CENTRALIZED AND DECENTRALIZED ASYNCHRONOUS OPTIMIZATION OF STOCHASTIC DISCRETE EVENT SYSTEMS

Felisa J. Vázquez-Abad *
Department of Computer Science and Operations Research
University of Montreal
Montréal, Quebec H3C 3J7

Christos G. Cassandras †
Department of Manufacturing Engineering
Boston University
Boston, MA 02215

Vibhor Julka
Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, MA 01003

October 20, 1998

ABSTRACT

We propose and analyze centralized and decentralized asynchronous control structures for the parametric optimization of stochastic Discrete Event Systems (DES) consisting of K distributed components. We use a stochastic approximation type of optimization scheme driven by gradient estimates of a global performance measure with respect to local control parameters. The estimates are obtained in distributed and asynchronous fashion at the K components based on local state information only. We identify two verifiable conditions for the estimators and show that if they, and some additional technical conditions, are satisfied, our centralized optimization schemes, as well as the fully decentralized asynchronous one we propose, all converge to a global optimum in a weak sense. All schemes have the additional property of using the entire state history, not just the part included in the interval since the last control update; thus, no system data are wasted. We include an application of our approach to a well-known stochastic scheduling problem and show explicit numerical results using some recently developed gradient estimators.

submitted to *IEEE Transactions on Automatic Control*, May 1995; revised June 1996, September 1997

*Supported in part by NSERC-Canada grants # WFA0139015 and FCAR-Québec grant # 93-ER-1654.
†Supported in part by NSF under Grants ECS-9311776 and EID-9212122, by the Air Force Rome Laboratory under Contracts F30602-94-C-0109 and F30602-97-C-0125, and by AFOSR under Contract F49620-95-1-0131.

1 Introduction

In this paper, we propose and analyze a class of centralized and decentralized asynchronous control and optimization schemes for stochastic Discrete Event Systems (DES), and include applications to a specific problem of interest. Our main objective is to develop a *decentralized* control structure and establish its convergence properties, our motivation being the following. It is often the case that a DES consists of a number of distributed components, with each component operating autonomously and contributing to the overall function of the system. Examples include the switches of a communication network, the processors in a distributed computer system, or workstations in a manufacturing system. While this decomposition conceptually provides opportunities for efficient control and optimization of the system, coordination and the transfer of information among components are costly and sometimes infeasible processes. It is, therefore, desirable to develop decentralized schemes which permit individual components to take control actions that contribute towards the optimization of a global performance criterion for the system. When a central controller structure is feasible, we also analyze some relevant optimization schemes and their convergence properties. The basic problem we consider is described next. Let u denote a real-valued controllable parameter vector and $J(u)$ a given performance measure (or cost) to be optimized. The DES under consideration consists of K components. Thus, the parameter vector is of the form $u = [u_1, \cdots, u_K]$, where u_i corresponds to the ith component, $i = 1, \cdots, K$, and may itself be a vector. Our objective is to determine a vector u^* that maximizes the performance criterion $J(u)$. When the DES operates in a stochastic environment, this criterion is usually of the form $J(u) = E[\mathcal{L}(u)]$, where $\mathcal{L}(u)$ is the cost obtained over a specific sample path. This problem is particularly hard due to the fact that closed-form expressions for $J(u)$ are seldom available. As a result, one must resort to various techniques for estimating $J(u)$ over all (or as many as possible) values of u in order to seek u^*. For control purposes, the most common approach for determining u^* is based on iterative schemes of the general form

$$u(n+1) = u(n) + \epsilon(n)Y(u(n)), \quad n = 0, 1, \cdots \tag{1}$$

where $Y(u(n)) = [Y_1(u(n)), \cdots, Y_K(u(n))]$ is usually an estimate of the negative of the gradient of $J(\cdot)$ with respect to $u(n)$. The factor $\epsilon(n)$ is referred to as the *step size* or *gain* or *learning rate* parameter. Such schemes are commonly referred to as Stochastic Approximation (SA) algorithms and they have been thoroughly studied in the literature (see [15, 16, 14, 21]). However, less attention has been paid to the use of SA algorithms for systems consisting of many components [19, 24, 23] in the context of several issues which are discussed below. In particular, there are two key issues that arise. First, there is the possibility of implementing a SA algorithm in a distributed fashion, i.e., by having components carry out separate computations using only local data. After these computations are performed, a second issue arises, i.e., whether the ensuing control actions are centralized or not. In the former case, a central controller collects the results of these computations and updates the control vector $u(n)$. Alternatively, each component may be able to take a control action, i.e., update $u(n)$ or part of it, using only the result of its local computation.

We examine next a number of issues that arise related to the general scheme (1) based on which we will be able to summarize the main contributions of this paper.

1. Gradient Estimation. We will limit ourselves to the case where $Y(u(n))$ is an estimate of the negative of the gradient of $J(u(n))$ with respect to $u(n)$. Thus, the first issue to consider is that of determining appropriate gradient estimates based on observable system data. Recent developments in the area of gradient estimation for DES include Perturbation Analysis (PA) (e.g., [13, 10]) and the Likelihood Ratio (LR) methodology (e.g., [20, 11]. Our analysis of the optimization schemes proposed in this paper depends on certain properties that the gradient estimates used must satisfy. As we will see, these properties are indeed satisfied by several types of PA estimators, including some recently developed in [5] and [2].

2. Convergence. Under a number of conditions on the set of admissible control parameter vectors $u(n)$, the step size sequence $\{\epsilon(n)\}$, and the estimates $Y(u(n))$, convergence w.p. 1 of the sequence $\{u(n)\}$ to a global optimum u^* can be established for the basic SA scheme (1). For the case of gradient estimators using Infinitesimal Perturbation Analysis (IPA), this has been shown in [6, 7, 8], applying the basic method in [15]. A weaker form of convergence can also be established, as in [17, 19] using the framework of [16]. However, when using (1) for decentralized optimization, the issue of convergence becomes significantly more complicated. We shall deal with it in the context of the convergence approach of [16] and [17].

3. Adaptivity. Convergence to a global optimum u^* is normally established for (1) by allowing the step size sequence $\{\epsilon(n)\}$ to go to zero over time. If, however, (1) is used on line as an adaptive control mechanism (as in [28]), then the scheme can obviously not respond to changes in the operating environment after the step size has reached zero. We are, therefore, often interested in the limiting behavior of SA schemes with some constant (normally small) step size (see [18]), which would permit the control vector to track various changes on line, usually at the expense of some oscillatory behavior around the value of a steady state performance measure. The framework we will use allows us to study this limiting behavior and apply the schemes we develop with a constant step size.

4. Distributed Estimation. In many DES, such as large communication networks, it is infeasible to transfer instantaneous state information from the ith system component to other components or to a central controller. Thus, it is highly desirable to develop *distributed* algorithms, whereby at least part of the necessary computation is carried out locally at each component. In the SA scheme (1), the main computational burden involves the gradient estimation process. One of our objectives, therefore, is to have each component locally evaluate an estimate of the derivative of $J(u)$ with respect to the local control parameter u_i.

5. Decentralized Control. Once the gradient estimates are evaluated, the simplest approach for executing an update in (1) is to have a central controller who collects all estimates and performs control updates. This approach, however, requires significant coordination among components, as well as the transfer of state information; this involves substantial communication overhead and delays which often render state information useless. More importantly, failure of a central controller implies failure of the entire system, which cannot sustain its proper operation without it. Therefore, a desirable alternative is to allow each individual component to separately update the global control vector $u(n)$ and transfer this information to all other components. Our analysis will cover both the centralized and decentralized control cases, but our primary goal is to study the latter.

6. Synchronization. In a fully synchronized scheme, there is an a priori mechanism based on which the updates of $u(n)$ take place. For instance, a central controller periodically requests estimates from all components in order to perform a control update. If the procedure is decentralized, however, a natural question is whether any component can be allowed to take a control action at any random point in time without any synchronizing mechanism. Such a feature is obviously highly desirable, since it requires virtually no coordination among components and it minimizes the amount of information that is transferred from one component to another.

7. Full utilization of system state history. A problem that frequently arises in SA schemes is that the estimator $Y(u(n))$ may not use all data collected over the history of the process. This typically arises in an asynchronous control update scheme, when a component being informed of a control update from another component may have to discard a partial local computation that it is in the process of performing. It is, therefore, desirable to develop a scheme using as much of the complete system history as possible and avoid having to re-initialize estimators, which essentially discards past history information.

We should point out that optimization algorithms for DES which use distributed computation have attracted a lot of attention, especially in the context of communication networks and computer systems. A number of such algorithms have been proposed and shown to converge to an optimal point under certain conditions (e.g., the distributed routing algorithm developed by Gallager [9] for minimizing the mean packet delay in data networks, and asynchronous versions of it [23]). These algorithms, however, are based on the assumption that the gradient of $J(u)$ is analytically available, so that no estimation is involved. The issue of convergence that arises when gradient *estimates* must replace their analytical counterparts in such distributed algorithms (e.g., see [4, 22]) is a much more challenging one.

In view of the issues identified above, the main contributions of this paper can be summarized as follows. First, we present and analyze centralized and fully decentralized optimization schemes based on distributed gradient estimation. For both types of schemes, we then establish convergence in the framework of [17], which involves fully developing a time scale argument for the decentralized control case. We also identify two verifiable conditions that the gradients estimators must satisfy in order for our convergence results to hold: *asymptotic unbiasedness* and *additivity*. The decentralized scheme presented has the added properties of being *fully asynchronous* and making use of *all* past state information. It is interesting, if not somewhat surprising, that such a scheme indeed converges, despite this very loose coordination among system components.

The paper is organized as follows. Section 2 provides a formulation of the stochastic optimization problem we consider and introduces some basic notation. In section 3 we present the mathematical framework for our analysis (section 3.1), based on which we describe the distributed gradient estimation process we shall use (section 3.2) and three separate control structures for solving the optimization problem (section 3.3). The first two are centralized but differ in the way the controller performs control updates; the first does so over a number of prespecified system events, while the second is an extension to random update times. The third scheme is a fully decentralized and asynchronous one. Section 4 is devoted to the detailed convergence analysis of these schemes. Throughout the paper, we use a well-known stochastic scheduling problem to illustrate our approach; in section 5 we present some representative numerical results from application of our estimation and optimization approach to this problem.

2 Basic Model and Optimization Problem Formulation

We consider a DES consisting of K components (e.g., nodes in a network or processors in a distributed computer system). Let $u = [u_1, \cdots, u_K]$ be a real-valued controllable parameter vector, where u_i represents the ith vector component. The optimization problem we are interested in is the determination of a vector u^* that maximizes a performance criterion $J(u) = E[\mathcal{L}(u)]$, where $\mathcal{L}(u)$ is the sample performance function. As already mentioned, we focus on problems where no analytical expression for $J(u)$ is available and we resort to an optimization scheme of the general form (1):

$$u(n+1) = u(n) + \epsilon(n) Y(u(n)), \quad n = 0, 1, \cdots$$

where $Y(u(n))$ is an estimate of the negative of the gradient of $J(u)$ with respect to $u(n)$. We assume that each system component only has access to local state information and can estimate a local performance criterion $J_i(u_i)$ and its gradient. For ease of notation, let us limit ourselves here to the case where u_i is a scalar. Given the structure of the system, the optimization problem we face is as follows:

$$\max_{u \in U} \sum_{i=1}^{K} \beta_i J_i(u_i) \quad s.t. \;\; g_1(u_1, \cdots, u_K) = c_1, \ldots, g_r(u_1, \cdots, u_K) = c_r$$

where β_i, $i = 1, \cdots, K$, are weights associated with the system components and $g_j(u_1, \cdots, u_K) = c_j$, $j = 1, \cdots, r$ are r *linear* constraints. Note that there may also be additional inequality constraints associated with the problem above; these can be taken into account by appropriately defining the admissible set U. Finally, let us assume that each system component has knowledge of all weights β_i and all linear constraints present. An unconstrained version of this optimization problem can be obtained as follows. By solving the r linear equations, it is generally possible to eliminate some of the K control parameters and solve for $q < K$ of them. Let C_q denote the reduced set of q system components. For any component $k \notin C_q$, we then have

$$u_k = a_k + \sum_{j \in C_q} b_{kj} u_j \tag{2}$$

for some constant coefficients a_k and b_{kj}, $j \in C_q$.

Remark. This formulation is typical of resource allocation problems often encountered in DES. For instance, consider a problem of allocating a single server (resource) over K queues in parallel over a discretized time line. Let u_i be the probability that the server is assigned to queue i at any time step. In this case, a single linear constraint of the form $u_1 + \cdots, +u_K = 1$ is used. The reduced set of components may be set to $\{1, \cdots, K-1\}$ with u_K given by $u_K = 1 - (u_1 + \cdots, u_{K-1})$.

With the above discussion in mind, we rewrite the problem in the following form, where the equality constraints have been eliminated:

$$\max_{u \in U} \left[\sum_{i \in C_q} \beta_i J_i(u_i) + \sum_{k \notin C_q} [\beta_k J_k(a_k + \sum_{j \in C_q} b_{kj} u_j)] \right] \tag{3}$$

Observing that the derivative of the objective function $J(u)$ with respect to u_i is given by

$$\frac{\partial J}{\partial u_i} = \beta_i \frac{dJ_i}{du_i} + \sum_{k \notin C_q} \beta_k b_{ki} \frac{dJ_k}{du_k}$$

and setting, for all $i \in C_q$,

$$\gamma_{ki} = \begin{cases} \beta_i & k = i \\ \beta_k b_{ki} & k \notin C_q \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

we have

$$\frac{\partial J}{\partial u_i} = \sum_{k=1}^{K} \gamma_{ki} \frac{dJ_k}{du_k} \tag{5}$$

Then, the optimization schemes we shall consider and analyze are of the general form

$$u_i(n+1) = u_i(n) + \epsilon(n) \sum_{k=1}^{K} \gamma_{ki} D_k(n) \qquad \text{for all } i \in C_q \tag{6}$$

where $D_k(n)$ is an estimate of the negative of the derivative dJ_k/du_k available at the end of the nth iteration. When new values $u_i(n+1)$ are evaluated for all $i \in C_q$, then new values $u_k(n+1)$ are also evaluated through (2) for all $k \notin C_q$. Note that when constraints are present, it is possible that (6) results in some $u_i(n) \notin U$. We will handle this issue by introducing an appropriate projection scheme as discussed in section 3.3.

Remark. Our analysis is not limited to the optimization problem setup above. For instance, the local performance criteria $J_i(\cdot)$ can be allowed to depend on the entire vector u rather than just u_i, in which case we can easily modify the definition of γ_{ki} and include derivative estimates $D_{ki}(n)$ of all $\partial J_k(u)/\partial u_i$. It is also possible to consider a non-additive performance measure structure. We choose to limit ourselves to the class of problems above, first to maintain manageable notation without affecting the key ideas of the analysis, and, second, because many of the applications we are familiar with fall into the class of problems above.

In the next section, we present two centralized and one decentralized stochastic optimization schemes based on (6). We develop an appropriate mathematical framework, describe the distributed derivative estimation process based on which all $D_k(n)$ above are evaluated, and finally describe the control structures we use in detail.

3 Stochastic Optimization Schemes and Control Structures

The basic SA scheme in (1) applied to DES with gradient estimates based on Infinitesimal Perturbation Analysis (IPA) has been studied in [6, 7]. In this case, convergence w.p. 1 has been shown under a number of technical conditions An alternative approach establishing a weaker form of convergence was recently presented in [17]. This method can be applied to time varying step sizes $\epsilon(n)$, $n = 1, 2, \ldots$, under the usual conditions on $\epsilon(n) \to 0$; however, it also allows us to study (1) under a constant step size $\epsilon > 0$. While the mathematical analysis is similar in both cases, the two approaches have different properties in applications, as explained next. If we are interested in simulation-based optimization, which is the case in many optimal design problems, then the use of time-varying gains under certain conditions yields w.p.1 convergence of the sequence $\{u(n)\}$ to the optimal point. This may be highly desirable as a numerical approximation scheme that ensures asymptotic optimality. On the other hand, in many on-line control problems the goal is to construct an iterative scheme capable of adjusting to the underlying (and generally unknown) dynamics of the system. In such problems, there may be no clear end to the optimization process as would be the case in a simulation. Instead, constant learning rates allow the schemes to *continuously revise the sensitivities estimated* and hence adjust towards the improvement of the objective function as external conditions change. If the latter remain constant, then the control process should, in some sense, approach the optimal value. As we will show, this approach is not limited to PA gradient estimators. Instead, it requires that gradient estimators satisfy certain properties (which we shall identify later) in order for the control process to converge. We will adopt this approach and will present an application based on a class of estimators that satisfy these properties.

In applying the weak convergence approach to (1), we begin with the basic recursive scheme

$$u^\epsilon(n+1) = u^\epsilon(n) + \epsilon Y(u^\epsilon(n)), \quad n = 0, 1, \cdots \tag{7}$$

where ϵ is initially fixed. Thus, this scheme gives rise to a sequence $\{u^\epsilon(n)\}$, parameterized by ϵ. Rather than attempting to prove that the sequence $\{u^\epsilon(n)\}$ converges to a fixed point u^* as $n \to \infty$, we concentrate on showing that the family of *functions* generated by the $u^\epsilon(n)$ recursion (7) approaches the tail of another function, $\tilde{\zeta}(t)$, as $\epsilon \to 0$ [16]. The property of $\tilde{\zeta}(t)$ is that it solves an ordinary differential equation (ODE):

$$\frac{d\zeta(t)}{dt} = -\frac{dJ}{d\zeta}(\zeta(t)) \tag{8}$$

where $J(\cdot)$ is the objective function of our original optimization problem. Intuitively, this ODE simply characterizes the "steepest descent" trajectory for a control $\zeta(t)$. When (and if) the right-hand-side above is zero, the solution of the ODE has in fact reached an asymptotic value, which also provides the optimal point u^* (under appropriate smoothness and convexity conditions; [26]). This approach permits us to study the convergence properties of a recursive scheme such as (7) by examining the asymptotic behavior of the solution to a "companion" ODE as in (8). Clearly, in the case of a stochastic recursion based on estimates of the derivative dJ/du requires a number of additional technical conditions and is highly dependent on the nature of these derivative estimates. These will be explicitly specified and discussed in the analysis that follows. However, the same basic idea is still applicable, i.e., determining a companion ODE to the recursive scheme used for solving the original optimization problem.

In this paper, our goal is to focus on decentralized asynchronous control update schemes, and to establish weak convergence of such schemes, building on the concepts that appeared in [24] and [17]. In section 4, this approach is used to prove convergence of a *decentralized asynchronous* scheme based on (6). First, however, in the remainder of this section we introduce the basic modeling framework for our analysis (section 3.1), followed by the distributed estimation process and its properties (section 3.2), and a presentation of three different control structures for optimization (section 3.3).

3.1 Modeling Framework

We will assume that the DES we consider is modeled as a stochastic timed automaton [3] or, equivalently, the process is a Generalized Semi-Markov Process (GSMP), a common framework encountered in the DES literature; for details, see [3, 10, 13]. Since our DES consists of K distributed components, the event set $\mathcal{E}$ is partitioned into $K+1$ subsets $\mathcal{E}^0, \mathcal{E}^1, \cdots, \mathcal{E}^K$ so that $\mathcal{E}^0$ contains all events (if any) directly observable by all system components, and $\mathcal{E}^k$ contains events which are directly observable by the kth component alone. Under a control parameter vector u, let $X_m(u)$ denote the state entered after the mth event occurrence; thus, we obtain a Markov chain (see also [12]) with transition probability $P_u(x, \cdot)$ defined on $\mathcal{B}$, the σ-algebra of subsets of the state space:

$$P_u(x, B) = P_u\{X_{m+1}(u) \in B | X_m(u) = x\}, \quad \forall B \in \mathcal{B}$$

We shall denote by E_u the expectation with respect to the measure P_u of the Markov chain $\{X_m(u)\}$. We assume that for each value of u, where u is defined over a set U, the invariant measure $\mu_u(\cdot)$ of the corresponding process exists and is unique. In this setting, we focus on the problem of finding a control value $u^* \in U$ that maximizes $J(u) = \int \mathcal{L}(x, u)\mu_u(dx)$. Within the framework of gradient based methods, we shall assume that $J(u)$ is differentiable and that all $\frac{\partial J(u)}{\partial u_k}, k = 1, \ldots, K$ are bounded and continuous.

