Multi-Process Control Using Queuing Theory

Magnus Egerstedt and Yorai Wardi
{magnus,yorai.wardi}@ece.gatech.edu
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

Abstract

We show how the problem of controlling multiple linear processes over a limited number of communication channels, i.e. by scheduling the control attention between the different processes, can be cast as a queuing problem. By defining Lyapunov-like functions for the individual subsystems, that can be interpreted as the queue lengths, we can produce suboptimal but computationally feasible solutions using standard results on optimal control of queues.

1 Introduction

The problem of controlling a collection of N unstable dynamical systems over $M < N$ channels has received considerable attention lately due to the emergence of sensor and actuator networks, embedded control systems, multi-agent robotic systems, and teleoperated actuator systems. (For a representative sample, see [2, 5, 6, 8, 9, 13].) The standard approach proposed in the literature has been to decouple the multi-process control problem into two parts, namely:

1. Control design of the different subsystems, using classic, control theoretic methods.
2. Scheduling of the order and duration at which the different systems are affected by the controller.

The first of these problems is simply the standard, stabilizing control design problem, while the second one calls for a new framework for distributing the control "attention" between the different processes. A lot of effort has been focused on formulating this scheduling problem as an optimal control problem ([7, 8, 10]). However, to produce computationally feasible solutions to these problems has been a goal that is hard to achieve, and in [8] the problem of finding the optimal, finite horizon scheduling strategy for discrete time LTI systems was shown to belong to the class of NP, using dynamic programming. In that work, heuristics for designing computationally more appealing methods, so called α-optimal methods based on tree-pruning, were suggested. However, no guarantees that these suboptimal solutions can be obtained in a computationally feasible way were given. In [6], another computational approach was taken by assuming that the different processes all get attention according to a fixed, alternation scheme, and the only thing that must be determined is the duration of each control period.

What is novel in this paper is that we complement the previous views on controller scheduling by exploiting a natural interpretation of the system performance within a *queuing theoretic* framework. The idea is that a Lyapunov-like function can be defined for each process, that basically measures how far from the desired equilibrium the process is (in some appropriate norm), and we show how this measure can be interpreted as the *queue-length* in a standard Possion arrival queue. For such systems there already exist a rich literature on optimal control (see for example [1, 3, 4, 12]), and by letting go of the continuous nature of the problem in order to get a problem that can be viewed as a queuing system, we immediately get closed form solutions to the scheduling problem. These solutions may be suboptimal, but they have a provably low computational complexity. In fact, this is the major contribution in this paper, i.e. a treatment of the multi-process control problem as a queuing problem for which there already exists a rich set of results that can be applied directly.

It should be noted already at this point that the problem of controlling multiple processes does not have to be decomposed into one control design part and one scheduling part, and, in fact, in [14] a solution is proposed that avoids this problem. However, in that particular limited-communications application, the cost for viewing the two problems simultaneously is a highly non-convex optimization problem.

The outline of this paper is as follows: In Section 2 we formulate the problem under consideration, and continue, in Section 3, by casting the controller scheduling problem within the queuing theory framework. In Section 4 we show how we can use the standard "μc-rule" for scheduling the control attention, and in Section 5 we discuss some possible extensions using "moving" zero-levels in the queues.

2 Preliminaries

Consider the problem of controlling a collection of N controllable, unstable processes

$$\dot{x}_i = A_i x_i + B_i u_i, \; i = 1, \ldots, N,$$

where $x_i \in \mathbb{R}^{n_i}, u_i \in \mathbb{R}^{p_i}, \; i = 1, \ldots, N$, and A_i, B_i are constant matrices of compatible dimensions. The subscript $i \in \{1, \ldots, N\}$ indicates which of the individual processes is under consideration.

