2

Event-Driven Control and Optimization in Hybrid Systems

Christos G. Cassandras
Boston University
Boston, MA, USA

CONTENTS

ABSTRACT The event-driven paradigm offers an alternative complementary approach to the time-driven paradigm for modeling, sampling, estimation, control, and optimization. This is largely a consequence of systems being increasingly networked, wireless, and consisting of distributed communicating components. The key idea is that control actions need not be dictated by time steps taken by a "clock"; rather, an action should be triggered by an "event," which may be a well-defined condition on the system state, including the possibility of a simple time step, or a random state transition. In this chapter, the event-driven paradigm is applied to control and optimization problems encountered in the general setting of hybrid systems where controllers are parameterized and the parameters are adaptively tuned online based on observable data. We present a general approach for evaluating (or estimating in the case of a stochastic system) gradients of performance metrics with respect to various parameters based on the infinitesimal perturbation analysis (IPA) theory originally developed for discrete event systems (DESs) and now adapted to hybrid systems. This results in an "IPA calculus," which amounts to a set of simple, event-driven iterative equations. The event-driven nature of this approach implies its scalability in the size of an event set, as opposed to the system state space. We also show how the event-based IPA calculus may be used in multi-agent systems for determining optimal agent trajectories without any detailed knowledge of environmental randomness.

2.1 Introduction

The history of modeling and analysis of dynamic systems is founded on the time-driven paradigm provided by a theoretical framework based on differential (or difference) equations. In this paradigm, time is an independent variable, and as it evolves, so does the state of the system. Conceptually, we postulate the existence of an underlying "clock," and with every "clock tick" a state update is performed, including the case where no change in the state occurs. The methodologies developed for sampling, estimation, communication, control, and optimization of dynamic systems have also evolved based on the same *time-driven* principle. Advances in digital technologies that occurred in the 1970s and beyond have facilitated the implementation of this paradigm with digital clocks embedded in hardware and used to drive processes for data collection or for the actuation of devices employed for control purposes.

As systems have become increasingly networked, wireless, and distributed, the universal value of this

point of view has understandably come to question. While it is always possible to postulate an underlying clock with time steps dictating state transitions, it may not be feasible to guarantee the synchronization of all components of a distributed system to such a clock, and it is not efficient to trigger actions with every time step when such actions may be unnecessary. The *event-driven* paradigm offers an alternative, complementary look at modeling, control, communication, and optimization. The key idea is that a clock should not be assumed to dictate actions simply because a time step is taken; rather, an action should be triggered by an "event" specified as a well-defined condition on the system state or as a consequence of environmental uncertainties that result in random state transitions. Observing that such an event could actually be defined to be the occurrence of a "clock tick," it follows that this framework may in fact incorporate time-driven methods as well. On the other hand, defining the proper "events" requires more sophisticated techniques compared to simply reacting to time steps.

The motivation for this alternative event-driven view is multifaceted. For starters, there are many natural DESs where the only changes in their state are dictated by event occurrences. The Internet is a prime example, where "events" are defined by packet transmissions and receptions at various nodes, causing changes in the contents of various queues. For such systems, a time-driven modeling approach may not only be inefficient, but also potentially erroneous, as it cannot deal with events designed to occur concurrently in time. The development of a rigorous theory for the study of DES in the 1980s (see, e.g., [1–5]) paved the way for event-based models of certain classes of dynamic systems and spurred new concepts and techniques for control and optimization. By the early 1990s, it became evident that many interesting dynamic systems are in fact "hybrid" in nature, i.e., at least some of their state transitions are caused by (possibly controllable) events [6–12]. This has been reinforced by technological advances through which sensing and actuating devices are embedded into systems allowing physical processes to interface with such devices which are inherently event driven. A good example is the modern automobile where an event induced by a device that senses slippery road conditions may trigger the operation of an antilock braking system, thus changing the operating dynamics of the actual vehicle. More recently, the term "cyber–physical system" [13] has emerged to describe the hybrid structure of systems where some components operate as physical processes modeled through time-driven dynamics, while other components (mostly digital devices empowered by software) operate in event-driven mode.

Moreover, many systems of interest are now networked and spatially distributed. In such settings, especially when energy-constrained wireless devices are involved, frequent communication among system components can be inefficient, unnecessary, and sometimes infeasible. Thus, rather than imposing a rigid time-driven communication mechanism, it is reasonable to seek instead to define specific events that dictate when a particular node in a network needs to exchange information with one or more other nodes. In other words, we seek to complement synchronous operating mechanisms with asynchronous ones, which can dramatically reduce communication overhead without sacrificing adherence to design specifications and desired performance objectives. When, in addition, the environment is stochastic, significant changes in the operation of a system are the result of random event occurrences, so that, once again, understanding the implications of such events and reacting to them is crucial. Besides their modeling potential, it is also important to note that event-driven approaches to fundamental processes such as sampling, estimation, and control possess important properties related to variance reduction and robustness of control policies to modeling uncertainties. These properties render them particularly attractive, compared to time-driven alternatives.

While the importance of event-driven behavior in dynamic systems was recognized as part of the development of DES and then hybrid systems, more recently there have been significant advances in applying event-driven methods (also referred to as "event-based" and "event-triggered") to classical feedback control systems; see [14–18] and references therein. For example, in [15] a controller for a linear system is designed to update control values only when a specific error measure (e.g., for tracking or stabilization purposes) exceeds a given threshold, while refraining from any updates otherwise. It is also shown how such controllers may be tuned and how bounds may be computed in conjunction with known techniques from linear system theory. Trade-offs between interevent times and controller performance are further studied in [19]. As another example, in [18] an event-driven approach termed "self-triggered control" determines instants when the state should be sampled and control actions taken for some classes of nonlinear control systems. Benefits of event-driven mechanisms for estimation purposes are considered in [20,21]. In [20], for instance, an event-based sampling mechanism is studied where a signal is sampled only when measurements exceed a certain threshold, and it is shown that this approach outperforms a classical periodic sampling process at least in the case of some simple systems.

In distributed systems, event-driven mechanisms have the advantage of significantly reducing

communication among networked components without affecting desired performance objectives (see [22–27]). For instance, Trimpe and D'Andrea [25] consider the problem of estimating the state of a linear system based on information communicated from spatially distributed sensors. In this case, each sensor computes the measurement prediction variance, and the event-driven process of transmitting information is defined by events such that this variance exceeds a given threshold. A scenario where sensors may be subject to malicious attacks is considered in [28], where event-driven methods are shown to lead to computationally advantageous state reconstruction techniques. It should be noted that in all such problems, one can combine event-driven and time-driven methods, as in [24] where a control scheme combining periodic (time-driven) and event-driven control is used for linear systems to update and communicate sensor and actuation data only when necessary in the sense of maintaining a satisfactory closed-loop performance. It is shown that this goal is attainable with a substantial reduction in communication over the underlying network. Along the same lines, combining event-driven and time-driven sensor information it is shown in [29] that stability can be guaranteed where the former methods alone may fail to do so.

In multi-agent systems, on the other hand, the goal is for networked components to cooperatively maximize (or minimize) a given objective; it is shown in [23] that an event-driven scheme can still achieve the optimization objective while drastically reducing communication (hence, prolonging the lifetime of a wireless network), even when delays are present (as long as they are bounded). Event-driven approaches are also attractive in receding horizon control, where it is computationally inefficient to reevaluate a control value over small time increments as opposed to event occurrences defining appropriate planning horizons for the controller (e.g., see [30]).

