Performance Regulation via Integral Control in a Class of Stochastic Discrete Event Dynamic Systems

C. Seatzu∗ Y. Wardi**

* Department of Electrical and Electronic Engineering, University of Cagliari, Italy (e-mail: seatzu @ diee.unica.it).
** School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA (e-mail: ywardi@ece.gatech.edu).

Abstract: This paper presents a performance-regulation method for a class of Discrete Event Dynamic Systems (DEDS). The main idea is to use an integral controller with a variable gain, adaptively computed so as to guarantee effective output-tracking of a given reference value. The computation of the gain is based on the Infinitesimal Perturbation Analysis (IPA) gradient of the plant function with respect to the control variable, and the resultant tracking can be quite robust with respect to modeling inaccuracies and gradient-estimation errors. The proposed technique is tested on two examples concerning the regulation of the loss rate in a queue, and of inventory levels in a manufacturing-system represented by a Petri net. The results suggest its potential efficacy for a broader class of DEDS.

Keywords: Infinitesimal perturbation analysis, stochastic hybrid systems, Petri nets, performance regulation.

1. INTRODUCTION

Infinitesimal Perturbation Analysis (IPA) has been extensively applied to compute sample-path gradients of performance functions, defined on the state space of Discrete Event Dynamic Systems (DEDS), as functions of continuous parameters. Its main use so far has been in optimization of expected-value performance functions in conjunction with stochastic approximation algorithms. For extensive presentations of the IPA technique and its applications, please see Ho (1991); Glasserman (1991); Cassandras and Laforange (1999) and references therein.

Recently IPA has been extended from DEDS to a class of stochastic hybrid systems, based on the Stochastic Flow Model (SFM) framework (see Cassandras et al. (2010); Wardi et al. (2010); Yao and Cassandras (2013); Wardi and Cassandras (2013) for recent surveys). The SFM paradigm, comprising a generalization of fluid queues, has several inherent features rendering it especially suitable to the application of IPA. In particular, IPA gradients in the setting of SFM are statistically unbiased in a far larger class of systems than those in the setting of DEDS, and they often admit simple, model-free formulas and algorithms. Furthermore, in situations where an SFM acts as an abstraction of a DEDS, the SFM-based IPA gradients can provide sensitivity estimates of expected-value performance functions defined on the DEDS, which DEDS-related IPA gradients fail to yield. An additional feature of IPA in the SFM setting is that convergence to local minima of stochastic approximation algorithms exhibit considerable robustness to gradient-estimation errors; see Cassandras et al. (2002); Sun et al. (2004); Cassandras (2006); Panayiotou and Cassandras (2006); Yao and Cassandras (2013) for simulation results, and Wardi and Cassandras (2013a) for an initial analysis.

This paper leverages on the aforementioned results to propose a role for IPA in applications beyond optimization, namely in performance regulation of DEDS. In the scenario that we examine, a given reference performance value has to be maintained in the face of unpredictable factors such as system-modeling inaccuracies, input variations, changes in the system’s characteristics, and the effects of noise. This can be achieved with a suitable feedback control law having an integrator in the loop. However, an integral control may result in inadequate stability margins or large output oscillations. Moreover, a controller with a fixed gain may yield inadequate performance under changing system’s characteristics. To get around these difficulties we propose an integrator with a time-varying gain, adaptively computed in a way that (for a class of systems) broadens the stability margins and yields a faster tracking of the reference input than fixed-gain integral controllers.

To set the stage for our problem, consider a stochastic timed DEDS defined over a probability space \((\Omega, F, P)\), suppose that its state variable evolves in a way that depends on a continuous parameter \(\theta \in R\), and hence denoted by \(x(\theta, t)\). We view \(x(\theta, t)\) as a realization of a random process defined on the underlying probability
space, and depend on a sample $\omega \in \Omega$. Let us partition the positive-time axis into a sequence of contiguous intervals, or cycles, $C_n$, $n = 1, 2, \ldots$, and denote their respective lengths by $T_n$, $n = 1, 2, \ldots$. Thus, $C_1 = [0, T_1)$, $C_2 = [T_1, T_1 + T_2)$, etc. Furthermore, let $L_n(\theta)$ be a random function dependent on the state evolution during the $n$th cycle $C_n$. Given a performance reference $r$, we are concerned with regulating the process $L_n(\theta)$, $n = 1, 2, \ldots$, so that it approaches the value of $r$

For example, consider a stable GI/G/1 queue whose service-time processes depend on a parameter $\theta$. Given $T > 0$, let $L_n(\theta)$ be the mean delay of jobs that arrive during the cycle $C_n := [(n-1)T, nT)$. The queue’s processes can be reset in various ways (or none at all) at the start of $C_n$, but that is not the point. We are concerned with regulating the functions $L_n$ to a given reference value $r$.