What we want to do is to drive all of these systems to the origin, which would of course be a completely trivial problem in the absence of communication constraints. However, in this paper we assume that at each time instant we can only "talk" to M of the different systems ($M < N$), since the communication has to be done over a network. We furthermore assume that we control each system using a stabilizing state-feedback law of the form $u_i(t) = -K_i x_i(t)$ if we talk to process i at time t, and set $u_i(t) = 0$ otherwise. By committing to this design of the stabilizing control laws we have ignored the question of how to find the appropriate K_i's for the time being. Instead, the main part of this paper will be focused on the scheduling aspect, i.e. on determining which M of the N systems should be controlled at each particular time instant. It should be noted that in [8], the control, u_i, was kept at its latest value when the attention was shifted away from plant i, but that does not significantly change the nature of the problem.

Now, in order to fully characterize a proposed scheduling policy, two different questions need be to addressed:

1. In which order should the different processes be attended to?
2. For how long should each process be attended to?

In the sections to follow, we will show how these two questions can be answered in an intuitively appealing way within the arrival-process context. In order to make this claim concrete, some comments about decay and growth rates must be made.

As pointed out in [6], when using the stabilizing state-feedback controller, $u_i(t) = -K_i x_i(t)$, to control system i, there exists a quadratic Lyapunov function $V_i = x_i^T P_i x_i$ such that

$$(A_i - B_i K_i)^T P_i + P_i (A_i - B_i K_i) = -Q_i \prec 0$$

for some $Q_i \succ 0$ (positive definite), as well as

$$\dot{V}_i \leq \xi_i V_i,$$

for some ξ_i that satisfies

$$0 > \xi_i \geq 2 \max\{\mathrm{Re}(\mathrm{eig}(A_i + B_i K_i))\}.$$

Here ξ_i is the *decay rate* that produces an upper bound on the rate at which system i decays in the norm induced by the positive definite P_i ($\|x_i\|_{P_i}^2 = x_i^T P_i x_i$). In a similar manner, when the process is not under active control, we get that the quadratic function $V_i(x_i)$ satisfies

$$\dot{V}_i \leq \lambda_i V_i,$$

for some λ_i, such that

$$\lambda_i \geq 2 \max\{\mathrm{Re}(\mathrm{eig}(A_i))\}.$$

We say that λ_i is the *growth rate* of system i.

3 A Queuing Theoretic Formulation

The purpose of this section is to show how we can interpret the decay and growth rates in the previous paragraph in order to generate the *arrival* and *service rates* in a standard single-server queuing system. Our first observation is that for any function

$$\dot{F} \leq \alpha F,$$

we have that F is dominated by G (i.e. $F(t) \leq G(t), \; \forall t \geq 0$) if

$$\begin{aligned} \dot{G} &= \alpha G \\ G(0) &= F(0). \end{aligned}$$

We can thus set

$$\dot{W}_i(t) = \begin{cases} \xi_i W_i(t) & \text{if system } i \text{ is talked to at time } t \\ \lambda_i W_i(t) & \text{otherwise} \end{cases}$$
$$W_i(0) = V_i(0).$$

Since W_i dominates V_i we have established that $W(t) \geq V(t), \; \forall t \geq 0$. Now, if $\dot{G} = \alpha G$ then

$$\ln\left(\frac{G(t)}{G(0)}\right) = \alpha t.$$

In other words we can set

$$z_i(t) = \ln\left(\frac{W_i(t)}{V_i(0)}\right),$$

such that

$$\dot{z}_i(t) = \begin{cases} \xi_i, & \text{if system } i \text{ is talked to at time } t \\ \lambda_i & \text{otherwise} \end{cases}$$
$$z_i(0) = 0.$$

Since $z_i \to -\infty$ only if $W_i \to 0$, which, in turn, implies that $V_i \to 0$, we can think of z_i as a measure of how far away from the origin system i is, due to the monotonicity of $\ln(\cdot)$. We state this fact as a proposition:

Proposition 3.1 *If we let z_i be given as*

$$\dot{z}_i(t) = \begin{cases} \xi_i, & \text{if system } i \text{ is talked to at time } t \\ \lambda_i & \text{otherwise} \end{cases}$$
$$z_i(0) = 0,$$

then $z_i \to -\infty$ implies that $x_i \to 0$ in the original LTI system.