In the remainder of this chapter, we limit ourselves to discussing how the event-driven paradigm is applied to control and optimization problems encountered in the general setting of hybrid systems. In particular, we consider a general-purpose control and optimization framework where controllers are parameterized and the parameters are adaptively tuned online based on observable data. One way to systematically carry out this process is through gradient information pertaining to given performance measures with respect to these parameters, so as to iteratively adjust their values. When the environment is stochastic, this entails generating gradient estimates with desirable properties such as unbiasedness. This gradient evaluation/estimation approach is based on the IPA theory [1,31] originally developed for DES and now adapted to hybrid system where it results in an "IPA calculus" [32], which amounts to a set of simple, event-driven iterative equations. In this approach, the gradient evaluation/estimation procedure is based on directly observable data, and it is entirely event driven. This makes it computationally efficient, since it reduces a potentially complex process to a finite number of actions. More importantly perhaps, this approach has two key benefits that address the need for scalable methods in large-scale systems and the difficulty of obtaining accurate models especially in stochastic settings. First, being event driven, it is scalable in the size of the event space and not the state space of the system model. As a rule, the former is much smaller than the latter. Second, it can be shown that the gradient information is often independent of model parameters, which may be unknown or hard to estimate. In stochastic environments, this implies that complex control and optimization problems can be solved with little or no knowledge of the noise or random processes affecting the underlying system dynamics.

This chapter is organized as follows. A general online control and optimization framework for hybrid systems is presented in Section 2.2, whose centerpiece is a methodology used for evaluating (or estimating in the stochastic case) a gradient of an objective function with respect to controllable parameters. This event-driven methodology, based on IPA, is described in Section 2.3. In Section 2.4, three key properties of IPA are presented and illustrated through examples. In Section 2.4, an application to multi-agent systems is given. In particular, we consider cooperating agents that carry out a persistent monitoring mission in simple one-dimensional environments and formulate this mission as an optimal control problem. Its solution results in agents operating as hybrid systems with parameterized trajectories. Thus, using the event-based IPA calculus, we describe how optimal trajectories can be obtained online without any detailed knowledge of environmental randomness.

2.2 A Control and Optimization Framework for Hybrid Systems

A hybrid system consists of both time-driven and event-driven components [33]. The modeling, control, and optimization of these systems is quite challenging. In particular, the performance of a stochastic hybrid system (SHS) is generally hard to estimate because of the absence of closed-form expressions capturing the dependence of interesting performance metrics on various design or control parameters. Most approaches rely on

approximations and/or using computationally taxing methods, often involving dynamic programming techniques. The inherent computational complexity of these approaches, however, makes them unsuitable for online control and optimization. Yet, in some cases, the structure of a dynamic optimization problem solution can be shown to be of parametric form, thus reducing it to a parametric optimization problem. As an example, in a linear quadratic Gaussian setting, optimal feedback policies simply depend on gain parameters to be selected subject to certain constraints. Even when this is not provably the case, one can still define parametric families of solutions which can be optimized and yield near-optimal or at least vastly improved solutions relative to ad hoc policies often adopted. For instance, it is common in solutions based on dynamic programming [34] to approximate cost-to-go functions through parameterized function families and then iterate over the parameters involved seeking near-optimal solutions for otherwise intractable problems.

With this motivation in mind, we consider a general-purpose framework as shown in Figure 2.1. The starting point is to assume that we can observe state trajectories of a given hybrid system and measure a performance (or cost) metric denoted by $L(\theta)$, where θ is a parameter vector. This vector characterizes a controller (as shown in Figure 2.1) but may also include design or model parameters. The premise here is that the system is too complex for a closed-form expression of $L(\theta)$ to be available, but that it is possible to measure it over a given time window. In the case of a stochastic environment, the observable state trajectory is a sample path of a SHS, so that $L(\theta)$ is a sample function, and performance is measured through $E[L(\theta)]$ with the expectation defined in the context of a suitable probability space. In addition to $L(\theta)$, we assume that all or part of the system state is observed, with possible noisy measurements. Thus, randomness may enter through the system process or the measurement process or both.

The next step in Figure 2.1 is the evaluation of the gradient $\nabla L(\theta)$. In the stochastic case, $\nabla L(\theta)$ is a random variable that serves as an estimate (obtained over a given time window) of $\nabla E[L(\theta)]$. Note that we require $\nabla L(\theta)$ to be evaluated based on available data observed from a *single* state trajectory (or sample path) of the hybrid system. This is in contrast to standard derivative approximation or estimation methods for $\frac{dL(\theta)}{d\theta}$ based on finite differences of the form $\frac{L(\theta+\Delta\theta)-L(\theta)}{\Delta\theta}$. Such methods require two state trajectories under θ and $\theta+\Delta\theta$, respectively, and are vulnerable to numerical problems when $\Delta\theta$ is selected to be small so as to increase the accuracy of the derivative approximation.

The final step then is to make use of $\nabla L(\theta)$ in a gradient-based adaptation mechanism of the general form $\theta_{n+1} = \theta_n + \eta_n \nabla L(\theta)$, where $n = 1, 2, \ldots$ counts the iterations over which this process evolves, and $\{\eta_n\}$ is a step size sequence which is appropriately selected to ensure convergence of the controllable parameter sequence $\{\theta_n\}$ under proper stationarity assumptions. After each iteration, the controller is adjusted, which obviously affects the behavior of the system, and the process repeats. Clearly, in a stochastic setting there is no guarantee of stationarity conditions, and this framework is simply one where the controller is perpetually seeking to improve system performance.

The cornerstone of this *online* framework is the evaluation of $\nabla L(\theta)$ based *only* on data obtained from the observed state trajectory. The theory of IPA [32,35] provides the foundations for this to be possible. Moreover, in the stochastic case where $\nabla L(\theta)$ becomes an estimate of $\nabla E[L(\theta)]$, it is important that this estimate possess desirable properties such as unbiasedness, without which the ultimate goal of achieving optimality cannot be provably attained. As we see in the next section, it is possible to evaluate $\nabla L(\theta)$ for virtually arbitrary SHS through a simple systematic event-driven procedure we refer to as the "IPA calculus." In addition, this gradient is characterized by several attractive properties under mild technical conditions.

In order to formally apply IPA and subsequent control and optimization methods to hybrid systems, we need to establish a general modeling framework. We use a standard definition of a hybrid automaton [33].

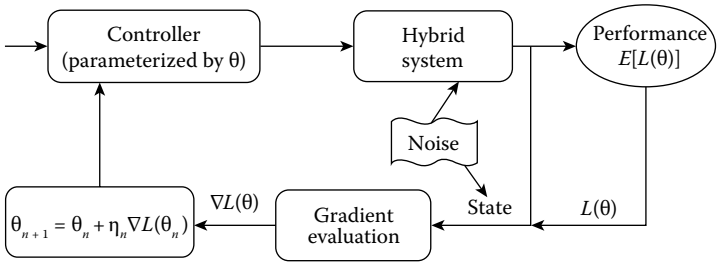

FIGURE 2.1

Online control and optimization framework for hybrid systems.

Thus, let $q \in Q$ (a countable set) denote the discrete state (or mode) and $x \in X \subseteq \mathbb{R}^n$ denote the continuous state of the hybrid system. Let $\upsilon \in \Upsilon$ (a countable set) denote a discrete control input and $u \in U \subseteq \mathbb{R}^m$ a continuous control input. Similarly, let $\delta \in \Delta$ (a countable set) denote a discrete disturbance input and $d \in D \subseteq \mathbb{R}^p$ a continuous disturbance input. The state evolution is determined by means of

- A vector field $f : Q \times X \times U \times D \to X$
- An invariant (or domain) set $Inv : Q \times \Upsilon \times \Delta \to 2^X$
- A guard set $Guard : Q \times Q \times \Upsilon \times \Delta \to 2^X$
- A reset function $r : Q \times Q \times X \times \Upsilon \times \Delta \to X$

A trajectory or sample path of such a system consists of a sequence of intervals of continuous evolution followed by a discrete transition. The system remains at a discrete state q as long as the continuous (time-driven) state x does not leave the set $Inv(q, \upsilon, \delta)$. If, before reaching $Inv(q, \upsilon, \delta)$, x reaches a set $Guard(q, q', \upsilon, \delta)$ for some $q' \in Q$, a discrete transition is allowed to take place. If this transition does take place, the state instantaneously resets to (q', x'), where x' is determined by the reset map $r(q, q', x, \upsilon, \delta)$. Changes in the discrete controls υ and disturbances δ are discrete events that either *enable* a transition from q to q' when $x \in Guard(q, q', \upsilon, \delta)$ or *force* a transition out of q by making sure $x \notin Inv(q, \upsilon, \delta)$. We also use $\mathcal{E}$ to denote the set of all events that cause discrete state transitions and will classify events in a manner that suits the purposes of perturbation analysis. In what follows, we provide an overview of the "IPA calculus" and refer the reader to [32] and [36] for more details.