Generally, we consider successive realizations of the functions $L_n(\theta)$ as the output sequence of a nonlinear, time-varying system whose input is $\theta$. Time is discrete and indicated by the counter $n$, and the time-variability is due to the sample path and the boundary condition (state) at the starting time of $C_n$. Furthermore, notwithstanding the fact that the state evolution during $C_n$ is inherently dynamic, we can view the system as memoryless by focusing merely on its input-output ($\theta-L_n(\theta)$) relations. This setting is natural for the forthcoming discussion of the integral controller that we consider. We will set the gain of the integrator during $C_n$ to the inverse of the IPA derivative $L_{n-1}^\prime(\theta)$, computed during the previous cycle, $C_{n-1}$. As will be explained in the sequel, the control law defined in this way yields effective regulation.

The rest of the paper is structured as follows. Section 2 describes the control system in an abstract setting. Section 3 provides a queuing example, and Section 4 presents a simulation example of balancing inventory and backorder in a Petri-net model of a manufacturing system. Section 5 concludes the paper and discusses potential extensions of the results derived therein.

2. REGULATION FRAMEWORK

Consider the discrete-time feedback system shown in Figure 1, where the counter $n = 1, 2, \ldots$ represents time. Both plant and controller are assumed to be single-input-single-output subsystems so that all the signals indicated in the figure are one-dimensional. Note that we use the unusual notation $\theta_n$ for the input to the plant, but it is common in the setting of IPA, which will be used to compute the sample gradients of performance functions with respect to $\theta_n$.

![Fig. 1. Basic regulation system](image)

Suppose that the plant is a nonlinear, memoryless, time-varying system represented by the functional relation $y_n = L_n(\theta_n)$, where the function $L_n : R \rightarrow R$ is assumed to be continuous in $\theta$. The purpose of the feedback system is to have the output signal $y_n$ asymptotically track a given reference value $r$.

To achieve such tracking, it is natural to have the controller contain an integrator. In its simplest form, an integral controller is a linear, time-invariant system with the transfer function

$$G_c(z) = A \frac{z^{-1}}{1-z^{-1}}.$$  

(1)

for a given gain $A > 0$, whose time-domain realization is defined by the equation $\theta_n = \theta_{n-1} + Ae_{n-1}$. However, a fixed-gain system may have the following two drawbacks. First, generally an integrator may result in a limited stability margin, and second, it may be impossible to determine a single gain which is adequate for every possible variation in the plant’s characteristics. For these reasons we explore a variable gain, $A_n$, $n = 1, 2, \ldots$, so that the closed-loop system is defined by the following equations:

$$\theta_n = \theta_{n-1} + A_ne_{n-1},$$  

(2)

$$y_n = L_n(\theta_n),$$  

(3)

and

$$e_n = r - y_n.$$  

(4)

The gains $A_n$ are computed adaptively according to the action of the system on the previous control variable, $\theta_{n-1}$. An effective choice for $A_n$ (for reasons that will become clear shortly) is $A_n = (L_n^\prime(\theta_{n-1}))^{-1}$, where “prime” denotes derivative with respect to $\theta$. However, this computation may not be exact, and only an approximation could be obtained. Thus, defining $K_n := (L_n^\prime(\theta_{n-1}))^{-1}$, $A_n$ has the form

$$A_n = K_n + \Delta K_n.$$  

(5)

Note that the computation of the last four equations is recursive if it is made in the order (5)-(2)-(3)-(4).

Consider now the simple scenario where the plant is time invariant and the computation of $K_n$ is exact, namely $L_n(\theta) = L(\theta)$ is independent of $n$, and $A_n = K_n$ in Equation (5). Then it can be seen that the system implements Newton’s method for solving the equation $L(\theta) = r$, for which there are well-known results guaranteeing that $\lim_{n \rightarrow \infty} e_n = 0$ (and hence $\lim_{n \rightarrow \infty} y_n = r$). This limit also is satisfied under the time-invariance assumption with inexact computation of $K_n$ in (5) as long as the relative error is less than $\gamma$ for a $\gamma \in (0, 1)$, namely $|\Delta K_n| \leq \gamma |K_n|$ for every $\theta_{n-1}$. Going a step further, suppose now that the plant is time varying in the sense that the functions $L_n(\theta)$ depend on $n$. In that case we cannot expect the error signal to converge to 0, but rather, to within a tolerance about 0 whose magnitude is a measure of the system’s variability.