The idea now is that it should be possible to interpret z_i as the queue length in a queuing system, by identifying the relevant service and arrival rates.

Consider a single-server queuing system with infinite storage capacity, where λ is the average arrival rate of customers, and μ is the average service rate, as shown in Figure 1. We want to identify these parameters with functions of the previously derived growth and decay rates, and if we think of λ_i as the arrival rate to the ith process and of $\mu_i = \lambda_i - \xi_i$ as the service rate, we can clearly model the system as a *deterministic birth-death chain.* It should be pointed out that the steady state analysis of any birth-death chain only deals with the expected inter-arrival and service times, i.e. our deterministic model can be investigated using steady state queuing theory as well.

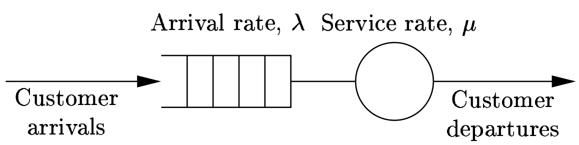


Figure 1: The standard, infinite capacity single-server queue.

We let the state of queue i be $S_i \in \{0, 1, 2, \ldots\}$, i.e. S_i is the length of queue i. Recall that the problem under investigation in this paper is how to drive the states of the original LTI systems to the origin. But, since this can only be guaranteed when $z_i = -\infty$, we need to establish bounds on z_i that gives us a performance that is "good enough", i.e. by letting the zero-level of the queue correspond to a small z_i-value we stabilize the control systems to a ball around the origin. As will be seen in Section 5, we can achieve asymptotic stability by shrinking the radius of this ball iteratively, but for the time being we have to define the following terms: We say that the *zero-level* of an individual queue is given by $\Omega \in \mathbb{R}$, i.e. we say that the queue length becomes zero as z_i intersects the set $(-\infty, \Omega]$ while decreasing. We furthermore, let the queue length remain zero until z_i becomes $\Omega + \Delta$, while increasing, and we interpret Δ as the *step-size* of the queue.[1] In other words, the discrete transition between $S_i = p$ and $S_i = p + 1$ occurs when

$$z_i \geq \Omega + (p+1)\Delta,$$

and the transition between $S_i = p$ and $S_i = p - 1$ takes place when

$$z_i \leq \Omega + (p-1)\Delta.$$

[1]Since the levels are the same for all queues, the arrival and service rates do not need to be scaled. It is, however, conceivable that different levels could be used for the different subsystems.

4 Scheduling

If we adopt the queuing theory language, the problem at hand consists of scheduling the activation of M out of N servers, working on N queues. The optimal solution to this well-studied problem is given by the *μc-rule*, i.e. if we assume that there is a cost c_i associated with the length S_i of queue i[2], we should always choose to service the M nonempty queues with the highest $\mu_i c_i$-value [11], and we state this results as a proposition for easy reference:

Proposition 4.1 (μc-Rule) *Given N queues, service should always be given to the M nonempty queues with the highest μc-value.*

The reason why this simple result is so powerful is that it directly gives us the scheduling policy as a function of the queue lengths, $S_1, \ldots, S_N$, which allows us to schedule the control attention without having to perform any intense computations. We illustrate the usefulness of this result with two examples:

