12

Distributed/Decentralized and Asynchronous Algorithms

12.0 Outline of Chapter

This chapter is concerned with decentralized and asynchronous forms of the stochastic approximation algorithms, a relatively new area of research. Compared with the rapid progress and extensive literature in stochastic approximation methods, the study of parallel stochastic approximations is still in its infancy. Perhaps the first work on the subject was [14], which dealt with a very particular class of algorithms, where similar computations were done by several processors and convex combinations were taken. The general ideas of weak convergence theory were applied to a fairly broad class of realistic algorithms in [111, 112]. The general ideas presented there and in [105, 171] form the basis of this chapter. Analogously to the problems in Chapter 8, those methods can handle correlated and state dependent noise, delays in communication, and asynchronous and distributed network forms. Various examples are given in Section 1. In the basic model, there are several processors; each one is responsible for the updating of only a part of the parameter vector. There might be overlaps in that several processors contribute to the updating of the same component of the parameter. Such models were treated in [105, 171, 172]. For a similar model, the problem of finding zeros of a nonlinear function with noisy observations via parallel processing methods and with random truncation bounds was treated in [201]. An attempt to get real-time implementable procedures via pipelining (see Section 1.2) of communication and computation for algorithms with "delayed" observations was in [203]. Pipelining is a potentially useful method for exploiting multiprocessors to accelerate convergence. Suppose that the system observes raw physical data, and then the value of Y_n is computed from that raw data. The actual time required for the computation of Y_n from the raw data might be relatively long in comparison with the potential rate at which the raw data can be made available. Then the computation can be sped up via pipelining. This is discussed further in Section 1.

A survey of some recent developments can be found in [194]. Reviews of decentralized and synchronous algorithms for traditional numerical analysis problems can be found in [129], and surveys and literature citations for asynchronous (almost entirely) deterministic problems are in the books [13] and [146].

In some applications, the input data are physically distributed; i.e., the underlying computation might use physically separated processors, estimators, trackers, controllers, and so on. For example, in the optimization of queueing networks, each node might be responsible for updating the "local" routing or service speed parameters. In such problems, there are generally communication delays in transferring information from one node to another. This might be because of the physical time required and the fact that different paths are used by each processor, and the path used by each processor is time dependent and random, or it might be due to the time required by other (perhaps higher-priority) demands placed on the network. In other applications, the time required to get the data needed to update a component of the parameter might be random, and we might not have complete control over which component is to be updated next. An example is the Q-learning problem in Chapter 2.

By means of parallel processing, one may be able to take advantage of state space decomposition. As a result, a large dimensional system may be split into small pieces so that each subsystem can be handled by one of the parallel processors.

The general problem of distributed algorithms can be treated by the methods of the previous chapters, provided they are synchronized or nearly synchronized, even if there are modest delays in the communication between processors. The problem becomes more difficult if the processors are not synchronized. This is particularly important if each processor takes a random length of time to complete its work on a single update but is able to continue processing updates for its set of parameters even if it does not have the latest results from the other processors. Then, in any real-time interval each processor completes a different (random) number of iterations. Because of this, one can no longer use the iterate number $n = 0, 1, \ldots,$ as a clock (time) to study the asymptotic properties. One needs to work in real time, or at least in an appropriately scaled real time.

Suppose there is only one processor (hence no synchronization problems), but that the actual computation of each update takes a random amount of time. This random time is irrelevant to the convergence, which is only concerned with what happens as the number of iterates goes to infinity. Thus, the previous chapters worked with what might be called "iterate time." To introduce the idea of working in real time, this problem is redone in Section 2 in a scaled real time, so that the limit mean ODE follows the progress not as the iterate number increases, but as real time increases. It will be seen that the analysis is exactly the same except for a simple time transformation, and that the limit mean ODE is just as before, except for a scale factor that accounts for the average rate at which the iterations progress in time. Analogously to the interpretation of the function $\bar{g}(\theta)$ in the ODE as a mean of the iterate values when the parameter is fixed at θ, the time is scaled at θ by the mean rate of updating at that parameter value. If the basic algorithm is of the gradient descent type, then the stable points are the same as for the centralized algorithm.

We concentrate on the weak convergence proofs, although analogs of the probability one results of Chapters 5 and 6 can also be proved. The proofs are actually only mild variations on those of Chapter 8. The notation might get involved at times, but it is constructed for versatility in the applications and so that the problem can be set up such that the proofs in Chapter 8 can be used, with only a time rescaling needed at the end to get the desired results. Section 3 deals with the constant step size algorithm, and the decreasing step size algorithm is treated in Section 4. Section 5 is concerned with the state dependent noise case and is based on the method of Section 8.4. The proofs of the rate of convergence and the stability methods in Sections 6 and 7 follow the lines of development of the analogous sections in Chapters 8 and 10, but work in "real time" rather than iterate time. Iterate averaging such as that used in Chapter 11 has the same advantages here but is omitted for lack of space. Finally, we return to the Q-learning problem of Section 2.3, and we use the differential inclusions form of the main theorem in Section 3 to readily prove the convergence to the optimal Q-values.

12.1 Examples

12.1.1 Introductory Comments

In the Q-learning example of Section 2.3, the parameter θ is the collection of Q-values, called $\{Q_{i\alpha}; i, \alpha\}$ there. While that algorithm is usually implemented in a centralized way and only one component is updated at each state transition, the algorithm is certainly asynchronous in the sense that the component updated at any time is random and the time between updates of any particular component is random. The methods of this chapter will be seen to be appropriate for that problem.

It is easy to construct examples of asynchronous algorithms by simply splitting the work among several processors and allowing them to function asynchronously. For example, in the animal learning example of Section 2.1, there might be several parameters updated at rates that are different and that depend on different aspects of the hunting experience. In the neural net training example of Section 2.2, the actual computation of the derivatives might be quite time-consuming relative to the rate at which new training inputs can be applied and the corresponding outputs measured. This suggests that the algorithm can be improved by "pipelining" in either a synchronized or an asynchronized way (see the example in the next subsection). The relative difficulty of the computation in comparison with the actual speed at which the network can be used suggests a general class of problems where pipelining can be useful. In many applications, one takes a "raw" observation, such as the input and the associated actual and desired outputs in the neural network training procedure. Then a time-consuming computation is done to get the actual Y_n or Y_n^ϵ, which is used in the stochastic approximation. A similar situation holds for the queue optimization problem of Section 2.5. In that case, the raw data are simply the sequences of arrival and service times. The computation of the pathwise derivative might take a (relatively) long time, particularly if the interarrival and service times are very small, the derivatives of the "inverse functions" which are often involved in getting pathwise derivatives are hard to compute, or the processors used must be shared (with perhaps a lower priority) with other tasks.

It is worth emphasizing that asynchronous algorithms commonly arise when the processing is distributed and some of the processors are interrupted at random times to do other work.

The pipelined algorithm described in the next subsection is not asynchronous. It is distributed and uses the notion of "concurrency," which is central to current work in distributed processing. If each processor takes a random length of time for its computation, one might consider the use of an asynchronous form of the pipelined algorithm, and then the methods of this chapter can be applied. In the synchronized case, the proofs of convergence of the pipelined algorithms can be handled by the machinery of Chapters 5–8. Subsection 2.3 concerns the optimization of a queueing network, where the various nodes of the network participate asynchronously in the computation. The example in Section 2.4 arises in telecommunications and is characteristic of a large class of problems. These few examples are illustrative of an ever-expanding set of possibilities.

12.1.2 Pipelined Computations

This example illustrates the usefulness of multiprocessors in stochastic approximation and is intended to be a simple illustration of the pipelining method. Consider the stochastic approximation algorithm as described in the previous chapters, with constant step size ϵ:

$$\theta_{n+1}^\epsilon = \Pi_H\left(\theta_n^\epsilon + \epsilon Y_n^\epsilon\right) = \theta_n^\epsilon + \epsilon Y_n^\epsilon + \epsilon Z_n^\epsilon. \tag{1.1}$$

The algorithm can often be thought of in terms of the following two-phase computation. In Phase 1, we evaluate the Y_n^ϵ from the raw observational data, and in Phase 2, we update the parameter by adding ϵY_n^ϵ to the current value of θ_n^ϵ (with a projection or a reflection term if needed) and observe new raw data.

Generally, most of the computation time is spent in Phase 1. Frequently, Phase 1 consists in doing an extensive computation based on physical measurements. Taking this fact into account, consider an alternative algorithm where several processors are lined up as on a production line and update interactively. After each iteration is completed, the new value of the parameter is passed to the next processor in the line.

For a concrete illustration, suppose that the time required for computing Y_n^ϵ is three units of time, after which θ_n^ϵ and ϵY_n^ϵ are added together and a new measurement is taken, where the addition and the taking of the new raw measurement requires one unit of time. Instead of using a single processor as in (1.1), where each iteration requires four units of time, four parallel processors will be used in a pipeline fashion. All processors perform the same kind of computation, but with different starting times. At a given instance n, suppose that Processor 1 has completed its evaluation based on physical data observed three units of time ago when the parameter value was θ_{n-3}^ϵ. In other words, it has computed Y_{n-3}^ϵ. Then, at time $n+1$, Processor 1 computes

$$\theta_{n+1}^\epsilon = \Pi_H(\theta_n^\epsilon + \epsilon Y_{n-3}^\epsilon),$$

where θ_n^ϵ is the current value of the parameter (actually just provided by Processor 4). Meanwhile Processor 2 is evaluating the increment based on physical measurements taken at time $n-2$, when the parameter was θ_{n-2}^ϵ. Thus, at time $n+2$, Processor 2 computes

$$\theta_{n+2}^\epsilon = \Pi_H(\theta_{n+1}^\epsilon + \epsilon Y_{n-2}^\epsilon)$$

and analogously for Processors 3 and 4.

In general, suppose that the evaluation of each Y_n requires d units of time. Then, using $d+1$ synchronized processors the algorithm is

$$\theta_{n+1}^\epsilon = \Pi_H(\theta_n^\epsilon + \epsilon Y_{n-d}^\epsilon). \tag{1.2}$$

Define the initial condition $\theta_n^\epsilon = \theta_0$ for $n = 0, \ldots, d$. Thus, following the format of the example, Processor 1 takes physical measurements at times $0, d+1, 2(d+1), \ldots$, and it computes $\theta_{d+1}^\epsilon, \theta_{2(d+1)}^\epsilon, \ldots$ For $1 \le v \le d+1$, Processor v ($v = 1, \ldots, d+1$) takes physical measurements at times $v-1, v-1+(d+1), v-1+2(d+1), \ldots$, and it computes $\theta_{d+v}^\epsilon, \theta_{v-1+2(d+1)}^\epsilon, \ldots$ Once an update θ_n^ϵ is computed, its value is passed to the next processor in the line. The algorithmic form (1.2) is covered by the theorems in Chapters 5–8 without any change, owing to the fact that the update times are synchronized.

The model can easily be extended to allow variable (but bounded or uniformly integrable) delays in communication between processors, so that θ_n^ϵ in (1.2) is replaced by some $\theta_{n-\mu_n}^\epsilon$ where μ_n is the delay. The delay can depend on the destination processor.

12.1.3 A Distributed and Decentralized Network Model

This example is taken from [105], but the notation is slightly different. Consider a system with r processors, the αth having the responsibility for updating the αth component of θ. We wish to minimize a function $F(\cdot)$ that takes the form $F(\theta) = \sum_{\gamma=1}^{r} F^\gamma(\theta)$, for real-valued and continuously differentiable $F^\gamma(\cdot)$. Let $F_\alpha^\gamma(\theta) = \partial F^\gamma(\theta)/\partial\theta^\alpha$. In our model, for each α and each $\gamma = 1, \ldots, r$, Processor γ produces estimates $Y_{n,\alpha}^{\epsilon,\gamma}$, for $n \geq 0$, which are sent to Processor α for help in estimating $F_\alpha^\gamma(\cdot)$, at whatever is the current parameter value. It also sends the current values of its own component θ^γ. The form of the function $F(\cdot)$ is not known.

An important class of examples that fits the above model is optimal adaptive routing or service time control in queueing networks. Let the network have r nodes, with θ being perhaps a routing or service speed parameter, where θ^α is the component associated with the αth node. [We note that it is customary in the literature to minimize a mean waiting time, but the mean waiting time is related to the mean queue lengths via Little's formula.] Let $F^\gamma(\theta)$ denote the stationary average queue length at node γ under parameter value θ. We wish to minimize the stationary average number of customers in the network $F(\theta) = \sum_{\gamma=1}^{r} F^\gamma(\theta)$. The problem arises in the control of telecommunication networks and has been treated in [171, 174]. The controller at node α updates the component θ^α of θ based on both its own observations and relevant data sent from other nodes. In one useful approach, called the *surrogate estimation method* in the references, each node γ estimates the pathwise derivative of the mean length of its own queue with respect to variations in external inputs to that node. Then one uses the mean flow equations for the system to get acceptable "pathwise" estimates of the derivatives $F_\alpha^\gamma(\theta)$. These estimates are transmitted to node α for use in estimating the derivative of $F(\theta)$ with respect to θ^α at the current value of θ and then updating the value of θ^α. After each transmission, new estimates are taken and the process is repeated. The communication from processor to processor might take random lengths of time, due to the different paths used or the requirements and priorities of other traffic.

The time intervals required for the estimation can depend heavily and randomly on the node; for example, they might depend on the random service and arrival times. The nodes would transmit their estimates in an asynchronous way. Thus the stochastic approximation is both decentralized and asynchronous. More generally, some nodes might require a vector parameter, but the vector case can be reduced to the scalar case by "splitting" the nodes. In a typical application of stochastic approximation, each time a new estimate of $F_\alpha^\gamma(\theta)$ (at the current value of θ at node γ) is received at node α, that estimate is multiplied by a step size parameter and subtracted (since we have a gradient descent algorithm) from the current value (at node α) of the state component θ^α. This "additive" structure allows us to represent the algorithm in a useful decomposed way, by writing the current value of the component θ^α as the sum of the initial value minus r terms. The γth such term is the product of an appropriate step size times the sum of the past transmissions from node γ to node α, where each transmission is an estimate of $F_\alpha^\gamma(\cdot)$ at the most recent value of the parameter available to node γ when the estimate was made.

As noted, the transmission of information might not be instantaneous. If the delays are not excessively long (say, the sequence of delays is uniformly integrable) and ϵ is small, then they cause no problems in the analysis. It should be noted that this insensitivity to delay is an *asymptotic* result and might require a small step size to hold in any particular application. In any given practical system, the possibility of instabilities due to delays needs to be taken seriously. To simplify the notation in the rest of this example, we will work under the assumption that there are no delays and that the parameters are updated as soon as new information is available.

Notation. *The notation is for this example only. The symbols might be given different meanings later in the chapter.* Let the step size be $\epsilon > 0$, a constant. Let $\delta\tau_{n,\alpha}^{\epsilon,\gamma}$ denote the real-time interval between the nth and $(n+1)$st transmissions from node γ to node α. Define

$$\tau_{n,\alpha}^{\epsilon,\gamma} = \epsilon \sum_{i=0}^{n-1} \delta\tau_{i,\alpha}^{\epsilon,\gamma}, \tag{1.3}$$

which is ϵ times the real time required for the first n transmissions from γ to α. Define

$$N_\alpha^{\epsilon,\gamma}(t) = \epsilon \times [\text{number of transmissions from } \gamma \text{ to } \alpha \text{ up to real time} t/\epsilon].$$

Let $\tau_\alpha^{\epsilon,\gamma}(t)$ be the piecewise constant interpolation of the $\tau_{n,\alpha}^{\epsilon,\gamma}$ with interpolation intervals ϵ and initial condition zero. Analogously to the situation to be encountered in Section 2, $N_\alpha^{\epsilon,\gamma}(\cdot)$ and $\tau_\alpha^{\epsilon,\gamma}(\cdot)$ are inverses of one another.

The stochastic approximation algorithm. The notation is a little complex but very natural, as seen later in the chapter. It enables us to use the results of Chapters 5–8 in a much more complex situation via several time change arguments and saves a great deal of work over a direct analysis. As mentioned in the preceding discussion, it is convenient to separate the various updates in the value of θ^α into components that come from the same node. This suggests the following decomposed representation for the stochastic approximation algorithm. Let $\hat\theta^\epsilon(\cdot) = \{\hat\theta_i^\epsilon(\cdot), i \le r\}$ denote the interpolation of the parameter values in the real-time (times ϵ) scale.

For each α, γ, let $c_\alpha^\gamma(\cdot)$ be a continuous and bounded real-valued function and define the sequence $\theta_{n,\alpha}^{\epsilon,\gamma}$ by

$$\theta_{n+1,\alpha}^{\epsilon,\gamma} = \theta_{n,\alpha}^{\epsilon,\gamma} + \epsilon c_\alpha^\gamma(\hat{\theta}^\epsilon(\tau_{n+1,\alpha}^{\epsilon,\gamma,-}))Y_{n,\alpha}^{\epsilon,\gamma}. \tag{1.4}$$

The role of the functions $c_\alpha^\gamma(\cdot)$, which are to be selected by the experimenter, is to adjust the step size to partially compensate for the fact that the frequency of the intervals between updates might depend on θ, α, and γ. If desired, one can always use $c_\alpha^\gamma(\theta) \equiv 1$. Note that, by the definitions, $\hat{\theta}(\tau_{n+1,\alpha}^{\epsilon,\gamma,-})$ is the value of the state just before the $(n+1)$st updating at Processor α with the contribution from Processor γ. Define

$$\theta_\alpha^{\gamma,\epsilon}(t) = \theta_{n,\alpha}^{\epsilon,\gamma} \text{ for } t \in [n\epsilon, (n+1)\epsilon),$$

and let $\hat{\theta}_\alpha^{\epsilon,\gamma}(\cdot)$ denote the piecewise constant interpolation in ϵ times the real-time scale.

We have the following important relationships:

$$\begin{aligned} \hat{\theta}_\alpha^{\epsilon,\gamma}(t) &= \theta_{N_\alpha^{\epsilon,\gamma}(t)/\epsilon,\alpha}^{\epsilon,\gamma} = \theta_\alpha^{\epsilon,\gamma}(N_\alpha^{\epsilon,\gamma}(t)), \\ \hat{\theta}_\alpha^{\epsilon,\gamma}(\tau_\alpha^{\epsilon,\gamma}(t)) &= \theta_\alpha^{\epsilon,\gamma}(t). \end{aligned} \tag{1.5}$$

We can now write the actual interpolated iterate in the appropriate real-time scale in terms of the components as

$$\hat{\theta}_\alpha^\epsilon(t) = \hat{\theta}_\alpha^\epsilon(0) + \sum_{\gamma=1}^{r} \hat{\theta}_\alpha^{\epsilon,\gamma}(t), \quad \hat{\theta}_\alpha^{\epsilon,\gamma}(0) = 0 \text{ for } \gamma \neq \alpha, \; \hat{\theta}_\alpha^{\epsilon,\alpha}(0) = \theta_{0,\alpha}^\epsilon. \tag{1.6}$$

A constraint might be added to (1.6). Under reasonable conditions on $Y_{n,\alpha}^{\epsilon,\gamma}$, the proofs of convergence are just adaptations of the arguments in the previous chapters modified by the time change arguments of Section 3 in [105], which are similar to those used in Sections 2 and 3.