In fact, Almoosa et al. (2012) showed that (under certain assumptions) for every $\epsilon > 0$ there exists $\delta > 0$ such that, if

$$|L_n(\theta_n) - L_{n-1}(\theta_{n-1})| < \delta$$  

for all $n$, then

$$\limsup_{n \rightarrow \infty} |e_n| < \epsilon.$$  

(7)

An important practical issue in a given control application is to compute the gain $A_n$ in real time. This was addressed in Almoosa et al. (2012,a) for regulating power.
and throughput in computer processors by the clock frequency of each core. In the case of throughput performance (Almoosa et al. (2012a)), the “plant” relating frequency to throughput is represented by a complicated multi-class queuing network, and the gain \( A_n \) is computed by an approximate IPA derivative. The resulting tracking was considerably better than those obtained for fixed-gain controllers.

In this paper we consider a general setting for regulation in DEDS, where \( A_n \) is computed by IPA. As described in Section 1, the time horizon is partitioned into \( N \) consecutive cycles (for a given \( N > 0 \)), \( C_1, \ldots, C_N \), and the plant-functions \( L_n(\theta) \) are represented by sample-performance functions defined on the system’s state trajectory during \( C_n \). The time variability of the system is induced by the particular sample path that is drawn during \( C_n \), and by the initial state in that cycle. Since the underlying system is discrete-event the derivative \( L'_n(\theta_{n-1}) \) can be highly biased and therefore its computation via IPA would not be useful. To get around this problem we use the IPA derivative-formulae obtained from an SFM abstraction of the system, driven by sample paths obtained from the discrete system. This setting will yield convergent tracking in the face of the changing plant. The next section presents two examples in detail.

### 3. APPLICATION TO A QUEUE

This section illustrates the aforementioned regulation framework by applying it to an example concerning the loss rate in an M/D/1 queue. The IPA derivative is highly biased and hence cannot be used, but instead we apply a formula which is derived from an SFM (fluid-queue) approximation to the sample paths obtained from the discrete queue. While this yields estimation errors due to modeling discrepancies, it guarantees the unbiasedness of IPA for the SFM and results in convergence of the regulation algorithm.

Consider an M/D/1/k queue with a finite buffer, where jobs arriving at a full queue are being discarded. Given an arrival rate, a buffer size, and a horizon period (cycle time), we control (regulate) the average loss rate per cycle to a given reference by adjusting the service times. Accordingly, we denote the arrival rate by \( \lambda \), the service times by \( s \), the buffer size (including the holding place at the server) by \( k \), and the horizon period by \( t_f \). The sample performance function of interest is the ratio of the number of discarded jobs during a cycle to the clock cycle \( t_f \).

Let us divide the time-axis into consecutive cycles, \( C_n \), each of duration \( t_f \) seconds. The control parameter is the service time, namely \( \theta = s \), and during \( C_n \) its value is denoted by \( \theta_n \). The sample-based function \( L_n(\theta) \) is the number of jobs discarded during \( C_n \) divided by \( t_f \), and the output at the end of \( C_n \) is \( y_n = L_n(\theta_n) \).

It is readily seen that the sample performance function is a monotone - nondecreasing step function. Furthermore, at a given \( \theta \), almost surely no arrival would occur at the same time when another job enters the server, and therefore (a.s.) \( L'_n(\theta) = 0 \) for every \( n \). This indicates that the IPA derivative \( L'_n(\theta) \) is biased, and \( A_n := (L'_n(\theta_{n-1}))^{-1} = \infty \) cannot be used in our regulation algorithm. Instead, we use the fluid-queue SFM paradigm as described in the next paragraph.

Consider a fluid queue with a finite buffer, a time-varying inflow rate, and a constant service rate. Suppose that the queue operates in a given time-interval \([0, t_f]\), where its instantaneous arrival rate, denoted by \( \alpha(t) \), is a random process. Denote its service rate by \( \beta \). Let \( \theta := \beta^{-1} \), and let \( \gamma(t, \theta) \) denote the instantaneous overflow (spillover) rate due to the limited buffer. Consider the random function \( L(\theta) \) defined as

\[
L(\theta) = \frac{1}{t_f} \int_0^{t_f} \gamma(t, \theta) dt; \tag{8}
\]

observe that \( L(\theta) \) is the average overflow rate in the interval \([0, t_f]\), and it can serve as an approximation to \( L(\theta) \) defined for the discrete queue. The precision of this approximation depends on the probability law underlying the process \( \alpha(t) \) as well as on the horizon time \( t_f \).