Example 1
Let $N = 3, M = 2$, and consider the systems

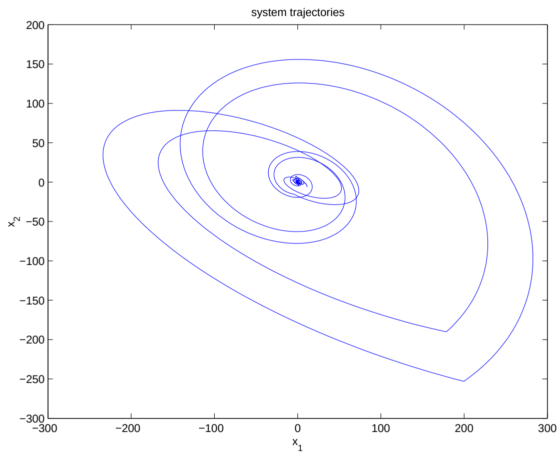
$$A_1 = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}, \; B_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$
$$K_1 = \begin{pmatrix} -0.366 & -2.91 \end{pmatrix}$$
$$\lambda_1 = 2, \; \mu_1 = 2.54, \; c_1 = 1, \; \mu_1 c_1 = 2.54$$

$$A_2 = \begin{pmatrix} 1 & 3 \\ -2 & 0 \end{pmatrix}, \; B_2 = \begin{pmatrix} -2 \\ 2 \end{pmatrix}$$
$$K_2 = \begin{pmatrix} -0.086 \; 1.45 \end{pmatrix}$$
$$\lambda_2 = 1, \; \mu_2 = 3.07, \; c_2 = 0.5, \; \mu_2 c_2 = 1.54$$

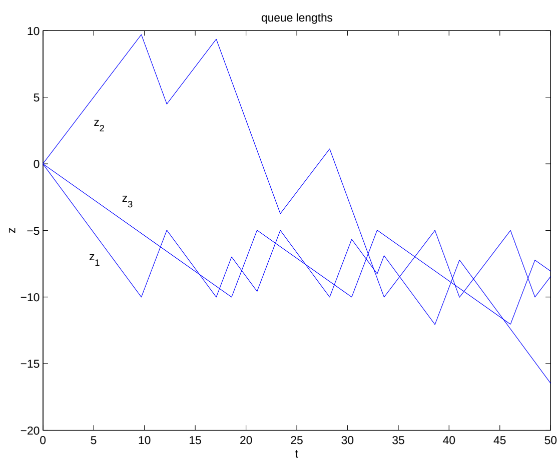
$$A_3 = \begin{pmatrix} 1 & -0.5 \\ 0 & -0.5 \end{pmatrix}, \; B_3 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$
$$K_3 = \begin{pmatrix} 5.22 & -1.53 \end{pmatrix}$$
$$\lambda_3 = 2, \; \mu_1 = 3.03, \; c_3 = 1.5, \; \mu_3 c_3 = 4.55$$

We see that queue 3 should be served before queue 2 and queue 1, while queue 1 has priority over queue 2 according to the μc-rule. In Figure 2 we see the effect of applying this scheduling strategy to this particular example. It should be noted that we do not drive the systems to zero since we interpret zero to occur at the zero-level, and we have set $\Omega = -10$, as well as $\Delta = 5$ in the example. As can be seen from Figure 2, when two or more of the queues have zero length then we simply apply the μc-rule again on these systems, which results in an asymptotic decay to zero of system 3, since that queue has the highest μc-value.

[2]If there is a cost $\gamma_i \|x_i\|^2_{P_i}$ associated with system i, then the cost $c_i z_i$ (to which $c_i S_i$ is directly related) bounds the original cost if $c_i = \ln(\gamma_i V_i(0))$.



(a)



(b)

Figure 2: In the upper figure, the trajectories of the three systems in Example 1 are depicted. Notice how system 2 gets further away from the origin initially since queue 2 has the lowest μc-value. In the lower figure, the Lyapunov-like z_i-functions are displayed as functions of time.