12.1.4 Multiaccess Communications

This example is motivated by the work of Hajek concerning the optimization of the ALOHA system [67]. In this system, r independent users share a common communication channel. At each time slot n, each user must decide whether or not to transmit assuming that there is a packet to transmit in that slot. Since there is no direct communication among users, there is a possibility that several users will transmit packets at the same time, in which case all such packets need to be retransmitted. The users can listen to the channel and will know whether or not there were simultaneous transmissions. Then they must decide when to retransmit the packet. One practical and common solution is to transmit or retransmit at random, where the conditional (given the past) probability of transmission or retransmission at any time is θ^α for user α. The determination of the optimal $\theta = (\theta^1, \ldots, \theta^r)$ is to be done adaptively, so that it can change appropriately as the operating conditions change.

Let $\hat{\theta}_{n,\alpha}$ denote the probability (conditioned on the past) that user α transmits a packet in time slot $n+1$, if a packet is available for transmission. Let I_n^α denote the indicator function of the event that user α transmits in time slot $n+1$, and J_n^α the indicator function of the event that other users also transmitted in that slot. Let $0 < a_\alpha < b_\alpha < 1, \alpha = 1, \ldots r$. For appropriate functions $G_\alpha(\cdot)$ the algorithm used in [67] is of the form

$$\hat{\theta}_{n+1,\alpha} = \Pi_{[a_\alpha, b_\alpha]} \left(\hat{\theta}_{n,\alpha} + \epsilon G_\alpha(\hat{\theta}_{n,\alpha}, J_n^\alpha) I_n^\alpha \right). \tag{1.7}$$

Thus, the update times of the transmission probability for each user are random; they are just the random times of transmission for that user. No update clock is synchronized among the users. The only information available concerning the actions of the other users is whether or not there were simultaneous transmissions.

12.2 Introduction: Real-Time Scale

The purpose of this section is to introduce and to motivate the basic issues of scaling for the asynchronous algorithm. To simplify the introduction of the "real-time" notation, assume that there is only one processor. The general case is in the following sections.

The time scale used in the proof of the convergence via the ODE method in Chapters 5–8 was determined by the iteration number and the step size only. For example, for the constant step size case, the interpolation $\theta^\epsilon(t)$ was defined to be θ_n^ϵ on the interpolated time interval $[n\epsilon, n\epsilon+\epsilon)$. For the decreasing step size case, we used the interpolations $\theta^n(\cdot), n = 0, 1, \ldots$, defined by $\theta^n(t) = \theta_{n+i}$ on the interpolated time interval $[t_{n+i} - t_n, t_{n+i+1} - t_n)$. These interpolations were natural choices for the problems of concern because our main interest was the characterization of the asymptotic properties of the sequences $\{\theta_n^\epsilon\}$ or $\{\theta_n\}$, and the increment in the iterate at the nth update was proportional to ϵ (ϵ_{n-1}, resp.). Of course, in applications one works in real (clock) time, but this real time is irrelevant for the purpose of proving convergence for the algorithms in Chapters 5–8. As was seen in the examples in Section 1, if the computation is split between several interacting processors that are not fully synchronized, then we might have to work in (perhaps scaled) real time, because there is no "iterate time" common to all the processors.

The situation is particularly acute if some of the interacting processors require (different) random times for the completion of their work on each update but can continue to update their part of the parameter even if there is a modest delay in receiving the latest information from the other processors. Of course, an algorithm that appears at first sight to be asynchronous might be rewritable as a fully synchronized algorithm or perhaps with only "slight" asynchronicities, and the user should be alert to this possibility.

When working in real time, the limit mean ODE or differential inclusion that characterizes the asymptotic behavior for the asynchronous algorithm will be seen to be very similar to what we have derived in Chapters 6–8. Indeed, it is the same except for rescalings that account for the different mean update speeds of the different processors. To prepare ourselves for these rescalings and to establish the type of notations and time scale to be used, we first obtain the limit mean ODE for a classical fully synchronized system as used in Chapters 5–8, but in a real-time scale.

Consider the algorithm used in Theorem 8.2.1, but without the constraint; namely,

$$\theta_{n+1}^\epsilon = \theta_n^\epsilon + \epsilon Y_n^\epsilon.$$

Assume the conditions of that theorem and that $\{\theta_n^\epsilon; \epsilon, n\}$ is bounded. Then Theorem 8.2.1 holds for the interpolations $\{\theta^\epsilon(\cdot), \theta^\epsilon(\epsilon q_\epsilon + \cdot)\}$, and the limit mean ODE is $\dot{\theta} = \bar{g}(\theta)$.

Suppose the computation of Y_n^ϵ requires a random amount of time $\delta\tau_n^\epsilon$. Thus $\delta\tau_n^\epsilon$ is the real-time interval between the nth and $(n+1)$st updating. Define the *interpolated real time*

$$\tau_n^\epsilon = \epsilon \sum_{i=0}^{n-1} \delta\tau_i^\epsilon,$$

and define

$$N^\epsilon(t) = \epsilon \times [\text{number of updatings by time } t/\epsilon].$$

Let $\tau^\epsilon(\cdot)$ be the interpolation of $\{\tau_n^\epsilon, n < \infty\}$ defined by $\tau^\epsilon(t) = \tau_n^\epsilon$ on $[\epsilon n, \epsilon n + \epsilon)$. See Figures 2.1 and 2.2.

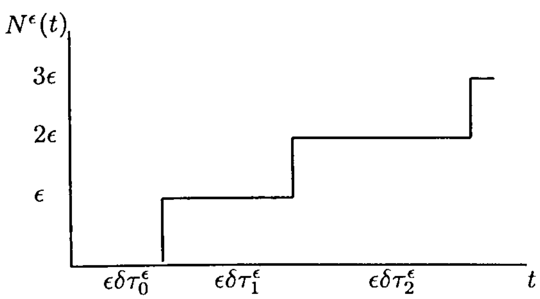

Figure 2.1. The function $N^\epsilon(\cdot)$.

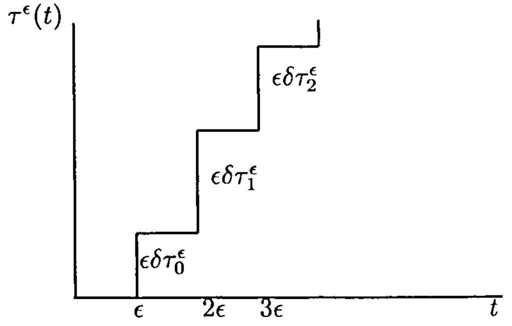

Figure 2.2. The function $\tau^\epsilon(\cdot)$.

Note that $\tau^\epsilon(\cdot)$ is the inverse of $N^\epsilon(\cdot)$ in the sense that

$$N^\epsilon(\tau^\epsilon(t)) = n\epsilon, \quad \text{for } t \in [n\epsilon, n\epsilon + \epsilon). \tag{2.1}$$

Define $\hat{\theta}^\epsilon(t) = \theta^\epsilon(N^\epsilon(t))$. This is the interpolation of the iterate sequence in the real-time scale, not in the iterate time scale. The weak convergence and characterization of the limit mean ODE for the $\{\hat{\theta}^\epsilon(\cdot)\}$ sequence is easily done by following the procedure used for Theorem 8.2.1; this will now be illustrated.

Refer to the notation and the structure of the setup for Theorem 8.2.1, and let $\mathcal{F}_n^\epsilon$ measure $\{\delta\tau_i^\epsilon, i < n\}$ as well. The notation $\dot{N}(t)$ denotes the derivative with respect to the argument. Thus, $\dot{N}(\tau(t))$ denotes the derivative of $N(\cdot)$ evaluated at $\tau(t)$. Equivalently,

$$\dot{N}(\tau(t)) = \frac{\partial}{\partial s} N(s)\big|_{s=\tau(t)}.$$

Suppose that

$$\{\delta\tau_n^\epsilon; \epsilon, n\} \ \text{ is uniformly integrable.} \tag{2.2}$$

Let there be measurable functions $u_n^\epsilon(\cdot)$ such that

$$E_n^\epsilon \delta\tau_n^\epsilon = u_n^\epsilon(\theta_n^\epsilon, \xi_n^\epsilon),$$

where we suppose that

$$\inf_{n,\epsilon,\theta,\xi} u_n^\epsilon(\theta, \xi) > 0 \tag{2.3}$$

and $u_n^\epsilon(\cdot, \xi)$ is continuous in θ uniformly in n, ϵ and in ξ on any compact set. Suppose there is a continuous function $\bar{u}(\cdot)$ such that

$$\frac{1}{m} \sum_{i=n}^{n+m-1} E_n^\epsilon u_i^\epsilon(\theta, \xi_i(\theta)) \to \bar{u}(\theta)$$

in mean for each θ as n and m go to infinity. Note that in applications, one often has some influence over the design of the system, and then the assumptions are reasonable requirements on that design.

The set $\{\theta^\epsilon(\cdot), \tau^\epsilon(\cdot), \hat{\theta}^\epsilon(\cdot), N^\epsilon(\cdot)\}$ is tight. The tightness assertion and the Lipschitz continuity of the weak sense limits of $\{\tau^\epsilon_\alpha(\cdot)\}$ are consequences of the uniform integrability (2.2). The strict positivity (2.3) implies that all weak sense limits of $\{\tau^\epsilon(\cdot)\}$ are strictly increasing. The tightness assertion and the Lipschitz continuity of the weak sense limits of $\{N^\epsilon(\cdot)\}$ is a consequence of these facts together with the property (2.1). [See the proof of Theorem 3.1 for additional detail.] For notational simplicity, we suppose that the original sequences indexed by ϵ converge weakly; otherwise take an appropriate subsequence. By the assumptions and the proof of Theorem 8.2.1, there are processes $(\theta(\cdot), \tau(\cdot), \hat{\theta}(\cdot), N(\cdot))$ such that

$$(\theta^\epsilon(\cdot), \tau^\epsilon(\cdot), \hat{\theta}^\epsilon(\cdot), N^\epsilon(\cdot)) \Rightarrow (\theta(\cdot), \tau(\cdot), \hat{\theta}(\cdot), N(\cdot)),$$

where $\hat{\theta}(t) = \theta(N(t))$ and

$$\tau(t) = \int_0^t \bar{u}(\theta(s))ds. \tag{2.4}$$

Owing to the strict positivity (2.3) and the "inverse" definitions of $N^\epsilon(\cdot)$ and $\tau^\epsilon(\cdot)$, it follows that both $\tau(\cdot)$ and $N(\cdot)$ are strictly increasing and differentiable with probability one and that $N(\tau(t)) = t$. Hence $N(t) = \tau^{-1}(t)$, where the -1 denotes the inverse function. Taking derivatives, we get $\dot{N}(\tau(t))\dot{\tau}(t) = 1$. Then using (2.4) and writing $s = \tau(t)$, we see that the slope of $N(\cdot)$ at s is $\dot{N}(s) = 1/\bar{u}(\theta(\tau^{-1}(s))) = 1/\bar{u}(\hat{\theta}(s))$. In other words, the mean rate of change of the number of updates is inversely proportional to the mean time between updates. We now have

$$N(t) = \int_0^t \frac{ds}{\bar{u}(\hat{\theta}(s))}. \tag{2.5}$$

By Theorem 8.2.1, $\theta(\cdot)$ satisfies $\dot{\theta} = \bar{g}(\theta)$. Using the relationships

$$\hat{\theta}^\epsilon(\cdot) = \theta^\epsilon(N^\epsilon(\cdot)) \Rightarrow \theta(N(\cdot)) \equiv \hat{\theta}(\cdot) \tag{2.6}$$

and $N(\tau(t)) = t$ yields

$$\dot{\hat{\theta}}(t) = [\dot{\theta}(N(t))]\dot{N}(t) = \frac{\bar{g}(\theta(N(t)))}{\bar{u}(\hat{\theta}(t))} = \frac{\bar{g}(\hat{\theta}(t))}{\bar{u}(\hat{\theta}(t))}. \tag{2.7}$$

Thus, the derivation of the limit mean ODE in the real-time scale is just that of Theorem 8.2.1 plus a time change argument.

Equation (2.7) says that the (asymptotic) progress of the convergence in real time is scaled by the (asymptotic) mean "rate of updating," as measured by the mean rate $1/\bar{u}(\hat{\theta}(t))$. The purpose of the time change argument is to avoid dealing with random interpolation intervals and the interaction of Y^ϵ_n and $\delta\tau^\epsilon_n$. The time change argument exploits the convergence of both the sequence of "time" processes $\{\tau^\epsilon(\cdot), N^\epsilon(\cdot)\}$ and the original sequence of interpolations $\{\theta^\epsilon(\cdot)\}$. The approach can be viewed as a "separation principle," which works with the iterate and real-time scales separately, doing the computations in each that are best suited to it. Similar time scalings will be used in the sequel to enable the general results of Chapter 8 to be carried over to the asynchronous algorithm with minimal effort.

Remark on the analysis for infinite time. The argument leading to (2.7) is for the sequence of processes $\{\theta^\epsilon(\cdot)\}$ that start at time zero and are defined on the interval $[0, \infty)$. This gives us a limit theorem for $\{\hat{\theta}^\epsilon(\cdot)\}$ for t in any large but bounded interval, or even for an interval $[0, T_\epsilon]$ where $T_\epsilon \to \infty$ slowly enough. It is good enough for most practical algorithms, when ϵ is small and ϵn might be large, but it will not go to infinity in the application.

In Chapters 5–8, to get limits as the interpolated time went to infinity, we worked with a sequence of processes that were actually the "tails" of the original process. For example, in Chapter 8, to get the limits of θ_n^ϵ as $\epsilon \to 0$ but $n\epsilon \to \infty$, we worked with the sequence of interpolations $\theta^\epsilon(\epsilon q_\epsilon + \cdot)$, where $\epsilon q_\epsilon \to \infty$. The natural analog of this procedure in the real-time scale uses a sequence of processes starting at a sequence of real times that go to infinity as $\epsilon \to 0$, as follows. Let $T_\epsilon \to \infty$ (real numbers) replace $\epsilon q_\epsilon \to \infty$ and work with $\{\hat{\theta}^\epsilon(T_\epsilon + \cdot)\}$.

The limits of $\hat{\theta}^\epsilon(T_\epsilon + \cdot)$ as $\epsilon \to 0$ are the limits of the iterate sequence in real time, as real time goes to infinity and $\epsilon \to 0$. Then the analysis is the same as above, except that the value at time zero (the new initial condition) of the real-time interpolated process shifted to the left by T_ϵ is

$$\hat{\theta}^\epsilon(T_\epsilon) = \theta^\epsilon(N^\epsilon(T_\epsilon)) = \theta^\epsilon_{N^\epsilon(T_\epsilon)/\epsilon}.$$

Recall that $N^\epsilon(t)/\epsilon$ is the index of the last iterate that occurs at or before real-time t and it is a random variable. The uniform integrability and tightness conditions must reflect this change of time origin. In particular, since $N^\epsilon(t)$ is a random variable, we will need to assume that

$$\{\xi^\epsilon_{N^\epsilon(T_\epsilon)/\epsilon+n}, \theta^\epsilon_{N^\epsilon(T_\epsilon)/\epsilon+n}; \epsilon, n \geq 0\} \text{ is tight} \tag{2.8}$$

and

$$\{Y^\epsilon_{N^\epsilon(T^\epsilon)/\epsilon+n}, \delta\tau^\epsilon_{N^\epsilon(T^\epsilon)/\epsilon+n}; \epsilon, n \geq 0\} \text{ is uniformly integrable.} \tag{2.9}$$

For each $\epsilon > 0$, the quantities in (2.9) are used to compute the iterates that occur at or after real-time T_ϵ. In fact, for each $\epsilon > 0$, (2.8) is just the set of observations and time intervals for the process starting at time T_ϵ with initial condition $\hat{\theta}^\epsilon(T_\epsilon)$. These assumptions would not generally hold as required if they held only for $T_\epsilon = 0$. But for the types of renewal processes that commonly define the update times, the verification of (2.8), (2.9) and their analogs in subsequent sections might not be more difficult than the verification for $T_\epsilon = 0$. Indeed, an examination of various cases suggests that (2.8) and (2.9) are not at all restrictive. Analogous conditions will be used in what follows. Unfortunately, the notation that needs to be used for the processes shifted left by an arbitrary real time is a little messy.

Note that in practice, we will start the procedure at some arbitrary time, which suggests that (2.8) and (2.9) are essential for a well-posed problem, whether or not we work in real time, because they are just the required conditions when we start at an arbitrary real time and desire convergence no matter what the starting time is.

12.3 The Basic Algorithms

12.3.1 Constant Step Size: Introduction

In this section, we work with a versatile canonical model, with a constant step size $\epsilon > 0$. The notation is, unfortunately, not simple. This seems to be unavoidable because the actual number of updates at the different processors at any time are only probabilistically related, and we wish to treat a form that covers many interesting cases and exposes the essential issues. The basic format of the convergence proofs is that of weak convergence. Extensions to the decreasing step size model are straightforward, being essentially notational, and are dealt with in the next section. In this and the next subsection, we develop the notation. Following the idea of the time scale changes used in Section 2, we first define each of the components of the parameter processes in "iterate time," and then we get the results for the "real-time" evolution from these by the use of appropriate time scale changes, with each component having its own time scale process. Thus we also need to deal with the weak convergence of the time scale processes. The method is first developed for the martingale difference noise case. The extension to the correlated noise case uses the same techniques and appropriate averaging conditions, analogous to what was done in Chapter 8. The basic theorems are for large but bounded time intervals, analogously to the first results in Section 2. The infinite time analysis is done in Subsection 4 and conditions analogous to those of (2.8) and (2.9) will be needed.

For descriptive purposes, it is convenient to suppose that there are r processors, each responsible for the updating one component of θ. Let $\delta\tau^\epsilon_{n,\alpha}$ denote the real time between the nth and $(n+1)$st updating of the αth component of θ. It is possible that $\delta\tau^\epsilon_{n,\alpha}$ has the same value for several values of α and all ϵ and n, and that the same processor is responsible for that set of components. However, that case is covered by the current assumption. Due to the working assumption of a separate processor for the updating of each component of θ, we will simplify further by supposing that the constraint set takes the form

$$H = [a_1, b_1] \times \ldots \times [a_r, b_r], \; -\infty < a_\alpha < b_\alpha < \infty. \tag{3.1}$$

This assumption on the form of the constraint set is not essential. If some processor is used to update several components, then the general constraints (A4.3.2) or (A4.3.3) can be used on that set. In fact, the algorithm can be written so that the entire vector is constrained, subject to current knowledge at each processor, and the limit mean ODE will be the correct one for the constrained problem. The proofs of convergence are all very close to those in Chapter 8; only details concerning the differences will be given.