Reference Cassandras et al. (2002) showed that the IPA derivative \( L'(\theta) \) is unbiased, and furthermore, it is computable by a simple, model-free formula (listed below) that can act on the sample paths of the discrete queue. Therefore, we will do the regulation in the following way. In Equation (2) we use \( A_n \) defined as

\[
A_n := (L_n(\theta_{n-1}))^{-1}, \tag{9}
\]

namely we apply the IPA derivative-formula, obtained from an analysis of the SFM, to the sample path of the discrete queue during \( C_{n-1} \). In Equation (3), \( y_n \) is computable from the discrete queue. The effectiveness of the resulting regulation algorithm will be related to the quality of the approximation of \( L_n(\theta) \) by \( L_n(\theta) \).

The IPA derivative \( L'(\theta) \) has the following form (Cassandras et al. (2002)). Suppose that there are \( Q \) lossy busy periods during the horizon interval \([0, t_f]\), indexed by \( q = 1, \ldots, Q \) in increasing order (a busy period is lossy if any positive amount of overflow is incurred throughout its duration). For the \( q \)th busy period, let \( u_q \) be the first time loss occurs during it, and let \( v_q \) be its end point; in other words, \( u_q \) is the first time in that busy period when the buffer becomes full, and \( v_q \) is the next time the buffer becomes empty. Then, under mild assumption,

\[
L'(\theta) = \frac{1}{t_f} \theta^2 \sum_{q=1}^{Q} (v_q - u_q). \tag{10}
\]

With \( A_n \) defined by (9), we ran a simulation with the following parameters: \( y_{ref} = 0.1, t_f = 4,000, k = 3, \lambda = 0.9 \), the initial parameter-value was \( \theta_1 = 1.5 \), and the number of cycles was \( N = 100 \). The resulting graph of \( y_n, n = 1, \ldots, 100 \) is shown in Figure 2, where we notice convergence of the tracking algorithm in 3 iterations, to a band around the target value of 1.0. Within this band \( y_n \) fluctuates between 0.04 and 0.25, except for a single value of \( n \) where \( y_n = 0.039 \). The mean of \( y_n \) in the range \( n = 5, \ldots, 100 \) was 0.1005. To reduce the variability we ran the simulation for \( t_f = 20,000 \), and the results, shown in Figure 3, exhibit an equally-fast convergence of the regulation algorithm with fluctuations in the range of \([0.091, 0.114]\), and with mean (over \( n = 5, \ldots, 100 \)) of 0.1000.
Fig. 2. Loss in an M/D/1/k queue. $t_f = 4,000$, $n = 1, \ldots, 100$.

Fig. 3. Loss in an M/D/1/k queue. $t_f = 20,000$, $n = 1, \ldots, 100$.

4. APPLICATION TO A PETRI NET

The IPA technique recently has been extended from fluid queueing networks to a class of continuous Petri nets (Xie (2002); Wardi et al. (2013b)). References Wardi et al. (2013b); Seatzu and Wardi (2013) applied the results to an optimization example of balancing part-inventories with product backorders in a single-stage manufacturing system, and tested the application of IPA in conjunction with a stochastic approximation algorithm. This section uses the same example to test our approach to regulation.

The considered manufacturing system consists of a machine that produces a sequence of single-type products. The production schedule is driven by products’ orders while parts’ inventories are maintained as safety stocks. To make a product, the system must have an available part and a product order; parts without orders accumulate in the form of inventories, while orders without parts result in cumulative backorders. Naturally both excessive inventories and backorders are undesirable, and References Wardi et al. (2013b); Seatzu and Wardi (2013) devise an IPA-based algorithm for optimally balancing them. The underlying model for the algorithm is comprised of the continuous (fluid) Petri net (event graph) shown in Figure 4.