2.3 IPA: Event-Driven IPA Calculus

In this section, we describe the general framework for IPA as presented in [37] and generalized in [32] and [36]. Let $\theta \in \Theta \subset \mathbb{R}^l$ be a global variable, henceforth called the *control parameter*, where Θ is a given compact, convex set. This may include system design parameters, parameters of an input process, or parameters that characterize a policy used in controlling this system. The disturbance input $d \in D$ encompasses various random processes that affect the evolution of the state (q, x) so that, in general, we can deal with a SHS. We will assume that all such processes are defined over a common probability space, $(\Omega, \mathcal{F}, P)$. Let us fix a particular value of the parameter $\theta \in \Theta$ and study a resulting sample path of the SHS. Over such a sample path, let $\tau_k(\theta)$, $k = 1, 2, \ldots$, denote the occurrence times of the discrete events in increasing order, and define $\tau_0(\theta) = 0$ for convenience. We will use the notation τ_k instead of $\tau_k(\theta)$ when no confusion arises. The continuous state is also generally a function of θ, as well as of t, and is thus denoted by $x(\theta, t)$. Over an interval $[\tau_k(\theta), \tau_{k+1}(\theta))$, the system is at some mode during which the time-driven state satisfies:

$$\dot{x} = f_k(x, \theta, t), \tag{2.1}$$

where $\dot{x}$ denotes $\frac{\partial x}{\partial t}$. Note that we suppress the dependence of f_k on the inputs $u \in U$ and $d \in D$ and stress instead its dependence on the parameter θ which may generally affect either u or d or both. The purpose of perturbation analysis is to study how changes in θ influence the state $x(\theta, t)$ and the event times $\tau_k(\theta)$ and, ultimately, how they influence interesting performance metrics that are generally expressed in terms of these variables. The following assumption guarantees that (2.1) has a unique solution w.p.1 for a given initial boundary condition $x(\theta, \tau_k)$ at time $\tau_k(\theta)$.

ASSUMPTION 2.1 W.p.1, there exists a finite set of points $t_j \in [\tau_k(\theta), \tau_{k+1}(\theta))$, $j = 1, 2, \ldots$, which are independent of θ, such that the function f_k is continuously differentiable on $\mathbb{R}^n \times \Theta \times ([\tau_k(\theta), \tau_{k+1}(\theta)) \setminus \{t_1, t_2 \ldots\})$. Moreover, there exists a random number $K > 0$ such that $E[K] < \infty$ and the norm of the first derivative of f_k on $\mathbb{R}^n \times \Theta \times ([\tau_k(\theta), \tau_{k+1}(\theta)) \setminus \{t_1, t_2 \ldots\})$ is bounded from above by K.

An event occurring at time $\tau_{k+1}(\theta)$ triggers a change in the mode of the system, which may also result in new dynamics represented by f_{k+1}, although this may not always be the case; for example, two modes may be distinct because the state $x(\theta, t)$ enters a new region where the system's performance is measured differently without altering its time-driven dynamics (i.e., $f_{k+1} = f_k$). The event times $\{\tau_k(\theta)\}$ play an important role in defining the interactions between the time-driven and event-driven dynamics of the system.

We now classify events that define the set $\mathcal{E}$ as follows:

- *Exogenous*: An event is exogenous if it causes a discrete state transition at time τ_k independent of the controllable vector θ and satisfies $\frac{d\tau_k}{d\theta} = 0$. Exogenous events typically correspond to uncontrolled random changes in input processes.

- *Endogenous*: An event occurring at time τ_k is endogenous if there exists a continuously differentiable function $g_k : \mathbb{R}^n \times \Theta \to \mathbb{R}$ such that

$$\tau_k = \min\{t > \tau_{k-1} \ : \ g_k(x(\theta, t), \theta) = 0\}. \tag{2.2}$$

The function g_k is normally associated with an invariant or a guard condition in a hybrid automaton model.

- *Induced*: An event at time τ_k is induced if it is triggered by the occurrence of another event at time $\tau_m \leq \tau_k$. The triggering event may be exogenous, endogenous, or itself an induced event. The events that trigger induced events are identified by a subset of the event set, $\mathcal{E}_I \subseteq \mathcal{E}$.

Although this event classification is sufficiently general, recent work has shown that in some cases, it is convenient to introduce further event distinctions [38]. Moreover, it has been shown in [36] that an explicit event classification is in fact unnecessary if one is willing to appropriately extend the definition of the hybrid automaton described earlier. However, for the rest of this chapter, we only make use of the above classification.

Next, consider a performance function of the control parameter θ:

$$J(\theta; x(\theta,0), T) = E[L(\theta; x(\theta,0), T)],$$

where $L(\theta; x(\theta,0), T)$ is a sample function of interest evaluated in the interval $[0,T]$ with initial conditions $x(\theta,0)$. For simplicity, we write $J(\theta)$ and $L(\theta)$. Suppose that there are N events, with occurrence times generally dependent on θ, during the time interval $[0,T]$ and define $\tau_0 = 0$ and $\tau_{N+1} = T$. Let $L_k : \mathbb{R}^n \times \Theta \times \mathbb{R}^+ \to \mathbb{R}$ be a function satisfying Assumption 2.1 and define $L(\theta)$ by

$$L(\theta) = \sum_{k=0}^{N} \int_{\tau_k}^{\tau_{k+1}} L_k(x,\theta,t)dt, \tag{2.3}$$

where we reiterate that $x = x(\theta,t)$ is a function of θ and t. We also point out that the restriction of the definition of $J(\theta)$ to a finite horizon T which is independent of θ is made merely for the sake of simplicity.

Returning to Figure 2.1 and considering (for the sake of generality) the stochastic setting, the ultimate goal of the iterative process shown is to maximize $E_\omega[L(\theta,\omega)]$, where we use ω to emphasize dependence on a sample path ω of a SHS (clearly, this is reduced to $L(\theta)$ in the deterministic case). Achieving such optimality is possible under standard ergodicity conditions imposed on the underlying stochastic processes, as well as the assumption that a single global optimum exists; otherwise, the gradient-based approach is simply continuously attempting to improve the observed performance $L(\theta,\omega)$. Thus, we are interested in estimating the gradient

$$\frac{dJ(\theta)}{d\theta} = \frac{dE_\omega[L(\theta,\omega)]}{d\theta},$$

by evaluating $\frac{dL(\theta,\omega)}{d\theta}$ based on *directly observed data*. We obtain θ^* (under the conditions mentioned above) by optimizing $J(\theta)$ through an iterative scheme of the form

$$\theta_{n+1} = \theta_n - \eta_n H_n(\theta_n; x(\theta,0), T, \omega_n), \quad n = 0,1,\ldots, \tag{2.4}$$

where $\{\eta_n\}$ is a step size sequence and $H_n(\theta_n; x(\theta,0), T, \omega_n)$ is the estimate of $\frac{dJ(\theta)}{d\theta}$ at $\theta = \theta_n$. In using IPA, $H_n(\theta_n; x(\theta,0), T, \omega_n)$ is the sample derivative $\frac{dL(\theta,\omega)}{d\theta}$, which is an unbiased estimate of $\frac{dJ(\theta)}{d\theta}$ if the condition (dropping the symbol ω for simplicity)

$$E\left[\frac{dL(\theta)}{d\theta}\right] = \frac{d}{d\theta}E[L(\theta)] = \frac{dJ(\theta)}{d\theta}, \tag{2.5}$$

is satisfied, which turns out to be the case under mild technical conditions to be discussed later. The conditions under which algorithms of the form (2.4) converge are well-known (e.g., see [39]). Moreover, in addition to being unbiased, it can be shown that such gradient estimates are independent of the probability laws of the stochastic processes involved and require minimal information from the observed sample path.