The algorithm of concern will be written in terms of its components. For $\alpha = 1, \ldots, r$, it takes the form

$$\theta^\epsilon_{n+1,\alpha} = \Pi_{[a_\alpha, b_\alpha]}\left(\theta^\epsilon_{n,\alpha} + \epsilon Y^\epsilon_{n,\alpha}\right) = \theta^\epsilon_{n,\alpha} + \epsilon Y^\epsilon_{n,\alpha} + \epsilon Z^\epsilon_{n,\alpha}. \tag{3.2}$$

Define the scaled *interpolated real time*

$$\tau^\epsilon_{n,\alpha} = \epsilon \sum_{i=0}^{n-1} \delta\tau^\epsilon_{i,\alpha},$$

and define

$$p^\epsilon_\alpha(\sigma) = \min\left\{ n : \sum_{i=0}^{n-1} \delta\tau^\epsilon_{i,\alpha} \geq \sigma \right\},$$

the index of the first update at Processor α at or after real time σ.

Define

$$N^\epsilon_\alpha(\sigma) = \epsilon \times [\text{number of updatings of the } \alpha\text{th component by time } \sigma/\epsilon].$$

Let $\theta^\epsilon_\alpha(\cdot)$ denote the interpolation of $\{\theta^\epsilon_{n,\alpha}, n < \infty\}$ on $[0, \infty)$ defined by $\theta^\epsilon_\alpha(t) = \theta^\epsilon_{n,\alpha}$ on $[n\epsilon, n\epsilon + \epsilon)$, and define $\tau^\epsilon_\alpha(\cdot)$ analogously, but using $\{\tau^\epsilon_{n,\alpha}\}$. The relations between some of the terms are illustrated in Figure 3.1.

Figure 3.1. The iterates up to real time σ for Processor α.

Analogously to the situation in Section 2, note that $\tau^\epsilon_\alpha(\cdot)$ is the inverse of $N^\epsilon_\alpha(\cdot)$ in the sense that

$$N^\epsilon_\alpha(\tau^\epsilon_\alpha(t)) = n\epsilon \text{ for } t \in [n\epsilon, n\epsilon + \epsilon). \tag{3.3}$$

Define the "real-time" interpolation $\hat{\theta}_\alpha^\epsilon(t)$ by

$$\hat{\theta}_\alpha^\epsilon(t) = \theta_{n,\alpha}^\epsilon,\ t \in [\tau_{n,\alpha}^\epsilon, \tau_{n+1,\alpha}^\epsilon), \tag{3.4}$$

and note the equalities

$$\theta_\alpha^\epsilon(t) = \hat{\theta}_\alpha^\epsilon(\tau_\alpha^\epsilon(t)),\ \hat{\theta}_\alpha^\epsilon(t) = \theta_\alpha^\epsilon(N_\alpha^\epsilon(t)). \tag{3.5}$$

Define the vector $\hat{\theta}^\epsilon(t) = (\hat{\theta}_\alpha^\epsilon(t), \alpha \le r)$.

12.3.2 Martingale Difference Noise

Notation. In Chapter 8, the σ-algebra $\mathcal{F}_n^\epsilon$ measured all the data available until, but not including, the computation of θ_{n+1}^ϵ. With E_n^ϵ denoting the expectation conditioned on $\mathcal{F}_n^\epsilon$, for the martingale difference noise case we had $E_n^\epsilon Y_n^\epsilon = g_n^\epsilon(\theta_n^\epsilon)$ plus possibly an asymptotically negligible error, called β_n^ϵ. By taking this conditional expectation, we were able to write Y_n^ϵ as the sum of the conditional mean and a martingale difference term. This decomposition was fundamental to the analysis. Thus, a key quantity in the analysis was the expectation of Y_n^ϵ conditioned on all the data needed to compute the past iterate values $\{\theta_i^\epsilon, i \le n\}$. Similar computations, centering the increments about appropriate conditional means, are also fundamental for the analysis of the asynchronous algorithm. Nevertheless, the problem is complicated by the fact that we need to deal with both $\hat{\theta}_\alpha^\epsilon(\cdot)$ and $\tau_\alpha^\epsilon(\cdot)$ for each α, and the full state can change between the updates at any given α.

The random variable $Y_{n,\alpha}^\epsilon$ is used at real time $\tau_{n+1,\alpha}^\epsilon$ to compute $\theta_{n+1,\alpha}^\epsilon$, as illustrated in Figure 3.2. Hence, when centering $Y_{n,\alpha}^\epsilon$ about a conditional mean (given the "past"), it is appropriate to let the conditioning data for processor α be all the data available on the open real-time interval $[0, \tau_{n+1,\alpha}^\epsilon)$. *Thus, we include $\delta\tau_{n,\alpha}^\epsilon$ in the conditioning data, as well as the value of the full state just before that update. Note that the value of the state just before that update at Processor α is $\hat{\theta}^\epsilon(\tau_{n+1,\alpha}^{\epsilon,-})$.*

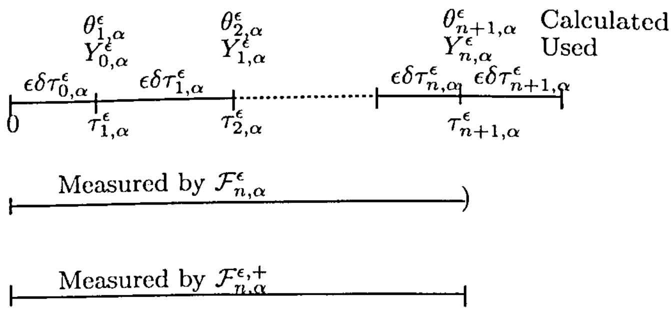

Figure 3.2. The definitions of the σ-algebras.

When centering $\delta\tau^\epsilon_{n,\alpha}$ about its conditional mean (given the "past"), it is appropriate to let the conditioning data for Processor α be all the data available on the closed real-time interval $[0, \tau^\epsilon_{n,\alpha}]$; this data includes $Y^\epsilon_{n-1,\alpha}$. Due to these considerations, different conditioning σ-algebras are used for the centering of the Y and $\delta\tau$ sequences.

For the reasons just explained, for each α, we need the nondecreasing families of σ-algebras $\mathcal{F}^\epsilon_{n,\alpha}$ and $\mathcal{F}^{\epsilon,+}_{n,\alpha}$ defined as follows. The σ-algebra $\mathcal{F}^\epsilon_{n,\alpha}$ measures at least the random variables

$$\begin{aligned}&\theta^\epsilon_0,\ \{Y^\epsilon_{j-1,\gamma}; j,\gamma \text{ such that } \tau^\epsilon_{j,\gamma} < \tau^\epsilon_{n+1,\alpha}\}, \text{ and}\\ &\{\tau^\epsilon_{j,\gamma}; j,\gamma \text{ such that } \tau^\epsilon_{j,\gamma} \le \tau^\epsilon_{n+1,\alpha}\}.\end{aligned}$$

Let $E^\epsilon_{n,\alpha}$ denote the expectation conditioned on $\mathcal{F}^\epsilon_{n,\alpha}$. The σ-algebra $\mathcal{F}^{\epsilon,+}_{n,\alpha}$ measures at least the random variables

$$\begin{aligned}&\theta^\epsilon_0,\ \{Y^\epsilon_{j-1,\gamma}; j,\gamma \text{ such that } \tau^\epsilon_{j,\gamma} \le \tau^\epsilon_{n+1,\alpha}\}, \text{ and}\\ &\{\tau^\epsilon_{j,\gamma}; j,\gamma \text{ such that } \tau^\epsilon_{j,\gamma} \le \tau^\epsilon_{n+1,\alpha}\}.\end{aligned}$$

Thus, $\mathcal{F}^\epsilon_{n,\alpha}$ measures the data available on the real-time interval $[0, \tau^\epsilon_{n+1,\alpha})$, and $\mathcal{F}^{\epsilon,+}_{n,\alpha}$ measures the data available on the real-time interval $[0, \tau^\epsilon_{n+1,\alpha}]$. Let $E^{\epsilon,+}_{n,\alpha}$ denote the expectation conditioned on $\mathcal{F}^{\epsilon,+}_{n,\alpha}$. The definitions are illustrated in Figure 3.2. The definitions allow considerable versatility, but one might wish to modify them appropriately (depending on the information patterns) in specific applications.

Assumptions. We will need the following analogs of (A8.1.1)–(A8.1.4). The $\Delta^\epsilon_{n,\alpha}$ and $\Delta^{\epsilon,+}_{n,\alpha}$ to be used in conditions (A3.2) and (A3.3) represent communication delays (times ϵ). Conditions (3.8) and (3.10) on these delays hold if the sequence of physical delays $\{\Delta^\epsilon_{n,\alpha}/\epsilon, \Delta^{\epsilon,+}_{n,\alpha}/\epsilon; \epsilon, n, \alpha\}$ is uniformly integrable, a condition that is not restrictive in applications. For notational convenience, we suppose in (3.6) and (3.9) that for each α all components of the argument $\hat\theta^\epsilon(\cdot)$ are delayed by the same amount. This is not necessary. Each component can have its own delay as long as it satisfies (3.8) and (3.10). The following conditions are to hold for each α.

(A3.1) $\{Y^\epsilon_{n,\alpha}, \delta\tau^\epsilon_{n,\alpha}; \epsilon, \alpha, n\}$ is uniformly integrable.

(A3.2) There are real-valued functions $g^\epsilon_{n,\alpha}(\cdot)$ that are continuous uniformly in n and ϵ, and random variables $\beta^\epsilon_{n,\alpha}$ and (non-negative) random variables $\Delta^\epsilon_{n,\alpha}$ such that

$$E^\epsilon_{n,\alpha} Y^\epsilon_{n,\alpha} = g^\epsilon_{n,\alpha}(\hat\theta^\epsilon(\tau^{\epsilon,-}_{n+1,\alpha} - \Delta^\epsilon_{n,\alpha})) + \beta^\epsilon_{n,\alpha}, \tag{3.6}$$

where

$$\{\beta^\epsilon_{n,\alpha}; n, \epsilon, \alpha\} \text{ is uniformly integrable} \tag{3.7}$$

and for each $T > 0$,

$$\sup_{n \le T/\epsilon} \Delta^\epsilon_{n,\alpha} \to 0 \tag{3.8}$$

in probability as $\epsilon \to 0$. (The $\Delta^{\epsilon}_{n,\alpha}$ represent the delays in the communication, multiplied by ϵ.)

(A3.3) There are real-valued functions $u^{\epsilon}_{n,\alpha}(\cdot)$ that are strictly positive in the sense of (2.3) and are continuous uniformly in n and ϵ, and non-negative random variables $\Delta^{\epsilon,+}_{n,\alpha}$ such that

$$E^{\epsilon,+}_{n,\alpha}\left[\delta\tau^{\epsilon}_{n+1,\alpha}\right] = u^{\epsilon}_{n+1,\alpha}(\hat{\theta}^{\epsilon}(\tau^{\epsilon}_{n+1,\alpha} - \Delta^{\epsilon,+}_{n+1,\alpha})) \tag{3.9}$$

and

$$\sup_{n\le T/\epsilon} \Delta^{\epsilon,+}_{n,\alpha} \to 0 \tag{3.10}$$

in probability as $\epsilon \to 0$. (The $\Delta^{\epsilon,+}_{n,\alpha}$ represent the delays in the communication multiplied by ϵ.)

(A3.4) There are continuous real-valued functions $\bar{g}_{\alpha}(\cdot)$ such that for each $\theta \in H$,

$$\lim_{m,n,\epsilon} \frac{1}{m}\sum_{i=n}^{n+m-1}\left[g^{\epsilon}_{i,\alpha}(\theta) - \bar{g}_{\alpha}(\theta)\right] = 0. \tag{3.11a}$$

(A3.5) There are continuous real-valued functions $\bar{u}_{\alpha}(\cdot)$ (which must be positive by (2.3)) such that for each $\theta \in H$,

$$\lim_{m,n,\epsilon} \frac{1}{m}\sum_{i=n}^{n+m-1}\left[u^{\epsilon}_{i,\alpha}(\theta) - \bar{u}_{\alpha}(\theta)\right] = 0. \tag{3.11b}$$

(A3.6) $\displaystyle\lim_{m,n,\epsilon} \frac{1}{m}\sum_{i=n}^{n+m-1} E^{\epsilon}_{n,\alpha}\beta^{\epsilon}_{i,\alpha} = 0$ in mean.

The $\lim_{m,n,\epsilon}$ means that the limit is taken as $m \to \infty, n \to \infty$ and $\epsilon \to 0$ simultaneously in any way. We can now write the algorithm for Processor α as

$$\theta^{\epsilon}_{n+1,\alpha} = \theta^{\epsilon}_{n,\alpha} + \epsilon g^{\epsilon}_{n,\alpha}(\hat{\theta}(\tau^{\epsilon,-}_{n+1,\alpha} - \Delta^{\epsilon}_{n,\alpha})) + \epsilon\delta M^{\epsilon}_{n,\alpha} + \epsilon\beta^{\epsilon}_{n,\alpha} + \epsilon Z^{\epsilon}_{n,\alpha},$$

where $\delta M^{\epsilon}_{n,\alpha} = Y^{\epsilon}_{n,\alpha} - E^{\epsilon}_{n,\alpha}Y^{\epsilon}_{n,\alpha}$, and there is an analogous decomposition for $\delta\tau^{\epsilon}_{n,\alpha}$ in terms of the conditional mean and a martingale difference term. Define

$$Z^{\epsilon}_{\alpha}(t) = \epsilon\sum_{i=0}^{t/\epsilon-1} Z^{\epsilon}_{i,\alpha}.$$

The comments below Theorem 8.2.1 concerning the limit set apply here, as well as in the other theorems of this section.

Theorem 3.1. *Assume* (A3.1)–(A3.6). *Then*

$$\{\theta^{\epsilon}_{\alpha}(\cdot), \tau^{\epsilon}_{\alpha}(\cdot), \hat{\theta}^{\epsilon}(\cdot), N^{\epsilon}_{\alpha}(\cdot), \alpha \le r\}$$

is tight in $D^{4r}[0,\infty)$. *Let* ϵ *(abusing terminology for simplicity) index a weakly convergent subsequence whose weak sense limit we denote by*

$$(\theta_\alpha(\cdot),\tau_\alpha(\cdot),\hat{\theta}(\cdot),N_\alpha(\cdot),\alpha\le r).$$

Then the limits are Lipschitz continuous with probability one and

$$\theta_\alpha(t)=\hat{\theta}_\alpha(\tau_\alpha(t)),\quad \hat{\theta}_\alpha(t)=\theta_\alpha(N_\alpha(t)) \tag{3.12}$$

$$N_\alpha(\tau_\alpha(t))=t. \tag{3.13}$$

Also,

$$\tau_\alpha(t)=\int_0^t \bar{u}_\alpha(\hat{\theta}(\tau_\alpha(s))ds, \tag{3.14}$$

$$\dot{\theta}_\alpha(t)=\bar{g}_\alpha(\hat{\theta}(\tau_\alpha(t)))+z_\alpha(t), \tag{3.15}$$

$$\dot{\hat{\theta}}_\alpha=\frac{\bar{g}_\alpha(\hat{\theta})}{\bar{u}_\alpha(\hat{\theta})}+\hat{z}_\alpha,\ \alpha=1,\ldots,r, \tag{3.16}$$

where the z_α *and* $\hat{z}_\alpha$ *serve the purpose of keeping the paths in the interval* $[a_\alpha,b_\alpha]$.

If L_H *is asymptotically stable in the sense of Liapunov, then for any* $\mu>0$ *there is a* $T_\mu>0$ *such that for* $t\ge T_\mu$, $\text{distance}(\hat{\theta}(t),L_H)<\mu$. *Also, for any* $T>T_\mu$,

$$\limsup_\epsilon P\left\{\sup_{T_\mu\le t\le T}\text{distance}(\hat{\theta}^\epsilon(t),L_H)\ge\mu\right\}=0. \tag{3.17}$$

For large T *and* $T_1>T$, $\hat{\theta}^\epsilon(\cdot),t\in[T,T_1]$ *spends nearly all of its time (the fraction going to one as* $\epsilon\to 0$*) in a small neighborhood of* L_H.

Now, drop the constraint set H *and suppose that* $\{\theta_n^\epsilon;\epsilon,n\}$ *is bounded with probability one. Then the above conclusions continue to hold with* $z_\alpha(t)=\hat{z}_\alpha(t)=0$ *and* L_H *replaced by some limit set of* (3.16).

Proof. The tightness assertion and the Lipschitz continuity of the paths of the weak sense limit processes of $\{\theta_\alpha^\epsilon(\cdot),\tau_\alpha^\epsilon(\cdot),Z_\alpha^\epsilon(\cdot)\}$ follow from the uniform integrability (A3.1) exactly as was done in the proof of Theorem 8.2.1. The strict positivity of $u_{n,\alpha}^\epsilon(\cdot)$ implies that with probability one any weak sense limit of $\{\tau_\alpha^\epsilon(\cdot)\}$ is strictly monotonically increasing and goes to infinity as $t\to\infty$. In fact, the slope of this weak sense limit process must be at least the lower bound in (2.3). These facts and (3.3) imply the tightness of $\{N_\alpha^\epsilon(\cdot)\}$ and the Lipschitz continuity of the paths of its weak sense limits. Finally, the tightness of $\{\theta_\alpha^\epsilon(\cdot),N_\alpha^\epsilon(\cdot)\}$, together with the Lipschitz property of their weak sense limits and (3.5), imply the tightness of $\{\hat{\theta}_\alpha^\epsilon(\cdot)\}$ and the fact that all of its weak sense limits are Lipschitz continuous with probability one.

Equations (3.12) and (3.13) are consequences of the weak convergence and definitions (3.3) and (3.5). The representation (3.16) follows from (3.12), (3.14), and (3.15) by a change of variable, exactly as (2.7) followed from $\dot{\theta} = \bar{g}(\theta)$, (2.4), and (2.6) in Section 1. In detail,

$$\begin{aligned}\dot{\hat{\theta}}_\alpha(t) &= \dot{\theta}_\alpha(N_\alpha(t))\dot{N}_\alpha(t)\\ &= \frac{\bar{g}_\alpha(\hat{\theta}(\tau_\alpha(N_\alpha(t)))}{\bar{u}_\alpha(\hat{\theta}(t))} + \text{reflection term}\\ &= \frac{\bar{g}_\alpha(\hat{\theta}(t))}{\bar{u}_\alpha(\hat{\theta}(t))} + \text{reflection term.}\end{aligned}$$

Equation (3.17) is a consequence of the weak convergence and the stability. Thus, we need only prove the representations (3.14) and (3.15).