Assumption 1: The transition probability $P_u(x, \cdot)$ is weakly continuous in (x, u) and the set of measures $\{\mu_u(\cdot); u \in A\}$ is tight for every compact set $A \subset U$.

Tightness is a concept analogous to compactness. In the case of a tight stochastic sequence, it implies that any subsequence has a further weakly convergent subsequence (see [17] for detailed definitions and discussion).

Example: To illustrate our modeling framework, consider an optimal scheduling problem where K nodes compete for a single server/resource. This is motivated by the well-known "transmission scheduling problem" arising in packet radio networks, where the resource is the communication channel, fixed length packets arrive at node k according to an arbitrary interarrival time distribution with rate λ_i, and a slotted time model is considered (with slot size equal to the packet transmission time δ). At each scheduling epoch, i.e., at the start of each time slot, the channel is assigned to a particular node (see Figure 1). The assignment is based on a *random polling* policy: the current time slot is allocated to the kth class with probability u_k.

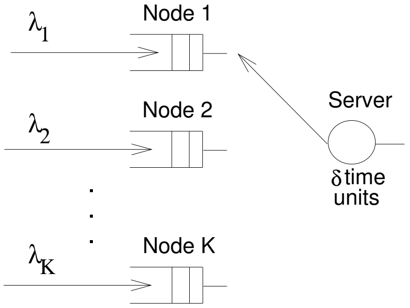

Figure 1: K-Node Polling System

The objective is to determine the optimal slot assignment probabilities so as to minimize the weighted average packet waiting time. The constrained optimization problem is then stated as:

$$\min_u \frac{1}{\sum_{j=1}^K \lambda_j} \sum_{k=1}^K \lambda_k J_k(u_k) \quad s.t. \quad \sum_{i=1}^K u_k \; = \; 1 \qquad \textbf{(P1)}$$

where $J_k(\cdot)$ is the average node k packet waiting time and $u = [u_1, \cdots, u_K]$ is assumed to be in the set of probability vectors such that $u_k > \lambda_k/\delta$, which ensures stability of each queue. This defines the set U over which the control vector u is defined. In the absence of any a priori information on the arrival processes, closed form expressions for this performance measure are unavailable. Thus, the gradual on-line adjustment of u is one attractive approach.

Let a_k denote a packet arrival event at node k and τ_k a packet transmission event when node k is assigned a time slot. The event set of this DES is then $\mathcal{E} = \{a_1, \cdots, a_K, \tau_1, \cdots, \tau_K\}$. Note that events a_k, τ_k are observed only by node k. The only way that a node $j \neq k$ can become aware of these events is if k explicitly transmits such information to j; this, however, not only entails communication overhead, but the information reaching j is also delayed. A natural partition of $\mathcal{E}$ consists of K sets, $\mathcal{E}^1, \cdots, \mathcal{E}^K$ with $\mathcal{E}^k = \{a_k, \tau_k\}$.

A simple way to implement a random polling policy characterized by the parameter vector u is to provide in advance each node with a common random number generator based on which all nodes can determine the node to which each time slot is assigned. This defines a transmission schedule common to all nodes. The state of the DES may be described by a vector $[x_1, \cdots, x_K, y]$, where $x_k \in \{0, 1, \cdots\}$ is the queue length at node k and $y \in \{0, 1, \cdots, K\}$ is the state of the channel: $y = k$ if node k is transmitting a packet and $y = 0$ if the channel is idle. Note that if a time slot is assigned to k, then only node k can determine the state of the channel, based on whether x_k is zero or positive; all other nodes have no direct access to this information. It should be clear why in this model centralized control is not only undesirable (because failure of a central control node results in failure of the network), but actually infeasible because state information cannot be instantaneously available to all nodes.

We now consider a more convenient state representation for our purposes, based on defining $x_k(p)$ to be the waiting time of the pth packet at node k. Each node behaves like a $GI/D/1$ queueing system with vacations (the time slots which are not assigned to k for transmission). Letting $\delta s_k(p)$ denote the time to serve the pth packet, then $\{s_k(p)\}$ is an i.i.d sequence of geometric random variables with parameter u_k. Letting also $\{\alpha_k(p)\}$ be the packet interarrival time sequence (which is independent of the vector u) and setting $\delta = 1$ for simplicity, $x_k(p)$ satisfies a standard Lindley recursion

$$x_k(p+1) = \max\{0, x_k(p) + s_k(p) - \alpha_k(p+1)\} \tag{9}$$

where $s_k(p)$ is associated with the occurrence of a transmission event τ_k and $\alpha_k(p+1)$ is associated with an arrival event a_k.

Let us now consider Assumption 1 for this model. The local processes behave as stable queues for each $u \in U$, ensuring existence of the invariant measure. For any k, it is clear from (9) that the process $\{x_k(j)\}$ has the Markovian property, and the transition probabilities are polynomial functions of u_k, verifying the weak continuity in (x, u). The remainder of Assumption 1 follows readily. Indeed, for any compact set $A \subset U$, the process is stochastically dominated by a process $\{\bar{x}_k(p)\}$ describing a queue where the services have geometric distribution with parameter $\bar{u}_k = \min\{u_k : u \in A\}$. The dominating process is constructed using common random variables for the generation of service times and the same sequence of interarrival times as the original process for any $u \in A$. Since the process $\{\bar{x}_k(p)\}$ is stable, it regenerates with a.s. finite regeneration cycle lengths, and it dominates all processes $\{x_k(p), u \in A\}$, yielding $P_u\{x_k(p+1) > C | x_k(0) = x\} \leq P_{\bar{u}_k}\{x_k(p+1) > C | x_k(0) = x\}$ for any real number $C < \infty$; this implies tightness of the invariant measures for all $u \in A$ at each local queue, using dominated convergence. Finally, the global process $\{X_m\}$ is a vector containing the values of the waiting times at each local node when the mth global event occurs. Therefore, $\{\mu_u(dx)\}_{u \in A}$ exist and are tight.

$\triangle$

In what follows, we shall use the notation $\mathbf{u}_m^\epsilon$ to denote the vector-valued control parameter in effect at the epoch of event m and ϵ is the value of the step size parameter in (7). Much of our subsequent analysis is based on carefully distinguishing between various natural time scales involved in the controlled process. Let us introduce the two main time scales we shall use and associated notation. We have a "fast" global time scale, defined by all events that drive the DES, and a "slow" time scale, defined by instants in time when control updates are performed according to (7). We will use the following notation to distinguish these two time scales:

$$
\begin{aligned}
m &= \text{global event index} \\
n &= \text{iteration index over global control updates}
\end{aligned}
$$

Thus, $\mathbf{u}_m^\epsilon$ is updated at selected events (at which time the index n is incremented) depending on the control structure selected, as described in section 3.3. Note that when the control is *centralized*, the controller gathers information from the various system components over an *update interval* and performs the global control update at the end of this interval. In this case, we also have an "intermediate" time scale defined at each system component $k = 1, \ldots, K$. In particular, the kth component collects local information over some *estimation interval*, at the end of which a derivative estimate is obtained and sent to the controller, subsequently to be used for global control updates. By indexing over these estimation intervals we define such an intermediate time scale. On the other hand, in the fully *decentralized* scheme, the global control updates are performed asynchronously by individual components and they coincide with the instants when any component completes a local estimation interval. Thus, the intermediate and slow time scales shall coincide.

3.2 Distributed Estimation

3.2.1 Local Derivative Estimators and their Properties

In both the centralized and decentralized cases which we analyze, the derivative estimators required are obtained in *distributed* fashion, i.e., each system component separately performs all estimation computation required, using only locally available state information. We emphasize that the issue of distributed estimation is distinct from that of control implementation, which can be centralized or decentralized.

We shall now present the construction of the local estimators for the fixed control process $\{X_m(u)\}$. In section 4, we shall explain how these estimators are used in the time-varying control parameter case. Let $j = 1, 2, \ldots$ index the sequence of *local* events at component k, i.e., all events in the set $\mathcal{E}^k$. Let $m_k(j)$ be the corresponding *global* event index (when no ambiguity arises, we will also use $m(j)$). We define $\rho_k(u)$ to be the invariant average rate of events at k. By the ergodicity assumption:

$$\rho_k(u) = \lim_{j \to \infty} \frac{j}{m_k(j)} \tag{10}$$

Assumption 2: For all $k = 1, \ldots, K$, $\rho_k(u)$ is continuous, and $\sup_{u \in U} \rho_k(u) < 1$, $\inf_{u \in U} \rho_k(u) > 0$.

The last part of this assumption ensures that the control variables do not change the topology of the system by "shutting off" a particular component. It is not essential, but it simplifies the notation, since if a particular component can be shut off (for example in a network where u represents a routing probability vector) for a value $u_0 \in U$, then the appropriate adjustments in the update equations to estimate sensitivities at u_0 have to be incorporated.

Example (continued): For the problem illustrated in Figure 1, it is easy to verify that Assumption 2 is satisfied. Indeed, under stability, for any $u \in U$ the rate of transmission events τ_k at each node k equals the arrival rate λ_k in steady state. Therefore, the invariant event rate $\rho_k(u)$ is independent of u for this example.

$\triangle$

A derivative estimator of the objective function $J(u)$ with respect to the control parameter u_k at component k is calculated from local observations of the state values over a period of time. Let j be some local event index and Δ a number of local events. Then, we define $d_k(j,\Delta)/\Delta$ to be the estimator of dJ_k/du_k obtained for the fixed-u process over the Δ local events $\{j,\ldots,j+\Delta-1\}$. The following assumption contains two key properties we require for our derivative estimators, based on which our convergence results will hold.

Assumption 3: All $d_k(\cdot,\cdot)$, $k=1,\cdots,K$, satisfy the following:
(i) Asymptotic Unbiasedness: For any random integer Δ_i such that $\Delta_i \to \infty$ as $i \to \infty$ w.p.1, and any x:

$$\lim_{i\to\infty} E_u \left\{ \frac{d_k(0,\Delta_i)}{\Delta_i} | X_{m_k(j)} = x \right\} = \frac{dJ_k(u_k)}{du_k} \quad \text{w.p.1} \tag{11}$$

(ii) Additivity: For any sequence of positive integers $\{\Delta_i\}$, and a sequence $\{L_l\}$ defined by $L_0 = 0$, $L_l = \sum_{i=1}^{l} \Delta_i$, we have:

$$d_k(0,L_l) = \sum_{i=0}^{l-1} d_k(L_i,\Delta_{i+1}) \tag{12}$$

Remark: The local estimation process is carried out over succesive subintervals $[0,L_1),[L_1,L_2),\ldots,$ $[L_{l-1},L_l)$, and $\Delta_i = L_i - L_{i-1}$ is the number of events sampled to obtain the ith estimate. Additivity ensures that the estimator obtained over the interval $[0,L_l) = \{0,\ldots,L_l-1\}$ is equivalent to adding the partial computations obtained over these subintervals. This is a particularly attractive property, since the estimators accumulate all past data and never waste any previously collected information. Finally, note that the two conditions in Assumption 3 are generally easy to verify.

Example (continued): For the problem illustrated in Figure 1, recall that each node is viewed as a $GI/D/1$ queueing system with vacations. One can then obtain estimators of $dJ_i(u_i)/du_i$ through Perturbation Analysis (PA) as shown in [5]. Under certain conditions, such estimators are consistent a.s. (e.g., see [10]); our analysis, however, will require condition (11). For such estimators to satisfy (12), we require that they are not reset to zero from one estimation interval to the next, but rather we keep on evaluating all cumulative effects. Many estimators based on PA (e.g., see [13, 5]), satisfy the additivity property. For this particular example, $d_k(j,\Delta)/\Delta$ is an estimator of the form of a sample average over local events (see [5] for details):

$$d_k(j,\Delta) = \sum_{i=j}^{j+\Delta} f_k(X_{m(i)},u) \tag{13}$$

where $X_{m(i)}$ is the state after the $m(i)$th global event and the function f_k depends on this state through all of its entries associated to the kth component. In particular, f_k is a function of the packet waiting times at node k which satisfy (9); we shall provide a more detailed form of this estimator in the next section. Note that (11) and (12) follow from construction (see also [5], [25] for details).

$\triangle$

3.2.2 Derivative Estimation Sequences

In this section, we present the most general framework for the distributed estimation we shall use, incorporating both asynchronous and parallel computation features. From Assumption 3, the local processors can evaluate their estimators by dividing the computation into *estimation intervals*. We shall now construct these intervals by choosing an appropriate increasing sequence $L_k(l)$, $l = 1, 2, \ldots$, of random stopping times with independently distributed increments:

$$\Delta_k(l+1) = L_k(l+1) - L_k(l)$$

for each component k. Thus, the l-th estimation interval at component k contains all local events $j \in \{L_k(l), \ldots, L_k(l+1)-1\}$. The resulting lth estimator at component k is

$$\tilde{d}_k(l) \equiv d_k(L_k(l), \Delta_k(l+1)) \tag{14}$$

In other words, we view the time line associated with component k as being partitioned into intervals defined by the sequence $\{L_k(l)\}$, the lth interval containing $\Delta_k(l+1)$ local events. Hence, a sequence of estimates $\{\tilde{d}_k(l)\}$ is defined.

Define $G_k(l)$ to be the *"G"lobal* event index corresponding to the *"L"ocal* event $L_k(l)$, that is, $G_k(l) = m(L_k(l))$. Figure 2 illustrates the relationship between the local and global indices. We shall now present conditions for establishing what we will refer to as the *renewal structure* of the sequence $\{G_k(l)\}$.

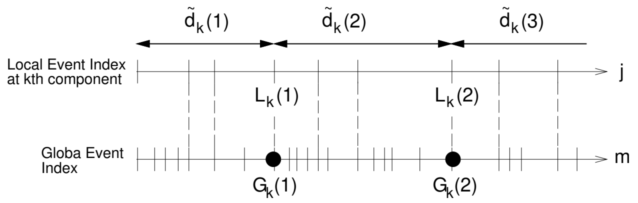

Figure 2: Correspondence between Local and Global Event Indices

It is customary to assume that the estimation intervals grow with l. However, in this work we shall show that this is an unnecessary condition. Instead, since the estimators satisfy Assumption 3, we shall focus on the simpler case where $P_u\{\Delta_k(l+1) = m | X_{G_k(l)}(u) = x\}$ is a continuous function of (x, u) independent of l and $L_k(l)$; in addition, $P\{\Delta_k(l) = 0\} = 0$. We shall also choose the increments so that for any compact $A \subset U$ and for any x, there is a $K(A) < \infty$ such that $\sup_{u \in A} E_u\{\Delta_k(l+1) | X_{G_k(l)}(u) = x\} < K(A)$. This condition will be implied by a stronger uniform tightness requirement made in Assumption 6 later on. In particular, this implies that $P_u\{\Delta_k(l+1) < \infty | X_{G_k(l)}(u) = x\} = 1$ for every x, l. From our continuity assumptions, it follows that:

$$\bar{M}_k(u) \equiv E^u[G_k(1)] = \lim_{l \to \infty} \frac{G_k(l)}{l} \tag{15}$$

where E^u denotes the expectation under the invariant measure of the fixed-u process, is continuous in u and uniformly bounded: $\sup_u \bar{M}_k(u) < \infty$, $\inf_u \bar{M}_k(u) > 0$.

Example (continued): Returning to the system of Figure 1, let us consider two possible ways of defining estimation intervals at any node k: (a) a fixed number N of local events, in which case $\bar{M}_k(u) = N/\rho_k(u)$, and (b) a fixed number N of local busy periods, in which case let $N_k(b)$ denote the number of events contained in the bth busy period at k and we get $\bar{M}_k(u) = NE^u[N_k(1)]/\rho_k(u)$.

For case (b) above, let us set $N = 1$ for notational simplicity and consider an estimator $\tilde{d}_k(l)$ as defined in (14). Based on the PA estimator in (13), $\tilde{d}_k(l)$ has the following general form:

$$\tilde{d}_k(l) = \frac{1}{u_k} \sum_{j \in A_l} f_k(X_{m(j)}, u_k) \tag{16}$$

where A_l is a particular set of local event indices and $f_k(X_{m(j)}, u_k)$ is a function of the of the state when the jth local event occurs (equivalently: the $m(j)$th global event occurs) and of the control parameter u_k. We will not discuss here the precise nature of A_l or of f_k (which may be found in [5, 25]), but only point out that f_k depends on the state through those entries associated with the kth component, as already mentioned in section 3.2.1. As in most pathwise derivative estimators, the terms $f_k(X_{m(j)}, u_k)$ represent individual "perturbations" which are easily evaluated functions of the system state.

$\triangle$

Before proceeding, it is important to note that even if the control values change within an estimation interval, the local estimators use the same functional form and continue the computation of the current estimate. We shall return to this point in section 4.

3.3 Control Structures

We shall deal with three control update structures: a central controller that performs updates every M global event epochs; a central controller that updates according to a random sequence of update epochs; and a fully decentralized control scheme where each component imposes a *global* control update on the entire system at the end of its individual local estimation intervals. Before proceeding, let us summarize the event indexing notation which is going to be used throughout this section:

$$\begin{aligned} L_k(l) &= \text{local event index at } k \text{ indicating the end of the } l\text{th estimation interval} \\ G_k(l) &= \text{global event index corresponding to } L_k(l)\text{, i.e., } G_k(l) = m(L_k(l)) \\ G(n) &= \text{global event index indicating the } n\text{th global control update} \end{aligned}$$

3.3.1 Central Controller with Fixed Update Events

In this centralized control setting, we assume that there exists a single controller (e.g., a preselected system component among $k = 1, \ldots, K$ acting as the global controller) who updates all control parameters u_k, $k = 1, \ldots, K$ at *update epochs* corresponding to global events $m = nM, n = 0, 1, \ldots$ Here, M is a deterministic integer. It represents the number of global events defining the control update interval. The value of the control parameter vector $\mathbf{u}_m^\epsilon$ is kept constant over the *update interval* defined by events $m \in \{nM, \ldots, (n+1)M - 1\}$; $\mathbf{u}_m^\epsilon$ changes only at event epochs $nM, n = 1, 2, \ldots$ according to a SA scheme that we now proceed to construct.

The control parameters are updated by making use of the local estimates $\tilde{d}_k(l)$ reported to the controller at time instants corresponding to the global events $m = G_k(l) - 1$, $l = 1, 2, \ldots$, $k = 1, \ldots, K$. Since, however, the control updates are carried out at $m = M, 2M, \ldots$ (where $nM \neq G_k(l)$ for some n, l in general), the actual estimates used at these time instants are denoted by $D_k(n, M)$. Thus, our first task is to consider an update event nM and specify how $D_k(n, M)$ is to be constructed from the local estimators $\tilde{d}_k(l)$ received by the controller prior to global event nM. For any component k which computes local derivative estimates, let

$$l_k(n) = \max\{l : G_k(l) \leq nM\}$$

i.e., $l_k(n)$ is the total number of estimates reported by k to the controller prior to the nth control update (performed when global event nM occurs). Thus, the estimates available to the controller from k over the nth update interval are given by $\tilde{d}_k(l)$, $l = l_k(n), \ldots, l_k(n+1) - 1$. The derivative estimator $D_k(n, M)$ used by the central controller is now defined as:

$$D_k(n, M) = \frac{1}{\rho_k(\mathbf{u}_{nM}^\epsilon) M} \sum_{l=l_k(n)}^{l_k(n+1)-1} \tilde{d}_k(l) \tag{17}$$

We can now specify the centralized control scheme based on the distributed estimation process described above. First, note that the estimate information is collected by the controller in *asynchronous* fashion, i.e., each component reports an estimate based on its own local event time scale. Thus, in (17), the estimates evaluated may contain partial computations using data prior to the current update interval. This provides some flexibility to the scheme and will turn out to be crucial in the decentralized control case to be discussed later. Next, recall from the discussion in section 2, that all components k obtain local derivative estimates, but only components $k \in C_q$ need to perform control updates dependent on these estimates; for all $k \notin C_q$, control updates are simply obtained through (2). The updates $u_k^\epsilon(n+1)$, for all $k \in C_q$, are made at the epochs of events $m = M, 2M, \ldots$ according to:

$$u_k^\epsilon(n+1) = u_k^\epsilon(n) + \epsilon Y_k^\epsilon(u^\epsilon(n)), \qquad k \in C_q \tag{18}$$

with

$$Y_k^\epsilon(u^\epsilon(n)) = -\sum_{j=1}^{K} \gamma_{jk} D_j(n, M) \tag{19}$$

where the coefficients γ_{ki} were defined in (4). The actual control is set to $\mathbf{u}_m^\epsilon = u^\epsilon(n)$ for all $m = nM, \ldots, (n+1)M-1$ (the case where the control is subject to constraints is discussed below).