The process through which IPA evaluates $\frac{dL(\theta)}{d\theta}$ is based on analyzing how changes in θ influence the state $x(\theta,t)$ and the event times $\tau_k(\theta)$. In turn, this provides information on how $L(\theta)$ is affected, because it is generally expressed in terms of these variables. Given $\theta = [\theta_1,\ldots,\theta_l]^T$, we use the Jacobian matrix notation:

$$x'(\theta,t) \equiv \frac{\partial x(\theta,t)}{\partial \theta}, \quad \tau_k' \equiv \frac{\partial \tau_k(\theta)}{\partial \theta}, \quad k = 1,\ldots,K,$$

for all state and event time derivatives. For simplicity of notation, we omit θ from the arguments of the functions above unless it is essential to stress this dependence. It is shown in [32] that $x'(t)$ satisfies

$$\frac{d}{dt}x'(t) = \frac{\partial f_k(t)}{\partial x}x'(t) + \frac{\partial f_k(t)}{\partial \theta}, \tag{2.6}$$

for $t \in [\tau_k(\theta), \tau_{k+1}(\theta))$ with boundary condition

$$x'(\tau_k^+) = x'(\tau_k^-) + [f_{k-1}(\tau_k^-) - f_k(\tau_k^+)]\tau_k', \tag{2.7}$$

for $k = 0,\ldots,K$. We note that whereas $x(t)$ is often continuous in t, $x'(t)$ may be discontinuous in t at the event times τ_k; hence, the left and right limits above are generally different. If $x(t)$ is not continuous in t at $t = \tau_k(\theta)$, the value of $x(\tau_k^+)$ is determined by the reset function $r(q,q',x,\upsilon,\delta)$ discussed earlier and

$$x'(\tau_k^+) = \frac{dr(q,q',x,\upsilon,\delta)}{d\theta}. \tag{2.8}$$

Furthermore, once the initial condition $x'(\tau_k^+)$ is given, the linearized state trajectory $\{x'(t)\}$ can be computed

in the interval $t \in [\tau_k(\theta), \tau_{k+1}(\theta))$ by solving (2.6) to obtain

$$x'(t) = e^{\int_{\tau_k}^{t} \frac{\partial f_k(u)}{\partial x} du} \left[\int_{\tau_k}^{t} \frac{\partial f_k(v)}{\partial \theta} e^{-\int_{\tau_k}^{t} \frac{\partial f_k(u)}{\partial x} du} dv + \xi_k \right], \tag{2.9}$$

with the constant ξ_k determined from $x'(\tau_k^+)$ in either (2.7) or (2.8).

In order to complete the evaluation of $x'(\tau_k^+)$ in (2.7), we need to also determine τ_k'. Based on the event classification above, $\tau_k' = 0$ if the event at $\tau_k(\theta)$ is exogenous and

$$\tau_k' = -\left[\frac{\partial g_k}{\partial x} f_k(\tau_k^-)\right]^{-1} \left(\frac{\partial g_k}{\partial \theta} + \frac{\partial g_k}{\partial x} x'(\tau_k^-)\right), \tag{2.10}$$

if the event at $\tau_k(\theta)$ is endogenous, that is, $g_k(x(\theta, \tau_k), \theta) = 0$, defined as long as $\frac{\partial g_k}{\partial x} f_k(\tau_k^-) \neq 0$. (Details may be found in [32].) Finally, if an induced event occurs at $t = \tau_k$ and is triggered by an event at $\tau_m \leq \tau_k$, the value of τ_k' depends on the derivative τ_m'. The event induced at τ_m will occur at some time $\tau_m + w(\tau_m)$, where $w(\tau_m)$ is a (generally random) variable which is dependent on the continuous and discrete states $x(\tau_m)$ and $q(\tau_m)$, respectively. This implies the need for additional state variables, denoted by $y_m(\theta, t)$, $m = 1, 2, \ldots$, associated with events occurring at times τ_m, $m = 1, 2 \ldots$. The role of each such a state variable is to provide a "timer" activated when a triggering event occurs. Triggering events are identified as belonging to a set $\mathcal{E}_I \subseteq \mathcal{E}$ and letting e_k denote the event occurring at τ_k. Then, define $F_k = \{m : e_m \in \mathcal{E}_I, m \leq k\}$ to be the set of all indices with corresponding triggering events up to τ_k. Omitting the dependence on θ for simplicity, the dynamics of $y_m(t)$ are then given by

$$\dot{y}_m(t) = \begin{cases} -C(t) & \tau_m \leq t < \tau_m + w(\tau_m),\ m \in F_m \\ 0 & \text{otherwise} \end{cases}, \tag{2.11}$$

$$y_m(\tau_m^+) = \begin{cases} y_0 & y_m(\tau_m^-) = 0,\ m \in F_m \\ 0 & \text{otherwise} \end{cases},$$

where y_0 is an initial value for the timer $y_m(t)$, which decreases at a "clock rate" $C(t) > 0$ until $y_m(\tau_m + w(\tau_m)) = 0$ and the associated induced event takes place. Clearly, these state variables are only used for induced events, so that $y_m(t) = 0$ unless $m \in F_m$. The value of y_0 may depend on θ or on the continuous and discrete states $x(\tau_m)$ and $q(\tau_m)$, while the clock rate $C(t)$ may depend on $x(t)$ and $q(t)$ in general, and possibly θ. However, in most simple cases where we are interested in modeling an induced event to occur at time $\tau_m + w(\tau_m)$, we have $y_0 = w(\tau_m)$ and $C(t) = 1$—that is, the timer simply counts down for a total of $w(\tau_m)$ time units until the induced event takes place. Henceforth, we will consider $y_m(t)$, $m = 1, 2, \ldots$, as part of the continuous state of the SHS, and we set

$$y_m'(t) \equiv \frac{\partial y_m(t)}{\partial \theta}, \quad m = 1, \ldots, N. \tag{2.12}$$

For the common case where y_0 is independent of θ and $C(t)$ is a constant $c > 0$ in (2.11), Lemma 2.1 facilitates the computation of τ_k' for an induced event occurring at τ_k. Its proof is given in [32].

Lemma 2.1

If in (2.11), y_0 is independent of θ and $C(t) = c > 0$ (constant), then $\tau_k' = \tau_m'$.

With the inclusion of the state variables $y_m(t)$, $m = 1, \ldots, N$, the derivatives $x'(t)$, τ_k', and $y_m'(t)$ can be evaluated through (2.6)–(2.11) and this set of equations is what we refer to as the "IPA calculus." In general, this evaluation is recursive over the event (mode switching) index $k = 0, 1, \ldots$. In other words, *the IPA estimation process is entirely event driven*. For a large class of problems, the SHS of interest does not involve induced events, and the state does not experience discontinuities when a mode-switching event occurs. In this case, the IPA calculus reduces to the application of three equations:

1. Equation 2.9:

$$x'(t) = e^{\int_{\tau_k}^{t} \frac{\partial f_k(u)}{\partial x} du} \left[\int_{\tau_k}^{t} \frac{\partial f_k(v)}{\partial \theta} e^{-\int_{\tau_k}^{t} \frac{\partial f_k(u)}{\partial x} du} dv + \xi_k \right],$$

   which describes how the state derivative $x'(t)$ evolves over $[\tau_k(\theta), \tau_{k+1}(\theta))$.

2. Equation 2.7:

$$x'(\tau_k^+) = x'(\tau_k^-) + [f_{k-1}(\tau_k^-) - f_k(\tau_k^+)]\tau_k',$$

   which specifies the initial condition ξ_k in (2.9).

3. Either $\tau_k' = 0$ or Equation 2.10:

$$\tau_k' = -\left[\frac{\partial g_k}{\partial x} f_k(\tau_k^-)\right]^{-1} \left(\frac{\partial g_k}{\partial \theta} + \frac{\partial g_k}{\partial x} x'(\tau_k^-)\right),$$

   depending on the event type at $\tau_k(\theta)$, which specifies the event time derivative present in (2.7).

