Scheduling of Event-Triggered Controllers on a Shared Network

Cervin, Anton; Henningsson, Toivo

2008

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

47th IEEE Conference on Decision and Control, Cancun, Mexico, December 9–11, 2008.

Scheduling of Event-Triggered Controllers on a Shared Network

Anton Cervin and Toivo Henningsson
Automatic Control LTH
Lund University
Sweden

***Abstract*— We consider a system where a number of independent, time-triggered or event-triggered control loops are closed over a shared communication network. Each plant is described by a first-order linear stochastic system. In the event-triggered case, a sensor at each plant frequently samples the output but attempts to communicate only when the magnitude of the output is above a threshold. Once access to the network has been gained, the network is busy for T seconds (corresponding to the communication delay from sensor to actuator), after which the control action is applied to the plant. Using numerical methods, we compute the minimum-variance control performance under various common MAC-protocols, including TDMA, FDMA, and CSMA (with random, dynamic-priority, or static-priority access). The results show that event-triggered control under CSMA gives the best performance throughout.**

I. INTRODUCTION

Networked feedback control systems are normally implemented using periodic sampling at the sensor nodes, combined with either time-triggered or event-triggered communication between the sensor, controller, and actuator nodes. Periodic sampling allows for standard sampled-data control theory (e.g. [3]) to be used, although network-induced delay and jitter may limit the performance [4].

In recent work [2], [6], [9], [7], event-triggered sampling has been proposed as a means for more efficient resource usage in networked control. The basic idea is to sample, communicate, and control only when something significant has occurred in the system. For first-order stochastic systems, it has been shown that event-based sampling can significantly reduce the output variance and/or the average control rate compared to periodic sampling [2]. A similar idea is to introduce a deadband in the sensor. The trade-off between network traffic and control performance for higher-order control loops with deadband sampling was studied in [8].

When multiple control loops are closed over a shared medium (like a communication bus or a wireless local-area network), a multiple access method such as TDMA (time division multiple access), FDMA (frequency division multiple access), or CSMA (carrier sense multiple access) is needed to multiplex the data streams. It is clear that the choice of access method can have a great impact on the control performance. Intuitively, TDMA should be suitable for time-triggered control loops, while CSMA, being a random-access method, would seem to be well suited for event-based control. FDMA provides a way to share the bandwidth without regard to synchronization among the loops, which could potentially be beneficial for both time-triggered and event-triggered control. At the same time, less bandwidth per control loop means longer transmission times and hence longer feedback delays.

Multi-loop networked control systems—taking into account issues such as clock synchronization, medium access, communication protocols, imperfect transmissions, delay and jitter, and event-triggered sampling, as well as the control algorithms themselves—are very complex systems. To facilitate analysis, great simplifications are needed. In this paper, we study a scenario where a number of independent control loops are closed over a shared network (see Fig. 1). Using very simple models for the plants, controllers, and network arbitration, we are able to numerically compute and compare the minimum-variance control performance under the various medium access protocols. In particular, we apply recent results in sporadic event-based control of first-order systems [7], [5] to model and analyze the interaction between control loops and medium-access schemes. Although far from an exhaustive study, the results offer some interesting insight into the suitability of the studied MAC-protocols for networked control.

The remainder of this paper is outlined as follows. In Section II, the system description is given. Section III reviews how to calculate the stationary variance under time-triggered and event-triggered sampling. In Section IV, we model the medium access schemes and describe the co-design problem associated with each scheme. Section V reports numerical results for symmetrical integrator plants. In Section VI, we digress and compare the achievable performance under global vs local scheduling decisions. Section VII contains a case study with three asymmetric plants. Finally, the conclusions are given in Section VIII.

II. SYSTEM DESCRIPTION

We consider a system where N control loops are closed over a shared network. Each plant $i \in 1 \dots N$ is described by a first-order stochastic differential equation

$$dx_i(t) = a_i x_i(t)dt + u_i(t)dt + \sigma_i dw_i(t), \quad x_i(0) = 0, \tag{1}$$

where x_i is the state, a_i is the process pole, u_i is the control signal, w_i is a Wiener process with unit incremental variance, and $\sigma_i > 0$ is the intensity of the noise. All noise processes are assumed independent.