Based on the above, we can summarize the centralized control scheme as follows:

Centralized Control Structure with Distributed Asynchronous Estimation:

- **Each Component** k: Evaluates an estimator $\tilde{d}_k(l)$ and reports it to the **Central Controller** at the epochs of its *local* events $L_k(l)$, $l = 1, 2, \ldots$
- **Central Controller:** At epochs of *global* events nM, $n = 1, 2, \ldots$:
  1. Evaluates the derivative estimator $D_k(n, M)$ through (17).
  2. Updates the control parameters by evaluating $u_k^\epsilon(n+1)$ for all $k \in C_q$ through (18)-(19); and for all $k \notin C_q$ through (2).

The distributed asynchronous nature of the derivative estimation process is evident from this description. The convergence properties of this scheme will be presented in section 4.5.

Recall that our actual optimization problem presented in section 2 allows the control vector u to be constrained in some set U, which we shall assume to be compact. Clearly, in the control scheme above it is possible that some $\mathbf{u}_m^\epsilon$ lies outside U, in which case we proceed as follows. Let $\Pi_U(\mathbf{u}_m^\epsilon)$ be the usual Cartesian projection of $\mathbf{u}_m^\epsilon$. Then, (a) Use $\Pi_U(\mathbf{u}_m^\epsilon)$ as the *actual* control value applied to the system based on which all dynamics are updated, but (b) Maintain $\mathbf{u}_m^\epsilon$ and use it (not $\Pi_U(\mathbf{u}_m^\epsilon)$) to evaluate the next control value through (18). Since, for any u, $\Pi_U(u)$ is unique, we may subsequently interpret the notation E_u, P_u, μ_u as the corresponding quantities for the fixed-u process at $\Pi_U(u)$ without any confusion.

Remark: Under the additivity assumption (12), it is also possible to implement another centralized scheme in which the central controller "requests" information from all components at its update epochs $m = M, 2M, \ldots$ They would then send all the derivative estimate information up to that time, and would keep a register to continue computing their local estimators for the next update interval. This, however, requires additional communication overhead in the form of the controller issuing requests to all components.

3.3.2 Central Controller with Random Update Events

In our distributed estimation scheme, we assumed that the system components estimate local derivatives using only local information. We used random stopping event indices based on the local event numbers to cover the general case where global event numbers may be unknown locally at the different components.

This may also be true for the central controller. We shall deal here with the extension of the previous scheme to random update event indices at the controller. Three important applications of this arise when the component acting as the central controller updates (a) at regular intervals containing a fixed number of its own local events, (b) in the case of queueing systems, at the end of busy periods, and (c) at regular intervals defined by time (say, every T seconds) instead of events, in which case we can introduce an event to occur at all time instants nT, $n = 1, 2, \ldots$ All of these schemes yield random update intervals determined by a sequence of stopping times, measurable with respect to the local history of the process.

The local estimates $\tilde{d}_k(l)$ are once again broadcast to the central controller at the global event epochs $G_k(l), l = 1, 2, \ldots$. Following a similar notation as in the previous subsection, call $G(n)$ the increasing sequence of stopping times that define the nth *update interval* at the central controller from event index $m = G(n)$ to $m = G(n+1)$, and let $M(n)$ denote the number of events included in this update interval. We choose this sequence so that it satisfies the renewal structure, and $0 < M(n) < \infty$ a.s., $\bar{M}(u) = E^u\{M(1)\}$ is continuous, $0 < \inf_u \bar{M}(u)$ and $\sup \bar{M}(u) < \infty$.

As before, $\mathbf{u}_m^\epsilon$ is kept constant over all events $m \in \{G(n), \ldots, G(n+1) - 1\}$. Let:

$$l_k(n) = \max\{l\colon G_k(l) \leq G(n)\}$$

which has the exact same meaning as in the last subsection. The estimator $D_k(n)$ used at the central controller is defined as:

$$D_k(n) = \frac{1}{\rho_k(\mathbf{u}_{G(n)}^\epsilon)} \sum_{l=l_k(n)}^{l_k(n+1)-1} \tilde{d}_k(l) \tag{20}$$

The control update equations for $u^\epsilon(n) = \mathbf{u}_{G(n)}^\epsilon$ are given by

$$u_k^\epsilon(n+1) = u_k^\epsilon(n) + \epsilon Y_k^\epsilon(u^\epsilon(n)), \qquad k \in C_q \tag{21}$$

with

$$Y_k^\epsilon(u^\epsilon(n)) = -\sum_{j=1}^{K} \gamma_{jk} D_j(n) \tag{22}$$

Clearly, when $M(n) = \bar{M}(u) \equiv M$ are deterministic and fixed, the two schemes (17) and (20) differ because in the latter, we are not dividing by $\bar{M}(u)$, and this affects the values of $Y_k^\epsilon(u^\epsilon(n))$. Consequently, the actual derivative estimator is obtained by scaling $D_k(n)$ above by $\bar{M}(u)$.

Note that in the case of control constraints, we can proceed exactly as in the previous section, i.e., by introducing the projection $\Pi_U(\mathbf{u}_m^\epsilon)$ and treating it as the actual control, while using $\mathbf{u}_m^\epsilon$ to evaluate the next control value through (21).

3.3.3 Decentralized Asynchronous Control

The decentralized structure we will use is described as follows. Each system component k of the DES becomes a *global* controller and can change the value of the local variable u_k, as well as the values of all u_i $i \neq k$ (as long as the constraints in (2) hold at all times.) In particular, at the end of the lth estimation interval, when the global event index is $G_k(l) = m + 1$, k becomes a controller and changes the value of the ith component of the vector $\mathbf{u}_m^\epsilon$ by adding to it an amount dependent on $\tilde{d}_k(l)$ as described below. In mathematical terms, if $\mathbf{u}_{m,i}^\epsilon$ denotes the ith component of the control vector at event epoch m, then

$$\mathbf{u}_{m+1,i}^\epsilon = \mathbf{u}_{m,i}^\epsilon + \epsilon Y_{m,i}^\epsilon(\mathbf{u}_m^\epsilon), \qquad i \in C_q \tag{23}$$

where now:

$$Y_{m,i}^{\epsilon}(\mathbf{u}_m^{\epsilon}) = -\sum_{k=1}^{K} \frac{\gamma_{ki}}{\rho_k(\mathbf{u}_m^{\epsilon})} \sum_{l=1}^{\infty} \tilde{d}_k(l) \mathbf{1}_{\{G_k(l)=m+1\}} \tag{24}$$

Due to our assumptions on $G_k(l)$, namely that $1 \leq \Delta_k(l)$ a.s, it follows that, for any fixed component k, for every m at most one value of l is such that $G_k(l) = m+1$, that is, $\sum_{l=1}^{\infty} \mathbf{1}_{\{G_k(l)=m+1\}} = 0$ or 1 a.s. Thus, whenever the $(m+1)$th global event coincides with the end of an update interval (say, l) at some component (say, k), the expression above yields $\gamma_{ki}\tilde{d}_k(l)/\rho_k(\mathbf{u}_m^{\epsilon})$. This is the amount by which the ith control parameter is changed at that time. Notice that in this scheme two or more controllers may simultaneously update the same components of the control vector.

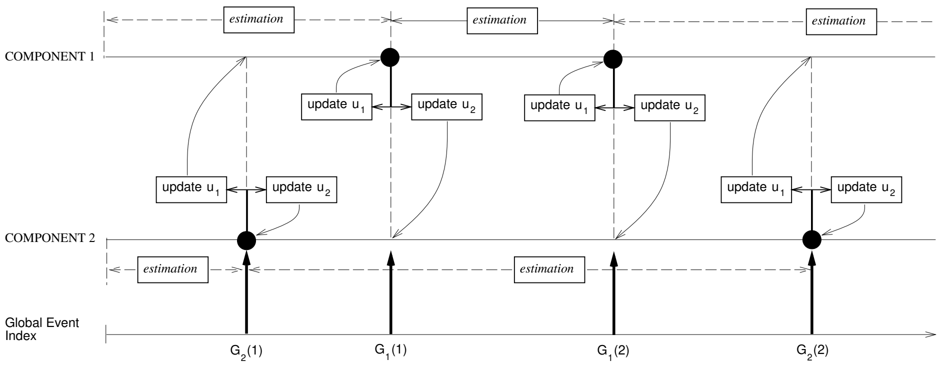

Figure 3: Illustrating the Decentralized Asynchronous Control Structure

This asynchronous decentralized control scheme can be summarized as follows:

Decentralized Asynchronous Control Structure:

- **Each Component** k:
  1. Evaluates a *local* estimator $\tilde{d}_k(l)$ over an interval of local events $j \in \{L_k(l), \ldots, L_k(l+1)-1\}$, $l = 0, 1, \ldots$
  2. At epochs of *local* events $L_k(l)$ [equivalently, global events $m = G_k(l)$], $l = 1, 2, \ldots$, updates all control parameters by evaluating $\mathbf{u}_{m+1,i}^{\epsilon}$ for all $i \in C_q$ through (23)-(24); and for all $i \notin C_q$ through (2).
  3. Sends the complete updated control vector $\mathbf{u}_{m+1}^{\epsilon}$ to all other system components.

To illustrate the operation of this structure, consider an example with two system components as shown in Figure 3. First, component 2 completes an estimation interval at global event $G_2(1)$. It updates both u_1 and u_2, based only on its local derivative estimate $\tilde{d}_2(1)$, and sends the information to component 1. Component 1 immediately adopts the new control value u_1, but continues its own estimation process without any other action. The next component that happens to complete an estimation interval after $G_2(1)$ happens to be component 1. This corresponds to global event $G_1(1)$. Component 1 evaluates $\tilde{d}_1(1)$, updates u_1 and u_2 and sends the new control values to component 2. It so happens that the next component completing an estimation interval is component 1 again (at global event $G_1(2)$), so this process repeats. Note, however, that component 2, while changing its control value u_2 twice, it goes on with its local estimation process accumulating state information without any estimator resetting action.

In order to keep the notation and subsequent analysis closer to the one introduced in previous sections, we will make use of some auxiliary (artificial) variables, denoted by $v_{ki}^{\epsilon}(n)$, with the following interpretation:

$v_{ki}^{\epsilon}(n)$ is the cumulative control change imposed by component k on component i by the instant when k becomes a global controller for the nth time. In other words, at the epoch of event $G_k(n)$, the current value of u_i has experienced a total change from the action of controller k given by $v_{ki}^{\epsilon}(n)$. The dynamics of these auxiliary variables are specified as follows. Let $v_{ki}^{\epsilon}(0)$ be such that $\sum_k v_{ki}^{\epsilon}(0) = u_i^{\epsilon}(0) = u_i$. Then define:

$$v_{ki}^{\epsilon}(n+1) = v_{ki}^{\epsilon}(n) + \epsilon Y_{ki}^{\epsilon}(n) \tag{25}$$

where

$$Y_{ki}^{\epsilon}(n) = -\frac{\gamma_{ki}}{\rho_k(\mathbf{u}_{G_k(n+1)-1}^{\epsilon})}\tilde{d}_k(n) \tag{26}$$

It should be clear that $Y_{ki}^{\epsilon}(n)$ is the amount by which component k imposes a change on the control parameter value at i at the $(n+1)$th time that k becomes a global controller (i.e., at global event index $G_k(n+1) = m(L_k(n+1))$).

As in the previous sections, in the case of control constraints of the form $u \in U$, we proceed by using the projection $\Pi_U(\mathbf{u}_m^{\epsilon})$ as the actual control, while using $\mathbf{u}_m^{\epsilon}$ to evaluate the next control value through (23). In order to make use of (25)-(26), we will update the auxiliary variables, which uniquely define $\mathbf{u}_m^{\epsilon}$ and hence the projection $\Pi_U(\mathbf{u}_m^{\epsilon})$.

Remark: At the beginning of section 3 we motivated the schemes through different time scales. However, there are no assumptions on the values of $\bar{M}_k(u)$ or $\bar{M}(u)$ that "force" the time scales to have any relationship at all. From a practical point of view, it is more reasonable to associate the time scales with a faster, an intermediate, and a slower one. The limiting behavior of the procedure, however, is not affected by the relationships between the different time scales $\bar{M}_k(u)$ and $\bar{M}(u)$, as long as the renewal structure is satisfied. In particular, suppose that for the central controller we choose $G_k(l) = G(l)$. This means that all components are synchronized and send the information when the central controller requests it. Then the intermediate and the slower time scale blend into one. On the other extreme, we may have $\Delta_k(l) = 1$, which for the decentralized structure (which actually becomes equivalent to a central one) imposes updates at every single event epoch. This case is covered by our model, and the intermediate and slower time scales coincide with the faster one. Notice that we do not consider the case where the estimation intervals tend to infinity, an approach commonly used in simulation optimization practice. We will show in section 4.5 that this is not necessary under the conditions in Assumption 3.

4 Weak Convergence Analysis

In this section, we address the issue of convergence of the three SA control schemes presented in section 3.3, (18), (21) and (23). To do so, we have to carefully consider how varying the control parameter vector u after selected events affects the underlying system state and hence the derivative estimates which drive these SA schemes. The first step in this process is to enlarge the original state vector X_m by defining an appropriate "augmented" state, denoted by ξ_m^{ϵ}. The choice of ξ_m^{ϵ} must be such that the resulting process $(\xi_m^{\epsilon}, \mathbf{u}_m^{\epsilon})$ is a MDP. While this may not always be possible for arbitrary systems and derivative estimators, the structure of our particular DES derivative estimators, as specified by Assumption 3 and illustrated in the example presented in section 3.2.1, allows us to accomplish this goal with relatively little effort.

Example (continued): We return to the system of Figure 1 discussed in the last section. We shall illustrate the enlargement of the state for the (harder) case of decentralized operation, where we choose estimation intervals at each node k defined by a fixed number N of busy periods at k. Thus, recall from section 3.3.3 that in this case the entire control vector is updated by any node k which completes an estimation interval of its own. For notational simplicity, let us set $N = 1$ and use the derivative estimator $\tilde{d}_k(l)$ presented in (16). Let $b_k(m)$ denote the index of the busy period at node k which includes the mth global event. The *partial computation* of the estimator above after the mth global event is denoted by $\psi_k(m)$ and is defined recursively as:

$$\psi_k(m+1) = \begin{cases} \psi_k(m) + \left[\sum_{j \in A_l} \phi_k^j(X_{m+1}, u_k)\right] & \text{if } b_k(m+1) = b_k(m) \\ 0 & \text{otherwise} \end{cases}$$

where $\phi_k^j(X_{m+1}, u_k)$ is a function of the state at the $(m+1)$th global event and of the control parameter u_k, and note that it depends on the local event index j (detailed expressions for this function may be found in [5, 25]). It can then be verified that if the lth busy period at k contains M_l local events, then $\psi_k(M_l) = \tilde{d}_k(l)$, i.e., the partial computation after the completion of a busy period recovers the derivative estimator (16) needed at that point to perform a control update.

It should now be clear that the enlarged state for this example is a vector ξ_m consisting of the original system state vector X_m and the K partial computations $\psi_k(m)$, $k = 1, \ldots, K$ (in some cases, it is possible that an additional auxiliary state variable has to be introduced; see, for example [25]). In general, the partial computations are a natural way to carry past information required in the derivative estimators of the particular form encountered in perturbation analysis and similar gradient estimation methods. The process $\{\psi_k(m)\}$ can be interpreted as the pathwise derivative process, which is adapted to the history of the underlying process itself. In this example, where the updates are performed as soon as a busy period is completed, the enlarged state ξ_m contains all the required information for the epoch and the amount of the updating.

Note that if we wish to keep the control vector fixed, then the corresponding enlarged process (system and derivative estimation process) $\{\xi_m(u)\}$ possesses the Markov property. Furthermore, using the domination argument invoked in section 3.1, for any compact set $A \subset U$, it follows that $\{\xi_m(u)\}$ is tight, the invariant measure exists and its marginal coincides with the invariant measure of the original process.

$\triangle$

In the next few sections, we proceed with the basic preliminary results used in the weak convergence method similar to [17]. In some texts, weak convergence (or convergence in distribution) is denoted by the symbols $\xrightarrow{\mathcal{D}}$, or $\Rightarrow$. We shall use the latter. Our goal is to use the modeling framework developed in the last section and show how the framework of [17] can be applied to the schemes (18), (21) and (23). Note that the corresponding fixed-u process $\{\xi_m(u)\}$ has weakly continuous transitions probabilities $P_u(\xi, B) = P\{\xi_{m+1}(u) \in B | \xi_m(u) = \xi\}$ and a unique, ergodic invariant measure μ_u satisfying the weak continuity and boundedness conditions of Assumption 1. Based on the discussion above, we assume that the process $(\xi_m^\epsilon, \mathbf{u}_m^\epsilon)$ is a MDP, satisfying:

$$P\{\xi_{m+1}^\epsilon \in B_1, \mathbf{u}_{m+1}^\epsilon \in B_2 | \xi_m^\epsilon = \xi, \mathbf{u}_m^\epsilon = u\} = P\{\mathbf{u}_{m+1}^\epsilon \in B_2 | \xi_m^\epsilon = \xi, \mathbf{u}_m^\epsilon = u\} P_u(\xi, B_1) \tag{27}$$

4.1 The Basic SA Framework

Under any of our three schemes in the last section, the updates of the control variable are of the general form:

$$\mathbf{u}_{m+1}^\epsilon = \mathbf{u}_m^\epsilon + \epsilon Y_m^\epsilon \tag{28}$$

Recall that in the case of control constraints $u \in U$, we still use this update scheme, but employ the projection $\Pi_U(\mathbf{u}_m^\epsilon)$ as the *actual control*, as discussed in sections 3.3.1-3.3.3.

Based on the analysis in section 3, it should be clear that this general form allows for Y_m^ϵ to be zero whenever $m+1$ is not an update epoch. In turn, this information is adapted to the evolution of the process; in our example, updates take place either after a fixed number of events, or after a fixed number of busy periods at a particular node (for the central controller case) or at each node (in the decentralized operation).

Before proceeding, we will present three more technical conditions used in our analysis. First, the derivative estimators at component k, introduced in section 3.2.1, are assumed to satisfy the following regularity conditions:

Assumption 4: For the process $\{\xi_m(u)\}$ and any integers j and Δ, $E_u\{d_k(j,\Delta)\ |\xi_{m(j)}(u) = \xi\}$ is a continuous function in (ξ, u). Furthermore, for every compact set $A \subset U$ there exists some $K(A)$ such that

$$\sup_{u\in A} E_u\{\|d_k(j,\Delta)\|\,|\xi_{m(j)}(u) = \xi\} < K(A)\Delta,$$

The latter condition is satisfied by all standard gradient estimators under mild technical conditions on the distributions of all event lifetimes affected by u.

Example (continued): Let us return to the derivative estimator (13) which was introduced in [5]. For simplicity, we limit ourselves to the regenerative form of this estimator in (16) (however, our analysis can be easily extended to the finite horizon implementation; see [25] for details in a similar example). We shall use the fact (established in [5]) that $f_k(X_{m(j)}, u_k) \leq (1/u_k)M_l$ (with $\delta = 1$), where M_l is the total number of local events in the lth busy period at node k, which we also used earlier. In addition, we shall assume that for any $u \in U$, $E^u[M_1^4] < \infty$.