From a computational standpoint, the IPA derivative evaluation process takes place iteratively at each event defining a mode transition at some time instant $\tau_k(\theta)$. At this point in time, we have at our disposal the value of $x'(\tau_{k-1}^+)$ from the previous iteration, which specifies ξ_{k-1}

in (2.9) applied for all $t \in [\tau_{k-1}(\theta), \tau_k(\theta))$. Therefore, setting $t = \tau_k(\theta)$ in (2.9) we also have at our disposal the value of $x'(\tau_k^-)$. Next, depending on whether the event is exogenous or endogenous, the value of τ_k' can be obtained: it is either $\tau_k' = 0$ or given by (2.10) since $x'(\tau_k^-)$ is known. Finally, we obtain $x'(\tau_k^+)$ using (2.7). At this point, one can wait until the next event occurs at $\tau_{k+1}(\theta)$ and repeat the process which can, therefore, be seen to be entirely *event driven*.

The last step in the IPA process involves using the IPA calculus in order to evaluate the IPA derivative $dL/d\theta$. This is accomplished by taking derivatives in (2.3) with respect to θ:

$$\frac{dL(\theta)}{d\theta} = \sum_{k=0}^{N} \frac{d}{d\theta} \int_{\tau_k}^{\tau_{k+1}} L_k(x, \theta, t) dt. \tag{2.13}$$

Applying the Leibnitz rule, we obtain, for every $k = 0, \ldots, N$,

$$\begin{aligned} &\frac{d}{d\theta} \int_{\tau_k}^{\tau_{k+1}} L_k(x, \theta, t) dt \\ &\quad = \int_{\tau_k}^{\tau_{k+1}} \left[\frac{\partial L_k}{\partial x}(x, \theta, t) x'(t) + \frac{\partial L_k}{\partial \theta}(x, \theta, t) \right] dt \\ &\qquad + L_k(x(\tau_{k+1}), \theta, \tau_{k+1}) \tau_{k+1}' - L_k(x(\tau_k), \theta, \tau_k) \tau_k', \end{aligned} \tag{2.14}$$

where $x'(t)$ and τ_k' are determined through (2.6)–(2.10). What makes IPA appealing is the simple form the right-hand-side in Equation 2.14 often assumes. As we will see, under certain commonly encountered conditions, this expression is further simplified by eliminating the integral term.

2.4 IPA Properties

In this section, we identify three key properties of IPA. The first one is important in ensuring that when IPA involves estimates of gradients, these estimates are unbiased under mild conditions. The second is a robustness property of IPA derivatives in the sense that they do not depend on specific probabilistic characterizations of any stochastic processes involved in the hybrid automaton model of a SHS. This property holds under certain sufficient conditions which are easy to check. Finally, under conditions pertaining to the switching function $g_k(x, \theta)$, which we have used to define endogenous events, the event-driven IPA derivative evaluation or estimation process includes some events that have the property of allowing us to decompose an observed state trajectory into cycles, thus greatly simplifying the overall computational effort.

2.4.1 Unbiasedness

We begin by returning to the issue of unbiasedness of the sample derivatives $\frac{dL(\theta)}{d\theta}$ derived using the IPA calculus described in the last section. In particular, the IPA derivative $\frac{dL(\theta)}{d\theta}$ is an unbiased estimate of the performance (or cost) derivative $\frac{dJ(\theta)}{d\theta}$ if the condition (2.5) holds. In a pure DES, the IPA derivative satisfies this condition for a relatively limited class of systems (see [1,31]). This has motivated the development of more sophisticated perturbation analysis methods that can still guarantee unbiasedness at the expense of additional information to be collected from the observed sample path or additional assumptions regarding the statistical properties of some of the random processes involved. However, in a SHS, the technical conditions required to guarantee the validity of (2.5) are almost always applicable.

The following result has been established in [40] regarding the unbiasedness of IPA:

Theorem 2.1

Suppose that the following conditions are in force: (1) For every $\theta \in \Theta$, the derivative $\frac{dL(\theta)}{d\theta}$ exists w.p.1. (2) W.p.1, the function $L(\theta)$ is Lipschitz continuous on Θ, and the Lipschitz constant has a finite first moment. Then, for a fixed $\theta \in \Theta$, the derivative $\frac{dJ(\theta)}{d\theta}$ exists, and the IPA derivative $\frac{dL(\theta)}{d\theta}$ is unbiased.

The crucial assumption for Theorem 2.1 is the continuity of the sample function $L(\theta)$, which in many SHSs is guaranteed in a straightforward manner. Differentiability w.p.1 at a given $\theta \in \Theta$ often follows from mild technical assumptions on the probability law underlying the system, such as the exclusion of co-occurrence of multiple events (see [41]). Lipschitz continuity of $L(\theta)$ generally follows from upper boundedness of $|\frac{dL(\theta)}{d\theta}|$ by an absolutely integrable random variable, generally a weak assumption. In light of these observations, the proofs of unbiasedness of IPA have become standardized, and the assumptions in Theorem 2.1 can be verified fairly easily from the context of a particular problem.

2.4.2 Robustness to Stochastic Model Uncertainties

Next, we turn our attention to properties of the estimators obtained through the IPA calculus which render them, under certain conditions, particularly simple and efficient to implement with minimal information required about the underlying SHS dynamics.

The first question we address is related to $\frac{dL(\theta)}{d\theta}$ in (2.13), which, as seen in (2.14), generally depends on information accumulated over all $t \in [\tau_k, \tau_{k+1})$. It is, however, often the case that it depends *only* on information related to the event times τ_k, τ_{k+1}, resulting in an IPA estimator that is simple to implement. Using the notation $L'_k(x,t,\theta) \equiv \frac{dL_k(x,t,\theta)}{d\theta}$, we can rewrite $\frac{dL(\theta)}{d\theta}$ in (2.13) as

$$\frac{dL(\theta)}{d\theta} = \sum_k \left[\tau'_{k+1} \cdot L_k(\tau^+_{k+1}) - \tau'_k \cdot L_k(\tau^+_k) + \int_{\tau_k}^{\tau_{k+1}} L'_k(x,t,\theta)dt \right]. \quad (2.15)$$

The following theorem provides two sufficient conditions under which $\frac{dL(\theta)}{d\theta}$ involves only the event time derivatives τ'_k, τ'_{k+1} and the "local" performance $L_k(\tau^+_{k+1})$, $L_k(\tau^+_k)$, which is obviously easy to observe. The proof of this result is given in [42].

Theorem 2.2

If condition $(C1)$ or $(C2)$ below holds, then $\frac{dL(\theta)}{d\theta}$ depends only on information available at event times $\{\tau_k\}$, $k = 0, 1, \ldots$.

$(C1)$ $L_k(x,t,\theta)$ is independent of t over $[\tau_k, \tau_{k+1})$ for all $k = 0, 1, \ldots$.

$(C2)$ $L_k(x,t,\theta)$ is only a function of x, and the following condition holds for all $t \in [\tau_k, \tau_{k+1})$, $k = 0, 1, \ldots$:

$$\frac{d}{dt}\frac{\partial L_k}{\partial x} = \frac{d}{dt}\frac{\partial f_k}{\partial x} = \frac{d}{dt}\frac{\partial f_k}{\partial \theta} = 0. \quad (2.16)$$

The implication of Theorem 2.2 is that (2.15), under either $(C1)$ or $(C2)$, reduces to

$$\frac{dL(\theta)}{d\theta} = \sum_k [\tau'_{k+1} \cdot L_k(\tau^+_{k+1}) - \tau'_k \cdot L_k(\tau^+_k)],$$

and involves *only* directly observable performance sample values at event times along with event time derivatives which are either zero (for exogenous events) or given by (2.10). The conditions in Theorem 2.2 are surprisingly easy to satisfy as the following example illustrates.