A sensor located at each plant i takes samples of the plant state at certain discrete time instants $\{t_{i,k}\}_{k=0}^{\infty}$:

$$x_{i,k} = x_i(t_{i,k}). \tag{2}$$

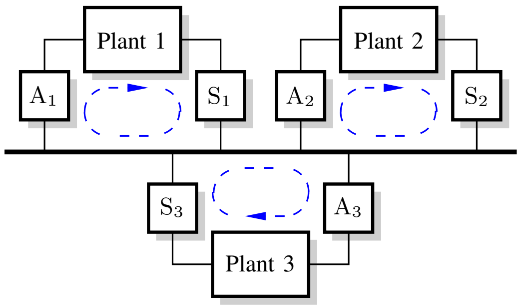

Fig. 1. Multiple control loops are closed over a shared communication medium. The controller in each loop may be co-located with either the sensor (S) or the actuator (A).

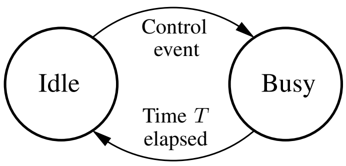

Fig. 2. Network state transitions. Control events may only be generated in the idle state.

The sampling can be either time-triggered or event-triggered, depending on the medium access scheme. After obtaining a sample, the sensor tries to initiate a control event by transmitting the value to the actuator. The network is however a shared resource that only one control loop may access at a time[1]. If two or more sensors attempt to transmit at the exact same time, a resolution mechanism determines who will gain access to the network. (The other nodes will simply discard their samples.) Once access has been gained, the network stays occupied for T seconds, corresponding to the transmission delay from sensor to actuator. During this interval, no new control events may be generated (see Fig. 2).

The controller in each loop may be co-located with either the sensor or the actuator; the network delay is assumed constant and known, so it does not matter which. The overall goal is to minimize the total cost

$$J = \sum_{i=1}^{N} J_i, \tag{3}$$

where the performance of loop i is measured by the stationary state variance

$$J_i = \lim_{t\to\infty} \frac{1}{t} \operatorname{E} \int_0^t (x_i(s))^2 ds. \tag{4}$$

In response to a sample taken at time $t_{i,k}$, the actuator is allowed to emit a Dirac pulse of size $u_{i,k}$. It is clear (see [5]) that minimum variance is achieved by driving the expected value of the state at time $t_{i,k} + T$ to zero, implying the deadbeat control law

$$u_{i,k} = -e^{a_i T} x_{i,k}. \tag{5}$$

[1]This is not true under FDMA. Under FDMA, we rather assume that each control loop has access to its own private network with lower bandwidth.

The control signal generated by actuator i is hence given by the pulse train

$$u_i(t) = \sum_{k=0}^{\infty} \delta\big(t - t_{i,k} - T\big) u_{i,k}. \tag{6}$$

While it may seem unrealistic to allow Dirac controls, it allows for a fair and straightforward comparison between time-triggered and event-triggered control. The Dirac pulse may be replaced by an arbitrary pulse shape of length no longer than T at the expense of slightly more complicated cost calculations.

III. Evaluation of Cost

We here briefly review how to compute the cost (4) under time-triggered and event-triggered sampling with a delay and minimum inter-event interval T. For more details, see [1], [7], [5]. For clarity, we here drop the plant index i.

A. Time-Triggered Sampling

Under time-triggered sampling, the stationary variance (4) can be calculated analytically. The sampling instants t_k are known a-priori and do not depend on the plant state, which will be normal distributed at all times. The (possibly irregularly) sampled closed-loop system becomes

$$x_{k+1} = w_k, \tag{7}$$

where $\{w_k\}_{k=0}^{\infty}$ are independent, zero-mean Gaussian variables with variance $P(t_{k+1} - t_k)$, where

$$P(t) = \begin{cases} \sigma^2 \frac{e^{2at}-1}{2a}, & a \neq 0, \\ \sigma^2 t, & a = 0. \end{cases} \tag{8}$$