Example 2
We now only change A_3 slightly from the value used in Example 1 (we let systems 1 and 2 remain the same)

$$A_3 = \begin{pmatrix} 1 & -0.5 \\ 0 & -5 \end{pmatrix}, \; B_3 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$
$$K_3 = \begin{pmatrix} 5.22 & -1.53 \end{pmatrix}$$
$$\lambda_3 = 2, \; \mu_1 = 2.05, \; c_3 = 1.5, \; \mu_3 c_3 = 3.07$$

The result of applying the μc-rule to these systems is shown in Figure 3 ($\Omega = -10, \Delta = 5$), and there it can be seen that the solution is in fact unstable. This means is that it should be possible to use queuing theory directly to arrive at sufficient conditions for stability of the original systems in terms of the arrival and service rates. In fact, the following well-known proposition (see for example [3]) gives necessary and sufficient conditions for stability of the scheduled queue systems:

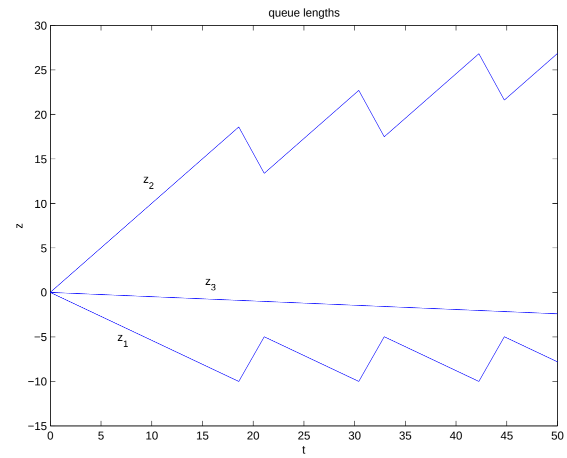


Figure 3: The unstable situation in Example 2 is shown.

Proposition 4.2 (Stability) *A queuing system consisting of N queues and $M < N$ active servers at each time instant can be stabilized if and only if*

$$\sum_{i=1}^{N} \frac{\lambda_i}{\mu_i} < M.$$

By comparing the results from Example 1 and Example 2 we see that in Example 1

$$\sum_{i=1}^{3} \frac{\lambda_i}{\mu_i} = 1.77 < 2,$$

while in Example 2

$$\sum_{i=1}^{3} \frac{\lambda_i}{\mu_i} = 2.09 > 2.$$

As a result of Proposition 4.2, the solution (optimal when interpreted within the queuing theory context), given by the μc-rule, will also result in a stable system, since otherwise it would no longer be optimal. So what we have in fact accomplished using the quantization induced by the choice of Ω and Δ is to give a solution that drives the original systems to the set bounded by $z_i \leq \Omega + \Delta$ if the systems can be stabilized under this particular choice of quantization levels.

However, we are not content to just drive the systems to a ball around the origin. Instead we want to drive the systems to the origin (if possible), and in the next section we investigate how this can be achieved.

5 Moving Queue-Levels

If we assume that

$$\sum_{i=1}^{N} \frac{\lambda_i}{\mu_i} < M,$$

i.e. that the queuing system can be stabilized, then we will have that

$$S_i(t) \in \{0,1\},\ \forall t \geq T,\ i = 1,\ldots,N,$$

for some $T \geq 0$. In fact, this is equivalent to having

$$z_i(t) \leq \Omega + \Delta,\ \forall t \geq T,\ i = 1,\ldots,N.$$

The reason for this can be seen from the following argument: Assume, without loss of generality, that $N = 2, M = 1$, and that

$$\frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} < 1.$$

Furthermore, assume that $\mu_1 c_1 \geq \mu_2 c_2$, and let t_1 denote the time at which $S_1(t_1)$ becomes 0, i.e. when $z_1(t_1) = \Omega$.