In our example, if the estimator is calculated starting at event $m(j)$ for Δ local events, then from the additivity property (12), Assumption 4 follows from the a.s. bound of $d_k(j,\Delta)$ in (13). In particular, letting $l(i)$ denote the index of the busy period where the local event i belongs,

$$\|d_k(j,\Delta)\| = \left\|\sum_{i=j}^{j+\Delta} f_k(X_{m(i)}, u_k)\right\| \leq \left\|\frac{1}{u_k}\sum_{i=j}^{j+\Delta} M_{l(i)}\right\|$$

The expectation in Assumption 4 can then be uniformly bounded for any compact set $A \in U$ using the value $\bar{u}_k = \min\{u_k : u \in A\}$ instead. Recalling the dominating process argument for any node k, which we first used in the example of section 3.1, let $\bar{M}_l$ denote the number of events in one busy period of this dominating process. Under the assumption that $E[\bar{M}_l^2] < \infty$, the above bound has an expectation which is linear in Δ.

$\triangle$

Assumption 5: The set of random variables $\{\xi_m^\epsilon, \mathbf{u}_m^\epsilon\}$ is tight. The sequence of random variables $\{Y_m^\epsilon; \epsilon > 0; n = 0, 1, \ldots\}$ is uniformly integrable.

Example (continued): Recall that in introducing our example in section 3.1, we used a common truncation argument to ensure that the resulting control variables are within a compact set A contained in the stable region U. Therefore, the varying control values u_m^ϵ all lie in a compact set completely contained in the stable region U. Tightness of $\{u_m^\epsilon\}$ follows directly from this fact. Also, for each node k, the queueing process with varying u_m^ϵ is stochastically dominated by a stable GI/G/1 queue with geometric service times at parameter $\bar{u}_k = \min\{u_k : u \in A\}$. As before, let $\bar{M}_l$ be the number of events in busy period l of the dominating process at node k, and consider the decentralized operation where each node performs a control update every $N = 1$ busy periods. From our previous bounds, $\psi_k^\epsilon(m) \leq (1/\bar{u}_k)\bar{M}_l^2$ a.s., with l denoting the current busy period. Similarly, the waiting time of a packet, $x_k^\epsilon(p)$, belonging to busy period l is always bounded by δM_l, that is, $x_k^\epsilon(p) \leq \bar{M}_l$ a.s., for $\delta = 1$. Since for any α we can find a sufficiently large K_α with $P\{\bar{M}_l^2 > K_\alpha\} < \alpha$, it follows that $\{X_m, \psi_k(m), k = 1, \ldots, K\}$ are tight and so is the sequence $\{\xi_m^\epsilon, u_m^\epsilon\}$. Uniform integrability of $\{Y_m^\epsilon\}$ follows from the assumption that $E\{\bar{M}_l^4\} < \infty$, implying that the variance of the estimators is uniformly bounded for any compact set contained in U.

$\triangle$

Recall that $\Delta_k(l)$ is the number of events contained in the lth estimation interval at component k, and $M(n)$ is the number of events contained in the nth update interval in the case of a centralized control structure. We shall assume the following properties of the stopping times associated with the estimation and control update processes.

Assumption 6: For all $k = 1, \ldots, K$, the sequences $\{\Delta_k^\epsilon(l)\}$ and $\{M^\epsilon(n)\}$ are uniformly integrable.

Example (continued): In our example, under regenerative estimation, $\Delta_k^\epsilon(l)$ is the number of packets transmitted in busy period l of node k, which, by assumption, has a uniformly bounded variance. This is shown using again the argument with the dominating stable queues. For the central controller, a similar argument applies to $M^\epsilon(n)$ for either a fixed number of service completions or a fixed number of local busy periods.

$\triangle$

4.2 The Interpolation Processes

When dealing with the notion of convergence, we implicitly assume a concept of a norm. The approaches that study a.s. convergence of the sequence $\{\mathbf{u}_m^\epsilon\}$ generally use the norm in $\mathbb{R}^K$. As motivated in the introduction of section 3, the approach taken in this work is the study of the behavior of the updates by taking a global view of the processes and establishing that it gets closer to the trajectory of the solution of an ODE, as $\epsilon \to 0$. The limiting process shall therefore be a *continuous time* process. The first step in analyzing weak convergence of SA schemes is to define "continuous time" processes from the event-driven sequence of control values. The time scale used in this definition is, of course, related to the gain parameter or learning rate ϵ. In our general framework, we have talked about a faster time scale that drives the process according to the events that trigger all state transitions, and a slower time scale according to which the control values change. We shall therefore begin by defining two important continuous time processes, as follows.

Let us start by considering $\{S^\epsilon(n), n \geq 0\}$ to be a sequence of random stopping event indices, measurable with respect to the filtration $\{\mathcal{F}_m^\epsilon\}$ of a MDP process $(\xi_m^\epsilon, \mathbf{u}_m^\epsilon)$. Then, set

$$\Delta^\epsilon(n) = S^\epsilon(n+1) - S^\epsilon(n)$$

In addition, let $(\chi^\epsilon(n), w^\epsilon(n)) = (\xi_{S^\epsilon(n)}^\epsilon, \mathbf{u}_{S^\epsilon(n)}^\epsilon)$ be a random sampling of the state of the process.

We now define the *ladder interpolation process* associated with $w^\epsilon(n) = \mathbf{u}_{S^\epsilon(n)}^\epsilon$:

$$\zeta^\epsilon(t) = w^\epsilon(n) \quad \text{for } t \in [n\epsilon, (n+1)\epsilon) \tag{29}$$

and the *natural interpolation process*:

$$\tilde{\zeta}^\epsilon(t) = \mathbf{u}_m^\epsilon \quad \text{for } t \in [m\epsilon, (m+1)\epsilon) \tag{30}$$

The first interpolation scales time with respect to *control update* intervals, and the second with respect to *global event* epochs. Figure 4 illustrates the construction of these processes. We begin with the piecewise constant process describing control updates as a function of the global event index m (thin solid line drawn on a $m\epsilon$ scale with jumps shown at *update* events). This defines the natural interpolation process $\tilde{\zeta}^\epsilon(t)$. This process is then sampled at a subset of event indices $\{S(0), S(1), \ldots\}$ with corresponding values $w^\epsilon(0), w^\epsilon(1), \ldots$, (thick solid line drawn on a $m\epsilon$ scale with jumps shown at *sampling* events) as shown in Figure 4. The ladder interpolation process $\zeta^\epsilon(t)$ is finally simply obtained by redrawing this piecewise constant function as a function of the control update index n on a $n\epsilon$ scale.

We shall now be interested in establishing some general properties of these processes when $\{S^\epsilon(n)\}$ is related to the the control update sequences corresponding to the three schemes defined in the last section: $\{nM\}$ for the centralized structure with update epochs every M global events; $\{G(n)\}$ for the centralized

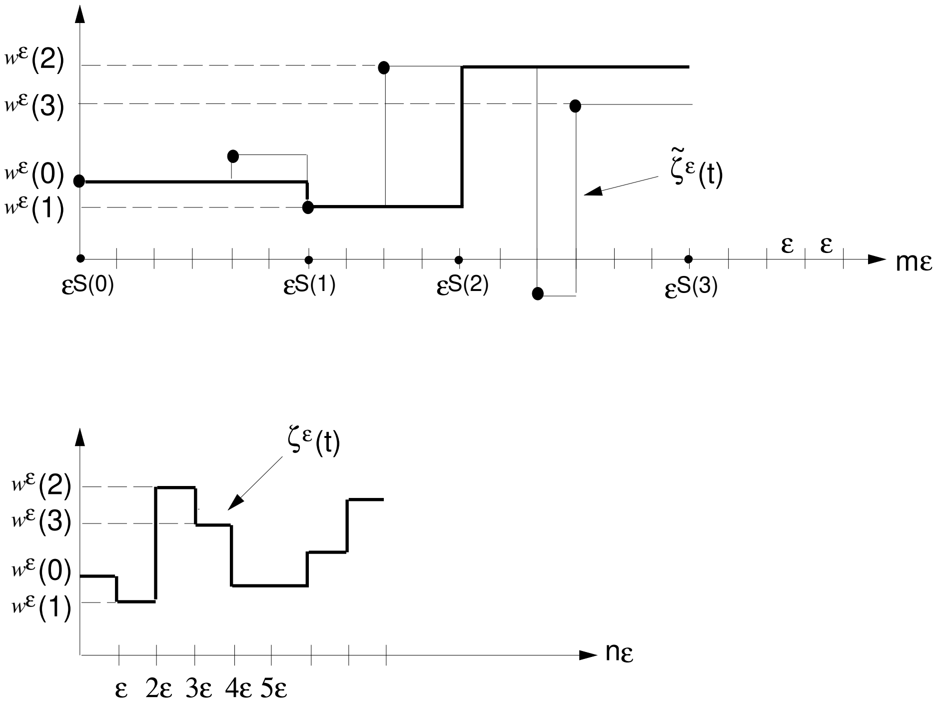

Figure 4: Illustrating the interpolation processes $\tilde{\zeta}^\epsilon(t)$ and $\zeta^\epsilon(t)$

structure with random update epochs; and $\{G_k(l)\}$ for the fully decentralized structure driven by individual components at epochs $G_k(l) = m(L_k(l))$.

The following result is needed to guarantee that, under Assumption 5, the interpolation processes we will work with satisfy a tightness condition. Recall that tightness of a sequence of stochastic processes (indexed by ϵ) is analogous to compactness and implies that any subsequence has a further weakly convergent subsequence (see [17] for detailed definitions and discussion). Therefore, this result allows us to work with weakly convergent subsequences of an interpolation process in order to characterize its limit as the solution of an ODE. In the analysis that follows, we will repeatedly exploit this fact.

Proposition 1 *Let $\{Y^\epsilon(n)\} \in I\!R$ be a sequence of uniformly integrable random variables and $\theta^\epsilon(n+1) = \theta^\epsilon(n) + \epsilon Y^\epsilon(n)$, with $\theta^\epsilon(0) = \theta(0)$ independent of ϵ. Call $\vartheta^\epsilon(t) = \theta^\epsilon(n)$ for $t \in [n\epsilon, (n+1)\epsilon)$ the corresponding interpolation process. Then the sequence of interpolations $\{\vartheta^\epsilon(\cdot); \epsilon > 0\}$ is tight in the space of piecewise constant, right continuous processes $D[0,\infty)$. Furthermore, all weak limits are Lipschitz continuous w.p.1.*

Proof : See Appendix.

Our first application of Proposition 1 gives us the following general result regarding the two interpolation processes in (29) and (30):

Corollary 1 *Under Assumption 5, if the sequence $\{\Delta^\epsilon(n); \epsilon > 0\}$ is uniformly integrable, then the interpolation processes $\{\zeta^\epsilon(\cdot), \tilde{\zeta}^\epsilon(\cdot); \epsilon > 0\}$ defined by (29) and (30) are tight and all joint weak limits are Lipschitz continuous w.p.1.*

Proof : Since $\mathbf{u}_m^\epsilon$ in (30) satisfies the recursion (28), it follows directly from Assumption 5 and Proposition 1 (applied to each component of the vector valued control process) that $\{\tilde{\zeta}^\epsilon(\cdot); \epsilon > 0\}$ are tight and all weak limits are Lipschitz contnuous w.p.1. In order to show that the same is true for the ladder interpolation process (29), define the *time scaling process*:

$$\tau^\epsilon(t) = \epsilon S^\epsilon(n) \quad \text{for } t \in [n\epsilon, (n+1)\epsilon)$$

and observe that $S^\epsilon(n)$ satisfies the recursion $S^\epsilon(n+1) = S^\epsilon(n) + \Delta^\epsilon(n)$. We can, therefore, apply Proposition 1 to this process and conclude that $\{\tau^\epsilon(\cdot); \epsilon > 0\}$ are tight and all weak limits are Lipschitz continuous w.p.1. Next, from the definition of $\tau^\epsilon(t)$ and (29)-(30), it follows that for every ϵ, if $t \in [n\epsilon, (n+1)\epsilon)$, then $\tau^\epsilon(t) = \epsilon S^\epsilon(n)$ and $\tilde{\zeta}^\epsilon[\tau^\epsilon(t)] = \tilde{\zeta}^\epsilon(\epsilon S^\epsilon(n)) = \mathbf{u}^\epsilon_{S^\epsilon(n)}$. Therefore,

$$\tilde{\zeta}^\epsilon[\tau^\epsilon(t)] = w^\epsilon(n) = \zeta^\epsilon(t) \quad \text{for } t \in [n\epsilon, (n+1)\epsilon)$$

which implies tightness of $\{\zeta^\epsilon(\cdot); \epsilon > 0\}$. Therefore, for any jointly weakly convergent subsequence of the processes $\{\tilde{\zeta}^\epsilon(\cdot), \tau^\epsilon(\cdot)\}$ (indexed also by ϵ) with limit $(\tilde{\zeta}(\cdot), \tau(\cdot))$, the sequence of processes $\zeta^\epsilon(\cdot)$ converges weakly to $\zeta(t) = \tilde{\zeta}[\tau(t)]$, which is Lipschitz continuous w.p.1.

□

4.3 The Averaging Result

In this section we apply the method first introduced in [18] and generalize the result of [17] for the random sampling of the process $(\chi^\epsilon(n), w^\epsilon(n))$. Our model is more restrictive only in the assumption of time homogeneity of the MDP $(\xi^\epsilon_m, \mathbf{u}^\epsilon_m)$ and can be appropriately extended to the non-homogeneous case, if desired. Notice that, according to the definition of the ladder interpolation process, it follows that $\zeta^\epsilon(n\epsilon) = w^\epsilon(n)$.

The main result in Proposition 2 below is to show that two averages are equivalent in the limit as $\epsilon \to 0$, under a set of technical conditions. In particular, for any continuous and bounded function $f(x,u)$, define:

$$\bar{f}^\epsilon(x_0, u_0) = \frac{1}{n_\epsilon} \sum_{n=l_\epsilon n_\epsilon}^{(l_\epsilon+1)n_\epsilon - 1} E\{f(\chi^\epsilon(n), u^\epsilon(n)) | \chi^\epsilon(l_\epsilon n_\epsilon) = x_0, w^\epsilon(l_\epsilon n_\epsilon) = u_0\} \tag{31}$$

This is the average over n_ϵ samples taken in an interval defined by $n \in \{l_\epsilon n_\epsilon, \ldots, (l_\epsilon+1)n_\epsilon - 1\}$. Note that at every sample the control value $u^\epsilon(n)$ generally varies. Next, define

$$\hat{f}^\epsilon(x_0, u_0) = \frac{1}{n_\epsilon} \sum_{m=0}^{n_\epsilon - 1} E_{u_0}\{f(\chi^\epsilon(m), u_0) | \chi^\epsilon(0) = x_0\} \tag{32}$$

which is another average over n_ϵ samples, this time taken in an interval defined by $\{0, \ldots, n_\epsilon - 1\}$, and note that in this case all control values $u^\epsilon(n)$ are fixed at u_0. The significance of the result that the two averages are equivalent as $\epsilon \to 0$ is revealed when we think back to the distributed estimation process discussed in section 3.2. This process was described under a fixed u_k throughout an estimation interval at component k. Yet, clearly, several control changes could be dictated by either a central controller or other components (in the decentralized case) within such an interval. If we think of $f(\cdot)$ above as a local estimator, Proposition 2 permits us to work with a fixed control process, as we did in section 3.2, because we are ultimately concerned only with the behavior of our system in the limit as $\epsilon \to 0$. In fact, Proposition 2 contains an even stronger statement involving the invariant measure of the fixed-u process. We will also see that the key condition in Proposition 2, i.e., (33) below, is verifiable for all cases of interest we consider in section 4.5.

Proposition 2 *Let $(\xi^\epsilon_m, \mathbf{u}^\epsilon_m)$ be a MDP satisfying Assumptions 1 and 5. Let $\{\Delta^\epsilon(n)\}$ be uniformly integrable and $S^\epsilon(n+1) = S^\epsilon(n) + \Delta^\epsilon(n)$, with $S^\epsilon(0) = 0$ be a sequence of stopping times, measurable with respect to the MDP process. Let $(\chi^\epsilon(n), w^\epsilon(n))$ be the random sampling process. Pick a weakly convergent subsequence*

$\{\zeta^\epsilon(\cdot)\}$ *also indexed by* ϵ*, with limit* $\zeta(\cdot)$*. Assume that along this subsequence of values of* $\epsilon \to 0$*, for any bounded, continuous function* $F(y)$*and any* (u, x)*,*

$$\int P\{\chi^\epsilon(n+1) \in dy | w^\epsilon(n) = u, \chi^\epsilon(n) = x\} F(y) = \int P_u\{\chi^\epsilon(n+1) \in dy | \chi^\epsilon(n) = x\} F(y) + r_n(\epsilon) \qquad (33)$$

where $|r_n(\epsilon)| \leq K\epsilon$ *for some constant* K*, and* P_u *denotes the measure with respect to the fixed-u process. Let* $s > 0$ *be any fixed number,* $\delta_\epsilon = \epsilon n_\epsilon$ *be such that* $n_\epsilon \to \infty, \delta_\epsilon \to 0$ *as* $\epsilon \to 0$*, and call* l_ϵ *the index such that* $l_\epsilon \delta_\epsilon \leq s < (l_\epsilon + 1)\delta_\epsilon$*. Then, using the definitions (31)- (32), for any continuous and bounded function* $f(x, u)$*:*

$$\lim_{\epsilon \to 0} E \left\| \bar{f}^\epsilon(\chi^\epsilon(l_\epsilon n_\epsilon), w^\epsilon(l_\epsilon n_\epsilon)) - \hat{f}^\epsilon(\chi^\epsilon(l_\epsilon n_\epsilon), w^\epsilon(l_\epsilon n_\epsilon)) \right\| = 0 \qquad (34)$$

where the expectation is w.r.t. to the distribution of the random variables $\chi^\epsilon(l_\epsilon n_\epsilon), w^\epsilon(l_\epsilon n_\epsilon)$*. Moreover,*

$$\lim_{\epsilon \to 0} E \left\| \bar{f}^\epsilon(\chi^\epsilon(l_\epsilon n_\epsilon), \zeta^\epsilon(s)) - \int \mu_{\zeta(s)}(dx) f(x, \zeta(s)) \right\| = 0 \qquad (35)$$

where $\mu_u(\cdot)$ *denotes the invariant measure of the fixed-u process.*

Proof : See Appendix.

We shall now introduce a time continuous process related to any continuous and bounded function $f(x, u)$ related to $\bar{f}^\epsilon(\cdot)$ and rephrase the result of Proposition 2 in a way that we shall frequently use later on.

Corollary 2 *Under the assumptions of Proposition 2, for any bounded and continuous function* $f(x, u)$*, if we define:*

$$\mathcal{G}^\epsilon(\tau) = \bar{f}^\epsilon(\chi^\epsilon(l n_\epsilon), w^\epsilon(l n_\epsilon)) \quad \text{for } l\delta_\epsilon \leq \tau < (l+1)\delta_\epsilon$$

and

$$\hat{f}(u) = \lim_{\epsilon \to 0} \hat{f}^\epsilon(x, u) = \int \mu_u(dx) f(x, u)$$

then $\lim_{\epsilon \to 0} E\|\mathcal{G}^\epsilon(s) - \hat{f}(\zeta(s))\| = 0$.

Proof : Under the ergodicity assumptions, the limit $\hat{f}$ of $\hat{f}^\epsilon$ is independent of the initial value x of the state and is the invariant average of the function $f(x, u)$. Then, by the definition of $\mathcal{G}^\epsilon(\cdot)$ and (34), we obtain the result.