(Note that the delay does not affect the state distribution at the sampling instants.) Sampling the cost function gives

$$\operatorname{E} \int_{t_k}^{t_{k+1}} x^2 ds = Q(T) \operatorname{E}(x_k)^2 + J_v(t_{k+1} - t_k), \tag{9}$$

where

$$Q(T) = \begin{cases} \frac{e^{2aT}-1}{2a}, & a \neq 0, \\ T, & a = 0 \end{cases} \tag{10}$$

is the state weight due to delay, while

$$J_v(t) = \begin{cases} \frac{e^{2at}-2at-1}{4a^2}, & a \neq 0, \\ \frac{t^2}{2}, & a = 0 \end{cases} \tag{11}$$

accounts for the inter-sample noise (see e.g. [1]). Finally, we know that $\operatorname{E} x^2(t_k) = P(t_k - t_{k-1})$. Using the expressions above, it is straightforward to evaluate the cost under any static cyclic schedule.

B. Event-Triggered Sampling

Under event-triggered sampling, control events may only be generated when the network is idle and $|x(t)| \geq r$, where r is the event detection threshold. The state will no longer be Gaussian, which complicates the calculation of $\operatorname{E} x^2(t_k)$. A useful and realistic approximation is to assume that the sensor does not measure x continuously, but rather uses

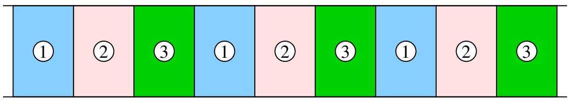

Fig. 3. Time division multiple access (TDMA). A static cyclic schedule determines which sensor node samples and transmits in which time slot.

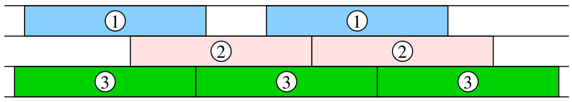

Fig. 4. Frequency division multiple access (FDMA). The bandwidth is divided into fixed shares, giving each loop a dedicated channel. Within each share, an event-triggered control loop is implemented.

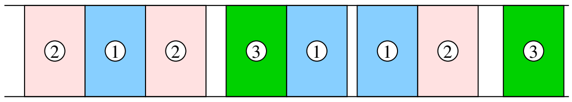

Fig. 5. Carrier sense multiple access (CSMA). Each loop is event-triggered. A static, dynamic, or random priority function determines who will transmit if many nodes try to access the network at the same time.

fast sampling with the interval $T_s \ll T$. The (irregularly) sampled closed-loop system then becomes

$$x_{k+1} = \begin{cases} e^{aT_s}x_k + w_k(T_s), & |x_k| < r \\ w_k(T), & |x_k| \geq r \ \& \ \text{won} \\ e^{aT}x_k + w_k(T), & |x_k| \geq r \ \& \ \text{lost} \end{cases} \tag{12}$$

where $\{w_k(t)\}_{k=0}^{\infty}$ is a sequence of independent, zero-mean Gaussian variables with variance $P(t)$; "won" means that the sensor node won the network arbitration, while "lost" means the opposite. Letting the system run in open loop between the fast samples, the expressions (8)–(11) for the sampled cost are still valid.

The update equation (12) is useful both for calculation of the state distribution and for Monte Carlo simulations. Because of the shared medium, the stationary probability distributions of $x_1, \ldots, x_N$ are not independent. To evaluate the cost using the first approach, it is hence necessary to find the multi-dimensional probability distribution $f(x_1, \ldots, x_N)$. This can in theory be done by gridding the state space and then iterating the distribution according to (12) until convergence. In practice, this can be done for a few dimensions, forcing us to rely on Monte Carlo simulations for $N \geq 3$ in this paper.

IV. Medium Access Schemes and Control Policies

In this section, we present simple scheduling and control models for three medium access schemes and discuss how to derive optimal schedules and control policies.

A. TDMA (Time Division Multiple Access)

In TDMA (see Fig. 3), a cyclic access schedule is determined off-line. In each slot in the schedule, one control loop has access to the network for T seconds. Since there is no cost associated with using the network in our problem formulation, it is obvious that no slot should be left empty, and that the sensor should always sample and transmit in its slot. Hence, the optimal control scheme associated with TDMA will be a pure time-triggered scheme.