Continuous Petri nets are hybrid Petri nets where the flow of fluid tokens through transitions is represented by piecewise-continuous rate processes; see, e.g., Silva and Recalde (2004); David and Alla (2005) for comprehensive presentations. With regard to our system shown in Figure 4, transitions $T_1$, $T_2$, and $T_3$ represent, respectively, the processes of product-orders, parts’ arrivals, and the machine’s operation. Each transition $T_i$, $i = 1, 2, 3$, is characterized by a maximum fluid-flow rate $V_i(t) > 0$, which acts as an upper bound on its actual flow rate, denoted by $v_i(t)$. The places $p_1$ and $p_2$ are used for holding fluid, and at time $t$ the amount of stored fluid is denoted by $m_1(t)$ and $m_2(t)$, respectively. The processes $\{V_i(t)\}$, $\{v_i(t)\}$ ($i = 1, 2, 3$) and $\{m_j(t)\}$ ($j = 1, 2$) can be viewed as random processes defined over a common probability space $(\Omega, \mathcal{F}, P)$.

The dynamics of the system are described by the following three equations relating the above processes: For $i = 1, 2$,

$$v_i(t) = V_i(t).$$

(11)

For $i = 3$, define $\varepsilon_3(t) := \{j \in \{1, 2\} : m_j(t) = 0\}$; then

$$v_3(t) = \begin{cases} V_3(t), & \text{if } \varepsilon_3(t) = \emptyset, \\ \min(v_i(t) : i \in \varepsilon_3(t)), & \text{if } \varepsilon_3(t) \neq \emptyset. \end{cases}$$

(12)

As for $m_j(t)$, $j = 1, 2$, we have that

$$m_j(t) = v_j(t) - v_3(t).$$

(13)

In the forthcoming discussion we assume that the system evolves in a given time-interval $[0, t_f]$ with a given set of initial conditions $m_1(0)$ and $m_2(0)$.

In the typical case where the three processes $\{V_i(t)\}$ are exogenous, the other network processes, $\{v_i(t)\}$ and $\{m_j(t)\}$, are defined in their terms via Equations (11)-(13). In other situations some of the processes $\{V_i(t)\}$ are exogenous while others are controlled, and in this case the equations describing the controls together with (11)-(13) define all of the network processes. In the example considered in Wardi et al. (2013b); Seatzu and Wardi (2013) the processes $\{V_1(t)\}$ and $\{V_2(t)\}$ are exogenous while $\{V_3(t)\}$ is controlled. Specifically, product orders are assumed to arrive in batches, and hence

$$V_1(t) = \sum_{k \geq 1} \alpha_k \delta(t - s_k),$$

(14)

where $\delta(\cdot)$ is the Dirac delta function, $s_k$, $k = 1, 2, \ldots$, are the arrival times, and $\alpha_k$ represent the quantities of the orders. The machine is assumed to have deterministic service times, and hence $V_3(t) = V_3$ for a given $V_3 > 0$. The parts’ arrival rates are controlled by the backorders via a threshold in the following fashion: $V_2(t)$ has a given low value if the backorder levels are below the threshold,
and a given higher value if the backorder levels are above the threshold. Formally, given a threshold \( \rho > 0 \), and given constants \( V_{2,1} \geq 0 \) and \( V_{2,2} > V_{2,1} \), \( V_2(t) \) is defined via

\[
V_2(t) = \begin{cases} 
V_{2,1}, & \text{if } m_1(t) \leq \rho \\
V_{2,2}, & \text{if } m_1(t) > \rho.
\end{cases}
\]  

(15)

We assumed that \( V_{2,1} \leq V_2 \leq V_{2,2} \). It is obvious that Equations (11)-(13) and (15) have a unique joint solution for every set of initial conditions.

Now let us consider the threshold \( \rho \) as the variational parameter, and according to standard notation in IPA we denote it by \( \theta = \rho \). Then the processes \( \{V_2(t)\}, \{y_j(t)\}, \) \( i = 2,3 \), and \( \{m_j(t)\}, j = 1,2 \) are functions of \( \theta \) as well, and hence are denoted by \( \{V_2(\theta, t)\}, \{y_j(\theta, t)\}, \) and \( \{m_j(\theta, t)\}, \) respectively. Assume that a particular value of \( \theta \) remains fixed throughout the evolution of the system in a given interval \([0, \tau_f]\). Consider the sample performance function \( L(\theta) \) defined as

\[
L(\theta) = \frac{1}{\tau_f} \int_0^{\tau_f} m_2(\theta, t) dt,
\]

(16)

for a given distribution of the initial conditions \( m_1(0), j = 1,2 \), and let \( J(\theta) := E(L(\theta)) \) denote its expected-value function. Observe that \( L(\theta) \) is the average workload at place \( p_2 \) over the time-interval \([0, \tau_f]\). References Wardi et al. (2013b); Seatzu and Wardi (2013) considered minimizing a weighted sum of \( J(\theta) \) and the expected-value of the average workload at \( p_1 \), thereby balancing inventories with backlogging. Using some of the simulation results of those references, we address the problem of regulating \( L(\theta) \), and we make the point that our principal objective is to have the computed values of \( L(\theta) \) settle to within a band around the reference value. The regulation problem is addressed entirely in the space of realizations (sample paths) and not in the steady state.