We continue to follow the approach of the proof of Theorem 8.2.1. Due to the asymptotic continuity of $\theta^\epsilon_\alpha(\cdot)$ and $\hat{\theta}^\epsilon(\cdot)$ and (3.8) and (3.10), the delays $\Delta^\epsilon_{n,\alpha}$ and $\Delta^{\epsilon p,+}_{n,\alpha}$ play no significant role and only complicate the notation. Without loss of generality, they will be dropped henceforth. Recall the definition $\delta M^\epsilon_{n,\alpha} = Y^\epsilon_{n,\alpha} - E^\epsilon_{n,\alpha}Y^\epsilon_{n,\alpha}$, and

$$M^\epsilon_\alpha(t) = \epsilon \sum_{i=0}^{t/\epsilon-1} \delta M^\epsilon_{i,\alpha}.$$

Define $B^\epsilon_\alpha(t)$ from $\{\beta^\epsilon_{n,\alpha}, n < \infty\}$ analogously, and set

$$\tilde{G}^\epsilon_\alpha(t) = \epsilon \sum_{i=0}^{t/\epsilon-1} \left[g^\epsilon_{i,\alpha}(\hat{\theta}^\epsilon(\tau^{\epsilon,-}_{i+1,\alpha})) - \bar{g}_\alpha(\hat{\theta}^\epsilon(\tau^{\epsilon,-}_{i+1,\alpha}))\right]$$

and

$$\bar{G}^\epsilon_\alpha(t) = \epsilon \sum_{i=0}^{t/\epsilon-1} \bar{g}_\alpha(\hat{\theta}^\epsilon(\tau^{\epsilon,-}_{i+1,\alpha})).$$

Define $W^\epsilon_\alpha(\cdot)$ by

$$W^\epsilon_\alpha(t) = \hat{\theta}^\epsilon_\alpha(\tau^\epsilon_\alpha(t)) - \theta^\epsilon_{0,\alpha} - \bar{G}^\epsilon_\alpha(t) - Z^\epsilon_\alpha(t) = \tilde{G}^\epsilon_\alpha(t) + M^\epsilon_\alpha(t) + B^\epsilon_\alpha(t).$$

Recall the terminology concerning $h(\cdot)$, $s_i \le t$, p, t, and τ used in (8.2.6), where $h(\cdot)$ is an arbitrary bounded real-valued function of its arguments, t and $\tau \ge 0$ are arbitrary real numbers, p is an arbitrary integer, and $s_i \le t, i \le p$, are real numbers. We can write

$$\begin{aligned}&Eh(\tau^\epsilon_\alpha(s_i), \hat{\theta}^\epsilon(\tau^\epsilon_\alpha(s_i)), i \le p)\left[W^\epsilon_\alpha(t+\tau) - W^\epsilon_\alpha(t)\right]\\ &-Eh(\tau^\epsilon_\alpha(s_i), \hat{\theta}^\epsilon(\tau^\epsilon_\alpha(s_i)), i \le p)\left[\tilde{G}^\epsilon_\alpha(t+\tau) - \tilde{G}^\epsilon_\alpha(t)\right]\\ &-Eh(\tau^\epsilon_\alpha(s_i), \hat{\theta}^\epsilon(\tau^\epsilon_\alpha(s_i)), i \le p)\left[M^\epsilon_\alpha(t+\tau) - M^\epsilon_\alpha(t)\right]\\ &-Eh(\tau^\epsilon_\alpha(s_i), \hat{\theta}^\epsilon(\tau^\epsilon_\alpha(s_i)), i \le p)\left[B^\epsilon_\alpha(t+\tau) - B^\epsilon_\alpha(t)\right] = 0.\end{aligned} \tag{3.18}$$

Note that the arguments of $h(\cdot)$ involve iterates at most up to the time of the (t/ϵ)th iterate of Processor α, and the terms in the square brackets involve iterates in the time interval from the $(t/\epsilon+1)$st iterate to the $((t+\tau)/\epsilon)$th iterate of Processor α.

The term involving $M_\alpha^\epsilon(\cdot)$ in (3.18) equals zero by the martingale difference property of the $\delta M_{n,\alpha}^\epsilon$. The term involving $B_\alpha^\epsilon(\cdot)$ goes to zero as $\epsilon \to 0$ by (A3.6). Finally, the term involving $\tilde{G}_\alpha^\epsilon(\cdot)$ goes to zero as $\epsilon \to 0$ by the uniform continuity of $g_{n,\alpha}^\epsilon(\cdot)$, (A3.4), and the tightness and Lipschitz continuity of the weak sense limits of $\{\hat{\theta}^\epsilon(\cdot), \tau_\alpha^\epsilon(\cdot), \alpha \le r\}$. Completing the argument exactly as in the proof of Theorem 8.2.1 yields that any weak sense limit $W_\alpha(\cdot), \alpha = 1, \ldots, r$, of $\{W_\alpha^\epsilon(\cdot)\}$ is a martingale (the reader can identify the associated σ-algebras) with Lipschitz continuous paths and the representation

$$W_\alpha(t) = \hat{\theta}_\alpha(\tau_\alpha(t)) - \theta_\alpha(0) - \bar{G}_\alpha(t) - Z_\alpha(t), \tag{3.19}$$

where

$$\bar{G}_\alpha(t) = \int_0^t \bar{g}_\alpha(\hat{\theta}(\tau_\alpha(s)))ds,$$

and $Z_\alpha(\cdot)$ is the weak sense limit of $\{Z_\alpha^\epsilon(\cdot)\}$.

Since the value of $W_\alpha(\cdot)$ is zero at time zero, it is identically zero for all time. The process $Z_\alpha(t)$ is characterized as in the proof of Theorem 8.2.1. Thus (3.15) holds. (3.14) is proved in a similar manner. □

Differential inclusions. In Theorems 6.8.1 and 8.2.5, the ODE was replaced by a differential inclusion. Such forms of the limit mean ODE arise when the local averages of the noise or dynamics vary with time. There is a complete analog for the asynchronous algorithm, and it might be even more important there. Suppose that the mean times between updates does not average out as required by (A3.5) and that the local averages varied over time. This would happen, for example, in the Q-learning problem if the sequence of distributions of the return times to each (state-action) pair has no particular regularity. For reasons of simplicity, the statement is restricted to the variations of the mean times between updates, which seems to be the most important application. However, the dynamical terms $\bar{g}_\alpha(\cdot)$ can also be elements of a set as in Theorems 6.8.1 and 8.2.5.

Theorem 3.2. *Assume the conditions of Theorem* 3.1, *but replace* (A3.5) *with the following: There are sets* $U_\alpha(\theta), \alpha = 1, \ldots, r$, *that are upper semicontinuous in the sense of* (4.3.2) *such that for each* $\theta \in H$,

$$\text{distance}\left[\frac{1}{m}\sum_{i=n}^{n+m-1} u_{i,\alpha}^\epsilon(\theta), U_\alpha(\theta)\right] \to 0 \tag{3.20}$$

in mean as n and m go to infinity and ϵ goes to zero. Then the conclusions of Theorem 3.1 *hold with* (3.14) *replaced by*

$$\tau_\alpha(t) = \int_0^t v_\alpha(s)ds, \quad \textit{where } v_\alpha(t) \in U_\alpha(\hat{\theta}(\tau_\alpha(t))), \tag{3.21}$$

and (3.16) *replaced by*

$$\dot{\hat{\theta}}_\alpha(t) = \frac{\bar{g}_\alpha(\hat{\theta}(t))}{v_\alpha(t)} + \hat{z}_\alpha(t), \quad \text{where } v_\alpha(t) \in U_\alpha(\hat{\theta}(t)). \tag{3.22}$$

Remark on the proof. The proof is essentially the same as that of Theorem 3.1. One need only prove (3.21), because (3.22) follows from (3.13), (3.15), and (3.21) by a time transformation. See also Theorems 6.8.1 and 8.2.5 for related results for the synchronized algorithm.

12.3.3 Correlated Noise

Recall that in Chapter 6 and Subsection 8.2.2, we introduced the "memory" random variables ξ_n^ϵ to account for correlated noise. The situation for the asynchronous algorithm is more complicated due to the asynchronicity and the fact that the appropriate "memories" for the $Y_{n,\alpha}^\epsilon$ and $\delta\tau_{n,\alpha}^\epsilon$ sequences might be quite different. For example, consider the queue optimization problem of Chapters 2 and 9, where the IPA estimators were used to compute pathwise derivatives with respect to the parameter. For the asynchronous form of this problem, the "memory" random variables needed to represent the conditional expectation (given the appropriate past) of the Y-terms might have the basic structure of that used in Chapter 9, but the memory random variables needed to represent the conditional expectation (given the appropriate past) of $\delta\tau_{n,\alpha}^\epsilon$ might not involve any of the terms that were used to compute the pathwise derivatives. They might only involve the minimal part of the past record, which is needed to compute the conditional mean of the time to the next update at α. For this reason, in what follows we introduce different memory random variables for the state and time processes. From the point of view of the proofs, this extra complication is purely notational. Note that the time argument in (A3.7) is the time just before the $(n+1)$st update at Processor α minus the delay, and that in (A3.8) is the time at the $(n+2)$nd update at Processor α minus the delay. There can be a separate delay for each component if we wish.

Keep in mind that the memory random variables need not be known to any of the processors. The use of these memory random variables is a convenient trick to obtain convergence proofs under general conditions.

Assumptions for the correlated noise case. Recall the definitions of $\mathcal{F}_{n,\alpha}^\epsilon$ and $\mathcal{F}_{n,\alpha}^{\epsilon,+}$ given in Subsection 3.2. We adapt the noise model used in

Subsection 8.2.2, and enlarge these σ-algebras as follows. For each α let $\xi^\epsilon_{n,\alpha}$ be random variables taking values in some complete separable metric space Ξ, and let $\mathcal{F}^\epsilon_{n,\alpha}$ also measure

$$\{\xi^\epsilon_{j,\gamma} : \gamma, j \text{ such that } \tau^\epsilon_{j,\gamma} < \tau^\epsilon_{n+1,\alpha}\}.$$

Let $\psi^\epsilon_{n,\alpha}$ be random variables taking values in some complete separable metric space Ξ^+ and let $\mathcal{F}^{\epsilon,+}_{n,\alpha}$ measure

$$\{\psi^\epsilon_{j,\gamma} : \gamma, j \text{ such that } \tau^\epsilon_{j,\gamma} \le \tau^\epsilon_{n+1,\alpha}\}.$$

The $\psi^\epsilon_{n,\alpha}$ are the "memory" random variables for the sequence of update times at α. The following conditions, for $\alpha \le r$, will be used in addition to (A3.1) and (A3.6).

(A3.7) There are real-valued measurable functions $g^\epsilon_{n,\alpha}(\cdot)$ and random variables $\beta^\epsilon_{n,\alpha}$ and non-negative (delay times ϵ) $\Delta^\epsilon_{n,\alpha}$ such that

$$E^\epsilon_{n,\alpha} Y^\epsilon_{n,\alpha} = g^\epsilon_{n,\alpha}(\hat{\theta}^\epsilon(\tau^{\epsilon,-}_{n+1,\alpha} - \Delta^\epsilon_{n,\alpha}), \xi^\epsilon_{n,\alpha}) + \beta^\epsilon_{n,\alpha},$$

and (3.7) and (3.8) hold.

(A3.8) There are strictly positive (in the sense of (2.3)) measurable functions $u^\epsilon_{n,\alpha}(\cdot)$ and non-negative (delay times ϵ) random variables $\Delta^{\epsilon,+}_{n,\alpha}$ such that

$$E^{\epsilon,+}_{n,\alpha} \delta\tau^\epsilon_{n+1,\alpha} = u^\epsilon_{n+1,\alpha}(\hat{\theta}^\epsilon(\tau^\epsilon_{n+1,\alpha} - \Delta^{\epsilon,+}_{n,\alpha}), \psi^\epsilon_{n+1,\alpha}),$$

and (3.10) holds.

(A3.9) For each compact set $A \subset \Xi$, $g^\epsilon_{n,\alpha}(\cdot, \xi)$ is continuous in θ uniformly in n, ϵ and in $\xi \in A$.

(A3.10) For each compact set $A^+ \subset \Xi^+$, $u^\epsilon_{n,\alpha}(\cdot, \psi)$ is continuous in θ uniformly in n, ϵ and in $\psi \in A^+$.

(A3.11) For each $\mu > 0$ there are compact sets $A_\mu \subset \Xi$ and $A^+_\mu \subset \Xi^+$ such that

$$\inf_{n,\epsilon} P\left\{\xi^\epsilon_{n,\alpha} \in A_\mu\right\} \ge 1 - \mu$$

and

$$\inf_{n,\epsilon} P\left\{\psi^\epsilon_{n,\alpha} \in A^+_\mu\right\} \ge 1 - \mu.$$

(A3.12) Each of the sets

$$\{g^\epsilon_{n,\alpha}(\theta, \xi^\epsilon_{n,\alpha}), u^\epsilon_{n,\alpha}(\theta, \psi^\epsilon_{n,\alpha}); \epsilon, n\}, \quad \text{for } \theta \in H \tag{3.23}$$

is uniformly integrable.

(A3.13) There are continuous functions $\bar{g}_\alpha(\cdot)$ such that for each $\theta \in H$,

$$\lim_{m,n,\epsilon} \frac{1}{m} \sum_{i=n}^{n+m-1} E^\epsilon_{n,\alpha} \left[g^\epsilon_{i,\alpha}(\theta, \xi^\epsilon_{i,\alpha}) - \bar{g}_\alpha(\theta)\right] = 0$$

in probability as n and m go to infinity and $\epsilon \to 0$.

(A3.14) There are continuous and positive valued functions $\bar{u}_\alpha(\cdot)$ such that for each $\theta \in H$,

$$\lim_{m,n,\epsilon} \frac{1}{m} \sum_{i=n}^{n+m-1} E_{n,\alpha}^{\epsilon,+} \left[u_{i+1,\alpha}^\epsilon(\theta, \psi_{i+1,\alpha}^\epsilon) - \bar{u}_\alpha(\theta)\right] = 0$$

in probability as n and m go to infinity and $\epsilon \to 0$.

We can now state the following theorem.

Theorem 3.3. *Assume* (A3.1) *and* (A3.6)–(A3.14). *Then the conclusions of Theorem* 3.1 *hold. The extension of Theorem* 3.1 *when the constraint set* H *is dropped also holds if* $\{\theta_{n,\alpha}^\epsilon; n, \epsilon\}$ *is bounded with probability one.*

Comment on the proof. Referring to the proof of Theorem 3.1, redefine $\tilde{G}_\alpha^\epsilon(\cdot)$ to be

$$\tilde{G}_\alpha^\epsilon(t) = \epsilon \sum_{i=0}^{t/\epsilon-1} \left[g_{i,\alpha}^\epsilon(\hat{\theta}^\epsilon(\tau_{i+1,\alpha}^{\epsilon,-} - \Delta_{i,\alpha}^\epsilon), \xi_{i,\alpha}^\epsilon) - \bar{g}_\alpha(\hat{\theta}^\epsilon(\tau_{i+1,\alpha}^{\epsilon,-} - \Delta_{i,\alpha}^\epsilon))\right],$$

or with the analogous changes required for treating the $\tau_\alpha^\epsilon(\cdot)$. Then the proof is an adaptation of that of Theorem 3.1 in the same way that the proof of Theorem 8.2.2 adapted that of Theorem 8.2.1; the details are omitted.

The next theorem is one useful "differential inclusions" extension.

Theorem 3.4. *Assume the conditions of Theorem* 3.3, *but replace* (A3.14) *with the following. There are sets* $U_\alpha(\theta), \alpha = 1, \ldots, r$, *that are upper semicontinuous in the sense of* (4.3.2) *such that*

$$\text{distance}\left[\frac{1}{m}\sum_{i=n}^{n+m-1} E_{n,\alpha}^{\epsilon,+} u_{i+1,\alpha}^\epsilon(\theta, \psi_{i+1,\alpha}^\epsilon), U_\alpha(\theta)\right] \to 0 \tag{3.24}$$

in probability as n *and* m *go to infinity and* $\epsilon \to 0$. *Then the conclusions of Theorem* 3.3 *hold with the mean limit ODEs being* (3.21) *and* (3.22).

12.3.4 Infinite Time Analysis

We now turn to the asymptotic properties of $\hat{\theta}^\epsilon(T+\cdot)$ as $T \to \infty$ and $\epsilon \to 0$. Recall the discussion in Section 2 (and the comments below the statement of Theorem 3.1) concerning the conditions (2.8) and (2.9) that were needed to deal with the convergence of $\hat{\theta}^\epsilon(t)$ as $t \to \infty$ and $\epsilon \to 0$. Those conditions were needed, because the analysis would be done on processes $\{\hat{\theta}^\epsilon(T_\epsilon + \cdot)\}$

that effectively "start" at large and arbitrary real times. The same issue is more serious in the asynchronous case. We require that the conditions of Theorem 3.3 or 3.4 hold for the processes starting at arbitrary real times. Recall the definition of $p^\epsilon_\alpha(\sigma)$ given below (3.2). The following notation for the process starting at real time σ will be needed.

Definitions. The concepts to be used are as before but under an arbitrary translation of the time origin. This accounts for the various terms that will now be defined. For each $\sigma > 0$, $n \geq 0$, and $\alpha \leq r$, define

$$\theta^{\epsilon,\sigma}_{n,\alpha} = \theta^\epsilon_{p^\epsilon_\alpha(\sigma)+n,\alpha},$$

$$Y^{\epsilon,\sigma}_{n,\alpha} = Y^\epsilon_{p^\epsilon_\alpha(\sigma)+n,\alpha},$$

and the conditional expectations

$$E^{\epsilon,\sigma}_{n,\alpha} = E^\epsilon_{p^\epsilon_\alpha(\sigma)+n,\alpha}, \quad E^{\epsilon,\sigma,+}_{n,\alpha} = E^{\epsilon,+}_{p^\epsilon_\alpha(\sigma)+n,\alpha}$$

and define

$$\delta\tau^{\epsilon,\sigma}_{n,\alpha},\ \beta^{\epsilon,\sigma}_{n,\alpha},\ \Delta^{\epsilon,\sigma}_{n,\alpha},\ \Delta^{\epsilon,\sigma,+}_{n,\alpha},\ \xi^{\epsilon,\sigma}_{n,\alpha},\ \psi^{\epsilon,\sigma}_{n,\alpha},\ g^{\epsilon,\sigma}_{n,\alpha}(\cdot),\ u^{\epsilon,\sigma}_{n,\alpha}(\cdot)$$

analogously. Define

$$\tau^{\epsilon,\sigma}_{n,\alpha} = \epsilon \sum_{i=0}^{n-1} \delta\tau^{\epsilon,\sigma}_{i,\alpha}.$$

Define the interpolations, for $t \geq 0$:

$$\begin{aligned}
\theta^{\epsilon,\sigma}_\alpha(t) &= \theta^{\epsilon,\sigma}_{n,\alpha}, \quad t \in [n\epsilon, n\epsilon + \epsilon),\\
\hat{\theta}^{\epsilon,\sigma}_\alpha(t) &= \theta^{\epsilon,\sigma}_{n,\alpha}, \quad t \in [\tau^{\epsilon,\sigma}_{n,\alpha}, \tau^{\epsilon,\sigma}_{n+1,\alpha}),\\
\tau^{\epsilon,\sigma}_\alpha(t) &= \tau^{\epsilon,\sigma}_{n,\alpha}, \quad t \in [n\epsilon, n\epsilon + \epsilon),\\
N^{\epsilon,\sigma}_\alpha(t) &= n\epsilon, \quad\ \ t \in [\tau^{\epsilon,\sigma}_{n,\alpha}, \tau^{\epsilon,\sigma}_{n+1,\alpha}).
\end{aligned}$$

Thus, $\hat{\theta}^\epsilon_\alpha(\cdot)$ "starts" at the time of the first update at Processor α at or after real time σ. Analogously to (3.3),

$$N^{\epsilon,\sigma}_\alpha(\tau^{\epsilon,\sigma}_\alpha(t)) = n\epsilon, \ t \in [n\epsilon, n\epsilon + \epsilon).$$

Definitions analogous to these are illustrated in Section 4 for the decreasing step size case.