For symmetric plants (with $a_i = a$, $\sigma_i = \sigma$, $\forall i$), a simple round-robin schedule is optimal. For asymmetric plants, an optimal schedule of length n can be found by evaluating the resulting cost for each possible schedule. (The search for an optimal schedule can be done more efficiently. The LQ-optimal cyclic scheduling and control problem for multiple higher-order plants is treated in [10].)

B. FDMA (Frequency Division Multiple Access)

In FDMA (see Fig. 4), the communication bandwidth is divided between the nodes, such that each loop receives a fixed fraction U_i of the total capacity $\sum_{i=1}^{N} U_i = 1$. Accounting for the lower transmission rate, the delay from sensor i to actuator i is now T/U_i.

It is previously known [7] that event-triggered sampling with a minimum inter-event interval T is superior to time-triggered sampling with the interval T, also when there is delay in the system. Hence, event-triggered control is the better choice for FDMA. The optimal event detection threshold and the associated optimal cost can be found numerically by sweeping r and computing the cost for each value.

For symmetric plants, an even division of the bandwidth is optimal. For asymmetric plants, the shares U_i can be found using optimization. Since the cost functions $J_i(U_i)$ are smooth and strictly decreasing, it is feasible to use standard nonlinear optimization tools to find the shares.

C. CSMA (Carrier Sense Multiple Access)

In CSMA (see Fig. 5), any node may try to access the network as soon as it becomes idle, making it suitable for event-triggered control loops. If many nodes want to transmit at the same time, some resolution mechanism must be used. In shared-medium Ethernet for instance, the collision detection and random back-off strategy will grant a random node access to the network (after some delay). In the Controller Area Network (CAN) on the other hand, access can be resolved based on either fixed (node) priorities or dynamic (message) priorities.

We will consider three different resolution mechanisms:

1) Random (CSMA-rand): As in Ethernet or WLAN, a random node will eventually win the contention. For simplicity, it is assumed that the resolution time is very small compared to the transmission time so that it can be neglected. The overall performance is optimized by selecting suitable event detection thresholds for the control loops. This is done by sweeping r_i and computing the cost for each value.

2) Static priority (CSMA-statprio): Each sensor node is assigned a static priority, which determines who will win the arbitration. Such a scheme can be useful for asymmetric plants where it is known that some plants are more sensitive to long access delays than others.

3) Dynamic Priority (CSMA-dynprio): For symmetric first-order plants, it can make sense to use the control error as a dynamic priority. (This idea was put forth in [11], where it was called the Maximum-Error-First (MEF) scheduling technique.) It is assumed that the network interface provides a mechanism (such as message priorities in CAN) so that priority access can be given to the node with the largest control error. It is obvious that this scheme will be better than random priorities. Again, the overall performance is optimized by selecting event thresholds for the loops.

V. Results for Symmetric Integrator Plants

We here present numerical results for N symmetric integrator plants with $a_i = 0$ and $\sigma_i = 1$. We assume that the network bandwidth scales in proportion to the number of plants, such that the transmission delay from sensor to actuator is $T = 1/N$ when the full bandwidth is utilized. For the numerical computations, we assume fast sampling with $T_s = T/100$.

Under TDMA, the optimal cyclic transmission schedule is $\{1, 2, \ldots, N\}$. The sampling period of each loop is 1 and the delay is $T = 1/N$, giving the following exact value for the cost per loop:

$$J_i = \left(J_v(T) + Q(T)\,\mathrm{E}\,x^2(t_k)\right)/T = \frac{1}{2} + \frac{1}{N}. \tag{13}$$

Under FDMA, each loop receives a share $U_i = 1/N$ of the bandwidth, implying the same performance regardless of the number of nodes. Computing the stationary state distribution under event-triggered sampling for different values of r, we find the optimal threshold $r = 1.06$, yielding the cost

$$J_i = 1.40. \tag{14}$$

For the CSMA case, we use Monte Carlo simulations to find the stationary variance of the plants under random or dynamic priority access. For each N, we sweep r to find the optimal threshold and the corresponding optimal cost. Each configuration was simulated for 10^8 time steps, corresponding to in the order of 10^6 simulated seconds. (The simulation time was around $15n$ seconds for each configuration on an Intel Core 2 CPU @1.83 GHz.)