Now, by letting z_1 increase until $z_1 = \Omega + \Delta$, and then decrease until $z_1 = \Omega$, the temporal duration of this response is

$$\frac{\Delta}{\mu_1 - \lambda_1} + \frac{\Delta}{\lambda_1}.$$

During that time, the total change in z_2 is

$$\frac{-(\mu_2 - \lambda_2)\Delta}{\lambda_1} + \frac{\lambda_2 \Delta}{\mu_1 - \lambda_1}.$$

But, by multiplying the expression above with λ_1/μ_2 we get

$$-\left(1 - \frac{\lambda_2}{\mu_2}\right) + \frac{\lambda_1}{\mu_1 - \lambda_1}\frac{\lambda_2}{\mu_2},$$

which is less than

$$-\left(1 - \frac{\lambda_2}{\mu_2}\right)^2 < 0.$$

The control attention can thus be distributed as follows:

Algorithm 5.1 *Among the $P \geq M$ queues such that $S_i \geq 1$, stabilize the M systems with the highest μc-value. If $P < M$, stabilize all those P queues as well as the $M - P$ queues with $S_j = 0$ that have the highest μc-value.*

We can thus state the following proposition:

Proposition 5.1 *If*

$$\sum_{i=1}^{N} \frac{\lambda_i}{\mu_i} < M$$

then Algorithm 1 drives the Lyapunov-like functions (the z_i's) to the set

$$z_i(t) \in (-\infty, \Omega + \Delta],\ \forall t \geq T,\ i = 1,\ldots,N,$$

for some $T > 0$.

We now observe that when the zero-levels have been obtained for all queues, it should be possible to simply shift the levels (Ω and Δ) appropriately in order to drive the original processes to the origin.

For instance, we can simply let Ω be shifted by a constant while decreasing the step-size in order to drive $z_i \to -\infty$, and one algorithm that achieves is as follows:

Algorithm 5.2 *Use Algorithm 5.1 until*

$$z_i \leq \Omega + \Delta,\ i = 1,\ldots,N.$$

Then shift the levels as

$$\begin{aligned} \Omega &:= \Omega - \kappa \\ \Delta &:= \tfrac{1}{\alpha}\Delta, \end{aligned}$$

where $\kappa > 0,\ \alpha \geq 1$. After the shift, repeat the process.

By using this algorithm, asymptotic stability can be achieved by repeating the Proposition 5.1 argument repeatedly.[3] Thus, for any $x_i(0)$ and $\epsilon > 0$ there exists a $T \geq 0$ such that $x_i(t + T) < \epsilon,\ \forall t > 0$. Hence, we do in fact have asymptotic stability, and we can state the following proposition:

Proposition 5.2 *Algorithm 5.2 drives $z_i \to -\infty,\ i = 1,\ldots,N$, as $t \to \infty$ if the queuing system satisfies the conditions in Proposition 4.2.*

Corollary 5.1 *Algorithm 5.2 drives $x_i \to 0,\ i = 1,\ldots,N$, as $t \to \infty$ if the conditions in Proposition 4.2 are satisfied.*

In Figure 4 we display the result of letting $\kappa = 10,\ \alpha = 2$, while using the system matrices in Example 1.

[3] Any choice of $\alpha > 1$ will in fact lead to chattering executions, and a simple remedy to this problem is to let $\alpha = 1$.

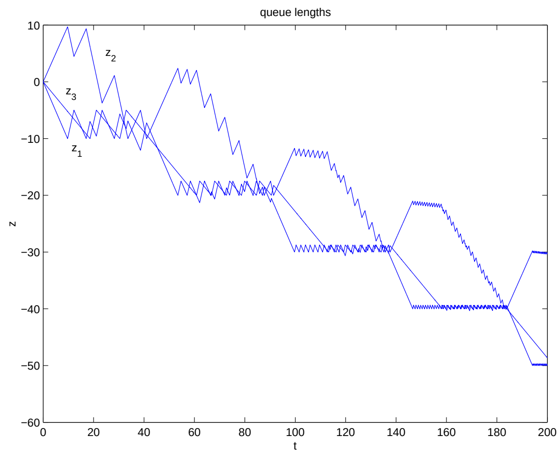


Figure 4: The effect of shifting the zero-levels and step-sizes in Example 1 is shown.