EXAMPLE 2.1 Consider a SHS whose time-driven dynamics at all modes are linear and of the form

$$\dot{x} = a_k x(t) + b_k u_k(\theta, t) + w_k(t), \quad t \in [\tau_{k-1}(\theta), \tau_k(\theta)),$$

where $u_k(\theta,t)$ is a control used in the system mode over $[\tau_k(\theta), \tau_{k+1}(\theta))$, which depends on a parameter θ and $w_k(t)$ is some random process for which no further information is provided. Writing $f_k = a_k x(t) + b_k u_k(\theta,t) + w_k(t)$, we can immediately see that $\frac{\partial f_k}{\partial x} = a_k$ and $\frac{\partial f_k}{\partial \theta} = \frac{\partial u_k(\theta,t)}{\partial \theta}$; hence, the second of the three parts of $(C2)$ is satisfied—that is, $\frac{d}{dt}\frac{\partial f_k}{\partial x} = 0$. Further, suppose that the dependence of $u_k(\theta,t)$ on t is such that $\frac{\partial u_k(\theta,t)}{\partial \theta}$ is also independent of t; this is true, for instance, if $u_k(\theta,t) = u_k(\theta)$, that is, the control is fixed at that mode, or if $u_k(\theta,t) = \gamma(\theta)t$, in which case $\frac{d}{dt}\frac{\partial f_k}{\partial \theta} = 0$, and the last part of $(C2)$ is also satisfied. Finally, consider a performance metric of the form

$$J(\theta) = E\left[\sum_{k=0}^{N} \int_{\tau_k}^{\tau_{k+1}} L_k(x,\theta,t)dt\right] = E\left[\sum_{k=0}^{N} \int_{\tau_k}^{\tau_{k+1}} x(t)dt\right],$$

where we have $\frac{\partial L_k}{\partial x} = 1$, thus satisfying also the first part of $(C2)$. It is worthwhile pointing out that the IPA calculus here provides unbiased estimates of $\frac{dJ(\theta)}{d\theta}$ without any information regarding the noise process $w_k(t)$. Although this seems surprising at first, the fact is that the effect of the noise is captured through the values of the observable event times $\tau_k(\theta)$ and the observed performance values $L_k(\tau^+_k)$ at these event times only: modeling information about $w_k(t)$ is traded against observations made online at event times only. In other words, while the noise information is crucial if one is interested in the actual performance $\int_{\tau_k}^{\tau_{k+1}} L_k(x,\theta,t)dt$ over an interval $[\tau_{k-1}(\theta), \tau_k(\theta))$, such information is not always required to estimate the *sensitivity* of the performance $\int_{\tau_k}^{\tau_{k+1}} L_k(x,\theta,t)dt$ with respect to θ.

We refer to the property reflected by Theorem 2.2 as "robustness" of IPA derivative estimators with respect to any noise process affecting the time-driven dynamics of the system. Clearly, that would not be the case if, for instance, the performance metric involved $x^2(t)$ instead of $x(t)$; then, $\frac{\partial L_k}{\partial x} = 2x(t)$ and the integral term in (2.15) would have to be included in the evaluation of $\frac{dL(\theta)}{d\theta}$. Although this increases the computational burden of the IPA evaluation procedure and requires the collection of sample data for $w_k(t)$, note that it still requires no prior modeling information regarding this random process.

Thus, one need not have a detailed model (captured by f_{k-1}) to describe the state behavior through $\dot{x} = f_{k-1}(x,\theta,t)$, $t \in [\tau_{k-1}, \tau_k)$ in order to estimate the effect of θ on this behavior. This explains why simple abstractions of a complex stochastic system are often adequate to perform sensitivity analysis and optimization, as long as the event times corresponding to discrete state transitions are accurately observed and the local system behavior at these event times, for example, $x'(\tau^+_k)$ in (2.7), can also be measured or calculated.

2.4.3 State Trajectory Decomposition

The final IPA property we discuss is related to the discontinuity in $x'(t)$ at event times, described in (2.7). This happens when endogenous events occur, since for exogenous events we have $\tau_k' = 0$. The next theorem identifies a simple condition under which $x'(\tau_k^+)$ is independent of the dynamics f before the event at τ_k. This implies that we can evaluate the sensitivity of the state with respect to θ without any knowledge of the state trajectory in the interval $[\tau_{k-1}, \tau_k)$ prior to this event. Moreover, under an additional condition, we obtain $x'(\tau_k^+) = 0$, implying that the effect of θ is "forgotten," and one can reset the perturbation process. This allows us to decompose an observed state trajectory (or sample path) into "reset cycles," greatly simplifying the IPA process. The proof of the next result is also given in [42].

Theorem 2.3

Suppose an endogenous event occurs at $\tau_k(\theta)$ with a switching function $g(x, \theta)$. If $f_k(\tau_k^+) = 0$, then $x'(\tau_k^+)$ is independent of f_{k-1}. If, in addition, $\frac{\partial g}{\partial \theta} = 0$, then $x'(\tau_k^+) = 0$.

The condition $f_k(\tau_k^+) = 0$ typically indicates a saturation effect or the state reaching a boundary that cannot be crossed, for example, when the state is constrained to be nonnegative. This often arises in stochastic flow systems used to model how parts are processed in manufacturing systems or how packets are transmitted and received through a communication network [43,44]. In such cases, the conditions of both Theorems 2.1 and 2.2 are frequently satisfied since (1) common performance metrics such as workload or overflow rates satisfy (2.16) and (2) flow systems involve nonnegative continuous states and are constrained by capacities that give rise to dynamics of the form $\dot{x} = 0$. This class of SHS is also referred to as stochastic flow models, and the simplicity of the IPA derivatives in this case has been thoroughly analyzed, for example, see [35,45]. We present an illustrative example below.

EXAMPLE 2.2 Consider the fluid single-queue system shown in Figure 2.2, where the arrival-rate process $\{\alpha(t)\}$ and the service-rate process $\{\beta(t)\}$ are random processes (possibly correlated) defined on a common probability space. The queue has a finite buffer, $\{x(t)\}$ denotes the buffer workload (amount of fluid in the buffer), and $\{\gamma(t)\}$ denotes the overflow of excess fluid when the buffer is full. Let the controllable parameter θ be the buffer size, and consider the sample performance function to be the loss volume during a given horizon interval $[0, T]$, namely,

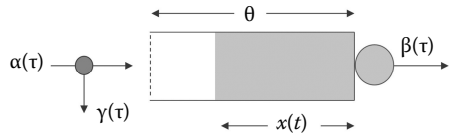

FIGURE 2.2
A simple fluid queue system for Example 2.2.

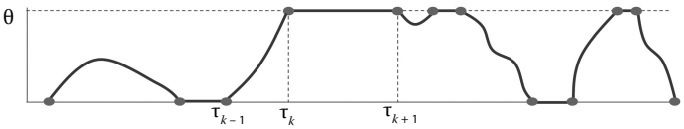

FIGURE 2.3
A typical sample path of the system in Figure 2.2.

$$L(\theta) = \int_0^T \gamma(\theta, t)dt. \tag{2.17}$$

We assume that $\alpha(t)$ and $\beta(t)$ are independent of θ, and note that the buffer workload and overflow processes certainly depend upon θ; hence, they are denoted by $\{x(\theta, t)\}$ and $\{\gamma(\theta, t)\}$, respectively. The only other assumptions we make on the arrival process and service process are that, w.p.1, $\alpha(t)$ and $\beta(t)$ are piecewise continuously differentiable in t (but need not be continuous), and the terms $\int_0^T \alpha(t)dt$ and $\int_0^T \beta(t)dt$ have finite first moments. In addition, to satisfy the first condition of Theorem 2.1, we assume that w.p.1 no two events can occur at the same time (unless one induces the other), thus ensuring the existence of $\frac{dL(\theta)}{d\theta}$.

The time-driven dynamics in this SHS are given by

$$\dot{x}(\theta, t) = \begin{cases} 0 & \text{if } x(\theta, t) = 0, \alpha(t) \le \beta(t) \\ 0 & \text{if } x(\theta, t) = \theta, \alpha(t) \ge \beta(t). \\ \alpha(t) - \beta(t) & \text{otherwise} \end{cases} \tag{2.18}$$

A typical sample path of the process $\{x(\theta, t)\}$ is shown in Figure 2.3. Observe that there are two endogenous events in this system: the first is when $x(\theta, t)$ increases and reaches the value $x(\theta, t) = \theta$ (as happens at time τ_k in Figure 2.3) and the second is when $x(\theta, t)$ decreases and reaches the value $x(\theta, t) = 0$. Thus, we see that the sample path is partitioned into intervals over which $x(\theta, t) = 0$, termed empty periods (EPs) since the fluid queue in Figure 2.2 is empty, and intervals over which $x(\theta, t) > 0$, termed nonempty periods (NEPs).