The optimal costs under the various policies described above for $N = 1 \ldots 10$ nodes are reported in Fig. 6, and the optimal thresholds under CSMA are shown in Figs. 7. It is seen that TDMA outperforms FDMA, except for $N = 1$ where sporadic event-based control has the edge over periodic control. In turn, both variants of CSMA outperform TDMA, CSMA with dynamic priorities performing slightly better than CSMA with random access. The results are not surprising, since CSMA with event-triggered sampling dynamically allocates the bandwidth to the loop(s) most in need. A higher event threshold is needed for the random priority scheme in order to be more selective about which plant to control.

It is possible to reason about what happens when $N \to \infty$ under the various access schemes. Under TDMA, the performance approaches $J_i = 1/2$, while under FDMA, the performance is unaffected by N and is constant $J_i = 1.40$. CSMA approaches aperiodic event-based control [2] when $N \to \infty$, regardless of the priority scheme used. For integrator plants, the optimal cost per plant approaches $J_i = 1/6$. Hence, CSMA asymptotically gives 67% lower cost than TDMA and 88% lower cost than FDMA when the number of control loops increases. Equivalently, one can reason about the network capacity needed to maintain the same performance as the number of integrator plants grows. Here, again, CSMA will asymptotically require 67% less bandwidth than TDMA and 88% less bandwidth than FDMA to achieve the same cost per loop.

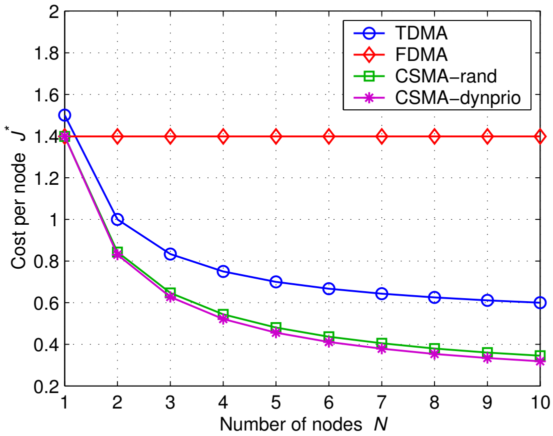

Fig. 6. Optimal cost per node vs number of nodes when controlling symmetric integrator plants.

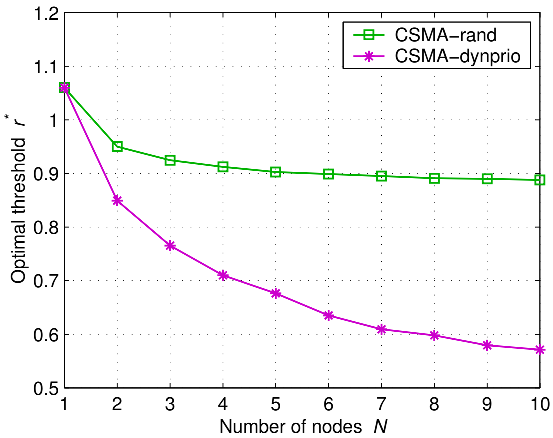

Fig. 7. Optimal threshold vs number of nodes for CSMA with random or dynamic priority access when controlling symmetric integrator plants.

VI. Local vs Global Knowledge

One important assumption in our model is that the decisions as to whether to transmit or not are taken locally at each sensor node. It was seen above that event-triggered control under CSMA with dynamic priority access gave the lowest cost among all the considered schemes. It is interesting to compare the performance to a controller with global knowledge of the plant states. Such a controller would

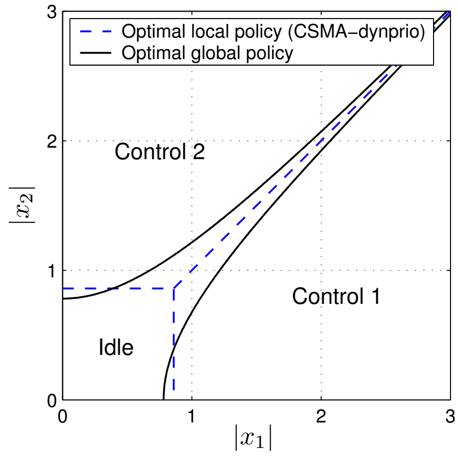

Fig. 8. Event-triggered control of two integrators: optimal global and local policies.

of course not be implementable in a networked setting but can provide a lower bound on the achievable cost.