□

4.4 The Integral Representation

Let $(\xi_m^\epsilon, \mathbf{u}_m^\epsilon)$ be a MDP satisfying Assumptions 1 and 5. Let $S^\epsilon(n)$ be a sequence of random stopping event indices measurable w.r.t to $\{\mathcal{F}_m^\epsilon\}$, and $\Delta^\epsilon(n+1) = S^\epsilon(n+1) - S^\epsilon(n)$ as before. Assume that the $\{\Delta^\epsilon(n)\}$ are uniformly integrable and let, as before, $(\chi^\epsilon(n), w^\epsilon(n))$ be the random sampling process. The ladder interpolation process $\zeta^\epsilon(\cdot)$ is defined as in (29) with respect to $\{S^\epsilon(n)\}$. Let $Y^\epsilon(n)$ be a real-valued function of $(\xi_m^\epsilon, \mathbf{u}_m^\epsilon)$ over the interval $m \in \{S^\epsilon(n), S^\epsilon(n+1) - 1\}$, and assume that $\{Y^\epsilon(n)\}$ are uniformly integrable. Let

$$\theta^\epsilon(n+1) = \theta^\epsilon(n) + \epsilon Y^\epsilon(n) \qquad (36)$$

and call $\vartheta^\epsilon(t)$ the real-valued ladder interpolation process:

$$\vartheta^\epsilon(t) = \theta^\epsilon(n) \quad \text{for } t \in [n\epsilon, (n+1)\epsilon) \qquad (37)$$

and finally, call $\tau^\epsilon(\cdot)$ the time scale process:

$$\tau^\epsilon(t) = \epsilon S^\epsilon(n) \quad \text{for } t \in [n\epsilon, (n+1)\epsilon) \tag{38}$$

Using Proposition 1, each subsequence of $\{\vartheta^\epsilon(\cdot), \zeta^\epsilon(\cdot)\}$ has a further weakly convergent subsequence with Lipschitz continuous limit process $\vartheta(\cdot), \zeta(\cdot)$. Our next result will characterize the limit process $\vartheta(\cdot)$ of such weakly convergent subsequences in terms of the solution of an ODE depending on the limit control process $\zeta(\cdot)$ along a chosen weakly convergent subsequence. In the following section, we shall identify $\vartheta^\epsilon(\cdot)$ with each component of the control processes $\zeta^\epsilon(\cdot)$ or functions of it. The time scale interpolations will therefore be defined with respect to different stopping time sequences, depending on the scheme itself.

We shall now develop the basis for the integral representation of the process $\vartheta(\cdot)$, a key ingredient in the proof that lends itself to the title of this section. Using a telescopic sum, from (36)-(37) we can write:

$$\vartheta^\epsilon(t+s) - \vartheta^\epsilon(t) = \epsilon \sum_{n=\lfloor t/\epsilon \rfloor}^{\lfloor (t+s)/\epsilon \rfloor} Y^\epsilon(n)$$

Following the method in [18, 17, 24], let $\delta_\epsilon = \epsilon n_\epsilon$, where $\delta_\epsilon \to 0$ and $n_\epsilon \to \infty$ as $\epsilon \to 0$. This corresponds to a "time scale change" device which we will repeatedly use in our analysis. Briefly, consider a time interval $[t, t+s]$ partitioned into N subintervals, each of length ϵ. With this "time scale" ϵ, the total length of the interval is then $N\epsilon$. Now set $\delta_\epsilon = \epsilon n_\epsilon$ such that $n_\epsilon \to \infty$ and $\delta_\epsilon \to 0$ as $\epsilon \to 0$. In this new "time scale" δ_ϵ, the total length of the interval is simply partitioned into (N/n_ϵ) subintervals, each of length δ_ϵ. Then, the sum in the expression above can be replaced by two sums: an inner one over all n_ϵ subintervals contained in an interval of length δ_ϵ, and an outer one over all the latter intervals. Thus, we have:

$$\vartheta^\epsilon(t+s) - \vartheta^\epsilon(t) \;=\; \epsilon \sum_{l=\lfloor t/\delta_\epsilon \rfloor}^{\lfloor (t+s)/\epsilon \rfloor} \sum_{n=ln_\epsilon}^{ln_\epsilon + n_\epsilon - 1} Y^\epsilon(n) + \mathcal{O}(\delta_\epsilon) = \sum_{l=\lfloor t/\delta_\epsilon \rfloor}^{\lfloor (t+s)/\delta_\epsilon \rfloor} \delta_\epsilon \frac{1}{n_\epsilon} \sum_{n=ln_\epsilon}^{ln_\epsilon + n_\epsilon - 1} Y^\epsilon(n) + \mathcal{O}(\delta_\epsilon) \tag{39}$$

where the term $\mathcal{O}(\delta_\epsilon)$ accounts for the end effects of the larger intervals of size δ_ϵ.

Let E_n denote the conditional expectation given all values up to time $S^\epsilon(n)$. From the MDP structure, this means that for any function $F(x,u)$:

$$E_n[F(\xi^\epsilon_m, \mathbf{u}^\epsilon_m)] \equiv E\{F(\xi^\epsilon_m, \mathbf{u}^\epsilon_m) | \chi^\epsilon(n), w^\epsilon(n)\}, \quad \text{for all } m \geq S^\epsilon(n)$$

whose distribution depends on the distribution of the initial conditions at time $S^\epsilon(n)$.

We now define an operator $\mathcal{A}$ on the space of piecewise constant functions of the form (37) as follows:

$$\mathcal{A}[\vartheta^\epsilon](s) = \frac{1}{n_\epsilon} \sum_{n=ln_\epsilon}^{ln_\epsilon + n_\epsilon - 1} E_{ln_\epsilon}[Y^\epsilon(n)] \quad \text{for } l\delta_\epsilon \leq s < (l+1)\delta_\epsilon \tag{40}$$

so that if we set $\mathcal{G}^\epsilon_\vartheta = \mathcal{A}[\vartheta^\epsilon]$, then $\mathcal{G}^\epsilon_\vartheta(\cdot)$ is also a real valued, piecewise constant stochastic process. It is defined through the average expected changes in the process $\vartheta^\epsilon(\cdot)$ over time intervals of size δ_ϵ. Since it is piecewise constant on intervals of length δ_ϵ, we have:

$$\tilde{\mathcal{G}}^\epsilon_\vartheta(t) = \int_0^t \mathcal{A}[\vartheta^\epsilon](s)\, ds = \sum_{l=\lfloor t/\delta_\epsilon \rfloor}^{\lfloor (t+s)/\delta_\epsilon \rfloor} \delta_\epsilon\, \mathcal{A}[\vartheta^\epsilon](l\delta_\epsilon)$$

which yields the integral representation of $\tilde{\mathcal{G}}^\epsilon_\vartheta$.

Proposition 3 *Under Assumptions 1 and 5 for $(\xi_m^\epsilon, \mathbf{u}_m^\epsilon)$, if there exists a continuous and bounded function $\hat{f}(\cdot)$ such that for every weakly convergent subsequence $\{\zeta^\epsilon(\cdot), \vartheta^\epsilon(\cdot)\}$ we have:*

$$\lim_{\epsilon \to 0} E\|\mathcal{A}[\vartheta^\epsilon](s) - \hat{f}[\zeta(s)]\| = 0 \tag{41}$$

uniformly in s, then the a.s. Lipschitz continuous limit $\vartheta(\cdot)$ along this subsequence satisfies:

$$\frac{\partial \vartheta(t)}{\partial t} = \hat{f}[\zeta(t)] \tag{42}$$

Proof : Define the process:

$$M^\epsilon(t) = \vartheta^\epsilon(t) - \vartheta^\epsilon(0) - \int_0^t \mathcal{A}[\vartheta^\epsilon](r)\, dr \tag{43}$$

We will now apply Proposition 1 to the processes $\{\zeta^\epsilon(\cdot), \vartheta^\epsilon(\cdot), \tilde{\mathcal{G}}_\vartheta^\epsilon(\cdot)\}$. We can do so for $\tilde{\zeta}^\epsilon(\cdot), \zeta^\epsilon(\cdot)$ directly from Corollary 1. Proposition 1 applies to $\{\vartheta^\epsilon(\cdot)\}$ in view of (36)-(37). We can also apply it to $\tilde{\mathcal{G}}_\vartheta^\epsilon(\cdot)\}$ by looking at its definition and observing that we can obtain a recursion by identifying in Proposition 1 ϵ with δ_ϵ and Y_n^ϵ with $\mathcal{G}_\vartheta^\epsilon(n\delta_\epsilon)$, and noting that $\delta_\epsilon \to 0$ as $\epsilon \to 0$. Therefore, we conclude that for any subsequence, there exists a jointly weakly convergent subsequence with a.s. Lipschitz continuous limits $\zeta(\cdot), \vartheta(\cdot)$, $\tilde{\mathcal{G}}_\vartheta(\cdot)$. Choose any such subsequence (indexed also by ϵ). Then, along this subsequence, $M^\epsilon(\cdot)$ converges to a Lipschitz continuous process $M(\cdot)$.

Let $\mathcal{F}_t^\epsilon$ denote the history σ-algebra of the process $\zeta^\epsilon(\cdot)$ up to time t. Then, using a conditioning argument on (39) we have:

$$\begin{aligned}
\mathcal{O}(\delta_\epsilon) &= E\left\{\vartheta^\epsilon(t+s) - \vartheta^\epsilon(t) - \sum_{l=\lfloor t/\delta_\epsilon \rfloor}^{\lfloor (t+s)/\delta_\epsilon \rfloor} \delta_\epsilon \frac{1}{n_\epsilon} \sum_{n=ln_\epsilon}^{ln_\epsilon + n_\epsilon - 1} Y^\epsilon(n) | \mathcal{F}_t^\epsilon \right\} \\
&= E\left\{\vartheta^\epsilon(t+s) - \vartheta^\epsilon(t) - \sum_{l=\lfloor t/\delta_\epsilon \rfloor}^{\lfloor (t+s)/\delta_\epsilon \rfloor} \delta_\epsilon \frac{1}{n_\epsilon} \sum_{n=ln_\epsilon}^{ln_\epsilon + n_\epsilon - 1} E_{ln_\epsilon}(Y^\epsilon(n)) | \mathcal{F}_t^\epsilon \right\} \\
&= E\left\{\vartheta^\epsilon(t+s) - \vartheta^\epsilon(t) - \int_t^{t+s} \mathcal{A}[\vartheta^\epsilon](r)\, dr | \mathcal{F}_t^\epsilon \right\}
\end{aligned}$$

where we have used (40) and the integral representation for $\tilde{\mathcal{G}}_\theta^\epsilon(\cdot)$. Using (41), the limiting process $M(t)$ along the weakly convergent subsequence, given by:

$$M(t) = \vartheta(t) - \vartheta(0) - \int_0^t \hat{f}[\zeta(s)]\, ds,$$

is a Lipschitz continuous martingale (from the Lipschitz continuity of the limit functions shown in Proposition 1), which implies that $M(t)$ has zero quadratic variation. Since $M(0) = 0$, it then follows that $M(t) \equiv 0$ with probability 1, therefore, using (43) the limit process satisfies:

$$\vartheta(t+s) - \vartheta(t) = \int_t^{t+s} \hat{f}[\zeta(\tau)]\, d\tau$$

and since $\hat{f}(\cdot)$ is a bounded, continuous function, the solution to this ODE is a deterministic and continuous function of t.

□

Remark: If $f(x,u)$ is given by:

$$f(x,u) = E\{Y^\epsilon(n) | \chi^\epsilon(n) = x, w^\epsilon(n) = u\} \tag{44}$$

then, following the notation in our previous subsection:

$$\hat{f}(u) = \int \mu_u(dx) f(x,u) \tag{45}$$

With this definition of $f(x,u)$ it follows that the expression for $\mathcal{G}^\epsilon(\cdot)$ introduced in Corollary 2 is equivalent to $\mathcal{A}[\vartheta^\epsilon](\cdot)$ in (40). Indeed, $E_n[Y_n] = f(\chi^\epsilon(n), w^\epsilon(n))$ and using conditional expectations, for all $n \geq ln_\epsilon$,

$$E_{ln_\epsilon}[Y^\epsilon(n)] = \int P\{\chi^\epsilon(n) \in dx, w^\epsilon(n) \in du | \chi^\epsilon(ln_\epsilon), w^\epsilon(ln_\epsilon)\} f(x,u) = E_{ln_\epsilon}[f(\chi^\epsilon(n), w^\epsilon(n))]$$

and thus:

$$\mathcal{A}[\vartheta^\epsilon](s) = \bar{f}^\epsilon(\chi^\epsilon(ln_\epsilon), w^\epsilon(ln_\epsilon)) \quad \text{for } l\delta_\epsilon \leq s < (l+1)\delta_\epsilon \tag{46}$$

where we have used the definition (31) of the previous subsection. Therefore, if Proposition 2 is applicable to (44), then condition (41) of Proposition 3 is satisfied and this, in turn, can be used to characterize the limiting ODE of the general form (42).

4.5 Convergence of the Algorithms

We shall now consider the update schemes in section 3.3. The DES is assumed to satisfy Assumptions 1 and 2. The estimators are assumed to satisfy Assumptions 3 and 4. Finally, the augmented MDP is assumed to satisfy Assumption 5, and the random stopping times satisfy Assumption 6. In particular, since the partial computations and the residual times related to the stopping times are part of the enlarged state space, for the fixed-u process the transition probabilities $P_u(x,B)$ are weakly continuous in u and there is a unique invariant measure $\mu_u(\cdot)$.

In the following, we present our main convergence results for the three update schemes in section 3.3. We shall limit ourselves to the most complex of the three, the decentralized asynchronous control case, since the remaining two can be similarly treated as special cases (detailed proofs can be found in [27]).

Before proceeding, we present next a lemma regarding a basic property of the derivative estimators we have defined in section 3.2 which are used in the three control schemes of section 3.3.

Lemma 1 *Under Assumptions 1-4, $\tilde{d}_k(n)$, defined in (14), satisfies the following:*

$$\lim_{m\to\infty} E_u \frac{1}{m} \sum_{n=0}^{m-1} E_u\{\tilde{d}_k(n) | \xi_{G_k(n)}(u)\} = \bar{M}_k(u)\rho_k(u)\frac{dJ_k(u_k)}{du_k} \tag{47}$$

Proof : Using the additivity property (12) in Assumption 3 we have:

$$\begin{aligned}
\frac{1}{m}\sum_{n=0}^{m-1} E_u\{\tilde{d}_k(n)|\xi_{G_k(n)}(u)\} &= \frac{1}{m} E_u\{d_k(0, L_k(m))|\xi_0\} \\
&= E_u\left\{\left(\frac{G_k(m)}{m}\right)\left(\frac{L_k(m)}{G_k(m)}\right)\frac{d_k(0,L_k(m))}{L_k(m)}|\xi_0\right\}
\end{aligned}$$

Since the global event index $G_k(m)$ is strictly increasing in m, then from (10) we have $\lim_{m\to\infty} L_k(m)/G_k(m) = \rho_k(u)$ a.s. for the fixed-u process. Moreover, from (15), $\lim_{m\to\infty} G_k(m)/m = \bar{M}_k(u)$, and using Assumption 3 we get

$$E_u\left\{\frac{d_k(0,L_k(m))}{L_k(m)}|\xi_0\right\} \to \frac{dJ_k(u_k)}{du_k} \quad \text{a.s.}$$

which, in view of Assumption 4, by the dominated convergence theorem yields the desired result.

$\square$

The next two results are stated as corollaries of the above lemma, since they can be proved in the same straightforward way:

Corollary 3 *Under Assumptions 1-4, $D_k(n,M)$, defined in (17), satisfies the following:*

$$\lim_{m\to\infty} \frac{1}{m} E_u \sum_{n=0}^{m-1} E_u\{D_k(n,M)|\xi_{nM}(u)\} = \frac{dJ_k(u_k)}{du_k} \tag{48}$$

Corollary 4 *Under Assumptions 1-4, $D_k(n)$, defined in (20), satisfies the following:*

$$\lim_{m\to\infty} \frac{1}{m} E_u \sum_{n=0}^{m-1} E_u\{D_k(n)|\xi_{G(n)}(u)\} = \bar{M}(u)\frac{dJ_k(u_k)}{du_k} \tag{49}$$

We now consider the decentralized asynchronous control update scheme (23) and for each k, the update scheme (25) for the vector $v_k^\epsilon(n)$ (with components $v_{ki}^\epsilon(n)$, $i = 1,\ldots,K$.) According to the discussion of section 3.3.3, the values of $v_k^\epsilon(n)$ are updated only at the local update epochs at processor k corresponding to global event indices $G_k^\epsilon(n)$. We deal with any fixed k, and let $S^\epsilon(n) = G_k(n)$ satisfy Assumption 6. Let $\nu_k^\epsilon(t)$ and $\tilde{\nu}_k^\epsilon(t)$ be the ladder and natural interpolation processes related to $v_k^\epsilon(n)$, as follows:

$$\begin{aligned} \nu_k^\epsilon(t) &= v_k^\epsilon(n) \quad \text{for } t \in [n\epsilon,(n+1)\epsilon) \\ \tilde{\nu}_k^\epsilon(t) &= v_k^\epsilon(n) \quad \text{for } t \in [G_k(n)\epsilon, G_k(n+1)\epsilon) \end{aligned}$$

where $\nu_k^\epsilon(\cdot)$ is a vector with components $\nu_{ki}^\epsilon(\cdot)$, and similarly for $\tilde{\nu}_k^\epsilon(\cdot)$. Recall that the kth component changes the value of the auxiliary control variable $v_k^\epsilon(n)$ only at local event indices $G_k(n)$, so that $\tilde{\nu}_k^\epsilon(t)$ is in fact the natural interpolation process corresponding to $v_k^\epsilon(n)$, following the general definition (30).

Let $\zeta_k^\epsilon(\cdot)$ denote the vector-valued ladder interpolation of the control process with respect to the indices $G_k(n)$. Recall that each processor k will now have its own local time scale, so that the sampling of the control process is done locally at different epochs for different processors. The natural interpolation process $\tilde{\zeta}(\cdot)$ is independent of the time scale.

The processes $\nu_k^\epsilon(\cdot)$ can be identified with components of the control process itself, in the sense that the control is uniquely defined by the relationship:

$$\tilde{\zeta}^\epsilon(t) = \sum_{k\in C_q} \tilde{\nu}_k^\epsilon(t)$$

for all t,ϵ. Indeed, from equations (25) and (24), since $\tilde{\nu}_k^\epsilon(\cdot)$ are piecewise constant and only change at the epochs corresponding to local updates, then the actual control value at the epoch of event m is the initial value $\mathbf{u}_0$ plus the total changes effected at the control updates. On the other hand, $\tilde{\nu}_k(\cdot) - \tilde{\nu}_k(0)$ contains the cumulative changes performed at controller k. Since $\tilde{\nu}_k(0) = v_k^\epsilon(0)$ with $\sum_k v_k^\epsilon(0) = \mathbf{u}_0$, then the control used at any time $t \in [\epsilon n, \epsilon(n+1))$ is given by $\tilde{\zeta}^\epsilon(t) = \sum_k \tilde{\nu}_k(t)$. In the case of control constraints, recall that the projection $\Pi_U(\mathbf{u}_m^\epsilon)$ is introduced as the actual control. Accordingly, we shall introduce $\tilde{z}^\epsilon(t) = \Pi_U(\tilde{\zeta}^\epsilon(t))$.

Define also the local time scaling process:

$$\tau_k^\epsilon(t) = \epsilon G_k(n) \quad \text{for } t \in [n\epsilon,(n+1)\epsilon)$$

Since $\tilde{\nu}_k^\epsilon(\cdot)$ is constant over local update intervals, $\nu_k^\epsilon(t) = \tilde{\nu}_k^\epsilon[\tau_k^\epsilon(t)]$ and $\zeta_k^\epsilon(t) = \tilde{\zeta}^\epsilon[\tau_k^\epsilon(t)]$.

Theorem 1 *Under Assumptions 1 through 6, the processes $\{\tilde{\zeta}^\epsilon(t)\}$ converge weakly as $\epsilon \to 0$ to a solution of the ODE:*

$$\frac{d\tilde{\zeta}(t)}{dt} = \nabla_u J(\Pi_U[\tilde{\zeta}(t)]) \tag{50}$$

If the ODE (50) has a unique solution for each initial condition, then the sequence $\tilde{\zeta}^\epsilon(\cdot) \Rightarrow \tilde{\zeta}(\cdot)$. Furthermore, if (50) has a unique stable point u^* in the interior of U such that $\nabla_u J(u^*) = 0$, then $\lim_{t\to\infty} \tilde{\zeta}(t) = u^*$. The corresponding true control value limit process $\tilde{z}(t) = \Pi_U(\tilde{\zeta}(t))$ satisfies the corresponing projected ODE, so that if there is a unique maximum at the boundary $u^* \in \delta U$, then $\lim_{t\to\infty} z(t) = u^*$.