Assumptions. The conditions for each α are just those for Theorem 3.3, but for arbitrary real starting times σ. The conditions are actually simpler than they might appear at first sight, particularly when the update times have a simple "renewal" character.

(A3.1′) $\{Y^{\epsilon,\sigma}_{n,\alpha}, \delta\tau^{\epsilon,\sigma}_{n,\alpha}; \epsilon, \alpha, n, \sigma\}$ is uniformly integrable.

(A3.6′) $\lim_{m,n,\epsilon,\sigma} \frac{1}{m} \sum_{i=n}^{n+m-1} E_{n,\alpha}^{\epsilon,\sigma} \beta_{i,\alpha}^{\epsilon,\sigma} = 0$ in mean.

(A3.11′) For each $\mu > 0$, there are compact sets $A_\mu \subset \Xi$ and $A_\mu^+ \subset \Xi^+$such that for each α,

$$\inf_{n,\epsilon,\sigma} P\left\{\xi_{n,\alpha}^{\epsilon,\sigma} \in A_\mu\right\} \geq 1 - \mu$$

and

$$\inf_{n,\epsilon,\sigma} P\left\{\psi_{n,\alpha}^{\epsilon,\sigma} \in A_\mu^+\right\} \geq 1 - \mu.$$

(A3.12′) For each α, the sets

$$\{\beta_{n,\alpha}^{\epsilon,\sigma}; \epsilon, n, \sigma\}$$

and

$$\{g_{n,\alpha}^{\epsilon,\sigma}(\theta, \xi_{n,\alpha}^{\epsilon,\sigma}), u_{n,\alpha}^{\epsilon,\sigma}(\theta, \psi_{n,\alpha}^{\epsilon,\sigma}); \epsilon, n, \sigma\}$$

are uniformly integrable for each $\theta \in H$.

(A3.13′) There are continuous real-valued functions $\bar{g}_\alpha(\cdot)$ such that for each $\theta \in H$,

$$\lim_{m,n,\epsilon,\sigma} \frac{1}{m} \sum_{i=n}^{n+m-1} E_{n,\alpha}^{\epsilon,\sigma} \left[g_{i,\alpha}^{\epsilon,\sigma}(\theta, \xi_{i,\alpha}^{\epsilon,\sigma}) - \bar{g}_\alpha(\theta)\right] = 0$$

in probability as n, m, and σ go to infinity and $\epsilon \to 0$.

(A3.14′) There are continuous and positive real-valued functions $\bar{u}_\alpha(\cdot)$ such that for each $\theta \in H$,

$$\lim_{m,n,\epsilon,\sigma} \frac{1}{m} \sum_{i=n}^{n+m-1} E_{n,\alpha}^{\epsilon,\sigma,+} \left[u_{i+1,\alpha}^{\epsilon,\sigma}(\theta, \psi_{i+1,\alpha}^{\epsilon,\sigma}) - \bar{u}_\alpha(\theta)\right] = 0$$

in probability as n, m, and σ go to infinity and $\epsilon \to 0$.

(A3.15a) $\lim_{\epsilon} \sup_{\sigma, 0 \leq n \leq T/\epsilon} \Delta_{n,\alpha}^{\epsilon,\sigma} = 0$ in probability, for each $T > 0$.

(A3.15b) $\lim_{\epsilon} \sup_{\sigma, 0 \leq n \leq T/\epsilon} \Delta_{n,\alpha}^{\epsilon,\sigma,+} = 0$ in probability, for each $T > 0$.

The next theorem is simply a restatement of the previous theorems except that time goes to infinity and hence the limit trajectories are supported essentially on the limit set of the ODE. The few extra details of the proof are omitted. The comments below Theorem 8.2.1 concerning the limit set apply here as well.

Theorem 3.5. *Assume the above conditions and* (A3.7)–(A3.10). *Then*

$$\{\theta_\alpha^{\epsilon,\sigma}(\cdot), \tau_\alpha^{\epsilon,\sigma}(\cdot), \hat{\theta}^{\epsilon,\sigma}(\cdot), N_\alpha^{\epsilon,\sigma}(\cdot); \alpha \leq r; \epsilon, \sigma\}$$

is tight in $D^{4r}[0,\infty)$. *Let* (ϵ,σ) *(abusing terminology for simplicity) index a weakly convergent (as* $\epsilon \to 0$ *and* $\sigma \to \infty$*) subsequence whose weak sense limit we denote by*

$$(\theta_\alpha(\cdot), \tau_\alpha(\cdot), \hat\theta(\cdot), N_\alpha(\cdot), \alpha \le r).$$

Equations (3.12)–(3.16) *hold.*

For any $\delta > 0$, *the fraction of time* $\hat\theta^\epsilon(\epsilon q_\epsilon + \cdot)$ *spends in* $N_\delta(L_H)$ *on* $[0,T]$ *goes to one (in probability) as* $\epsilon \to 0$ *and* $T \to \infty$. *Let* $\epsilon q_\epsilon \to \infty$ *as* $\epsilon \to 0$. *Then for almost all* ω, *the path* $\hat\theta(\omega,\cdot)$ *lies in an invariant set of* (3.16). *If* L_H *is asymptotically stable in the sense of Liapunov, then that invariant set is in* L_H.

For the case of differential inclusions, replace (A3.14′) *and* (3.24) *by*

$$\text{distance}\left[\frac{1}{m}\sum_{i=n}^{n+m-1} E_{n,\alpha}^{\epsilon,\sigma,+} u_{i+1,\alpha}^{\epsilon,\sigma}(\theta, \psi_{i+1,\alpha}^{\epsilon,\sigma}), U_\alpha(\theta)\right] \to 0 \qquad (3.25)$$

in probability, where $U_\alpha(\cdot)$ *is upper semicontinuous in the sense of* (4.3.2). *Then the above assertions hold with the ODE* (3.22).

Now, drop the constraint set H *and suppose that* $\{\theta_{n,\alpha}^\epsilon; n, \epsilon, \alpha\}$ *is bounded with probability one. Then the above conclusions continue to hold with* $z(t) = 0$ *and* L_H *replaced by a bounded limit set of the unconstrained ODE.*

12.4 Decreasing Step Size

The decreasing step size algorithm is treated in virtually the same way as the constant step size algorithm. The main differences either are notational or they concern the choice of step size. We will choose a "canonical form," which will be seen to be quite versatile.

Let $\delta\tau_{n,\alpha}$ denote the time between the nth and $(n+1)$st updates for Processor α. Define (not scaled by the step sizes)

$$T_{n,\alpha} = \sum_{i=0}^{n-1} \delta\tau_{i,\alpha}, \qquad (4.1)$$

Thus, $T_{n,\alpha}$ is the real time of the nth update for Processor α. For $\sigma \ge 0$, define

$$p_\alpha(\sigma) = \min\{n : T_{n,\alpha} \ge \sigma\}. \qquad (4.2)$$

The step size sequence. There are several essentially equivalent sequences of step sizes that can be used. The basic issue in the choice is notational. We will adopt a choice that is versatile and for which the notation is not too complex. There are many variations for which the basic notational structure can be used. The basic step size will be a function of real time, and as such will be the same for all processors. More precisely, we suppose that there is a positive real-valued function $\epsilon(\cdot)$ defined on $[0, \infty)$ such that the step size for the α-processor at the $(n+1)$st update is

$$\epsilon_{n,\alpha} = \frac{1}{\delta\tau_{n,\alpha}} \int_{T_{n,\alpha}}^{T_{n,\alpha}+\delta\tau_{n,\alpha}} \epsilon(s)ds. \tag{4.3}$$

The algorithm can then be written as

$$\theta_{n+1,\alpha} + \Pi_{[a_\alpha,b_\alpha]}\left(\theta_{n,\alpha} + \epsilon_{n,\alpha}Y_{n,\alpha}\right), \quad \text{for } \alpha \le r. \tag{4.4}$$

The selected form is actually not very restrictive, and by suitable transformations it will model other reasonable sequences. For example, consider a more traditional alternative, where a separate sequence $\{\epsilon_{n,\alpha}\}$ is given *a priori* for each Processor α. Hence, the step size is indexed by the processor and the update number at that processor. Then the ratio of the step sizes used at the first update at or after real time s for Processors α and γ is

$$\frac{\epsilon_{p_\alpha(s),\alpha}}{\epsilon_{p_\gamma(s),\gamma}} = \frac{\epsilon_{s[p_\alpha(s)/s],\alpha}}{\epsilon_{s[p_\gamma(s)/s],\gamma}}. \tag{4.5}$$

Let there be positive numbers v_α such that $\lim_s p_\alpha(s)/s = v_\alpha$ in the sense of convergence in probability. Then, for large s, (4.5) is approximately $\epsilon_{v_\alpha s,\alpha}/\epsilon_{v_\gamma s,\gamma}$. If we use the common form $\epsilon_{n,\alpha} = K_\alpha/n^a, a \in (0,1]$, for some positive numbers K_α, then this last ratio will not depend on s. Suppose that the ratios in (4.5) do not (asymptotically) depend on s for any pair (α, γ). Then for large real time, the step sizes for different components essentially differ by a constant factor. If these constant factors are incorporated into the Y_n^α terms, then for all practical purposes there is a common $\epsilon(\cdot)$ whose use will yield the same asymptotic results as the use of the separate $\epsilon_{n,\alpha}$ sequences. This common step size can take the form $\epsilon(s) = \epsilon_{v_\alpha s,\alpha}$ for any particular value of α.

Comment on different time scales for different components. Suppose that for some (but not all) α either $p_\alpha(s)/s \to 0$ or $[p_\alpha(T_s + s) - p_\alpha(T_s)]/s \to 0$ for some sequence $T_s \to \infty$ with positive probability. Then the sequence $\{\delta\tau_{n,\alpha}\}$ will not be uniformly integrable. The problem takes on a singular character in that some components are on a different time scale than others. The algorithm can still be analyzed and convergence proved under conditions similar to what we use. The techniques are similar, but they require taking into account the fact that some processors work much faster than others. The part of the limit mean ODE corresponding to the faster components is replaced by the limit point as a function of the slower components assuming that the slower components are held fixed. Then in the part of the ODE corresponding to the slower components, the faster components are replaced by these limit points. At this time there is little practical reason for further analysis of such singular stochastic approximation algorithms; see [97] for the analysis of many types of singularly perturbed stochastic systems and [80] for the analysis of deterministic singularly perturbed systems.

Definitions. In Chapters 5, 6, and 8, the analysis of the algorithm with decreasing step size used the shifted processes $\theta^n(\cdot)$, which started at the nth iteration. Since the iteration times are not synchronized among the processors, as for the constant step size case, we continue to work in real time. The interpolated process $\theta^n(\cdot)$ will be replaced by the "shifted" process whose time origin is $\sigma > 0$. The following definitions will be needed. Keep in mind that $p_\alpha(\sigma) + n$ is the index of the update, which is the $(n+1)$st update at or after time σ and $T_{p_\alpha(\sigma)+n,\alpha}$ is the real time at which that update occurs. Figures 4.1–4.5 illustrate the following definitions.

For each $\sigma \geq 0$ and $n \geq 0$, define

$$\begin{aligned} \delta\tau^\sigma_{n,\alpha} &= \delta\tau_{p_\alpha(\sigma)+n,\alpha}, \\ \epsilon^\sigma_{n,\alpha} &= \epsilon_{p_\alpha(\sigma)+n,\alpha}, \\ t^\sigma_{n,\alpha} &= \sum_{i=0}^{n-1} \epsilon^\sigma_{i,\alpha}, \end{aligned} \tag{4.6}$$

and

$$\begin{aligned} \tau^\sigma_{n,\alpha} &= \sum_{i=0}^{n-1} \epsilon^\sigma_{i,\alpha} \delta\tau^\sigma_{i,\alpha}, \\ Y^\sigma_{n,\alpha} &= Y_{p_\alpha(\sigma)+n,\alpha}. \end{aligned} \tag{4.7}$$

Figure 4.1. The iterates up to time σ for Processor α.

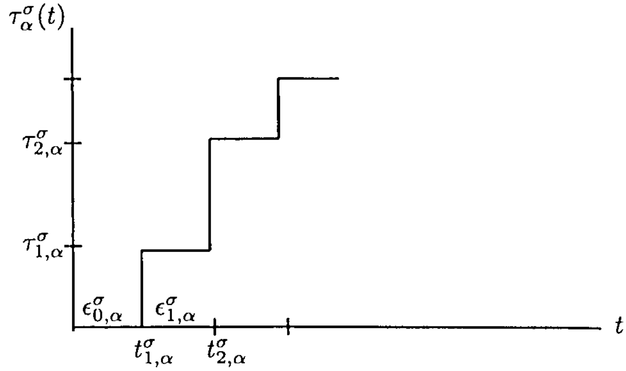

Figure 4.2. The function $\tau_\alpha^\sigma(t)$.

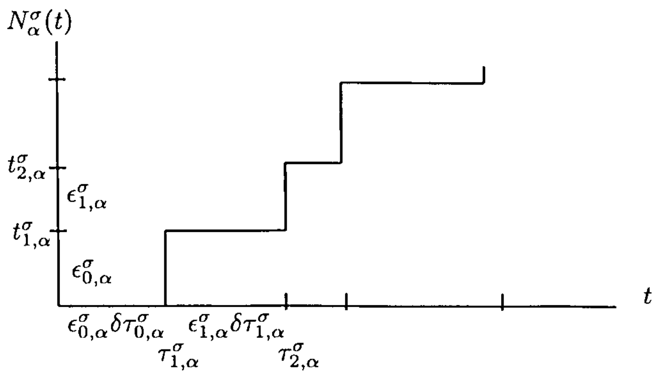

Figure 4.3. The function $N_\alpha^\sigma(t)$.

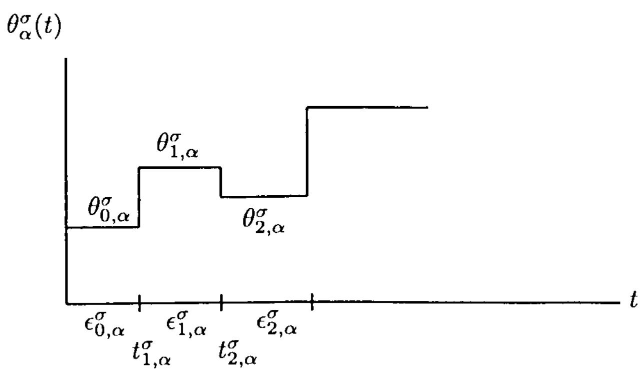

Figure 4.4. The function $\theta_\alpha^\sigma(t)$.

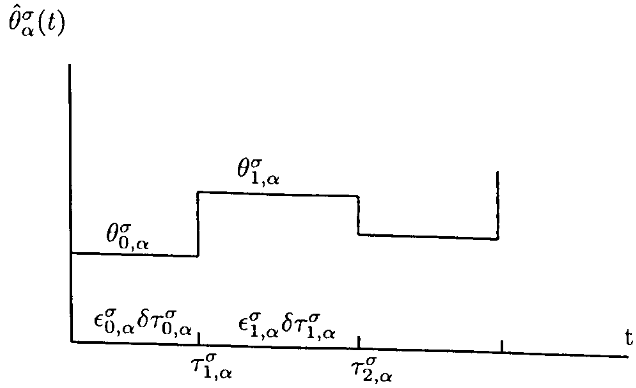

Figure 4.5. The function $\hat{\theta}^\sigma_\alpha(t)$.

Thus, $\delta\tau^\sigma_{0,\alpha}, \delta\tau^\sigma_{1,\alpha}, \ldots$, are the interupdate times for Processor α starting at the first update at or after σ, and $\epsilon^\sigma_{0,\alpha}, \epsilon^\sigma_{1,\alpha}, \ldots$, are the associated step sizes. Also, $\tau^\sigma_{n,\alpha}$ is the integral of $\epsilon(\cdot)$ from the time of the first update at Processor α at or after time σ until the time of the nth update there. In general, for each $\sigma \geq 0$ and a sequence $\{X_{n,\alpha}\}$, we define $X^\sigma_{n,\alpha} = X_{p_\alpha(\sigma)+n,\alpha}$.

The sequence of updates for Processor α starting at the first update at or after time σ can be written as follows. Define the sequence $\{\theta^\sigma_{n,\alpha}\}$ by $\theta^\sigma_{0,\alpha} = \theta_{p_\alpha(\sigma),\alpha}$, its value at the first update at or after time σ. For $n \geq 0$, we have

$$\theta^\sigma_{n+1,\alpha} = \Pi_{[a_\alpha,b_\alpha]}\left(\theta^\sigma_{n,\alpha} + \epsilon^\sigma_{n,\alpha}Y^\sigma_{n,\alpha}\right). \tag{4.8}$$

Define the interpolations

$$\theta^\sigma_\alpha(t) = \theta^\sigma_{n,\alpha},\ t \in [t^\sigma_{n,\alpha}, t^\sigma_{n+1,\alpha}), \tag{4.9}$$

$$\hat{\theta}^\sigma_\alpha(t) = \theta^\sigma_{n,\alpha},\ t \in [\tau^\sigma_{n,\alpha}, \tau^\sigma_{n+1,\alpha}), \tag{4.10}$$

$$N^\sigma_\alpha(t) = t^\sigma_{n,\alpha},\ t \in [\tau^\sigma_{n,\alpha}, \tau^\sigma_{n+1,\alpha}), \tag{4.11}$$

$$\tau^\sigma_\alpha(t) = \tau^\sigma_{n,\alpha},\ t \in [t^\sigma_{n,\alpha}, t^\sigma_{n+1,\alpha}). \tag{4.12}$$

Define the vector $\hat{\theta}^\sigma(\cdot) = \{\hat{\theta}^\sigma_\alpha(\cdot), \alpha \leq r\}$.

Analogously to the case in Section 3,

$$N^\sigma_\alpha(\tau^\sigma_\alpha(t)) = t^\sigma_{n,\alpha},\ t \in [t^\sigma_{n,\alpha}, t^\sigma_{n+1,\alpha}), \tag{4.13}$$

$$\theta^\sigma_\alpha(t) = \hat{\theta}^\sigma_\alpha(\tau^\sigma_\alpha(t)),\ \hat{\theta}^\sigma_\alpha(t) = \theta^\sigma_\alpha(N^\sigma_\alpha(t)). \tag{4.14}$$

The processes $\theta^\sigma_\alpha(\cdot)$ and $\hat{\theta}^\sigma_\alpha(\cdot)$ are interpolations of the sequences $\{\theta^{\epsilon,\sigma}_{n,\alpha}, n \geq 0\}$, starting at the first update at or after σ. The interpolation $\theta^\epsilon_\alpha(\cdot)$ is in the "iterate" scale (scaled by $\epsilon(\cdot)$), and the interpolation $\hat{\theta}^\epsilon_\alpha(\cdot)$ is in a scaled (by the $\epsilon(\cdot)$ process) real time. Since the real time at which the first update at or after σ occurs might differ among the components, the processes $\hat{\theta}^\sigma_\alpha(\cdot)$,

$\alpha = 1, \ldots, r$, are not quite properly aligned with each other in time, but the misalignment goes to zero as $\sigma \to \infty$.