We consider the special case of $N = 2$ symmetric integrator plants with the minimum inter-control interval and delay $T = 1/2$. The optimal local scheme under CSMA with dynamic priorities was computed above, giving the optimal cost $J_i = 0.834$ for the threshold $r = 0.85$. For the global scheme, we gridded the plant state space in the two dimensions and applied dynamic programming to derive the optimal control policy. For each state (x_1, x_2), the controller has the choice to control to the first plant, the second plant, or to idle. The resulting optimal global control policy is shown in Fig. 8, together with the local CSMA policy with dynamic priorities. It is seen that the control policies are quite similar. One difference is that the global controller will idle if both plants have about the same error magnitude, waiting to see where the processes will go next. The resulting cost under the global policy is found to be $J_i = 0.828$, which is only one percent lower than the cost for the optimal local scheme.

VII. Results for Three Asymmetric Plants

As a final numerical example, we consider a case where three asymmetric first-order systems should be controlled: one asymptotically stable plant, one integrator, and one unstable plant. The plant parameters are $\sigma_i = 1$ and

$$a_1 = -0.5, \quad a_2 = 0, \quad a_3 = 0.5.$$

Further, we let $T = 1/3$. Here, intuition tells us that more resources should be allocated to the unstable plant (Plant 3) while the stable plant (Plant 1) can manage with less resources.

For TDMA, the total cost is computed for all possible cyclic schedules of length $n = 2, \ldots, 12$. Since the unstable plant must be controlled at least once per cycle, we fix the first entry in the schedule to 3, leaving about 3^{n-1} schedules to test per value of n (including "necklace duplicates"). The optimal schedule for each value of n is reported in Table I. It is seen that the best schedule is of length 6: $\{3, 2, 3, 2, 3, 1\}$,

TABLE I

Optimal cyclic schedules for the three asymmetric plants.

Length n	Cyclic schedule	Total cost J
2	$\{3, 2\}$	2.651
3	$\{3, 3, 2\}$	2.708
4	$\{3, 2, 3, 1\}$	2.588
5	$\{3, 2, 3, 2, 1\}$	2.650
6	$\{3, 2, 3, 2, 3, 1\}$	2.563
7	$\{3, 2, 3, 3, 2, 3, 1\}$	2.589
8	$\{3, 2, 3, 2, 3, 2, 3, 1\}$	2.567
9	$\{3, 2, 3, 3, 2, 3, 2, 3, 1\}$	2.591
10	$\{3, 2, 3, 2, 3, 1, 3, 2, 3, 1\}$	2.573
11	$\{3, 2, 3, 3, 2, 3, 1, 3, 2, 3, 1\}$	2.588
12	$\{3, 2, 3, 2, 3, 1, 3, 2, 3, 2, 3, 1\}$	2.563

TABLE II

Optimal costs for the three asymmetric plants under the various medium access schemes.

Scheme	J_1	J_2	J_3	$J = \sum J_i$
TDMA	0.690	0.889	0.984	2.56
FDMA	1.000	1.177	1.319	3.49
CSMA-rand	0.554	0.618	0.772	1.94
CSMA-statprio	0.562	0.641	0.723	1.92

giving a total cost of $J = 2.56$. In the optimal schedule, the stable plant is controlled once per cycle, the integrator twice, and the unstable plant three times per cycle.

For FDMA, we optimize over the bandwidths U_1, U_2, U_3 to find the lowest total cost. For each plant, we first approximate the cost function $J_i(U)$ by sweeping r for each value of U. We then apply nonlinear optimization to find the optimal shares, yielding $U_1 = 0$, $U_2 = 0.397$, $U_3 = 0.603$ and the total cost $J = 3.49$. It is interesting to note that the long delay associated with FDMA apparently makes it pointless to control the stable plant.

For CSMA, we consider two arbitration mechanisms: random access and static priorities. For the random access scheme, we sweep the three thresholds to find the minimum cost, giving $r_1 = 1.12$, $r_2 = 0.92$, $r_3 = 0.77$, and the total cost $J = 1.96$. The three loops occupy the network on average 14%, 22%, and 38% of the time, while it is idle 26% of the time. The relative shares for the loops are not that different from the ones generated by the optimal cyclic schedule.