Proof : We work with any one system component k and show that the associated ladder interpolation process $\nu_k^\epsilon(\cdot)$ satisfies an ODE. Since the analysis is the same for every k, it will follow that for all k, the limit ladder processes will satisfy each a similar ODE. Let $S^\epsilon(n) = G_k^\epsilon(n)$ satisfy Assumption 6. Applying Corollary 1 of Proposition 1, given any subsequence of $\tilde{\zeta}^\epsilon(\cdot)$, we can choose a further subsequence so that the corresponding processes $\nu_k^\epsilon(\cdot)$, $\tau_k^\epsilon(\cdot)$ and $\zeta_k^\epsilon(\cdot)$ converge weakly. Let the random sampling process be $(\chi^\epsilon(n), w^\epsilon(n)) = (\xi^\epsilon_{S^\epsilon(n)}, \mathbf{u}^\epsilon_{S^\epsilon(n)})$. Set

$$f_{ki}(x,u) = E\{Y_{ki}^\epsilon(n) | \chi^\epsilon(n) = x, w^\epsilon(n) = u\}$$

for each $i = 1, \ldots, K$, where $Y_{ki}^\epsilon(n)$ are given by (26) and represent the changes in the artificial control component v_{ki}^ϵ at its jump epochs.

We shall apply Proposition 3, identifying the process $\vartheta^\epsilon(\cdot)$ with each of the components $k \in C_q$ of the artificial control process $\nu_{ki}^\epsilon(\cdot)$. Proposition 3 can be applied if we can apply Proposition 2 to these processes. We show in the Appendix as Lemma 2 that (33) is satisfied. It follows from (26) and from (47) in Lemma 1 that:

$$\hat{f}_{ki}(u) = \lim_{m\to\infty} \frac{1}{m} \sum_{n=0}^{m-1} E_u\{Y_{ki}^\epsilon(n) | \chi^\epsilon(0) = x\} = \gamma_{ki} \bar{M}_k(\Pi_U(u)) \frac{dJ_k}{du_k}(\Pi_U(u))$$

for all u. Recall that we now interpret E_u as the expectation for the fixed control process that operates at true value $\Pi_U(u)$. In order to apply Proposition 3, we shall show that (41) is satisfied. To shorten notation, call E_n the expectation conditioned on $(\chi^\epsilon(n), w^\epsilon(n))$, from (40) and (25) we have:

$$\mathcal{A}[\nu_{ki}^\epsilon](s) = \frac{1}{n_\epsilon} \sum_{n=ln_\epsilon}^{ln_\epsilon + n_\epsilon - 1} E_{ln_\epsilon}[Y_{ki}^\epsilon(n)] \quad \text{for } l\delta_\epsilon \le s < (l+1)\delta_\epsilon$$

If U is a bounded compact set, then by Assumption 4 $f_{ki}(x,u)$ is continuous and uniformly bounded. Then Corollary 2 applies using $\mathcal{G}_{ki}^\epsilon = \mathcal{A}[\nu_{ki}^\epsilon]$, which yields:

$$\lim_{\epsilon\to 0} E\|\mathcal{A}[\nu_{ki}^\epsilon](s) - \gamma_{ki}\bar{M}_k(\Pi_U[\zeta_k(s)]) \frac{dJ_k}{du_k}(\Pi_U[\zeta_k(s)])\| = 0 \tag{51}$$

If U is not bounded (or no projection is used), a truncation argument can be used as in [17]. Specifically, for every constant $B > 0$, let

$$f_B(x,u) = \begin{cases} -B & \text{if } f(x,u) < -B \\ f(x,u) & \text{if } -B \le f(x,u) \le B \\ B & \text{if } f(x,u) > B \end{cases}$$

denote the truncation of the function. Corollary 2 yields for any $B < \infty$:

$$\lim_{\epsilon\to 0} E\|\mathcal{A}_B[\nu_{ki}^\epsilon](s) - \hat{f}_{ki,B}(\zeta_k(s))\| = 0$$

where now

$$\mathcal{A}_B[\nu_{ki}^\epsilon](s) = \frac{1}{n_\epsilon} \sum_{n=ln_\epsilon}^{ln_\epsilon + n_\epsilon - 1} E_{ln_\epsilon} f_{ki_B}(\chi^\epsilon(n), w^\epsilon(n))] \quad \text{for } l\delta_\epsilon \le s < (l+1)\delta_\epsilon$$

and $\hat{f}_{ki,B}(u) = \int \mu_u(dx) f_{ki,B}(x,u)$. Use now the dominated convergence theorem to make $B \to \infty$, and obtain:

$$\lim_{B\to\infty} \hat{f}_{ki,B}(\zeta(s) = \gamma_{ki} \bar{M}_k(\Pi_U[\zeta_k(s)]) \frac{dJ_k}{du_k}(\Pi_U[\zeta_k(s)])$$

We assumed that $\bar{M}_k(u), \nabla_u J(u)$ were continuous and bounded functions. Since the projection is continuous and bounded, the RHS will be so too, as required. Since $\mathcal{A}_B[\nu^\epsilon_{ki}](s) \to \mathcal{A}[\nu^\epsilon_{ki}](s)$, as $B \to \infty$, then (51) holds.

We can now apply Proposition 3 to each of the components of the artificial control processes to obtain:

$$\frac{d\nu_{ki}(t)}{dt} = \bar{M}_k(\Pi_U[\zeta_k(t)])\gamma_{ki} \frac{dJ_k}{du_k}(\Pi_U[\zeta_k(t)]) \quad i = 1, \ldots, K \tag{52}$$

In order to characterize the dynamics of the limit process $\tilde{\zeta}(\cdot)$ we shall study the time scale process $\tau^\epsilon_k(\cdot)$ generated by the chosen k. Let $\Delta^\epsilon_k(n) = G_k(n+1) - G_k(n)$ be the lengths of the local update intervals and write the telescopic sum for $\tau^\epsilon_k(t)$:

$$\tau^\epsilon_k(t+s) - \tau^\epsilon_k(t) = \sum_{n=\lfloor t/\epsilon \rfloor}^{\lfloor (t+s)/\epsilon \rfloor} \Delta^\epsilon_k(n) = \sum_{l=\lfloor t/\delta_\epsilon \rfloor}^{\lfloor (t+s)/\delta_\epsilon \rfloor} \delta_\epsilon \frac{1}{n_\epsilon} \sum_{n=ln_\epsilon}^{ln_\epsilon + n_\epsilon - 1} \Delta^\epsilon_k(n) + \mathcal{O}(\delta_\epsilon)$$

so that

$$E\left\{ \tau^\epsilon_k(t+s) - \tau^\epsilon_k(t) - \int_t^{t+s} \mathcal{A}[\tau^\epsilon_k](s)\, ds | \mathcal{F}^\epsilon_t \right\} = \mathcal{O}(\epsilon)$$

where, by (46):

$$\mathcal{A}[\tau^\epsilon_k](s) = \frac{1}{n_\epsilon} \sum_{n=ln_\epsilon}^{(l+1)n_\epsilon} E_{l_\epsilon n_\epsilon}[\Delta^\epsilon_k(n)] \quad \text{for } s \in [ln_\epsilon, (l+1)n_\epsilon)$$

represents the expected average change in the time scale process over a time interval of size δ_ϵ. From the renewal structure, it follows that we can also apply Corollary 2 to obtain:

$$\lim_{\epsilon\to 0} E\|\mathcal{A}[\tau^\epsilon_k](s) - \bar{M}_k[\Pi_U[\zeta(s)]]\| = 0$$

and therefore, the limit function $\tau_k(\cdot)$ satisfies:

$$\frac{d\tau_k(t)}{dt} = \bar{M}_k(\Pi_U[\zeta(t)])$$

which is a deterministic, continuous and monotone function. Therefore it has an inverse function $\tau_k^{-1}(t)$ that satisfies:

$$1 = \frac{d\tau_k[\tau_k^{-1}(t)]}{dt} = \frac{d\tau_k}{dt}[\tau_k^{-1}(t)] \frac{d\tau_k^{-1}(t)}{dt} = \bar{M}_k[\Pi_U[\zeta(\tau_k^{-1}(t))]] \frac{d\tau_k^{-1}(t)}{dt} = \bar{M}_k[\Pi_U[\tilde{\zeta}(t)]] \frac{d\tau_k^{-1}(t)}{dt}$$

so that:

$$\frac{d\tau_k^{-1}(t)}{dt} = \frac{1}{\bar{M}_k[\Pi_U[\tilde{\zeta}(t)]]} \tag{53}$$

where we have used the fact that $\tilde{\zeta}^\epsilon[\tau_k(t)] = \zeta^\epsilon_k(t)$, and $\tilde{\zeta}^\epsilon(t) \Rightarrow \tilde{\zeta}(t) \equiv \zeta_k[\tau_k^{-1}(t)]$ along the convergent subsequence, which follows from Corollary 1.

Since for every ϵ, we have $\tilde{\nu}^\epsilon_k(\tau_k(t)) = \nu^\epsilon_k(t)$, using the chain rule for the derivatives, the limit process $\tilde{\nu}_k(\cdot)$ along this convergent subsequence satisfies for each component:

$$\frac{d\tilde{\nu}_{ki}(t)}{dt} = \frac{d\nu_{ki}}{dt}(\tau_k^{-1}) \frac{d\tau_k^{-1}(t)}{dt} = \bar{M}_k[\Pi_U[\tilde{\zeta}(t)]]\gamma_{ki} \frac{dJ_k}{du_k}(\Pi_U[\tilde{\zeta}(t)]) \frac{1}{\bar{M}_k[\Pi_U[\tilde{\zeta}(t)]]} = \gamma_{ki} \frac{dJ_k}{du_k}(\Pi_U[\tilde{\zeta}(t)]) \tag{54}$$

The argument above is the same for all k, therefore along any jointly weakly convergent subsequence of $\{\tilde{\nu}_k^\epsilon(\cdot), \tau_k^\epsilon(\cdot), \tilde{\zeta}^\epsilon(\cdot), k = 1, \dots, K\}$, the limit processes satisfy (54) and (53). For every ϵ, t the natural interpolation process satisfies $\tilde{\zeta}^\epsilon(t) = \sum_k \tilde{\nu}_k^\epsilon(t)$, therefore in the limit, along the convergent subsequence, we obtain for each component $\tilde{\zeta}_i(t) \in I\!R$ recalling (5):

$$\frac{d\tilde{\zeta}_i}{dt} = \sum_{k=1}^{K} \frac{d\nu_{ki}}{dt} = \sum_{k=1}^{K} \gamma_{ki} \frac{dJ_k}{du_k}(\Pi_U[\tilde{\zeta}(t)]) = \frac{\partial J}{\partial u_i}(\Pi_U[\tilde{\zeta}(t)])$$

and therefore (50) is satisfied for the chosen subsequence of the original weakly convergent subsequence. If this ODE has a unique solution for each initial condition, since $\zeta^\epsilon(0) = u(0)$, then this limit is independent of the chosen subsequence and therefore all subsequences have further convergent subsequences with the same limit, thus $\zeta^\epsilon(\cdot) \Rightarrow \zeta(\cdot)$. The remaining statements follow using continuity of the projection operator and stability of the ODE.

□

Remark: For each system component k, the local time scale τ_k is related to its "slower" update epochs. Working component by component allows us to apply the weak convergence method as in [16] locally. However, the only time scale which is common to all processors is the natural time scale. Therefore, we have first obtained the limiting ODE's satisfied by the natural interpolation of the artificial processes that are local to each processor in order to finally characterize the limit of the true control process $\tilde{\zeta}(t)$.

Although in the limit the equation satisfied by the natural interpolation processes in the central controller with random update times and in the fully decentralized operation are the same, the choice of $\epsilon > 0$ and the sequences $\Delta_k(l)$ are of practical importance in applications where we keep a fixed learning rate. The compound effect of $\sum_k \gamma_{ki} \frac{\partial J_k}{\partial u_k}(u)$ for u close to the optimal value induces small changes at the update epochs of the central controller. However, the decentralized control schemes act separately, thus we can expect the control process $\tilde{\zeta}^\epsilon(\cdot)$ to present more noticeable oscillations in the decentralized operation even close to the optimal value. This is indeed the case in the results shown in section 5.

We also state below two Corollaries of the Theorem that correspond to the centralized scheme with fixed and random update events of sections 3.3.1 and 3.3.2 (for details see [27]).

Corollary 5 *For the central scheme (18), under Assumptions 1 through 6, the processes $\{\zeta^\epsilon(t)\}$ converge weakly as $\epsilon \to 0$ to a solution of the ODE:*

$$\frac{d\zeta(t)}{dt} = -\nabla_u J(\Pi_U[\zeta(t)]) \tag{55}$$

Corollary 6 *For the central scheme (21), under Assumptions 1 through 6, the processes $\{\tilde{\zeta}^\epsilon(t)\}$ converge weakly as $\epsilon \to 0$ to a solution of the ODE:*

$$\frac{d\tilde{\zeta}(t)}{dt} = -\nabla_u J(\Pi_U[\tilde{\zeta}(t)]) \tag{56}$$

5 An Application

In this section, we provide an application of the optimization schemes developed and obtain experimental results for the scheduling problem introduced in section 3.1 as shown in Figure 1. We shall illustrate the use of distributed derivative estimation and compare the convergence behavior of the three control structures. Recall that in our model, K nodes compete for a single server/resource (e.g., the channel in a packet radio network). Fixed length packets arrive at node i according to an arbitrary interarrival time distribution with rate λ_i. We consider a slotted time model with slot size $\delta = 1$, where at the start of each time slot, the server is assigned to a particular node (see Figure 1). The current time slot is allocated to the ith class with probability u_i. The objective is to minimize the weighted average packet waiting time.

In the example presented here, we have simulated a model with $K = 3$, so that the constrained optimization problem is stated as:

$$\min_u \frac{1}{\sum_{j=1}^{3} \lambda_j} \sum_{i=1}^{3} \lambda_i J_i(u_i) \quad s.t. \quad \sum_{i=1}^{3} u_i \; = \; 1 \qquad \textbf{(P1)}$$

where $J_i(\cdot)$ is the average node i packet waiting time and $u = [u_1, u_2, u_3]$.

As described in section 2, we first convert **(P1)** to an unconstrained problem with $C_q = \{1, 2\}$, corresponding to the independent control variables. Due to the nature of the problem, each queue $i = 1, 2, 3$ can be modeled as a single server with deterministic service time. From the point of view of the ith queue, the server is on vacation (i.e., serving some other queue) at any one time slot with probability $(1 - u_i)$. Notice then that the average packet waiting time at queue i can be estimated locally without the need for information from the other queues, and the sensitivity with respect to u_i can also be estimated locally. In our experiments, these sensitivities were estimated via the *Phantom Slot (PS)* method of [5], based on the admission control derivative estimation of [2]. For brevity, we omit here details on the *PS* estimator. As described earlier, the estimation is done in distributed fashion. All estimators are constructed depending on the state values that can be measured locally at the epochs of service completions at the given queues. Therefore, our global event epochs are simply counting time slots. Under uniform stability, the stationary throughput of each node equals its arrival rate and the factor $\rho_k(u)$ in (10) is given by $\rho_k = \lambda_k \delta$ independent of u. It is worth noting that if the rates λ_k are unknown, it is still possible to estimate ρ_k on line.

In what follows, we implement the three control structures defined in section 3.3 to perform a single run optimization using the basic scheme (7) with fixed learning rate ϵ for the 3-node polling system, and compare their respective performance. For this example, our method ensures convergence of the control processes of our three schemes to the ODE:

$$\frac{du_1(t)}{dt} = \frac{dJ_1(u_1(t))}{du_1} - \frac{dJ_3(u_3(t))}{du_3} \tag{57}$$

$$\frac{du_2(t)}{dt} = \frac{dJ_2(u_2(t))}{du_2} - \frac{dJ_3(u_3(t))}{du_3} \tag{58}$$

$$u_3(t) = 1 - u_1(t) - u_2(t) \tag{59}$$

which, in the limit as $t \to \infty$, has an asymptotic value $u(t) \to u^*$ that satisfies the Kuhn-Tucker conditions for optimality.

In the simulations performed, we considered a network with Poisson arrivals and *symmetric* traffic. As long as the model parameters are selected to ensure that the optimal point is an interior one, this allows us to know that the optimal control vector is $u^* = [1/3, 1/3, 1/3]$. The system parameters considered are $\delta = 1, \epsilon = 10^{-6}$ and:

- Symmetric Traffic: $\lambda_1 = \lambda_2 = \lambda_3 = 0.1$
- Initial Slot Assignment: $u_1(0) = 0.5, u_2(0) = u_3(0) = 0.25$
- Central controller with Fixed Update Events: $M = 500$ time slots,
- Central controller with Random Update Events: $B = 100$ busy periods,
- Decentralized controller: $\Delta = 500$ local service completions.

1. Centralized Control with Fixed Update Events: In the centralized control scheme used here, a control update is performed by a central controller (any node can be *a priori* selected to be the controller) at update epochs $nM, n = 0, 1, \ldots$ where M is a deterministic number of slots. For simplicity, we assume that the estimation interval for each node k is identical to the controller update epochs. In our earlier notation (see section 3.3.1) this simply means that $l_k(n) = n$ and there is only one local sensitivity estimate of each node k reported over the n-th update interval. Thus, as described in section 3.3.1, at the epoch of global event nM, each node k transmits its estimate $\tilde{d}_k(n)$ to the central controller. The central controller then calculates $D_k(n, M)$ as defined in (17) with just one term in the sum, and does an update based on:

$$\begin{aligned} u_1(n+1) &= u_1(n) - \epsilon[D_1(n,M) - D_3(n,M)] \\ u_2(n+1) &= u_2(n) - \epsilon[D_2(n,M) - D_3(n,M)] \end{aligned}$$

and $u_3(n+1) = 1 - u_1(n+1) - u_2(n+1)$. Note that because of the slotted nature of the model, each node can independently recognize a global update epoch without explicit solicitation from the central controller. Following Theorem 1, our method of convergence predicts that the limit of the ladder interpolation process $\zeta(t)$ satisfies (57)-(59). Recall that for this case, the limit of the natural interpolation process satisfies $\tilde{\zeta}(Mt) = \zeta(t)$, so they are related by a time scale change.

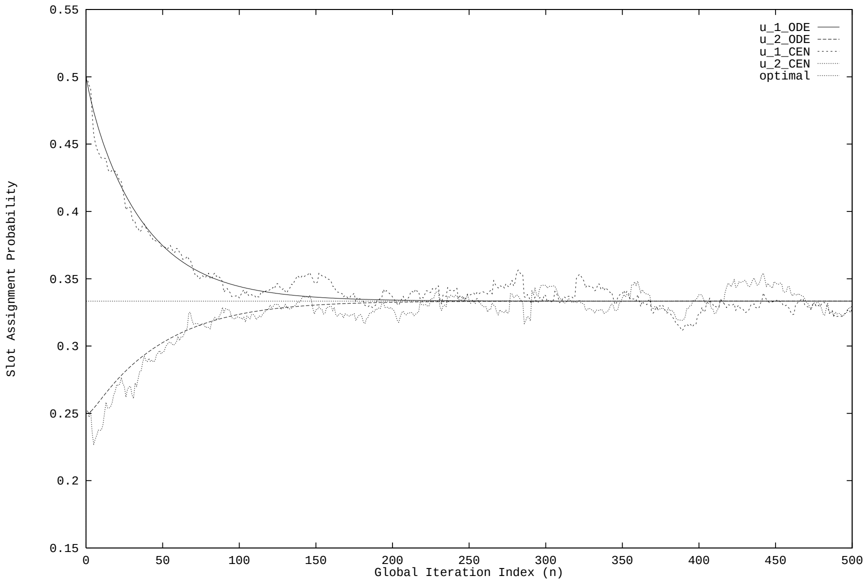

Figure 5: Comparison of Centralized Control vs. Limiting Numerical Solution to ODE

Figure 5 shows a plot of the solution of the companion ODE (57)-(59), obtained numerically via a Newton-Raphson method, and the corresponding interpolation process $\zeta^\epsilon(t)$ for the centralized control structure with fixed update events. In Figure 5 we verify that, for the given system parameters, the sequence of controls $\mathbf{u}_m^\epsilon$ not only approaches the tail of the solution to the companion ODE, but is able to accurately track the trajectory of the limiting solution. As predicted by our theoretical results, the limit processes approach this solution as $\epsilon \to 0$. In the plot, ϵ was kept fixed and it shows fluctuations around the limit ODE, as expected.