Definitions. Analogously to what was done in Subsection 3.2, for each α we will need the nondecreasing families of σ-algebras $\mathcal{F}_{n,\alpha}$ and $\mathcal{F}^+_{n,\alpha}$ defined as follows. The σ-algebra $\mathcal{F}_{n,\alpha}$ measures at least the random variables

$$\begin{aligned}&\theta_0, \ \{Y_{j-1,\gamma}; j, \gamma \text{ such that } T_{j,\gamma} < T_{n+1,\alpha}\}, \quad \text{and} \\ &\{T_{j,\gamma}; j, \gamma \text{ such that } T_{j,\gamma} \leq T_{n+1,\alpha}\}.\end{aligned}$$

Let $E_{n,\alpha}$ denote the expectation conditioned on $\mathcal{F}_{n,\alpha}$. The σ-algebra $\mathcal{F}^+_{n,\alpha}$ measures at least the random variables

$$\begin{aligned}&\theta_0, \ \{Y_{j-1,\gamma}; j, \gamma \text{ such that } T_{j,\gamma} \leq T_{n+1,\alpha}\}, \quad \text{and} \\ &\{T_{j,\gamma}; j, \gamma \text{ such that } T_{j,\gamma} \leq T_{n+1,\alpha}\}.\end{aligned}$$

The associated conditional expectation is $E^+_{n,\alpha}$. $E_{n,\alpha}$ replaces $E^\epsilon_{n,\alpha}$ and $E^+_{n,\alpha}$ replaces $E^{\epsilon,+}_{n,\alpha}$.

Assumptions. Except for those concerning the step size, the assumptions are obvious modifications of those used in the previous theorems, and the reader should have little difficulty in making the notational changes.

The step size assumptions are:

(A4.1) $\int_0^\infty \epsilon(s)ds = \infty, \quad 0 < \epsilon(s) \to 0.$

(A4.2) There are real $T(s) \to \infty$ as $s \to \infty$ such that

$$\lim_{s\to\infty} \sup_{0\leq t\leq T(s)} \left| \frac{\epsilon(s)}{\epsilon(s+t)} - 1 \right| = 0.$$

Theorem 4.1. Assume (A4.1)–(A4.2) *and the obvious modifications of the conditions of Theorem* 3.5. *Then the conclusions of that theorem hold.*

12.5 State Dependent Noise

The (Markov) state dependent noise case is treated by the methods of Section 8.4, modified by the techniques of this chapter to deal with the asynchronicity. The notation and assumptions will be stated. The proof is a straightforward combination of the methods of Section 8.4 and Section 3 and the details will not be given. The setup will be based on the approach for Theorem 8.4.1, but the approach of Theorem 8.4.3 or 8.4.4 can also be used.

As noted in Section 3, there are two "memory random variables," called $\xi^\epsilon_{n,\alpha}$ and $\psi^\epsilon_{n,\alpha}$. The first (resp., second) was used to represent the expectation of the observation (resp., time between the observations) at Processor α, given the "past." The same arrangement is useful for the state dependent noise case. However, to adapt the setup in Section 8.4 to this situation there must be two sets of "fixed-θ processes"; namely, $\xi_{n,\alpha}(\theta)$ and $\psi_{n,\alpha}(\theta)$, $\alpha \le r$. This helps explain the following conditions. Since the $\xi^\epsilon_{n,\alpha}$ and $\psi^\epsilon_{n,\alpha}$ concern the "effective memory" at the update times, we have some control over them in the design of the stochastic approximation procedure.

We will work with the constant step size algorithm, use the terminology of Subsection 3.3, and use (A3.1), (A3.6)–(A3.8) and (A3.11), as well as the following assumptions, where $\alpha = 1, \ldots, r$. The delays in (5.1) and (A5.4) can depend on the component of $\hat\theta$.

(A5.1) There are transition functions $P^\epsilon_{n,\alpha}(\cdot,\cdot|\theta)$ such that $P^\epsilon_{n,\alpha}(\cdot, A|\cdot)$ is measurable for each Borel set A in the range space Ξ of $\xi^\epsilon_{n,\alpha}$ and

$$P\left\{\xi^\epsilon_{n+1,\alpha} \in \cdot | \mathcal{F}^\epsilon_{n,\alpha}\right\} = P^\epsilon_{n,\alpha}(\xi^\epsilon_{n,\alpha}, \cdot | \hat\theta^\epsilon(\tau^{\epsilon,-}_{n+1,\alpha} - \Delta^\epsilon_{n,\alpha})). \tag{5.1}$$

(A5.2) For each fixed θ, there is a transition function $P_\alpha(\xi,\cdot|\theta)$ such that

$$P^\epsilon_{n,\alpha}(\xi,\cdot|\theta) \Rightarrow P_\alpha(\xi,\cdot|\theta), \quad \text{as } n \to \infty, \epsilon \to 0, \tag{5.2}$$

where the limit is uniform on each compact (θ,ξ) set; that is, for each bounded and continuous real-valued function $F(\cdot)$ on Ξ,

$$\int F(\tilde\xi) P^\epsilon_{n,\alpha}(\xi, d\tilde\xi|\theta) \to \int F(\tilde\xi) P_\alpha(\xi, d\tilde\xi|\theta) \tag{5.3}$$

uniformly on each compact (θ,ξ)-set, as $n \to \infty$ and $\epsilon \to 0$.

(A5.3) $P_\alpha(\xi,\cdot|\theta)$ is weakly continuous in (θ,ξ).

(A5.4) The analogs of (A5.1)–(A5.3) hold for the $\psi^\epsilon_{n,\alpha}$, where the (multiplied by ϵ) delays are $\Delta^{\epsilon,+}_{n,\alpha}$. Let $P^+(\cdot|\theta)$ denote the analog of $P(\cdot|\theta)$.

The fixed-θ Markov chains. For each α and fixed θ, $P_\alpha(\cdot,\cdot|\theta)$ is the transition function of a Markov chain with state space Ξ, analogous to the situation in Section 8.4. This chain is referred to as the *fixed-θ chain* and the random variables of the chain are denoted by $\xi_{n,\alpha}(\theta)$. Unless otherwise specified, when using the expression $E^\epsilon_{n,\alpha}F(\xi_{n+j,\alpha}(\theta))$ for $j \ge 0$, the conditional expectation is for the fixed-θ chain

$$\{\xi_{n+j,\alpha}(\theta), j \ge 0;\ \xi_{n,\alpha}(\theta) = \xi^\epsilon_{n,\alpha}\}, \tag{5.4}$$

which starts at time n with initial condition $\xi_{n,\alpha}(\theta) = \xi^\epsilon_{n,\alpha}$ and θ is given. These fixed-θ chains play the same role here that they do in the synchronized case of Section 8.4. The fixed-θ chains $\psi_{n,\alpha}(\theta)$ are defined analogously, using $P^+(\cdot|\theta)$.

(A5.5) $g^\epsilon_{n,\alpha}(\cdot)$ and $u^\epsilon_{n,\alpha}(\cdot)$ are continuous on each compact (θ,ξ)-set, uniformly in ϵ and n.

(A5.6) For any compact sets $A_\alpha \subset \Xi$ and $A^+_\alpha \subset \Xi^+$ and for each $\theta \in H$,

$$\{g^\epsilon_{n+j,\alpha}(\theta, \xi_{n+j,\alpha}(\theta)); j \geq 0, \text{ all } \xi_{n,\alpha}(\theta) \in A_\alpha;\ n \geq 0\} \text{ and} \tag{5.5}$$

$$\{u^\epsilon_{n+j,\alpha}(\theta, \psi_{n+j,\alpha}(\theta)); j \geq 0, \text{ all } \psi_{n,\alpha}(\theta) \in A^+_\alpha;\ n \geq 0\} \tag{5.6}$$

are uniformly integrable.

(A5.7) There are continuous functions $\bar{g}_\alpha(\cdot)$ and $\bar{u}_\alpha(\cdot)$ such that for each compact set $A_\alpha \subset \Xi$ and $A^+_\alpha \subset \Xi^+$,

$$\frac{1}{m}\sum_{i=n}^{n+m-1} E^\epsilon_{n,\alpha}\left[g^\epsilon_{i,\alpha}(\theta, \xi_{i,\alpha}(\theta)) - \bar{g}_\alpha(\theta)\right] I_{\{\xi^\epsilon_{n,\alpha} \in A_\alpha\}} \to 0, \tag{5.7}$$

$$\frac{1}{m}\sum_{i=n}^{n+m-1} E^{\epsilon,+}_{n,\alpha}\left[u^\epsilon_{i,\alpha}(\theta, \psi_{i,\alpha}(\theta)) - \bar{u}_\alpha(\theta)\right] I_{\{\psi^\epsilon_{n,\alpha} \in A^+_\alpha\}} \to 0 \tag{5.8}$$

in the mean for each θ, as n and m go to infinity and $\epsilon \to 0$.

(A5.8) For each compact set $A_\alpha \subset \Xi$ and $A^+_\alpha \subset \Xi^+$ and each $\mu > 0$, there are compact sets $A_{\alpha,\mu} \subset \Xi$ and $A^+_{\alpha,\mu} \subset \Xi^+$ such that

$$P\left\{\xi_{n+j,\alpha}(\theta) \in A_{\alpha,\mu} \middle| \xi_{n,\alpha}(\theta)\right\} \geq 1 - \mu,$$

for all $\theta \in H, j > 0, n > 0$, and $\xi_{n,\alpha}(\theta) \in A_\alpha$, and

$$P\left\{\psi_{n+j,\alpha}(\theta) \in A^+_{\alpha,\mu} \middle| \psi_{n,\alpha}(\theta)\right\} \geq 1 - \mu,$$

for all $\theta \in H, j > 0, n > 0$, and $\psi_{n,\alpha}(\theta) \in A^+_\alpha$.

The proof of the following theorem uses the methods of Section 8.4 adjusted by the scaling techniques of this chapter; the details are left to the reader. The comments below Theorem 8.2.1 concerning the limit set hold here as well.

Theorem 5.1. *Assume* (A3.1), (A3.6)–(A3.8), (A3.11) *and* (A5.1)–(A5.8). *Then the conclusions of Theorem* 3.3 *hold. The conclusions of Theorem* 3.5 *hold if the assumptions are replaced by those on the σ-shifted sequences analogous to what was done in Theorem* 3.5. *The replacement for* (3.24) *is: For each compact set $A^+_\alpha \subset \Xi$,*

$$\text{distance}\left[\frac{1}{m}\sum_{i=n}^{n+m-1} E^+_{n,\alpha} u^\epsilon_{i,\alpha}(\theta, \psi_{i,\alpha}(\theta)), U_\alpha(\theta)\right] I_{\{\psi^\epsilon_{n,\alpha} \in A^+_\alpha\}} \to 0 \tag{5.9}$$

in the mean uniformly in the initial condition, as n and m go to infinity.

12.6 Rate of Convergence: The Limit Rate Equations

The rate of convergence analysis of Chapter 10 can be applied to the algorithms of the previous sections. The basic methods of proof combine the techniques of Chapter 10 with those used previously in this chapter. The previous results of this chapter were proved by working basically in "iterate time" for each component and then using a time change argument. To obtain the rate of convergence results, it is simpler if we work directly in (scaled) real time. Because of this, it is convenient to restrict the problem slightly. The restriction is of no practical consequence.

It will be supposed that there is some small number so that the updates at any processor can occur only at times that are integral multiples of this number. There is no loss of generality in doing this since the number can be as small as desired, and the procedure is insensitive to small variations. Having made this assumption, without loss of generality we can suppose that the basic interval is one unit of time. The basic structure of Subsection 3.3 will be retained. Thus we work with a constant step size and weak convergence. There are also full analogs of what was done in Chapter 10 for the decreasing step size case under either weak or probability one convergence. In Section 10.1, we were concerned with the asymptotic behavior of $(\theta_n - \bar{\theta})/\sqrt{\epsilon}$ as $\epsilon \to 0$ and $n\epsilon \to \infty$ fast enough, and we will treat the analogous situation here.

With the above assumption that the updates can occur only at integral units of time, let $\hat{\theta}^\epsilon_n = \{\hat{\theta}^\epsilon_{n,\alpha}, \alpha = 1, \ldots, r\}$ denote the value of the iterate at real time n. This is not to be confused with the iteration number. We retain the other notations of the previous sections. Recall, in particular, that $\tau^\epsilon_{n,\alpha}$ is the real time of the nth update at Processor α. Let $I^\epsilon_{n,\alpha}$ denote the indicator of the event that the αth component is updated at time $n+1$; that is, that $\hat{\theta}^\epsilon_{n+1,\alpha}$ differs from $\hat{\theta}^\epsilon_{n,\alpha}$. Define the random variables $\hat{\xi}^\epsilon_{n,\alpha}$, $\hat{Y}^\epsilon_{n,\alpha}$ and the functions $\hat{g}^\epsilon_{n,\alpha}(\cdot)$ by

$$\left.\begin{array}{l} \hat{\xi}^\epsilon_{n,\alpha} = \xi^\epsilon_{i,\alpha},\ \hat{Y}^\epsilon_{n,\alpha} = Y^\epsilon_{i,\alpha},\ \hat{g}^\epsilon_{n,\alpha}(\cdot) = g^\epsilon_{i,\alpha}(\cdot) \\ \delta \hat{M}^\epsilon_{n,\alpha} = \delta M^\epsilon_{i,\alpha} \end{array}\right\} \text{ for } n \in (\tau^\epsilon_{i,\alpha}, \tau^\epsilon_{i+1,\alpha}]. \tag{6.0}$$

Thus, these newly defined discrete parameter random variables are constant in the random interval $(\tau^\epsilon_{i,\alpha}, \tau^\epsilon_{i+1,\alpha}]$.

Following the usage in Subsection 10.1.2, it is convenient to define the observation $Y^\epsilon_{n,\alpha}(\bar{\theta})$ that would be obtained if the parameter were fixed at $\bar{\theta}$. Let $\hat{\mathcal{F}}^\epsilon_n$ denote the σ-algebra that measures all the data used to calculate the iterates up to and including $\hat{\theta}^\epsilon_n$, as well as $I^\epsilon_{n,\alpha}, \alpha \le r$ and $\{\hat{Y}^\epsilon_{i,\alpha}(\bar{\theta}); \alpha \le r, i < n\}$. Let $\hat{E}^\epsilon_n$ denote the expectation conditioned on $\hat{\mathcal{F}}^\epsilon_n$. Then the algorithm can be written as

$$\hat{\theta}^\epsilon_{n+1,\alpha} = \Pi_{[a_i,b_i]}\left(\theta^\epsilon_{n,\alpha} + \epsilon I^\epsilon_{n,\alpha} \hat{Y}^\epsilon_{n,\alpha}\right), \quad \alpha = 1, \ldots, r,$$

which we rewrite as

$$\hat{\theta}^{\epsilon}_{n+1,\alpha} = \hat{\theta}^{\epsilon}_{n,\alpha} + \epsilon I^{\epsilon}_{n,\alpha}\left(\hat{Y}^{\epsilon}_{n,\alpha} + \hat{Z}^{\epsilon}_{n,\alpha}\right), \quad \alpha = 1, \ldots, r. \tag{6.1}$$

Assumptions. Following the "division of labor" approach used in Chapter 10, which was used there to reduce the proofs and ideas to their essential components, we will start by assuming (A6.2), the tightness of the normalized iterates, and then we will prove this tightness in the next section.

The notation concerning the shift by σ in Subsection 3.4 will be used. The physical delays in communication $(\Delta^{\epsilon}_{n,\alpha}/\epsilon, \Delta^{\epsilon,+}_{n,\alpha}/\epsilon)$ are dropped for notational simplicity only. The theorem remains true if they are retained but assumed to be uniformly integrable. Compare the following assumptions with those used in Subsection 10.1.2.

(A6.1) Let $\bar{\theta}$ be an isolated stable point of the ODE (3.16) in the interior of H. Let N_ϵ be a sequence of integers such that $\epsilon N_\epsilon \to \infty$. Then $\{\hat{\theta}^{\epsilon}(\epsilon N_\epsilon + \cdot)\}$ converges weakly to the process with constant value $\bar{\theta}$.

(A6.2) There are integers $p_\epsilon \to \infty$, which we can take to be $\geq N_\epsilon$, such that

$$\{(\hat{\theta}^{\epsilon}_{p_\epsilon+n} - \bar{\theta})/\sqrt{\epsilon};\ \epsilon > 0, n \geq 0\} \quad \text{is tight.}$$

(A6.3) Define $\delta M^{\epsilon}_{n,\alpha}(\bar{\theta}) = Y^{\epsilon}_{n,\alpha}(\bar{\theta}) - E^{\epsilon}_{n,\alpha} Y^{\epsilon}_{n,\alpha}(\bar{\theta})$. Then

$$E\left|\delta M^{\epsilon,\sigma}_{n,\alpha} - \delta M^{\epsilon,\sigma}_{n,\alpha}(\bar{\theta})\right|^2 \to 0$$

as $n, \sigma \to \infty$ and $\epsilon \to 0$.

(A6.4) For each $\alpha \leq r$, the sequence of real-valued process $W^{\epsilon,\sigma}_{\alpha}(\cdot)$ defined on $(-\infty, \infty)$ by

$$W^{\epsilon,\sigma}_{\alpha}(t) = \sqrt{\epsilon} \sum_{i=0}^{t/\epsilon - 1} Y^{\epsilon,\sigma}_{i,\alpha}(\bar{\theta}), \text{ with } Y^{\epsilon,\sigma}_{i,\alpha}(\bar{\theta}) = g^{\epsilon,\sigma}_{i,\alpha}(\bar{\theta}, \xi^{\epsilon,\sigma}_{i,\alpha}) + \delta M^{\epsilon,\sigma}_{i,\alpha}(\bar{\theta})$$

(for $t \geq 0$, with the analogous definition for $t < 0$) converges weakly (as $\sigma \to \infty$ and $\epsilon \to 0$) to a real-valued Wiener process $W_\alpha(\cdot)$ with variance parameter denoted by Σ_α. Also, for each positive t and τ,

$$E^{\epsilon,\sigma}_{n,\alpha}\left[W^{\epsilon,\sigma}_{\alpha}(t+\tau) - W^{\epsilon,\sigma}_{\alpha}(t)\right] \to 0 \tag{6.2}$$

in mean as $\epsilon \to 0$ and $\sigma \to \infty$.