In the considered example, \( V_3 = 6, V_{2,1} = 2.15, \) and \( V_{2,2} = 6 \); these numbers are taken from Wardi et al. (2013b); Seatzu and Wardi (2013). The product-orders process \( \{V_1(t)\}, \) defined by (14), consists of equally-spaced arrivals every 50 seconds (deterministic), and each arrival brings in an amount of fluid that is uniformly distributed in the \([30, 70]\)-range. The reference value to be tracked is \( J_{ref} = 758.70 \), and it is the computed value of \( J \) obtained for the aforementioned optimization problem in Seatzu and Wardi (2013). The IPA derivative \( L'(\theta) \) is computable via a recursive algorithm constructed according to the event-calculation framework defined in Cassandras et al. (2010); Wardi et al. (2013b); a detailed presentation thereof can be found in Seatzu and Wardi (2013).

We ran the regulation algorithm, consisting of Equations (2)-(4), for a predetermined number \( N \) of iterations. Each cycle corresponds to a \( \tau_f \)-seconds simulation which is used to compute \( L_n(\theta_{n-1}) \) and the IPA derivative \( L_n'(\theta_{n-1}) \), and the integrator’s gain is set to be \( A_n := (L_n'(\theta_{n-1}))^{-1} \). In our simulation runs we used the values \( N = 100 \), and \( \tau_f = 1,000 \).

Results of two typical runs are shown in Figure 5 for two values of the initial control parameter, \( \theta_1 = 35 \) and \( \theta_1 = 15 \). The respective graphs of \( y_n := L_n(\theta_n) \) are plotted by the dashed curve and solid curve, and both indicate convergence to a band around \( y_{ref} = 758.70 \) after 3 iterations. This band has a maximum range of 47.17, and the averages of the outputs \( y_n \), taken over \( n = 20, \ldots, 100 \), are 758.55 for the dashed plot, and 758.73 for the solid plots. Although these results indicate convergence of the outputs’ average to \( y_{ref} \), variations in the output values are discernable. These variations, as well as those in the IPA derivatives, yield fluctuations in the values of \( \theta_n, n = 1,2, \ldots, \) as can be seen in Figure 6. We believe that the major cause of these variations is in the variance of \( L(\theta) \) and it is not inherent in the regulation algorithm. To test this point, we ran 100 independent simulations of the system at the fixed value of \( \theta = 24.8 \), which is close to the average of \( \theta_n, n = 20, \ldots, 100 \) obtained by the regulation algorithm. The results, shown in Figure 7, indicate a persistent variation with a maximum range of 42, which is comparable to the range obtained from Figure 7.

To further test the convergence rate of the regulation algorithm we chose more extreme starting values of the control parameter, namely \( \theta_1 = 45 \) and \( \theta_1 = 5 \). The corresponding values of \( y_n \) are 635 and 883, which are more extreme than those obtained in Figure 7, but nonetheless the regulation algorithm converged to a similar band around \( y_{ref} \) in 3 iterations.

5. CONCLUSIONS

This paper presents a framework for performance regulation in a class of DEDS, which is based on an integral control with adaptive gain. The gain is computed by the IPA derivative of the plant function with respect to the control parameter. Whereas previous works applied this regulation scheme to controlling power and throughput in multi-core processors, this paper tests the framework on a queue and a Petri net. The underscoring idea is that the IPA derivative need not be computed exactly nor does it have to be unbiased for the regulation to work well, and its computation may incur fairly large relative errors. This
Fig. 7. Independent simulations of the Petri net may render our regulation technique applicable in real-time situations where it is beneficial to trade off a high degree of precision with small computing times.

A number of theoretical and practical problems present themselves for future research. On the theoretical side, the main question is how to obtain a-priori tight upper bounds on the relative error of the IPA derivative that guarantee effective tracking. On the practical side, we foresee the testing of the technique in a number of application domains including sensor networks and transportation systems.

REFERENCES