2. Centralized Control with Random Update Events: This control scheme is similar to the previous case, where now instead of performing a global update at the end of a deterministic number of time slots, the central controller counts the number of busy periods at all nodes (assuming, just for this case, that the controller can actually detect busy periods at all nodes). Let B_k be the number of busy periods at node k

within an update interval and B a given integer. Then, the controller performs an update when it observes that $B_1 + B_2 + B_3 = B$. At the end of each such estimation interval, the central controller solicits from each node k its estimate $\tilde{d}_k(n)$, with $l_k(n) = n$ as before. It then constructs $D_k(n)$ as defined in (20), and then proceeds to perform an update based on:

$$\begin{aligned} u_1(n+1) &= u_1(n) - \epsilon[D_1(n) - D_3(n)] \\ u_2(n+1) &= u_2(n) - \epsilon[D_2(n) - D_3(n)] \end{aligned}$$

and $u_3(n+1) = 1 - u_1(n+1) - u_2(n+1)$. In this case, as stated in Theorem 2, the limit of the natural interpolation process $\tilde{\zeta}(t)$ satisfies (57)-(59).

3. Decentralized Control: In the decentralized asynchronous control structure, each node k asynchronously performs a control update at the end of its local estimation interval $[L_k(l), L_k(l+1))$ $l = 0, 1, ...,$ where the interval length is given by a deterministic number Δ of service completions at node k. Since each queue is a M/D/1 server with vacations, a fixed number of local service completions yields nonetheless a random number of slots depending on u_k. Let m be the global index and let node k be the node that initiates its l-th update at event $m + 1 = G_k(l)$. Then, node k updates the ith component of the control vector according to (23) as follows (recall that $\rho_k = \lambda_k$ for $\delta = 1$ is independent of the value of the control and is known to each queue):

$$\begin{aligned} \mathbf{u}_{m+1,1} &= \mathbf{u}_{m,1} + \epsilon \frac{\gamma_{k1} \tilde{d}_k(l)}{\rho_k} \\ \mathbf{u}_{m+1,2} &= \mathbf{u}_{m,2} + \epsilon \frac{\gamma_{k2} \tilde{d}_k(l)}{\rho_k} \end{aligned}$$

and $\mathbf{u}_{m+1,3} = 1 - \mathbf{u}_{m+1,1} - \mathbf{u}_{m+1,2}$, where $\gamma_{11} = 1 = \gamma_{22}, \gamma_{12} = \gamma_{21} = 0, \gamma_{31} = \gamma_{32} = -1$ and $\tilde{d}_k(l)$ is the estimate at node k over the local interval $[L_k(l), L_k(l+1))$. Finally, the complete updated control vector $\mathbf{u}_{m+1}$ is sent to all other system components $j \neq k$. The procedure therefore updates as follows: every time node 1 (or node 2) has an estimate $\tilde{d}_1(l)$ (or $\tilde{d}_2(l)$), it adds to u_1 (or u_2) the corresponding term weighted by the factor ρ_k and adjusts u_3. When node 3 has an estimate $\tilde{d}_3(l)$, it subtracts it from both u_1 and u_2, and adjusts u_3. The compound effects, as shown in Theorem 3, yield convergence of the natural interpolation $\tilde{\zeta}^\epsilon(t)$ to the solution of (57)-(58).

In the simulation results that follow, in order to compare the convergence behavior of all three schemes on a common basis provided by the common underlying ODE associated with all schemes, time is appropriately scaled in order to plot $\tilde{\zeta}^\epsilon(t)$ in all cases. Thus, in the the central control scheme with fixed update times, we adjust the time scale by a factor M.

Figure 6 shows a plot of the slot assignment probabilities as a function of the global event index (or equivalently the simulation length in time units) for each of the update schemes, where the control parameter values are plotted at discrete sample points defined by the global event indices nM, $n = 1, 2, \ldots$. In other words, this plot shows a sample of the natural interpolation processes $\tilde{\zeta}^\epsilon(t)$. In Figure 6, as expected, we observe that the control processes of the three schemes approximate the behavior of the predicted ODE, which has the optimal value u^* as an asymptote. Recall that our results establish convergence in the distribution of the interpolated control processes to the degenerate distribution of the deterministic solution of the ODE. If we were to perform a series of simulations, each with fixed but decreasing learning rate ϵ, we would see a closer and closer fit in the corresponding plots to the smooth curve shown in Figure 5.

Moreover, the performance of the decentralized scheme seems almost identical to that under a centralized scheme. This is attributed to the discrete sampling (i.e., every nM time units) of the control processes: as expected, the decentralized version of the scheme compensates the individual updates in time, yielding a compound effect similar to the central schemes. Rather than sampling at long time intervals, if we plot the control at the actual update epoch under the given control structure, we expect to see a smoother behavior in a centralized scheme than in the decentralized scheme. In particular, Figure 7 shows a magnified comparison between the decentralized and centralized with fixed update events schemes over the time horizon indicated. As noted in the Remark at the end of the previous section, we observe a visibly oscillatory behavior under the decentralized scheme. Similar results to those seen in Figures 5-7 were obtained for different parameter settings in this model (including asymmetric traffic cases) not included here.

Finally, a few comments on the choice of ϵ are worth making. First, as discussed in the Introduction, we have chosen a fixed value of ϵ to illustrate the behavior of the three optimization schemes motivated by the need to equip them with "adaptivity" properties. In the context of simulation optimization, we can easily allow for a gradual reduction of ϵ to zero so as to eliminate the small oscillations observed around the "optimal" reference line in Figures 5-7. A problem related to the value of ϵ arises because an adjustment induced by any one of our schemes may result in an infeasible value of the probability vector (typically, a value greater than 1). In the example of this section, the value of ϵ was chosen such that feasibility and stability constraints were never violated. Clearly, there is a number of different methods to handle this problem, including various projection techniques; this is the subject of ongoing research.

6 Conclusions

We have presented and analyzed centralized and decentralized asynchronous control structures for the parametric optimization of stochastic Discrete Event Systems (DES) consisting of K distributed components. We have used a stochastic approximation type of optimization scheme driven by gradient estimates of a global performance measure with respect to local control parameters. The estimates are obtained in distributed and asynchronous fashion at the K components based on local state information only. If the conditions specified in Assumption 3 (section 3.2) for the estimators are satisfied, i.e., asymptotic unbiasedness and additivity, and some additional technical conditions hold, we have shown that two centralized optimization schemes (one with a fixed and one with a random number of events contained in the update intervals), as well as the fully decentralized asynchronous scheme, all converge to a global optimum in a weak sense. Our schemes have the additional property of using the entire state history, not just the part included in the interval since the last control update; thus, no system data are wasted. Regarding Assumption 3, the nature of the performance measure given in section 2 determines the ease or difficulty associated with the derivation and verification of asymptotic unbiasedness for our estimators. It is, therefore, of great interest to study derivative estimators for classes of problems with different characteristics, such as objective functions which do not have the additive structure considered in this paper (which imposes limited coupling over the system components) or cases where a performance measure $J_i(\cdot)$ depends on control parameters other than just u_i.

Finally, as already pointed out, the choice of learning rate ϵ and of the length of the estimation and control update intervals remains a challenging issue, and can be critical in some applications. In particular, in the presence of control constraints, it is essential to incorporate mechanisms to handle the possibility of an infeasible control value resulting from an iteration.

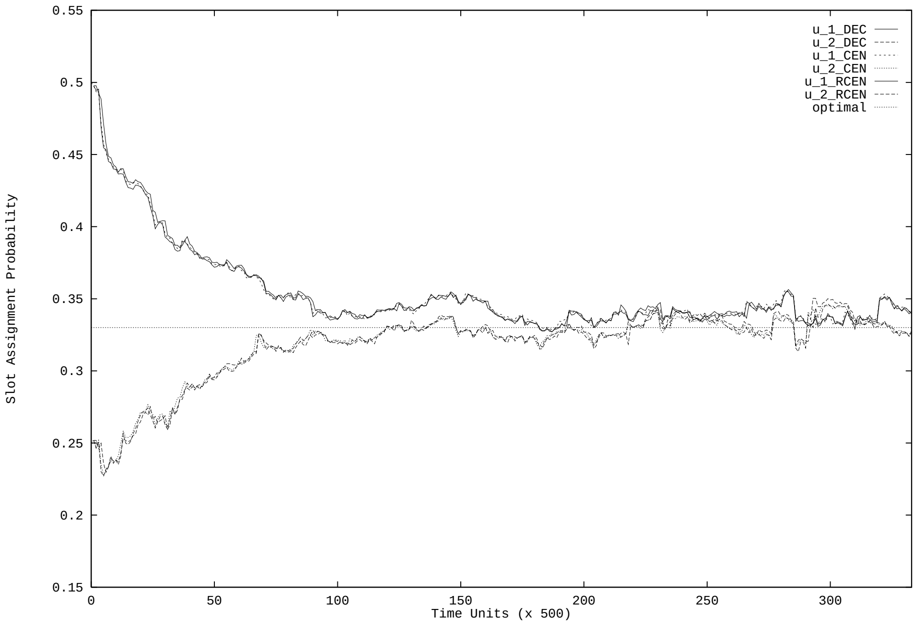

Figure 6: Comparison of Centralized vs. Decentralized Control Structure

APPENDIX

Proposition 1:

Proof : Since the sequence of initial values $\{\vartheta^\epsilon(0); \epsilon > 0\}$ is tight, using theorem 15.5 in [1] it suffices to show that for all $\nu > 0$ and $\eta > 0$, there are $\delta > 0$ and ϵ_0 such that:

$$P\left\{\sup_{t\leq T, |s|<\delta} \|\vartheta^\epsilon(t+s) - \vartheta^\epsilon(t)\| \geq \nu\right\} \leq \eta, \quad \text{forall } \epsilon \leq \epsilon_0$$

Uniform integrability of Y_n^ϵ is defined as $\sup_{n,\epsilon} \lim_{B\to\infty} E[|Y_n^\epsilon| \mathbf{1}_{\{|Y_n^\epsilon > B|\}}] = 0$, where $\mathbf{1}_{\{A\}}$ is the indicator function of the event A. Let $\nu > 0, \eta > 0$ be any given positive constants. Then for any number $B < \infty$ (to be determined later) we have:

$$\begin{aligned} P\left\{\sup_{t\leq T,|s|<\delta} \epsilon \sum_{n=\lfloor t/\epsilon\rfloor}^{\lfloor (t+s)/\epsilon\rfloor} |Y_n^\epsilon| \geq \nu\right\} \quad &\leq \quad P\left\{\sup_{t\leq T,|s|<\delta} \epsilon \sum_{n=\lfloor t/\epsilon\rfloor}^{\lfloor (t+s)/\epsilon\rfloor} |Y_n^\epsilon| \mathbf{1}_{\{|Y_n^\epsilon \leq B|\}}\right. \\ &+ \quad \left.\sup_{t\leq T,|s|<\delta} \epsilon \sum_{n=\lfloor t/\epsilon\rfloor}^{\lfloor (t+s)/\epsilon\rfloor} |Y_n^\epsilon| \mathbf{1}_{\{|Y_n^\epsilon > B|\}} \geq \frac{\nu}{2}\right\} \end{aligned}$$

For all $\epsilon < \epsilon_0 < \delta$, the number of terms in the sum is bounded by s/ϵ, and $|s| < \delta$. Since the first term in the previous expression involves the sum of random variables that are uniformly bounded by B, then for all $\delta < \nu/B$ the first term in the bracket is smaller than ν w.p.1., so that for $\delta < \nu/B$:

$$P\left\{\sup_{t\leq T,|s|<\delta} \epsilon \sum_{n=\lfloor t/\epsilon\rfloor}^{\lfloor (t+s)/\epsilon\rfloor} |Y_n^\epsilon| \geq \nu\right\} \leq P\left\{\sup_{t\leq T,|s|<\delta} \epsilon \sum_{n=\lfloor t/\epsilon\rfloor}^{\lfloor (t+s)/\epsilon\rfloor} |Y_n^\epsilon| \mathbf{1}_{\{|Y_n^\epsilon > B|\}} \geq \nu\right\}$$

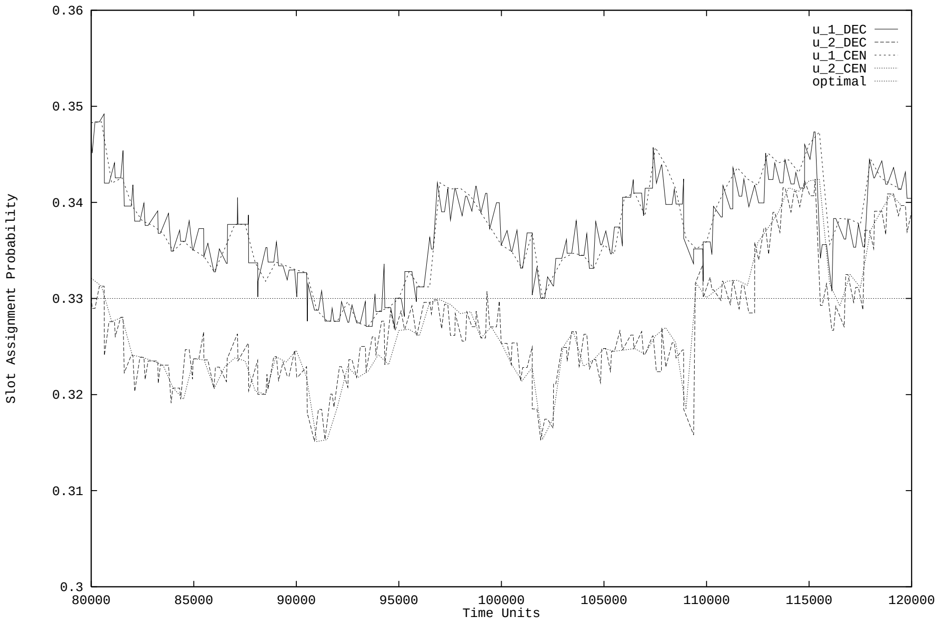

Figure 7: Comparison of Centralized vs. Decentralized Control Structure

Therefore, we must choose B, $\delta \leq \nu/B$ and $\epsilon_0 < \delta$ in order for the r.h.s. to be bounded by η for all $\epsilon \leq \epsilon_0$. Using the corollary of Theorem 8.3 in [1], we can partition $[0,T]$ into a finite number $r \approx \mathcal{O}(T/\delta)$ of subintervals whose widths are smaller than δ so that:

$$P\left\{\sup_{t\leq T,|s|<\delta} \epsilon \sum_{n=\lfloor t/\epsilon \rfloor}^{\lfloor (t+s)/\epsilon \rfloor} |Y_n^\epsilon| \mathbf{1}_{\{|Y_n^\epsilon > B|\}} \geq \nu \right\} \leq \sum_{j=0}^{r-1} P\left\{\sup_{|s|<\delta} \epsilon \sum_{n=\lfloor j\delta/\epsilon \rfloor}^{\lfloor (j\delta+s)/\epsilon \rfloor} |Y_n^\epsilon| \mathbf{1}_{\{|Y_n^\epsilon > B|\}} \geq \frac{\nu}{3}\right\}$$

$$\leq \sum_{j=0}^{\lfloor T/\delta \rfloor} P\left\{\epsilon \sum_{n=\lfloor j\delta/\epsilon \rfloor}^{\lfloor (j+1)\delta/\epsilon \rfloor} |Y_n^\epsilon| \mathbf{1}_{\{|Y_n^\epsilon > B|\}} \geq \frac{\nu}{3}\right\} \leq \frac{3}{\nu} \sum_{j=0}^{\lfloor T/\delta \rfloor} E\left\{\sup_{|s|<\delta} \epsilon \sum_{n=\lfloor j\delta/\epsilon \rfloor}^{\lfloor (j+1)\delta/\epsilon \rfloor} |Y_n^\epsilon| \mathbf{1}_{\{|Y_n^\epsilon > B|\}}\right\}$$

where we have used Markov's inequality in the last step. From the uniform integrability of Y_n^ϵ, for all $\eta' > 0$ there is a constant B such that $E|Y_n^\epsilon| \mathbf{1}_{\{|Y_n^\epsilon| > B\}} \leq \eta'$. Choose B so that $E|Y_n^\epsilon| \mathbf{1}_{\{|Y_n^\epsilon| > B\}} \leq \eta(\nu/3T)$. Then choose $\delta \leq \nu/B$. Since the number of terms in the inner sum is bounded by δ/ϵ, then we finally obtain that:

$$P\left\{\sup_{t\leq T,|s|<\delta} \epsilon \sum_{n=\lfloor t/\epsilon \rfloor}^{\lfloor (t+s)/\epsilon \rfloor} |Y_n^\epsilon| \geq \nu \right\} \leq \frac{3}{\nu}\left(\frac{T}{\delta}\right) \epsilon \times \left(\frac{\delta}{\epsilon}\right) \eta \frac{\nu}{3T} = \eta$$

which proves the assertion.

$\square$

Proposition 2:

Proof : From the definition of the indices l_ϵ and n_ϵ, if $l_\epsilon n_\epsilon \leq n < (l_\epsilon + 1)n_\epsilon$ we have $|\epsilon n - s| \leq \delta_\epsilon$. By the definition of the ladder interpolation process we have for all such n that:

$$\begin{aligned} \|w^\epsilon(n+1) - w^\epsilon(n)\| &= \|\zeta^\epsilon(\epsilon n + \epsilon) - \zeta^\epsilon(\epsilon n)\| \\ &\leq \|\zeta^\epsilon(\epsilon n + \epsilon) - \zeta(\epsilon n + \epsilon)\| + \|\zeta(\epsilon n + \epsilon) - \zeta(\epsilon n)\| + \|\zeta(\epsilon n) - \zeta^\epsilon(\epsilon n)\| \end{aligned}$$

Using Theorem 2.3 in [16], we can use Skorohod imbedding to change the underlying probability space and assume w.l.o.g. and invoking Corollary 1 that the chosen subsequence $\zeta^\epsilon(\cdot) \to \zeta(\cdot)$ w.p.1. For the first and third terms, we use a.s. convergence of $\zeta^\epsilon(\cdot)$ and for the term in the middle, the Lipschitz continuity of the limit function, to conclude that the values of $w^\epsilon(n)$ lie in a compact set w.p.1, for all $n \in [l_\epsilon n_\epsilon, (l_\epsilon+1)n_\epsilon)$. That is, along this subsequence there is an ϵ_0 such that $P\{\|w^\epsilon(n+1) - w^\epsilon(n)\| > B(\epsilon)|w^\epsilon(n)\} = \mathcal{O}(\epsilon)$ where $0 \le B(\epsilon) = \mathcal{O}(\epsilon)$, for all $\epsilon \le \epsilon_0$.

Let $f(x,u)$ be an arbitrary bounded and continuous function. By time homogeneity of the MDP, for $n \ge l_\epsilon n_\epsilon$, if $m \equiv n - l_\epsilon n_\epsilon$, we have:

$$E\{f(\chi^\epsilon(n), w^\epsilon(n))|\chi^\epsilon(l_\epsilon n_\epsilon) = x_0, w^\epsilon(l_\epsilon n_\epsilon) = u_0\} = E\{f(\chi^\epsilon(m), w^\epsilon(m)|\chi^\epsilon(0) = x_0, w^\epsilon(0) = u_0\} \tag{60}$$

Let

$$P^\epsilon_m(dx, du; x_0, w_0) = P\{\chi^\epsilon(m) \in dx, w^\epsilon(m) \in du|\chi^\epsilon(0) = x_0, w^\epsilon(0) = u_0\}$$

denote the m-step joint transition measure of the random sample process $(\chi^\epsilon(n), w^\epsilon(n))$. Similarly, let:

$$P_m(dx; x_0, u_0) = P\{\chi(m) \in dx|\chi(0) = x_0\}$$

denote the m-step transition measure of the process $\{\chi_m\} = \{\xi_{S(n)}(u_0)\}$ when the control is fixed at the value u_0.