6 Conclusions and Future Research

The main contribution in this paper is the intuitively appealing idea of connecting multi-process control with queuing theory. It should be noted that this direction of research was indicated in [6], but the connection between the decay and growth rates with the arrival and service rates was never made. By making this connection, we show how to solve the scheduling problem by implicitly imposing a quantization on the system, which allows us to cast the problem as an optimal control problem for queues. We can thus apply the μc-rule in order to produce scheduling policies that require a minimal computational effort. By shifting the quantization levels iteratively, we furthermore achieve asymptotic stability, under some rate conditions.

An additional issue that needs to be addressed is the construction of the individual stabilizing control laws. We notice that Proposition 4.2 tells us that we should minimize the ratio λ_i/μ_i for each system in order to get the most out of our controllers from a stability point of view. In other words, by solving the minimization problem (once again indicated in [6])

$$\min_{K_i,P_i,\lambda_i,\xi_i} \frac{\lambda_i}{\lambda_i - \xi_i},$$

subject to

$$\begin{aligned}(A_i - B_iK_i)^T P_i + P_i(A_i - B_iK_i) < \xi_i P_i \\ A_i^T P_i + P_i A_i < \lambda_i P_i \\ P_i \succ 0,\ \xi_i < 0, \lambda_i > 0 \\ K_i \in \mathcal{K},\end{aligned}$$

we would find the best possible K_i from a scheduling point of view. The reason why we need to constrain $K_i \in \mathcal{K}$ is that otherwise the minimization problem might not have a solution (corresponding to infinitely high feedback gains). This line of research will be pursued in the future, even though it falls slightly outside the scope of this paper.

References

[1] D.P. Bertsekas. *Dynamic Programming and Optimal Control, Vols. 1 and 2.* Athena Scientific, Belmont, MA, 1995.

[2] R. Brockett. Stabilization of Motor Networks. *IEEE Conference on Decision and Control*, pp. 1484–1488, 1995.

[3] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems.* Kluwer Academic Publishers, Boston, MA, 1999.

[4] D. Hanson, D. Armbruster, and T. Taylor. On the Stability of Re-entrant Manufacturing Systems. In *Mathematical Theory of Networks and Systems*, Padova, Italy, 1998.

[5] D. Hristu and K. Morgansen. Limited Communication Control. *Systems & Control Letters*, No. 37, pp. 193–205, 1999.

[6] D. Hristu-Varsakelis. Feedback Control Systems as Users of Shared Network: Communication Sequences that Guarantee Stability. *IEEE Conference on Decision and Control*, pp. 3631–3631, Orlando, FL, 2001.

[7] B. Lincoln and B. Bernhardsson. Efficient Pruning of Search Trees in LQR Control of Switched Linear Systems. *IEEE Conference on Decision and Control*, Sidney, Australia, Dec. 2000.

[8] B. Lincoln and A. Rantzer. Optimizing Linear Systems Switching. *IEEE Conference on Decision and Control*, pp. 2063–2068, Orlando, FL, 2001.

[9] H. Rehbinder and M. Sanfirdson. Scheduling of a Limited Communication Channel for Optimal Control. *IEEE Conference on Decision and Control*, Sidney, Australia, Dec. 2000.

[10] E. Skafidas, R. Evans, and I. Marsells. Optimal Controller Switching for Stochastic Systems. *IEEE Conference on Decision and Control*, pp. 3950–3955, 1997.

[11] S. Stidham, Jr. Optimal Control of Admission to a Queueing System. *IEEE Transactions on Automatic Control*, Vol. AC-30, pp. 705–713, 1985.

[12] J. Walrand. *An Introduction to Queueing Networks.* Prentice-Hall, Englewood Cliffs, NJ, 1988.

[13] G. Walsh, H. Ye, and L. Bushnell. Stability Analysis of Networked Control Systems. *American Control Conference.* pp. 2876–2880, 1999.

[14] L. Xiao, M. Johansson, H. Hindi, S. Boyd, and A. Goldsmith. Joint Optimization of Communication Rates and Linear Systems. *IEEE Conference on Decision and Control*, pp. 2321–2326, Orlando, FL, 2001.