For the static priority CSMA case, we assume that the unstable plant has the highest priority, the integrator has medium priority, while the stable plant has the lowest priority. Again sweeping the three thresholds and evaluating the costs gives the optimal thresholds $r_1 = 0.95$, $r_2 = 0.87$, $r_3 = 0.77$, and the total cost $J = 1.94$. The priorities allow for tighter thresholds to be utilized. The three loops occupy the network on average 15%, 25%, and 38% of the time, while it is now idle 22% of the time.

The results under the various access schemes are summarized in Table II. We can again conclude that CSMA can provide better control performance than both TDMA and FDMA. For this example, CSMA gives 23% percent lower total cost than TDMA and 44% lower cost than FDMA. We further note that there is only a very modest improvement by using priorities, which is good news for wireless systems where random access schemes may be the only realistic choice for the implementation.

VIII. Discussion and Conclusion

This paper has studied a prototypical networked control co-design problem, where both the control policy and network scheduling policy have been taken into account. Although very simple mathematical models were used, some interesting conclusions regarding the various medium access schemes could be drawn. CSMA with event-triggered sampling was the superior scheme in all presented examples, while FDMA performed poorly due to the long transmission delay.

The simulation-based design approach adopted in this paper is conceptually easy to extend to higher-order plants and controllers. We have noted that the simulation time required to evaluate the cost with a given accuracy grows slower than the number of states in the system. Rather, the main problem with more realistic systems is the number of controller parameters that need to be optimized. For higher-order systems, it is probably necessary to impose restrictions on the controller structure and only optimize over a small subset of the parameters.

Another interesting approach would be to develop a way to characterize the performance of an event-triggered control loop as a function of its network resource usage pattern. Integrating several control loops, it should be possible to provide guarantees on the worst-case performance of each controller. Apart from higher-order plants and controllers, several other extensions to the work in this paper are possible to imagine, including

- having the controller located in a separate node, meaning that both the transmission from sensor to controller and from controller to actuator need to be scheduled.
- having more detailed models of real network protocols, including, e.g., the random back-offs in CSMA/CD.
- allowing MIMO systems, where each sensor and actuator may reside on a different node in the network.
- modeling measurement noise, variable transmission times, and lost packets.

Acknowledgment

This work has been supported by the Swedish Research Council (VR).

References

[1] K. J. Åström. *Introduction to Stochastic Control Theory*. Academic Press, New York, Jan. 1970.

[2] K. J. Åström and B. Bernhardsson. Comparison of periodic and event based sampling for first-order stochastic systems. In *Preprints 14th World Congress of IFAC*, volume J, pages 301–306, Beijing, P.R. China, July 1999.

[3] K. J. Åström and B. Wittenmark. *Computer-Controlled Systems*. Prentice Hall, Jan. 1997.

[4] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Årzén. How does control timing affect performance? *IEEE Control Systems Magazine*, 23(3):16–30, June 2003.

[5] A. Cervin and E. Johannesson. Sporadic control of scalar systems with delay, jitter and measurement noise. In *Proc. 17th IFAC World Congress*, Seoul, Korea, July 2008.

[6] D. Hristu-Varsakelis and P. R. Kumar. Interrupt-based control over a shared communication medium. In *Proc. IEEE Conference on Decision and Control*, 2002.

[7] E. Johannesson, T. Henningsson, and A. Cervin. Sporadic control of first-order linear stochastic systems. In *Proc. 10th International Conference on Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science 4416, Pisa, Italy, Apr. 2007. Springer-Verlag.

[8] P. G. Otanez, J. R. Moyne, and D. M. Tilbury. Using deadbands to reduce communication in networked control systems. In *Proc. American Control Conference*, 2002.

[9] M. Rabi. *Packet based Inference and Control*. PhD thesis, Institute for Systems Research, University of Maryland, 2006. URL: http://hdl.handle.net/1903/6612.

[10] H. Rehbinder and M. Sanfridson. Scheduling of a limited communication channel for optimal control. *Automatica*, 30(3):491–500, 2004.

[11] G. C. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems. In *Proc. American Control Conference*, 1999.