For this function f and $0 \le m < n_\epsilon$, define:

$$A_m(x_0, u_0) = \int P^\epsilon_m(dx, du; x_0, w_0) f(x,u) = E\{f(\chi^\epsilon(m), w^\epsilon(m)|\chi^\epsilon(0) = x_0, w^\epsilon(0) = u_0\} \tag{61}$$

Then, we have:

$$A_m(x_0, u_0) = \int_{(y,v)} P^\epsilon_1(dy, dv; x_0, u_0) \int_{(x,u)} P^\epsilon_{m-1}(dx, du; y, v) f(x,u) = \int_{(x,y)} P^\epsilon_1(dy, dv; x_0, u_0) A_{m-1}(y, v)$$

Since f is bounded and continuous, it follows from the weak continuity of $P_u(\cdot)$ in Assumption 1 that $A_{m-1}(y,v)$ is a continuous and bounded function of v, and for $\|v - u_0\| \le B(\epsilon)$, $\|A_{m-1}(y,v) - A_{m-1}(y,u_0)\| = \mathcal{O}(\epsilon)$, thus:

$$A_m(x_0, u_0) = \int_{(y,v)} P^\epsilon_1(dy, dv; x_0, u_0) A_{m-1}(y, u_0) + \mathcal{O}(\epsilon)$$

where the integrand $A_{m-1}(y, u_0)$ is independent of v, therefore, integrating over v and using assumption (33), we have:

$$A_m(x_0, u_0) = \int_y P_1(dy; x_0, u_0) A_{m-1}(y, u_0) + R_m(\epsilon)$$

where the remainder term satisfies $|R_m(\epsilon)| = \sum_{k=1}^m R_k(\epsilon) = \mathcal{O}(\epsilon)$. Proceeding by induction, we obtain:

$$\begin{aligned}
A_m(x_0, u_0) &= \int P_1(dy; x_0, u_0) A_{m-1}(y, u_0) + R_m(\epsilon) \\
&= \int_y P_1(dy; x_0, u_0) \int_w P_1(dw; y, u_0) A_{m-2}(w, u_0) + R_m(\epsilon) + R_{m-1}(\epsilon) \\
&= \int P_2(dy; x_0, u_0) A_{m-2}(y, u_0) + R_m(\epsilon) + R_{m-1}(\epsilon) \\
&= \sum_{k=1}^m R_k(\epsilon) + \int P_m(dx; x_0, u_0) f(x, u_0)
\end{aligned}$$

Therefore,

$$\frac{1}{n_\epsilon}\sum_{m=0}^{n_\epsilon-1} A_m(x_0,u_0) = \frac{1}{n_\epsilon}\sum_{m=0}^{n_\epsilon-1}\mathcal{R}_m(\epsilon) + \frac{1}{n_\epsilon}\sum_{m=0}^{n_\epsilon-1} E_{u_0}\{f(\chi^\epsilon(m),u_0)|\chi^\epsilon(0)=x_0\}$$

Combining (60) and (61) and recalling the definitions in (31) and (32), this equation becomes

$$\bar{f}^\epsilon(\chi^\epsilon(l_\epsilon n_\epsilon), w^\epsilon(l_\epsilon n_\epsilon)) = \frac{1}{n_\epsilon}\sum_{m=0}^{n_\epsilon-1}\mathcal{R}_m(\epsilon) + \hat{f}^\epsilon(\chi^\epsilon(l_\epsilon n_\epsilon), w^\epsilon(l_\epsilon n_\epsilon))$$

Since $|\mathcal{R}_m| = \mathcal{O}(m\epsilon)$, the remainder term in the average above is of the order $\mathcal{O}(\epsilon n_\epsilon) = \mathcal{O}(\delta_\epsilon) \to 0$ as $\epsilon \to 0$ along the chosen subsequence. It follows that

$$\lim_{\epsilon\to 0} E\left\|\bar{f}^\epsilon(\chi^\epsilon(l_\epsilon n_\epsilon), w^\epsilon(l_\epsilon n_\epsilon)) - \hat{f}^\epsilon(x_0,u_0)\right\| = 0$$

which establishes (34).

Next, by Assumption 5 the sequence $\{\chi^\epsilon(l_\epsilon n_\epsilon), w^\epsilon(l_\epsilon n_\epsilon)\}$ is tight and therefore every sequence (and in particular the chosen one) has a further subsequence such that its joint distribution $P^\epsilon(dx,du)$ converges as $\epsilon \to 0$ to $P(dx,du)$. Since $w^\epsilon(l_\epsilon n_\epsilon) = \zeta^\epsilon(s) \to \zeta(s)$, the distribution of $w^\epsilon(l_\epsilon n_\epsilon)$ converges to that of the limiting random variable $\zeta(s)$ along all such subsequences.

Given any value of $\zeta(s) = u$, set:

$$\hat{f}(x,u) = \lim_{n\to\infty}\frac{1}{n}\sum_{m=0}^{n-1} E_u\{f(\chi^\epsilon(m))|\chi^\epsilon(0)=x\} \tag{62}$$

therefore: $\hat{f}^\epsilon(x,u) \to \hat{f}(x,u)$ as $\epsilon \to 0$, where (x,u) have the limiting distribution $P(dx,du)$.

Under the ergodicity in Assumption 1, since the invariant measure of the fixed-u process exists, then the Cesaro sum in $\hat{f}(x,u)$ satisfies:

$$\hat{f}(x,u) = \int \mu_u(dx) f(x,u)$$

where $\mu_u(s)$-a.s. for every $x$and is independent of x. By the fact that the limit is independent of the chosen subsequence of $P^\epsilon(dx,du)$, (35) is satisfied.

□

Lemma 2 *For the decentralized control structure, under Assumptions 1 through 6, for any weakly convergent subsequence of the process $\tilde{\zeta}^\epsilon(\cdot) \Rightarrow \tilde{\zeta}^\epsilon(\cdot)$, (33) is satisfied for any continuous and bounded function $F(x)$.*

Proof : Call A_k the subset of the state space such that if $\xi_m^\epsilon \in A_k$ then m is a local update epoch for processor k (see section 3.2.2), that is: $P\{\sum_{l=1}^\infty \mathbf{1}_{\{S^\epsilon(l)=m\}}|\xi_m^\epsilon \in A_k\} = 1$ and $P\{\sum_{l=1}^\infty \mathbf{1}_{\{S^\epsilon(l)=m\}}|\xi_m^\epsilon \notin A_k\} = 0$. Let $F(x)$ be any bounded and continuous function defined on $x \in A_k$. Then, using time homogeneity of the MDP and (27), we have for any $x \in A_k$:

$$\begin{aligned}
&E\{F(\chi^\epsilon(n+1))|\chi^\epsilon(n) = x_0, w^\epsilon(n) = u_0\} = \\
&= \sum_{m\geq 1} P\{S^\epsilon(1) = m|\chi^\epsilon(0) = x, w^\epsilon(0) = u\} E\{F(\xi_m^\epsilon)|\xi^\epsilon(0) = x, \mathbf{u}^\epsilon(0) = u\} \\
&= \int_{x_1\in A_k} F(x_1) P\{\xi_1^\epsilon \in dx_1|\xi_0^\epsilon = x_0, \mathbf{u}_0^\epsilon = u_0\}
\end{aligned}$$

$$+\int_{x_2\in A_k}\int_{u_1}F(x_2)\int_{x_1\notin A_k}P\{\xi_2^\epsilon\in dx_2|\xi_1^\epsilon=x_1,\mathbf{u}_1^\epsilon=u_1\}P\{\xi_1^\epsilon\in dx_1,\mathbf{u}_1^\epsilon\in du_1|\xi^\epsilon(0)=x_0,\mathbf{u}^\epsilon(0)=u_0\}+\dots$$

$$+\int_{u_1,\dots,u_m}\int_{x_m\in A_k}\int_{x_{m-1},\dots x_1\notin A_k}F(x_m)P\{\xi_m^\epsilon\in dx_m|\xi_{m-1}^\epsilon=x_{m-1},\mathbf{u}_{m-1}^\epsilon=u_{m-1}\}\times$$
$$\prod_{j=1}^{m-1}P\{\xi_j^\epsilon\in dx_j,\mathbf{u}_j^\epsilon\in du_j|\xi_{j-1}^\epsilon=x_{j-1},\mathbf{u}_{j-1}^\epsilon=u_{j-1}\}+\dots$$

$$=\sum_{m\geq 1}\int_{u_1,\dots,u_m}\int_{x_m\in A_k}\int_{x_{m-1},\dots x_1\notin A_k}F(x_m)P_{u_{m-1}}(x_{m-1},dx_m)\times$$
$$\prod_{j=1}^{m-1}P\{\xi_j^\epsilon\in dx_j,\mathbf{u}_j^\epsilon\in du_j|\xi_{j-1}^\epsilon=x_{j-1},\mathbf{u}_{j-1}^\epsilon=u_{j-1}\}$$

$$=\sum_{m\geq 1}\int_{u_1,\dots,u_m}\int_{x_m\in A_k}\int_{x_{m-1},\dots x_1\notin A_k}F(x_m)P_{u_{m-1}}(x_{m-1},dx_m)\times$$
$$\prod_{j=1}^{m-1}P_{u_{j-1}}(x_{j-1},dx_j)P\{\mathbf{u}_j^\epsilon\in du_j|\xi_{j-1}^\epsilon=x_{j-1},\mathbf{u}_{j-1}^\epsilon=u_{j-1}\}$$

We use now the fact that along the convergent subsequences, under Skorohod representation $\tilde{\zeta}^\epsilon(\cdot)\to\tilde{\zeta}(\cdot)$ a.s., and $P\{\|\mathbf{u}_j^\epsilon-u_{j-1}\|<B(\epsilon)|\mathbf{u}_{j-1}^\epsilon=u_{j-1}\}=\mathcal{O}(\epsilon)$ as $\epsilon\to 0$ along this subsequence. The transition probability $P_u(x,B)$ is weakly continuous in (x,u). Therefore for any bounded and continuous function $f(x)$, for v such that $\|v-u\|\leq B(\epsilon)$,

$$\int_B f(y)P_v(x,dy)=\int_B f(y)P_u(x,dy)+\rho(\epsilon)P_u(x,B)=\int_B(f(y)+\rho(\epsilon))P_u(x,dy)$$

where $|\rho(\epsilon)|\leq K\epsilon$ and K depends on the bound of f. Therefore, for the term $m=2$ above, we can replace u_1 by u_0 to get:

$$\int_{x_2\in A_k}\int_{u_1}F(x_2)\int_{x_1\notin A_k}P_{u_1}(x_1,dx_2)P_{u_0}(x_0,dx_1)P\{\mathbf{u}_1^\epsilon\in du_1|\xi_0^\epsilon=x_0,\mathbf{u}_0^\epsilon=u_0\}$$
$$=\int_{x_2\in A_k}\int_{u_1}\int_{x_1\notin A_k}[F(x_2)+\rho(\epsilon)]P_{u_0}(x_1,dx_2)P_{u_0}(x_0,dx_1)P\{\mathbf{u}_1^\epsilon\in du_1|\xi_0^\epsilon=x_0,\mathbf{u}_0^\epsilon=u_0\}$$
$$=\int_{x_2\in A_k}\int_{x_1\notin A_k}[F(x_2)+\rho(\epsilon)]P_{u_0}(x_1,dx_2)P_{u_0}(x_0,dx_1)$$

where we have integrated over u_1 in the last step. Proceeding in the same manner, for $m=3$ we replace first u_2 by u_1 and then u_1 by u_0 to obtain:

$$\int_{x_3\in A_k}\int_{u_1,u_2}F(x_3)\int_{x_2,x_1\notin A_k}P_{u_2}(x_2,dx_3)P_{u_1}(x_1,dx_2)P_{u_0}(x_0,dx_1)\times$$
$$P\{\mathbf{u}_2^\epsilon\in du_2|\xi_1^\epsilon=x_1,\mathbf{u}_1^\epsilon=u_1\}P\{\mathbf{u}_1^\epsilon\in du_1|\xi_0^\epsilon=x_0,\mathbf{u}_0^\epsilon=u_0\}$$
$$=\int_{x_3\in A_k}\int_{u_1}\int_{x_2,x_1\notin A_k}[F(x_3)+\rho(\epsilon)]P_{u_1}(x_2,dx_3)P_{u_1}(x_1,dx_2)P_{u_0}(x_0,dx_1)\times$$
$$P\{\mathbf{u}_1^\epsilon\in du_1|\xi_0^\epsilon=x_0,\mathbf{u}_0^\epsilon=u_0\}$$
$$=\int_{x_3\in A_k}\int_{u_1}\int_{x_2,x_1\notin A_k}[F(x_3)+\rho(\epsilon)]P_{u_1}\{\xi_3^\epsilon\in dx_3|\xi_1^\epsilon=x_1\}P_{u_0}(x_0,dx_1)P\{\mathbf{u}_1^\epsilon\in du_1|\xi_0^\epsilon=x_0,\mathbf{u}_0^\epsilon=u_0\}$$

$$= \int_{x_3 \in A_k} \int_{x_2, x_1 \notin A_k} [F(x_3) + 2\rho(\epsilon)] P_{u_0}\{\xi_3^\epsilon \in dx_3 | \xi_1^\epsilon = x_1\} P_{u_0}\{\xi_1^\epsilon \in dx_1 | \xi^\epsilon(0) = x_0\}$$

$$= \int_{x_3 \in A_k} \int_{x_2, x_1 \notin A_k} F(x_3) P_{u_0}\{\xi_3^\epsilon \in dx_3 | \xi_0^\epsilon = x_0\} + 2 \int_{x_3 \in A_k} \int_{x_2, x_1 \notin A_k} \rho(\epsilon) P_{u_0}\{\xi_3^\epsilon \in dx_3 | \xi_0^\epsilon = x_0\}$$

Proceeding by induction, we have:

$$\begin{aligned} E\{F(\chi^\epsilon(n+1)) \quad |\chi^\epsilon(n) = x_0, w^\epsilon(n) = u_0\} &= \sum_{m \geq 1} \int_{x_m \in A_k} \int_{x_{m-1}, \ldots, x_1 \notin A_k} F(x_m) P_{u_0}\{\xi_m^\epsilon \in dx_m | \xi_0^\epsilon = x_0\} \\ &\quad + \sum_{m \geq 1} \int_{x_{m-1}, \ldots, x_1 \notin A_k} (m-1)\rho(\epsilon) P_{u_0}\{\xi_m^\epsilon \in dx_m | \xi_0^\epsilon = x_0\} \\ &= \sum_{m \geq 1} P_{u_0}\{S^\epsilon(1) = m | \xi_0^\epsilon = x_0\} E_{u_0}\{F(\xi_m^\epsilon) | \xi_0^\epsilon = x_0\} \\ &\quad + \sum_{m \geq 1} (m-1)\rho(\epsilon) P_{u_0}\{S^\epsilon(1) = m | \xi_0^\epsilon = x_0\} \\ &= E_{u_0}\{F(\chi^\epsilon(n+1)) | \chi^\epsilon(0) = x_0\} + [\bar{M}(u_0) - 1]\rho(\epsilon) \end{aligned}$$

which shows (33).

□

References

[1] P. Billingsley. *Convergence of Probability Measures.* John Wiley, New York, 1968.

[2] P. Brémaud and F.J. Vázquez-Abad, 'On the Pathwise Computation of Derivatives with Respect to the Rate of a Point Process: The Phantom RPA Method,' *Queueing Systems: Theory and Applications*, vol. 10, pp. 249–270, 1992

[3] C.G. Cassandras, *Discrete Event Systems: Modeling and Performance Analysis*, Irwin Publishers, 1993.

[4] C.G. Cassandras, M.V. Abidi, and D. Towsley, 'Distributed Routing with On-line Marginal Delay Estimation,' *IEEE Transactions on Communications*, vol. COM-38, no. 3, pp. 348-359, 1990.

[5] C.G. Cassandras and V. Julka, 'Scheduling Policies Using Marked/Phantom Slot Algorithms,' *Queueing Systems: Theory and Applications*, vol. 20, pp. 207-254, 1995..

[6] E. K. P. Chong and P. J. Ramadge, 'Convergence of Recursive Optimization Algorithms using IPA Derivative Estimates,' *Journal of Discrete Event Dynamics Systems*, vol. 1, pp. 339-372, 1992.

[7] E. K. P. Chong and P. J. Ramadge, 'Optimization of Queues using an Infinitesimal Perturbation Analysis-based Stochastic Algorithm with General Update Times,' *SIAM J. on Control and Optimization*, vol 31, pp. 698-732, 1993.

[8] M. C. Fu. 'Convergence of the stochastic approximation algorithm for the GI/G/1 queue using infinitesimal perturbation analysis,' *J. of Optimization Theory and Applications*, no 65, pp. 149–160, 1990.

[9] R.G. Gallager, 'A Minimum Delay Routing Algorithm Using Distributed Computation,' *IEEE Transactions on Communications*, vol. COM-25, pp. 73-85, 1977.

[10] P. Glasserman, *'Gradient Estimation via Perturbation Analysis'*, Kluwer Academic Publishers, Boston, 1991.

[11] P. Glynn, 'Likelihood Ratio Gradient Estimation: An Overview,' *Proceedings of the 1987 Winter Simulation Conference*, pp. 336-375, 1987.

[12] P. Glynn, 'A GSMP Formalism for Discrete Event Systems,' *Proceedings of IEEE*, pp. 14-23, 1989.

[13] Y.C. Ho and X. Cao, *Perturbation Analysis of Discrete Event Dynamic Systems*, Kluwer Academic Publishers, Boston, 1991.

[14] J. Kiefer and J. Wolfowitz, 'Stochastic Estimation of the Maximum of a Regression Function,' *Annals of Mathematical Statistics*, 23, pp. 462-466, 1952.

[15] H.J. Kushner and D.S. Clark. *Stochastic Approximation for Constrained and Unconstrained Systems*, Springer Verlag, Berlin and New York, 1978.

[16] H.J. Kushner, *Approximation and Weak Convergence Methods for Random Processes with Applications to Stochastic System Theory*, MIT Press, Cambridge, MA, 1984.

[17] H.J. Kushner and F.J. Vázquez-Abad, 'Stochastic Approximation Methods for Systems of Interest Over an Infinite Horizon,' *SIAM J. on Control and Optim.*, vol 34, 2, pp. 712-756, 1996. *to appear in SIAM J. on Control and Optimization.*

[18] H.J. Kushner and A. Shwartz. 'Weak convergence and asymptotic properties of adaptive filters with constant gains.' *IEEE Trans. on Information Theory*, vol IT-30, pp. 177–182, 1984.

[19] H.J. Kushner and G. Yin. 'Stochastic approximation algorithms for parallel and distributed processing.' *Stochastics*, vol 22, pp. 219–250, 1987.

[20] M. Reiman and A. Weiss, 'Sensitivity Analysis for Simulations via Likelihood Ratios,' *Operations Research*, vol. 37, pp. 830-844, 1989.

[21] H. Robbins and S. Monro, 'A Stochastic Approximation Method,' *Annals of Mathematical Statistics*, 22, pp. 400-407, 1951.

[22] A. Segall, 'The Modeling of Adaptive Routing in Data Communication Networks,' *IEEE Transactions on Communications*, vol. COM-25, pp. 85-95, 1977.

[23] J.N. Tsitsiklis and D.P. Bertsekas, 'Distributed Asynchronous Optimal Routing in Data Networks,' *IEEE Transactions on Autom. Control*, vol. AC-31, 4, pp. 325-332, 1986.

[24] F.J Vázquez-Abad, 'Stochastic Recursive Algorithms for Optimal Routing in Queueing Networks,' PhD thesis, Brown University, 1989.

[25] F.J Vázquez-Abad and K. Davis, 'Strong Points of Weak Convergence: A Study Using RPA Gradient Estimation for Automatic Learning', submitted to *Automatica*, also Publication 1025, Département d'iro, Umonersité de Montréal, April 1996.

[26] F.J. Vázquez-Abad and L.G. Mason, 'Adaptive Decentralized Control Under Non-uniqueness of the Optimal Control' submitted to *J DEDS.*

[27] F.J. Vázquez-Abad, C.G. Cassandras, and V. Julka, 'Centralized and Decentralized Asynchronous Optimization of Stochastic Discrete Event Systems' Dept. of Manufacturing Engineering Technical Report, Boston University, 1997.

[28] H. Yan, G. Yin, and S.X.C. Lou. 'Using stochastic approximation to determine threshold values for control of unreliable manufacturing systems.' *J. Optimization Theory and Applications*, vol. 83, pp. 511-539, 1994.