(A6.5) $g^{\epsilon}_{n,\alpha}(\cdot, \xi)$ is continuously differentiable for each n, ϵ, α, and ξ and can be expanded as

$$g^{\epsilon}_{n,\alpha}(\theta, \xi) = g^{\epsilon}_{n,\alpha}(\bar{\theta}, \xi) + (g^{\epsilon}_{n,\alpha,\theta}(\bar{\theta}, \xi))'(\theta - \bar{\theta}) + (y^{\epsilon}_{n,\alpha}(\theta, \xi))'(\theta - \bar{\theta}),$$

where

$$y_{n,\alpha}^{\epsilon}(\theta,\xi) = \int_0^1 \left[g_{n,\alpha,\theta}^{\epsilon}(\bar\theta + s(\theta-\bar\theta),\xi) - g_{n,\alpha,\theta}^{\epsilon}(\bar\theta,\xi)\right] ds,$$

and if $\delta_n^\epsilon \to 0$ as $\epsilon \to 0$ and $n \to \infty$, then

$$E\left|y_{n,\alpha}^{\epsilon,\sigma}(\theta_{n,\alpha}^{\epsilon,\sigma},\xi_{n,\alpha}^{\epsilon,\sigma})\right| I_{\{|\theta_{n,\alpha}^{\epsilon,\sigma}-\bar\theta|\le\delta_n^\epsilon\}} \to 0$$

as $\epsilon \to 0$ and $n, \sigma \to \infty$.

(A6.6) The set

$$\{g_{n,\alpha,\theta}^{\epsilon,\sigma}(\bar\theta,\xi_{n,\alpha}^{\epsilon,\sigma});\ \alpha,\epsilon,\sigma\} \quad \text{is uniformly integrable.}$$

(A6.7) There is a Hurwitz matrix A with rows $A_\alpha, \alpha = 1,\ldots,r$ (considered as row vectors), such that

$$\frac{1}{m}\sum_{i=n}^{n+m-1} E_{n,\alpha}^{\epsilon,\sigma}\left[(g_{i,\alpha,\theta}^{\epsilon,\sigma}(\bar\theta,\xi_{i,\alpha}^{\epsilon,\sigma}))' - A_\alpha\right] \to 0$$

in probability as $\epsilon \to 0$ and n, m, and $\sigma \to \infty$, where the gradient $g_{i,\alpha,\theta}^{\epsilon,\sigma}(\bar\theta,\xi_{i,\alpha}^{\epsilon,\sigma})$ is a column vector.

Following the approach taken in Subsection 10.1.2, let q_ϵ be integers such that $\epsilon(q_\epsilon - p_\epsilon) \to \infty$ as $\epsilon \to 0$. Define $\hat U_n^\epsilon = (\hat\theta_{q_\epsilon+n}^\epsilon - \bar\theta)/\sqrt{\epsilon}$, and define the process $\hat U^\epsilon(\cdot)$ on $[0,\infty)$ by $\hat U^\epsilon(t) = \hat U_n^\epsilon$ on $[n\epsilon, n\epsilon+\epsilon)$. Define the process $\hat W^\epsilon(\cdot)$ by

$$\hat W_\alpha^\epsilon(t) = \sqrt{\epsilon}\sum_{i=q_\epsilon}^{q_\epsilon+t/\epsilon-1} \hat Y_{i,\alpha}^\epsilon(\bar\theta).$$

Theorem 6.1. *Assume conditions* (A3.1′), (A3.7), (A3.8) (*with the delays and the* $\beta_{n,\alpha}^\epsilon$ *dropped*), (A3.10), (A3.11′), (A3.14′) (*for* $\theta = \bar\theta$ *only*), *and* (A6.1)–(A6.7). *Then the sequence* $\{\hat U^\epsilon(\cdot), \hat W^\epsilon(\cdot)\}$ *converges weakly in* $D^r[0,\infty) \times D^r(-\infty,\infty)$ *to* $(\hat U(\cdot), \hat W(\cdot))$, *where* $\hat W(\cdot) = \{\hat W_\alpha(\cdot), \alpha \le r\}$ *is a Wiener process with* $E[\hat W_\alpha(1)]^2 = \Sigma_\alpha/\bar u_\alpha(\bar\theta)$, *and* $\hat U(\cdot) = \{\hat U_\alpha(\cdot), \alpha \le r\}$ *is the stationary solution to*

$$d\hat U_\alpha = \frac{A_\alpha \hat U}{\bar u_\alpha(\bar\theta)} dt + d\hat W_\alpha,\ \alpha = 1,\ldots,r. \tag{6.3}$$

Proof: Part 1. Formulation of the problem. By the assumed weak convergence (A6.1), for any $T > 0$ and $\mu > 0$,

$$\limsup_\epsilon P\left\{\sup_{-T/\epsilon \le i \le T/\epsilon} \left|\hat\theta_{q_\epsilon+i}^\epsilon - \bar\theta\right| \ge \mu/2\right\} = 0. \tag{6.4}$$

Hence, by modifying $\{I^\epsilon_{q_\epsilon+n,\alpha}\hat{Y}^\epsilon_{q_\epsilon+n,\alpha};\alpha,|n|\le T/\epsilon\}$ on a set of arbitrarily small measure for each ϵ, without altering the assumptions, it can be supposed (justified by Theorem 7.3.6) that

$$\sup_{-T/\epsilon\le i\le T/\epsilon}\left|\hat{\theta}^\epsilon_{q_\epsilon+i}-\bar{\theta}\right|<\mu \tag{6.5}$$

for small ϵ and for any $T>0$ and $\mu>0$ of concern. Thus, without loss of generality, the Z-term in (6.1) can be dropped, and we can suppose that (6.5) holds for any given T and μ.

Let us use $\hat{y}$ for the integral in (A6.5) if $\hat{g}$ replaces g in the integrand. Now we can rewrite (6.1) in the form

$$\begin{aligned}\hat{\theta}^\epsilon_{n+1,\alpha}=\hat{\theta}^\epsilon_{n,\alpha}&+\epsilon I^\epsilon_{n,\alpha}\left[\hat{g}^\epsilon_{n,\alpha}(\bar{\theta},\hat{\xi}^\epsilon_{n,\alpha})+\delta\hat{M}^\epsilon_{n,\alpha}(\bar{\theta})+(\hat{g}^\epsilon_{n,\alpha,\theta}(\bar{\theta},\hat{\xi}^\epsilon_{n,\alpha}))'(\hat{\theta}^\epsilon_n-\bar{\theta})\right]\\&+\epsilon I^\epsilon_{n,\alpha}\left[(\hat{y}^\epsilon_{n,\alpha}(\hat{\theta}^\epsilon_n,\hat{\xi}^\epsilon_{n,\alpha}))'(\hat{\theta}^\epsilon_n-\bar{\theta})+\left(\delta\hat{M}^\epsilon_{n,\alpha}(\bar{\theta})-\delta\hat{M}^\epsilon_{n,\alpha}\right)\right],\end{aligned}$$

where $\hat{g}^\epsilon_{n,\alpha}(\cdot)$, $\delta\hat{M}^\epsilon_{n,\alpha}$, and $\hat{\xi}^\epsilon_{n,\alpha}$ are defined in (6.0). Recalling that $\hat{U}^\epsilon_n=(\hat{\theta}^\epsilon_{q_\epsilon+n}-\bar{\theta})/\sqrt{\epsilon}$, we can write for each α,

$$\begin{aligned}\hat{U}^\epsilon_{n+1,\alpha}&=\hat{U}^\epsilon_{n,\alpha}+\epsilon I^\epsilon_{q_\epsilon+n,\alpha}A_\alpha\hat{U}^\epsilon_n+\sqrt{\epsilon}I^\epsilon_{q_\epsilon+n,\alpha}\hat{Y}^\epsilon_{q_\epsilon+n,\alpha}(\bar{\theta})\\&+\epsilon I^\epsilon_{q_\epsilon+n,\alpha}\left[(\hat{g}^\epsilon_{q_\epsilon+n,\alpha,\theta}(\bar{\theta},\hat{\xi}^\epsilon_{q_\epsilon+n,\alpha}))'-A_\alpha\right]\hat{U}^\epsilon_n\\&+\epsilon I^\epsilon_{q_\epsilon+n,\alpha}(\hat{y}^\epsilon_{q_\epsilon+n,\alpha}(\hat{\theta}^\epsilon_{q_\epsilon+n},\hat{\xi}^\epsilon_{q_\epsilon+n,\alpha}))'\hat{U}^\epsilon_n\\&+\sqrt{\epsilon}I^\epsilon_{q_\epsilon+n,\alpha}\left[\delta\hat{M}^\epsilon_{q_\epsilon+n,\alpha}(\bar{\theta})-\delta\hat{M}^\epsilon_{q_\epsilon+n,\alpha}\right].\end{aligned}$$

Recall the truncation function $q_M(\cdot)$ and truncated process $U^{\epsilon,M}(\cdot)$ used in Theorem 10.1.1. Introducing the same idea here leads to the following equation for the truncated process:

$$\begin{aligned}\hat{U}^{\epsilon,M}_{n+1,\alpha}&=\hat{U}^{\epsilon,M}_{n,\alpha}+\epsilon I^\epsilon_{q_\epsilon+n,\alpha}A_\alpha\hat{U}^{\epsilon,M}_n q_M(\hat{U}^{\epsilon,M}_n)+\sqrt{\epsilon}I^\epsilon_{q_\epsilon+n,\alpha}\hat{Y}^\epsilon_{q_\epsilon+n,\alpha}(\bar{\theta})\\&+\epsilon I^\epsilon_{q_\epsilon+n,\alpha}\left[(\hat{g}^\epsilon_{q_\epsilon+n,\alpha,\theta}(\bar{\theta},\hat{\xi}^\epsilon_{q_\epsilon+n,\alpha}))'-A_\alpha\right]\hat{U}^{\epsilon,M}_n q_M(\hat{U}^{\epsilon,M}_n)\\&+\epsilon I^\epsilon_{q_\epsilon+n,\alpha}(\hat{y}^\epsilon_{q_\epsilon+n,\alpha}(\hat{\theta}^\epsilon_{q_\epsilon+n},\hat{\xi}^\epsilon_{q_\epsilon+n,\alpha}))'\hat{U}^{\epsilon,M}_n q_M(\hat{U}^{\epsilon,M}_n)\\&+\sqrt{\epsilon}I^\epsilon_{q_\epsilon+n,\alpha}\left[\delta\hat{M}^\epsilon_{q_\epsilon+n,\alpha}(\bar{\theta})-\delta\hat{M}^\epsilon_{q_\epsilon+n,\alpha}\right].\end{aligned}\tag{6.6}$$

Let $\hat{U}^{\epsilon,M}_\alpha(\cdot)$ denote the interpolated process with values $\hat{U}^{\epsilon,M}_\alpha(t)=\hat{U}^{\epsilon,M}_{n,\alpha}$ on the interval $[\epsilon n,\epsilon n+\epsilon)$.

Part 2. Tightness of $\{\hat{U}^{\epsilon,M}_\alpha(\cdot)\}$ and the weak sense limit process. The set of initial conditions $\{\hat{U}^\epsilon_0\}$ is tight by assumption (A6.2). The sequence of processes defined by

$$\epsilon\sum_{i=0}^{t/\epsilon-1}I^\epsilon_{q_\epsilon+i,\alpha}A_\alpha\hat{U}^{\epsilon,M}_i q_M(\hat{U}^{\epsilon,M}_i)$$

is tight by the uniform boundedness of $U_n^{\epsilon,M} q_M(\hat{U}_n^{\epsilon,M})$.

Define the processes $\hat{W}_\alpha^\epsilon(\cdot)$ by

$$\hat{W}_\alpha^\epsilon(t) = \sqrt{\epsilon} \sum_{i=q_\epsilon}^{q_\epsilon+t/\epsilon-1} I_{i,\alpha}^\epsilon \hat{Y}_{i,\alpha}^\epsilon(\bar{\theta}) \tag{6.7}$$

and (see Subsection 3.4 for the σ-notation)

$$W_\alpha^{\epsilon,\sigma}(t) = \sqrt{\epsilon} \sum_{i=0}^{t/\epsilon-1} Y_{i,\alpha}^{\epsilon,\sigma}(\bar{\theta}). \tag{6.8}$$

Then, with $\sigma = q_\epsilon$ we have the following relationship:

$$\hat{W}_\alpha^\epsilon(t) = W_\alpha^{\epsilon,q_\epsilon}(N_\alpha^{\epsilon,q_\epsilon}(t)). \tag{6.9}$$

Now $\{\tau_\alpha^{\epsilon,q_\epsilon}(\cdot)\}$ is tight by (A3.1′). Since $\hat{\theta}^{\epsilon,q_\epsilon}(\cdot)$ converges weakly to the process with constant value $\bar{\theta}$ by (A6.1), $\tau_\alpha^{\epsilon,q_\epsilon}(\cdot)$ converges weakly to the process with values $\bar{u}_\alpha(\bar{\theta})t$. Similarly, $N_\alpha^{\epsilon,q_\epsilon}(\cdot)$ converges weakly to the process with values $t/\bar{u}_\alpha(\bar{\theta})$. These convergences, the representation (6.9) and the weak convergence assumed in (A6.4) imply the tightness of $\{\hat{W}_\alpha^\epsilon(\cdot)\}$ and that the weak sense limit $\hat{W}_\alpha(\cdot)$ has the representation $\hat{W}_\alpha(t) = W_\alpha(t/\bar{u}_\alpha(\bar{\theta}))$, which is a real-valued Wiener process with variance parameter $\Sigma_\alpha/\bar{u}_\alpha(\bar{\theta})$.

Define the process

$$\hat{y}_\alpha^\epsilon(t) = \epsilon \sum_{i=q_\epsilon}^{q_\epsilon+t/\epsilon-1} I_{i,\alpha}^\epsilon \hat{y}_{i,\alpha}^\epsilon(\hat{\theta}_{i,\alpha}^\epsilon, \hat{\xi}_i^\epsilon). \tag{6.10}$$

Since, for purposes of the proof we can suppose that $\hat{\theta}_{q_\epsilon+n}^\epsilon \to \bar{\theta}$ as $\epsilon \to 0$ uniformly in n, the last part of condition (A6.5) implies that the sequence defined by (6.10) converges weakly to the "zero" process.

The tightness of the sequence defined by

$$\epsilon \sum_{i=0}^{t/\epsilon-1} I_{q_\epsilon+i,\alpha}^\epsilon \left[(\hat{g}_{q_\epsilon+i,\alpha,\theta}^\epsilon(\bar{\theta}, \hat{\xi}_{q_\epsilon+i,\alpha}^\epsilon))' - A_\alpha \right] \hat{U}_i^{\epsilon,M} q_M(\hat{U}_i^{\epsilon,M}) \tag{6.11}$$

is a consequence of the uniform integrability in (A6.6) and the boundedness of the $\hat{U}_n^{\epsilon,M} q_M(\hat{U}_n^{\epsilon,M})$. By (A6.3), the interpolations of the last term on the right of (6.6) converges weakly to the "zero process." Putting these pieces together, we have the tightness of $\{\hat{U}_\alpha^\epsilon(\cdot)\}$ for each α. Then the fact that the weak sense limit of the processes defined in (6.11) is the "zero" process follows from this tightness and (A6.7).

Let $(\hat{U}^M(\cdot), \hat{W}(\cdot))$ with $\hat{U}^M(\cdot) = \{\hat{U}_\alpha^M(\cdot), \alpha \le r\}$ and $\hat{W}(\cdot) = \{\hat{W}_\alpha(\cdot), \alpha \le r\}$ denote the weak sense limit of a weakly convergent subsequence of $\{\hat{U}^{\epsilon,M}(\cdot), \hat{W}^\epsilon(\cdot)\}$. Then the process defined by

$$\sum_{i=0}^{t/\epsilon-1} \epsilon I_{q_\epsilon+i,\alpha}^\epsilon A_\alpha \hat{U}_i^{\epsilon,M} q_M(\hat{U}_i^{\epsilon,M})$$

converges weakly (along the chosen subsequence) to

$$\int_0^t \frac{A_\alpha \hat{U}_\alpha^M(s) q_M(\hat{U}^M(s))}{\bar{u}_\alpha(\bar{\theta})} ds.$$

While we have shown that $\hat{W}_\alpha(\cdot)$ is a real-valued Wiener process for each α, we have not shown that $\{\hat{W}_\alpha(\cdot), \alpha \leq r\}$ is a Wiener process. The joint distributions are Gaussian, since the marginal distributions are Gaussian, but we need to show that the increments are mutually independent. This independence (equivalently, orthogonality, due to the Gaussian property and the zero mean value) follows from the conditional expectation condition in (6.2) and the weak convergence (which implies that $\hat{W}(\cdot)$ is a martingale); the details are left to the reader.

The rest of the details are a combination of those used in the proofs of Theorems 10.1.1, 10.1.2, 3.1 and 3.3. As in the proof of Theorem 10.1.1, one shows that the truncation is not needed. The stationarity is proved exactly as in Theorem 10.1.1. □

12.7 Stability and Tightness of the Normalized Iterates

12.7.1 *The Unconstrained Algorithm*

The result in this section is an extension of that in Section 10.5.1 for the unconstrained algorithm. It illustrates one typical Liapunov function-based approach for getting stability for the asynchronous problem. For simplicity, the delays will be dropped. Delays can be handled, but one needs to obtain bounds on the changes in the Liapunov functions during the "delay" intervals. Without loss of generality, we retain the structure and notation of the last section, where the updates occur only at the integer times $1, 2, \ldots$

Write the algorithm (6.1) in unconstrained form as

$$\hat{\theta}^\epsilon_{n+1,\alpha} = \hat{\theta}^\epsilon_{n,\alpha} + \epsilon I^\epsilon_{n,\alpha} \hat{Y}^\epsilon_{n,\alpha}. \tag{7.1}$$

The limit mean ODE for the unconstrained algorithm is

$$\dot{\hat{\theta}}_\alpha = \frac{\bar{g}_\alpha(\hat{\theta})}{\bar{u}_\alpha(\hat{\theta})}, \quad \alpha = 1, \ldots, r. \tag{7.2}$$

Define the vector $\bar{\gamma}(\theta) = \{\bar{g}_\alpha(\theta)/\bar{u}_\alpha(\theta), \alpha = 1, \ldots, r\}$.

Assumptions. The assumptions are to hold for each α and are analogs of those used in Subsection 10.5. The K_i are arbitrary positive real numbers whose values might be different in different usages.

(A7.1) $\bar{\theta}$ is an asymptotically stable point of the ODE $\dot{\theta} = \bar{\gamma}(\theta)$. The non-negative continuously differentiable function $V(\cdot)$ is a Liapunov function for the ODE, its second-order partial derivatives are bounded, and $V(\theta) \to \infty$ as $|\theta| \to \infty$.

(A7.2) $|V_\theta(\theta)|^2 \le K_1(V(\theta)+1)$.

(A7.3) There is a $\lambda > 0$ such that

$$V_\theta'(\theta)\bar{\gamma}(\theta) \le -\lambda V(\theta).$$

(A7.4) There is a $K_1 > 0$ such that for each $K > 0$

$$\sup_n E|I_{n,\alpha}^\epsilon \hat{Y}_{n,\alpha}^\epsilon|^2 I_{\{|\hat{\theta}_n^\epsilon - \bar{\theta}| \le K\}} \le K_1 E(V(\hat{\theta}_n^\epsilon)+1) I_{\{|\hat{\theta}_n^\epsilon - \bar{\theta}| \le K\}}.$$

(A7.5) Condition (A3.7) holds with the delay and $\beta_{n,\alpha}^\epsilon$ dropped; equivalently,

$$\hat{E}_n^\epsilon \hat{Y}_{n,\alpha}^\epsilon I_{n,\alpha}^\epsilon = \hat{g}_{n,\alpha}^\epsilon(\hat{\theta}_n^\epsilon, \hat{\xi}_{n,\alpha}^\epsilon) I_{n,\alpha}^\epsilon,$$

where $\hat{E}_n^\epsilon$ is the expectation conditioned on $\hat{\mathcal{F}}_n^\epsilon$.