We can immediately see that Theorem 2.3 applies here for endogenous events with $g(x, \theta) = x$, which occur when an EP starts at some event time τ_k. Since $\frac{\partial g}{\partial \theta} = 0$ and $f_k(\tau_k^+) = 0$ from (2.18), it follows that $x'(\tau_k^+) = 0$

and remains at this value throughout every EP. Therefore, the effect of the parameter θ in this case need only be analyzed over NEPs.

Next, observe that $\gamma(\theta,t)>0$ only when $x(\theta,t)=\theta$. We refer to any such interval as a full period (FP) since the fluid queue in Figure 2.2 is full, and note that we can write $L(\theta)$ in (2.17) as

$$L(\theta)=\sum_{k\in\Psi_T}\int_{\tau_k}^{\tau_{k+1}}[\alpha(t)-\beta(t)]dt,$$

where $\Psi_T=\{k: x(\theta,t)=\theta \text{ for all } t\in[\tau_k(\theta),\tau_{k+1}(\theta))\}$ is the set of all FPs in the observed sample path over $[0,T]$. It follows that

$$\frac{dL(\theta)}{d\theta}=\sum_{k\in\Psi_T}[\alpha(\tau_{k+1}^-)-\beta(\tau_{k+1}^-)]\tau_{k+1}^{'}-\sum_{k\in\Psi_T}[\alpha(\tau_k^+)-\beta(\tau_k^+)]\tau_k^{'}, \tag{2.19}$$

and this is a case where condition $(C2)$ of Theorem 2.2 holds: $\frac{d}{dt}\frac{\partial L_k}{\partial x}=\frac{d}{dt}[\alpha(t)-\beta(t)]=0$ and $\frac{d}{dt}\frac{\partial f_k}{\partial x}=\frac{d}{dt}\frac{\partial f_k}{\partial\theta}=0$ since $f_k=\alpha(t)-\beta(t)$ from (2.18). Thus, the evaluation of $\frac{dL(\theta)}{d\theta}$ reduces to the evaluation of $\tau_{k+1}^{'}$ and $\tau_k^{'}$ at the end and start, respectively, of every FP. Observing that $\tau_{k+1}^{'}=0$ since the end of a FP is an exogenous event depending only on a change in sign of $[\alpha(t)-\beta(t)]$ from nonnegative to strictly negative, it only remains to use the IPA calculus to evaluate $\tau_k^{'}$ for every endogenous event such that $g(x(\theta,\tau_k),\theta)=x-\theta$. Applying (2.10) gives:

$$\tau_k^{'}=\frac{1-x'(\tau_k^-)}{\alpha(\tau_k^-)-\beta(\tau_k^-)}.$$

The value of $x'(\tau_k^-)$ is obtained using (2.9) over the interval $[\tau_{k-1}(\theta),\tau_k(\theta))$:

$$x'(\tau_k^-)=e^{\int_{\tau_{k-1}}^{\tau_k^-}\frac{\partial f_k(u)}{\partial x}du}\times\left[\int_{\tau_{k-1}}^{\tau_k^-}\frac{\partial f_k(v)}{\partial\theta}e^{-\int_{\tau_{k-1}}^{\tau_k^-}\frac{\partial f_k(u)}{\partial x}du}dv+x'(\tau_{k-1}^+)\right],$$

where $\frac{\partial f_k(u)}{\partial x}=\frac{\partial f_k(u)}{\partial x}=0$ and $\frac{\partial f_k(v)}{\partial\theta}=0$. Moreover, using (2.7) at $t=\tau_{k-1}$, we have $x'(\tau_{k-1}^+)=x'(\tau_{k-1}^-)+[f_{k-1}(\tau_{k-1}^-)-f_k(\tau_{k-1}^+)]\tau_{k-1}'=0$, since the start of a NEP is an exogenous event so that $\tau_{k-1}'=0$ and $x'(\tau_{k-1}^-)=0$ as explained earlier. Thus, $x'(\tau_k^-)=0$, yielding

$$\tau_k^{'}=\frac{1}{\alpha(\tau_k^-)-\beta(\tau_k^-)}.$$

Recalling our assumption that w.p.1 no two events can occur at the same time, $\alpha(t)$ and $\beta(t)$ can experience no discontinuities (exogenous events) at $t=\tau_k$ when the endogenous event $x(\theta,t)=\theta$ takes place, that is, $\alpha(\tau_k^-)-\beta(\tau_k^-)=\alpha(\tau_k^+)-\beta(\tau_k^+)=\alpha(\tau_k)-\beta(\tau_k)$. Then, returning to (2.19) we get

$$\frac{dL(\theta)}{d\theta}=-\sum_{k\in\Psi_T}\frac{\alpha(t)-\beta(t)}{\alpha(t)-\beta(t)}=-|\Psi_T|,$$

where $|\Psi_T|$ is simply the number of observed NEPs that include a "lossy" interval over which $x(\theta,t)=\theta$. Observe that this expression for $\frac{dL(\theta)}{d\theta}$ does not depend in any functional way on the details of the arrival or service rate processes. Furthermore, it is simple to compute, and in fact amounts to a simple counting process.

2.5 Event-Driven Optimization in Multi-Agent Systems

Multi-agent systems are commonly modeled as hybrid systems with time-driven dynamics describing the motion of the agents or the evolution of physical processes in a given environment, while event-driven behavior characterizes events that may occur randomly (e.g., an agent failure) or in accordance to control policies (e.g., an agent stopping to sense the environment or to change directions). As such, a multi-agent system can be studied in the context of Figure 2.1 with parameterized controllers aiming to meet certain specifications or to optimize a given performance metric. In some cases, the solution of a multi-agent *dynamic* optimization problem is reduced to a policy that is naturally parametric. Therefore, the adaptive scheme in Figure 2.1 provides a solution that is (at least locally) optimal. In this section, we present a problem known as "persistent monitoring," which commonly arises in multi-agent systems and where the event-driven approach we have described can be used.

Persistent monitoring tasks arise when agents must monitor a dynamically changing environment that cannot be fully covered by a stationary team of available agents. Thus, all areas of a given mission space must be visited infinitely often. The main challenge in designing control strategies in this case is in balancing the presence of agents in the changing environment so that it is covered over time optimally (in some well-defined sense) while still satisfying sensing and motion constraints. Examples of persistent monitoring missions include surveillance, patrol missions with unmanned vehicles, and environmental applications where routine sampling of an area is involved. Control and motion

planning for this problem have been studied in the literature, for example, see [46–49]. We limit ourselves here to reviewing the optimal control formulation in [49] for a simple one-dimensional mission space taken to be an interval $[0, L] \subset \mathbb{R}$. Assuming N mobile agents, let their positions at time t be $s_n(t) \in [0, L]$, $n = 1, \ldots, N$, following the dynamics

$$\dot{s}_n(t) = g_n(s_n) + b_n u_n(t), \tag{2.20}$$

where $u_n(t)$ is the controllable agent speed constrained by $|u_n(t)| \leq 1$, $n = 1, \ldots, N$, that is, we assume that the agent can control its direction and speed. Without loss of generality, after some rescaling with the size of the mission space L, we further assume that the speed is constrained by $|u_n(t)| \leq 1$, $n = 1, \ldots, N$. For the sake of generality, we include the additional constraint,

$$a \leq s(t) \leq b, \ a \geq 0, \ b \leq L, \tag{2.21}$$

over all t to allow for mission spaces where the agents may not reach the endpoints of $[0, L]$, possibly due to the presence of obstacles. We associate with every point $x \in [0, L]$ a function $p_n(x, s_n)$ that measures the probability that an event at location x is detected by agent n. We also assume that $p_n(x, s_n) = 1$ if $x = s_n$, and that $p_n(x, s_n)$ is monotonically nonincreasing in the distance $|x - s_n|$ between x and s_n, thus capturing the reduced effectiveness of a sensor over its range which we consider to be finite and denoted by r_n. Therefore, we set $p_n(x, s_n) = 0$ when $|x - s_n| > r_n$.

