered the design of gains for the PI controller, showing that the most aggressive gains gave the best performance, even if a slightly more conservative (i.e., longer) play-back delay was necessary. Thus, we can conclude that (when an exact model is available), the play-back delay is the most important parameter.

Our results here are far from complete. Future work should consider more aspects of co-design of the controller and the play-back buffer. For example, we only considered choosing gains for the PI controller strictly based on performance in a system free from model errors, disturbances, and sensor noise. The most aggressive gains may not be optimal under such conditions. Also, other aspects of Liberatore's full integrated play-back algorithm for networked control [11], including the integration of variable sampling times, adaptive play-back delays, and the use of a contingency control, should be studied. Additionally, because the buffer at the actuator could have other uses (as in [5]), other future work should consider the combination of such control strategies. Finally, more complete analysis of the randomness in the delay could determine a more accurate cost than the approximation method as presented (e.g., by calculating the true expected value of the cost), and this would also assess the accuracy of the approximation method.

VI. Acknowledgments

The research in this paper was partially supported by the NSF (grant CCR-0309910). Partial support was also provided by the NiuNiu Ji Fellowship for BS/MS students researching the area of systems, control, and signal processing at CWRU. The research in this paper does not necessarily reflect the views of either institution. We would also like to acknowledge many helpful discussions with Ahmad Al-Hammouri.

References

[1] G. W. Alldredge. *PID and Model Predictive Control in a Networked Environment*. Master's thesis, Dept. of Electrical Engineering and Computer Science, Case Western Reserve University, June, 2007.

[2] K. J. Åström and T. Hägglund. *Advanced PID Control*. The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC, 2005.

[3] M. S. Branicky, S. M. Phillips, and W. Zhang Scheduling and feedback co-design for networked control systems. In *Proc. IEEE Conf. on Decision and Control*, Dec. 2002.

[4] M. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. In *Proceedings of SIGMETRICS'96: The ACM International Conf. on Measurement and Modeling of Computer Systems.*, May 1996.

[5] M. Epstein, L. Shi, S. Di Cairano, and R. M. Murray. Control over a network: using actuation buffers to reduce transmission frequency. *European Control Conference.*, July, 2007.

[6] L. M. Eriksson and M. Johansson. PID controller tuning rules for varying time-delay systems. In *Proc. American Control Conf.*, pp. 619–625, July 2007.

[7] K. Gu, V. L. Kharitonov, and J. Chen. *Stability of Time-Delay Systems*. Birkhäuser, 2003.

[8] A. Hassibi, S.P. Boyd, and J.P. How Control of asynchronous dynamical systems with rate constraints on events. In *Proc. IEEE Conf. on Decision and Control*, pp. 1345–1351, Dec. 1999.

[9] D. Hristu-Varsakelis and W.S. Levine, editors. *Handbook of Networked and Embedded Control Systems*. Birkhäuser, 2005.

[10] H. N. Koivo and A. Reijonen. Tuning of PID controllers for varying time-delay systems. In *Proc. IEEE International Conf. on Mechatronics*, pp. 446–451, June 2004.

[11] V. Liberatore. Integrated play-back, sensing, and networked control. In *IEEE Infocom*, 2006.

[12] L. Mirkin and Z. J. Palmor. Control issues in systems with loop delays. In Hristu-Varsakelis and Levine [9], pp. 627–649.

[13] A. Mukherjee. On the dynamics and significance of low frequency components of Internet load. *Internetworking: Research and Experience*, 5:13-205, Dec. 1994.

[14] J. Nilsson. *Real-Time Control Systems with Delays*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, 1998.

[15] A. O'Dwyer. *Handbook of PI and PID Controller Tuning Rules*. Imperial College Press, London, England, Second edition, 2006.

[16] L. L. Peterson and B. S. Davie. *Computer Networks*. Morgan Kaufmann, 2000.

[17] M. Pohjola. *PID Controller Design in Networked Control Systems*. Master's thesis, Department of Automation and Systems Technology, Helsinki University of Technology, 2006.

[18] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne. Adaptive playout mechanisms for packetized audio applications in wide-area networks. In *IEEE Infocom* (2), pp. 680–688, 1994.

[19] L. Shi, M. Epstein, and R. M. Murray. Towards robust control over a packet dropping network. In *Proc. of the 17th Int'l Symposium on Mathematical Theory of Networks and Systems.*, 2006.

[20] G. Silva, A. Datta, and S. P. Bhattacharyya. *PID Controllers for Time-Delay Systems*. Birkhäuser, 2004.

[21] G. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems. In *Proc. American Control Conf.*, pp. 2876–2880, June 1999.

[22] L. Xiao, A. Hassibi, and J.P. How. Control with random communication delays via a discrete-time jump system approach. In *Proc. American Control Conf.*, pp. 2199–2204, June 2000.

[23] W. Zhang, M.S. Branicky, and S.M. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, 21:84–89, Feb. 2001.