(A7.6) For each n, let $I_{i,\alpha}^\epsilon(\theta, n)$ denote the indicator of the event that there is an update at time $i \ge n$ under the condition that the parameter is held fixed at the value θ for all time at or after time n. [Note that $\hat{E}_n^\epsilon I_{n,\alpha}^\epsilon(\hat{\theta}_n^\epsilon, n) = I_{n,\alpha}^\epsilon$.] Let the random function

$$\Gamma_{n,\alpha}^{\epsilon,d}(\theta) = \epsilon \sum_{i=n}^{\infty} (1-\epsilon)^{i-n} \hat{E}_n^\epsilon \left[\hat{g}_{i,\alpha}^\epsilon(\theta, \hat{\xi}_{i,\alpha}^\epsilon) - \bar{g}_\alpha(\theta) \right] I_{i,\alpha}^\epsilon(\theta, n) \tag{7.3}$$

be well defined for each θ, α, n and small $\epsilon > 0$, in that the sum of the absolute values of the summands is integrable. Define $\Gamma_n^{\epsilon,d}(\theta) = \{\Gamma_{n,\alpha}^{\epsilon,d}(\theta), \alpha \le r\}$. Then

$$E|\Gamma_{n,\alpha}^{\epsilon,d}(\hat{\theta}_n^\epsilon)|^2 = O(\epsilon^2)\left(EV(\hat{\theta}_n^\epsilon) + 1\right). \tag{7.4}$$

(A7.7) $E\left|\Gamma_{n+1,\alpha}^{\epsilon,d}(\hat{\theta}_{n+1}^\epsilon) - \Gamma_{n+1,\alpha}^{\epsilon,d}(\hat{\theta}_n^\epsilon)\right|^2 = O(\epsilon^4)\left(EV(\hat{\theta}_n^\epsilon) + 1\right).$

(A7.8) Let the random function

$$\Lambda_{n,\alpha}^{\epsilon,d}(\theta) = \epsilon \sum_{i=n}^{\infty} (1-\epsilon)^{i-n} \hat{E}_n^\epsilon \left[I_{i,\alpha}^\epsilon(\theta, n) - \frac{1}{\bar{u}_\alpha(\theta)} \right] \bar{g}_\alpha(\theta) \tag{7.5}$$

be well defined for each θ, α, n, and small $\epsilon > 0$, in that the sum of the absolute values of the summands is integrable. Also,

$$E|\Lambda_{n,\alpha}^{\epsilon,d}(\hat{\theta}_n^\epsilon)|^2 = O(\epsilon^2)\left(EV(\hat{\theta}_n^\epsilon) + 1\right). \tag{7.6}$$

Define $\Lambda_n^{\epsilon,d}(\theta) = \{\Lambda_{n,\alpha}^{\epsilon,d}(\theta), \alpha \le r\}$.

(A7.9) $E\left|\Lambda^{\epsilon,d}_{n+1,\alpha}(\hat\theta^\epsilon_{n+1}) - \Lambda^{\epsilon,d}_{n+1,\alpha}(\hat\theta^\epsilon_n)\right|^2 = O(\epsilon^4)\left(EV(\hat\theta^\epsilon_n)+1\right).$

Comment on $\Gamma^{\epsilon,d}_{n,\alpha}(\theta)$ and $\Lambda^{\epsilon,d}_{n,\alpha}(\theta)$. The proof will use a perturbed Liapunov function method. There are two random effects to average via the perturbations. The first are those of $g^\epsilon_{n,\alpha}(\theta,\xi^\epsilon_{n,\alpha})$; they are handled with a perturbation using $\Gamma^{\epsilon,d}_{n,\alpha}(\theta)$, which will help us to replace this "noise" term with $\bar g_\alpha(\theta)$ plus a "small" error. The indicator function $I^\epsilon_{i,\alpha}(\theta,n)$ in the perturbation function is used simply as a way to keep track of the actual (random) number of updates on any real-time interval. Otherwise the perturbation is similar to what has been used before (say in Theorem 10.5.1). The second random effects to be averaged are those due to the random time between updates at any of the processors. This is handled by a perturbation based on $\Lambda^{\epsilon,d}_{n,\alpha}(\theta)$, which works "on top of" the first perturbation, in that it uses the mean $\bar g_\alpha(\cdot)$ and not $g^\epsilon_{n,\alpha}(\cdot)$. This perturbation is "centered" at $1/\bar u_\alpha(\theta)$, which is the mean rate of increase of $\sum I^\epsilon_{i,\alpha}(\theta)$.

Theorem 7.1. *Consider algorithm* (7.1) *and assume conditions* (A7.1)–(A7.9). *Let there be a positive definite and symmetric matrix P such that*

$$V(\theta) = (\theta-\bar\theta)'P(\theta-\bar\theta) + o(|(\theta-\bar\theta)|^2) \tag{7.7}$$

for small $|\theta-\bar\theta|$. Then there are $n_\epsilon<\infty$ such that $\{(\hat\theta^\epsilon_n-\bar\theta)/\sqrt{\epsilon}; n\ge n_\epsilon,\epsilon\}$ is tight. In fact, n_ϵ satisfies $e^{-\lambda_1\epsilon n_\epsilon}|\theta_0-\bar\theta| = O(\sqrt{\epsilon})$, where $\lambda_1<\lambda$, but is arbitrarily close to it. If there is a positive definite and symmetric P_1 such that

$$V(\theta) \ge (\theta-\bar\theta)'P_1(\theta-\bar\theta), \tag{7.8}$$

then

$$\limsup_{\epsilon\to 0}\sup_{n\ge n_\epsilon} E\frac{|\theta^\epsilon_n-\bar\theta|^2}{\epsilon} < \infty. \tag{7.9}$$

Proof. The proof is very similar to that of Theorem 10.5.1. Recall the fact that $\hat E^\epsilon_n I^\epsilon_{n,\alpha}(\hat\theta^\epsilon_n,\hat\xi^\epsilon_{n,\alpha}) = I^\epsilon_{n,\alpha}$. Start, as usual, by expanding the Liapunov function:

$$\begin{aligned}\hat E^\epsilon_n V(\theta^\epsilon_{n+1}) - V(\theta^\epsilon_n) &= \epsilon\sum_\alpha V_{\theta^\alpha}(\hat\theta^\epsilon_n)\hat E^\epsilon_n \hat Y^\epsilon_{n,\alpha} I^\epsilon_{n,\alpha} \\ &\quad + O(\epsilon^2)\textstyle\sum_\alpha \hat E^\epsilon_n |\hat Y^\epsilon_{n,\alpha} I^\epsilon_{n,\alpha}|^2.\end{aligned} \tag{7.10}$$

Two perturbations will be used. The first is

$$\delta V^\epsilon_{n,1}(\hat\theta^\epsilon_n) = V'_\theta(\hat\theta^\epsilon_n)\Gamma^{\epsilon,d}_n(\hat\theta^\epsilon_n).$$

We can write

$$\begin{aligned}&\hat{E}_n^\epsilon \delta V_{n+1,1}^\epsilon(\hat{\theta}_{n+1}^\epsilon) - \delta V_{n,1}^\epsilon(\hat{\theta}_n^\epsilon) \\ &\quad = -\epsilon \sum_\alpha V_{\theta^\alpha}(\hat{\theta}_n^\epsilon)\left[\hat{g}_{n,\alpha}^\epsilon(\hat{\theta}_n^\epsilon, \hat{\xi}_{n,\alpha}^\epsilon) - \bar{g}_\alpha(\hat{\theta}_n^\epsilon)\right] I_{n,\alpha}^\epsilon \\ &\quad + \hat{E}_n^\epsilon \left[V_\theta'(\hat{\theta}_{n+1}^\epsilon)\Gamma_{n+1}^{\epsilon,d}(\hat{\theta}_{n+1}^\epsilon) - V_\theta'(\hat{\theta}_n^\epsilon)\Gamma_{n+1}^{\epsilon,d}(\hat{\theta}_n^\epsilon)\right] \\ &\quad + \epsilon^2 V_\theta'(\hat{\theta}_n^\epsilon)\hat{E}_n^\epsilon \Gamma_{n+1}^{\epsilon,d}(\hat{\theta}_n^\epsilon).\end{aligned} \tag{7.11}$$

Define the first perturbed Liapunov function $V_{n,1}^\epsilon(\hat{\theta}_n^\epsilon) = V(\hat{\theta}_n^\epsilon) + \delta V_{n,1}^\epsilon(\hat{\theta}_n^\epsilon)$. As in the proof of Theorem 10.5.1, adding (7.10) and (7.11) yields

$$\begin{aligned}\hat{E}_n^\epsilon V_{n+1,1}^\epsilon(\hat{\theta}_{n+1}^\epsilon) - V_{n,1}^\epsilon(\hat{\theta}_n^\epsilon) &= \epsilon \sum_\alpha V_{\theta^\alpha}(\hat{\theta}_n^\epsilon)\bar{g}_\alpha(\hat{\theta}_n) I_{n,\alpha}^\epsilon \\ &\quad + \text{error terms},\end{aligned} \tag{7.12}$$

where the error terms are the last term on the right side of (7.10) and the last two lines of (7.11).

Next, define the second perturbation

$$\delta V_{n,2}^\epsilon(\hat{\theta}_n^\epsilon) = V_\theta'(\hat{\theta}_n^\epsilon)\Lambda_n^\epsilon(\hat{\theta}_n^\epsilon),$$

and define the final perturbed Liapunov function

$$V_n^\epsilon(\hat{\theta}_n^\epsilon) = V(\hat{\theta}_n^\epsilon) + \delta V_{n,1}^\epsilon(\hat{\theta}_n^\epsilon) + \delta V_{n,2}^\epsilon(\hat{\theta}_n^\epsilon) = V_{n,1}^\epsilon(\hat{\theta}^\epsilon) + \delta V_{n,2}^\epsilon(\hat{\theta}_n^\epsilon).$$

We can write

$$\begin{aligned}&\hat{E}_n^\epsilon \delta V_{n+1,2}^\epsilon(\hat{\theta}_{n+1}^\epsilon) - \delta V_{n,2}^\epsilon(\hat{\theta}_n^\epsilon) \\ &\quad = -\epsilon \sum_\alpha V_{\theta^\alpha}(\hat{\theta}_n^\epsilon)\left[\bar{g}_\alpha(\hat{\theta}_n) I_{n,\alpha}^\epsilon - \bar{\gamma}_\alpha(\hat{\theta}_n^\epsilon)\right] \\ &\quad + \hat{E}_n^\epsilon \left[V_\theta'(\hat{\theta}_{n+1}^\epsilon)\Lambda_{n+1}^{\epsilon,d}(\hat{\theta}_{n+1}^\epsilon) - V_\theta'(\hat{\theta}_n^\epsilon)\Lambda_{n+1}^{\epsilon,d}(\hat{\theta}_n^\epsilon)\right] \\ &\quad + \epsilon^2 V_\theta'(\hat{\theta}_n^\epsilon)\hat{E}_n^\epsilon \Lambda_{n+1}^{\epsilon,d}(\hat{\theta}_n^\epsilon),\end{aligned} \tag{7.13}$$

where $\bar{\gamma}_\alpha(\cdot)$ is defined below (7.2). Finally, putting the above expansions together and using the bounds in (A7.1)–(A7.9) yields for small ϵ (analogously to what was done in the proof of Theorem 10.5.1):

$$\hat{E}_n^\epsilon V_{n+1}^\epsilon(\hat{\theta}_{n+1}^\epsilon) - V_n^\epsilon(\hat{\theta}_n^\epsilon) \le -\epsilon\lambda_1 V(\hat{\theta}_n^\epsilon) + O(\epsilon^2), \tag{7.14}$$

where $\lambda_1 < \lambda$ but is arbitrarily close to it for small ϵ. Follow the proof of Theorem 10.5.1 from this point on to get the theorem. □

The constrained algorithm and other extensions. There are more or less obvious versions of the results in Subsection 10.5.2, and of the results based on probability one convergence in Section 10.4. Similarly, the soft constraint idea of Chapter 5 can be used.

12.8 Convergence for Q-Learning: Discounted Cost

The differential inclusions form of Theorem 3.5 will be applied to prove convergence of the Q-learning algorithm of Section 2.3 for the discounted case $\beta < 1$. The unconstrained algorithm for the decreasing step size case is (2.3.3) or (2.3.4); the reader is referred to that section for the notation. We will work with a practical constrained form and let the step size be ϵ, a constant. The proof for the decreasing step size case is virtually the same. The values of $Q_{n,id}$ will be truncated at $\pm B$ for large B. Thus $H = [-B, B]^r$, where r is the number of possible (state, action) pairs. Define $C = \max\{\bar{c}_{id}\}$. Then the Q-values under any policy are bounded by $C/(1-\beta)$. [This does not imply that the Q-values given by an *unconstrained* stochastic approximation will be no greater than $C/(1-\beta)$.] Let $B > C/(1-\beta)$.

Let ψ_n^ϵ denote the state of the controlled chain at time n. The state will generally depend on the step size ϵ, since it depends on the control policy that, in turn, depends on the current estimate of the optimal Q-values. Recall that the Q-value for the (state, action) pair (i,d) is updated at time $n+1$ (after the next state value is observed) if that pair (i,d) occurs at time n. Thus, we can suppose that the updates are at times $n = 1, 2, \ldots$. Let $\hat{Q}^\epsilon_{n,id}$ denote the Q-values at real time n and let $I^\epsilon_{n,id}$ be the indicator of the event that (i,d) occurs at real time n.

The algorithm can be written as

$$\hat{Q}^\epsilon_{n+1,id} = \Pi_{[-B,B]}\left[\hat{Q}^\epsilon_{n,id} + \epsilon\left(c_{n,id} + \beta \min_{v\in U(\psi^\epsilon_{n+1})} \hat{Q}^\epsilon_{n,\psi^\epsilon_{n+1}v} - \hat{Q}^\epsilon_{n,id}\right) I^\epsilon_{n,id}\right] \tag{8.1}$$

or, equivalently, as

$$\hat{Q}^\epsilon_{n+1,id} = \Pi_{[-B,B]}\left[\hat{Q}^\epsilon_{n,id} + \epsilon\left(T_{id}(\hat{Q}^\epsilon_n) - \hat{Q}^\epsilon_{n,id} + \delta M^\epsilon_n\right) I^\epsilon_{n,id}\right], \tag{8.2}$$

where the operator T_{id} is defined below (2.3.3) and the noise δM^ϵ_n is defined above (2.3.4).

In the absence of the constraint, the general stability results of Section 7 can be used to prove tightness of $\{\hat{Q}^\epsilon_{n,id}; n, i, d, \epsilon\}$, which can then be used to get the convergence.

Suppose that there are $n_0 < \infty$ and $\delta_0 > 0$ such that for each state pair i, j,

$$\inf P\left\{\psi^\epsilon_{n+k} = j, \text{ some } k \le n_0 \middle| \psi^\epsilon_n = i\right\} \ge \delta_0, \tag{8.3}$$

where the inf is over the time n and all ways in which the experimenter might select the controls. Let $\delta\tau^\epsilon_{n,id}$ denote the time interval between the nth and $(n+1)$st occurrences of the (state, action) pair (i,d), and let $E^{\epsilon,+}_{n,id}$ denote the expectation, conditioned on all the data up to and including the

$(n+1)$st update for the pair (i,d). Define $u^\epsilon_{n,id}$ by

$$E^{\epsilon,+}_{n,id}\delta\tau^\epsilon_{n+1,id} = u^\epsilon_{n+1,id},$$

and suppose that the (possibly random) $\{u^\epsilon_{n,id}; n, \epsilon\}$ are uniformly bounded by a real number $\bar{u}_{id}$ (which must be ≥ 1) and that $\{\delta\tau_{n,id}; n, i, d\}$ is uniformly integrable.

Let $\hat{Q}^\epsilon_{id}(\cdot)$ denote the piecewise constant interpolation of $\{\hat{Q}^\epsilon_{n,id}; n < \infty\}$ in scaled real time, that is, with interpolation intervals of width ϵ. Define $\hat{Q}^\epsilon(\cdot) = \{\hat{Q}^\epsilon_{id}(\cdot);\ i, d\}$. The problem is relatively easy to deal with since the noise terms δM^ϵ_n are martingale differences. Under the given conditions, for any sequence of real numbers T_ϵ, $\{Q^\epsilon_{id}(T_\epsilon + \cdot)\}$ is tight, and (if $T_\epsilon \to \infty$) it will be seen that it converges weakly (as $\epsilon \to 0$) to the process with constant value $\bar{Q}$, the optimal value defined by (2.3.2).

The limit mean ODE will be shown to be

$$\dot{Q}_{id} = D_{id}(t)\,(T_{id}(Q) - Q_{id}) + z_{id}, \quad \text{all } i, d, \tag{8.4}$$

where the $z_{id}(\cdot)$ serve the purpose of keeping the values in the interval $[-B, B]$ and will be shown to be zero. The values of $D_{id}(\cdot)$ lie in the intervals $[1/\bar{u}_{id}, 1]$. We next show that all solutions of (8.4) tend to the unique limit point $\bar{Q}$. Suppose that $Q_{id}(t) = B$ for some t and i, d pair. Note that, by the lower bound on B,

$$\sup_{Q:Q_{id}=B} [T_{id}(Q) - Q_{id}] \leq C + \beta\sum_j p_{ij}(d)B - B = C + (\beta - 1)B < 0.$$

This implies that $\dot{Q}_{id}(t) < 0$ if $Q_{id}(t) = B$. Analogously, $\dot{Q}_{id}(t) > 0$ if $Q_{id}(t) = -B$. Thus, the boundary of the constraint set H is not accessible by a trajectory of (8.4) from any interior point. Now, dropping the z_{id} terms, by the contraction property of $T(\cdot)$, $\bar{Q}$ is the unique limit point of (8.4).

We need only prove that (8.4) is the limit mean ODE by verifying the conditions of the differential inclusions part of Theorem 3.5. The expectation of the observation in (8.2) used at time $(n+1)$, given the past and the fact that the pair (i,d) occurred at time n, is

$$\bar{g}_{id}(\hat{Q}^\epsilon_n) \equiv T_{id}(\hat{Q}^\epsilon_n) - \hat{Q}^\epsilon_{id}.$$

Thus, the $g^\epsilon_{n,id}(\cdot)$ of Theorem 3.5 are all $\bar{g}_{id}(\cdot)$. Thus, (A3.7) holds, with the $\beta^\epsilon_{n,\alpha}$-terms and the delays being zero. [Delays can be allowed, provided that they are uniformly integrable, but in current Q-learning applications, delays are of little significance.]

Condition (A3.9) is obvious (there are no $\xi^\epsilon_{n,\alpha}$). Also the conditions for the differential inclusion result in Theorem 3.5 hold, where $U_{id}(\theta) = U_{id} = [1, \bar{u}_{id}]$. Condition (A3.1′) is obvious. All the other conditions in the theorem are either obvious or not applicable. Thus, Theorem 3.5 holds.