Next, consider a partition of $[0, L]$ into M intervals whose center points are $\alpha_i = \frac{(2i-1)L}{2M}$, $i = 1, \ldots, M$. We associate a time-varying measure of uncertainty with each point α_i, which we denote by $R_i(t)$. Without loss of generality, we assume $0 \leq \alpha_1 \leq \cdots \leq \alpha_M \leq L$ and, to simplify notation, we set $p_n(x, s_n(t)) = p_{n,i}(s_n(t))$ for all $x \in [\alpha_i - \frac{L}{2M}, \alpha_i + \frac{L}{2M}]$. Therefore, the joint probability of detecting an event at location $x \in [\alpha_i - \frac{L}{2M}, \alpha_i + \frac{L}{2M}]$ by all the N agents simultaneously (assuming detection independence) is

$$P_i(\mathbf{s}(t)) = 1 - \prod_{n=1}^{N} [1 - p_{n,i}(s_n(t))], \tag{2.22}$$

where we set $\mathbf{s}(t) = [s_1(t), \ldots, s_N(t)]^{\mathrm{T}}$. We define uncertainty functions $R_i(t)$ associated with the intervals $[\alpha_i - \frac{L}{2M}, \alpha_i + \frac{L}{2M}]$, $i = 1, \ldots, M$, so that they have the following properties: (1) $R_i(t)$ increases with a prespecified rate A_i if $P_i(\mathbf{s}(t)) = 0$, (2) $R_i(t)$ decreases with a fixed rate B if $P_i(\mathbf{s}(t)) = 1$, and (3) $R_i(t) \geq 0$ for all t. It is then natural to model uncertainty so that its decrease is proportional to the probability of detection. In particular, we model the dynamics of $R_i(t)$, $i = 1, \ldots, M$, as follows:

$$\dot{R}_i(t) = \begin{cases} 0 & \text{if } R_i(t) = 0, \ A_i \leq BP_i(\mathbf{s}(t)) \\ A_i - BP_i(\mathbf{s}(t)) & \text{otherwise} \end{cases}, \tag{2.23}$$

where we assume that initial conditions $R_i(0)$, $i = 1, \ldots, M$, are given and that $B > A_i > 0$ (thus, the uncertainty strictly decreases when there is perfect sensing $P_i(\mathbf{s}(t)) = 1$.) Note that A_i represents the rate at which uncertainty increases at α_i which may be random. We will start with the assumption that the value of A_i is known and will see how the robustness property of the IPA calculus (Theorem 2.2) allows us to easily generalize the analysis to random processes $\{A_i(t)\}$ describing uncertainty levels at different points in the mission space.

The goal of the optimal persistent monitoring problem is to control the movement of the N agents through $u_n(t)$ in (2.20) so that the cumulative uncertainty over all sensing points $\{\alpha_i\}$, $i = 1, \ldots, M$, is minimized over a fixed time horizon T. Thus, setting $\mathbf{u}(t) = [u_1(t), \ldots, u_N(t)]$, we aim to solve the following optimal control problem:

$$\min_{\mathbf{u}(t)} \ J = \frac{1}{T} \int_0^T \sum_{i=1}^{M} R_i(t) dt, \tag{2.24}$$

subject to the agent dynamics (2.20), uncertainty dynamics (2.23), control constraint $|u_n(t)| \leq 1$, $t \in [0, T]$, and state constraints (2.21), $t \in [0, T]$.

Using a standard calculus of variations analysis, it is shown in [49] that the optimal trajectory of each agent n is to move at full speed, that is, $u_n(t)$, until it reaches some switching point, dwell on the switching point for some time (possibly zero), and then switch directions. Consequently, each agent's optimal trajectory is fully described by a vector of switching points $\theta_n = [\theta_{n,1}, \ldots, \theta_{n,\Gamma_n}]^{\mathrm{T}}$ and $w_n = [w_{n,1} \ldots, w_{n,\Gamma_n}]^{\mathrm{T}}$, where $\theta_{n,\xi}$ is the ξth control switching point and $w_{n,\xi}$ is the waiting time for this agent at the ξth switching point. Note that Γ_n is generally not known a priori and depends on the time horizon T. It follows that the behavior of the agents operating under optimal control is fully described by hybrid dynamics, and the problem is reduced to a *parametric* optimization one, where θ_n and w_n need to be optimized for all $n = 1, \ldots, N$. This enables the use of the IPA calculus and, in particular, the use of the three equations, (2.9), (2.7), and (2.10), which ultimately leads to an evaluation of the gradient $\nabla J(\theta, w)$ with $J(\theta, w)$ in (2.24) now viewed as a function of the parameter vectors θ, w.

In order to apply IPA to this hybrid system, we begin by identifying the events that cause discrete state transitions from one operating mode of an agent to another. Looking at the uncertainty dynamics (2.23), we define an

event at time τ_k such that $\dot{R}_i(t)$ switches from $\dot{R}_i(t) = 0$ to $\dot{R}_i(t) = A_i - BP_i(\mathbf{s}(t))$ or an event such that $\dot{R}_i(t)$ switches from $\dot{R}_i(t) = A_i - BP_i(\mathbf{s}(t))$ to $\dot{R}_i(t) = 0$. In addition, since an optimal agent trajectory experiences switches of its control $u_n(t)$ from ± 1 to 0 (the agent comes to rest before changing direction) or from 0 to ± 1, we define events associated with each such action that affects the dynamics in (2.20). Denoting by $\tau_k(\theta, w)$ the occurrence time of any of these events, it is easy to obtain from (2.24):

$$\nabla J(\theta, w) = \frac{1}{T} \sum_{i=1}^{M} \sum_{k=0}^{K} \int_{\tau_k(\theta,w)}^{\tau_{k+1}(\theta,w)} \nabla R_i(t) dt,$$

which depends entirely on $\nabla R_i(t)$. Let us define the function

$$G_{n,i}(t) = B \prod_{d \neq n} (1 - p_i(s_d(t))) \left(\frac{\partial p_i(s_n)}{\partial s_n} \right) (t - \tau_k), \tag{2.25}$$

for all $t \in [\tau_k(\theta, w), \tau_{k+1}(\theta, w))$ and observe that it depends only on the sensing model $p_i(s_n(t))$ and the uncertainty model parameter B. Applying the IPA calculus (details are provided in [49]), we can then obtain

$$\frac{\partial R_i}{\partial \theta_{n,\xi}}(t) = \frac{\partial R_i(\tau_k^+)}{\partial \theta_{n,\xi}} - \begin{cases} 0 & \text{if } R_i(t) = 0, A_i < BP_i(\mathbf{s}(t)) \\ G_{n,i}(t) \frac{\partial s_n(\tau_k^+)}{\partial \theta_{n,\xi}} & \text{otherwise} \end{cases}, \tag{2.26}$$

and

$$\frac{\partial R_i}{\partial w_{n,\xi}}(t) = \frac{\partial R_i(\tau_k^+)}{\partial w_{n,\xi}} - \begin{cases} 0 & \text{if } R_i(t) = 0, A_i < BP_i(\mathbf{s}(t)) \\ G_{n,i}(t) \frac{\partial s_n(\tau_k^+)}{\partial w_{n,\xi}} & \text{otherwise} \end{cases}, \tag{2.27}$$

for all $n = 1, \ldots, N$ and $\xi = 1, \ldots, \Gamma_n$. It remains to derive event-driven iterative expressions for $\frac{\partial R_i(\tau_k^+)}{\partial \theta_{n,\xi}}$, $\frac{\partial R_i(\tau_k^+)}{\partial w_{n,\xi}}$ and $\frac{\partial s_n(\tau_k^+)}{\partial w_{n,\xi}}$, $\frac{\partial s_n(\tau_k^+)}{\partial \theta_{n,\xi}}$ above. These are given as follows (see [49] for details):

1. If an event at time τ_k is such that $\dot{R}_i(t)$ switches from $\dot{R}_i(t) = 0$ to $\dot{R}_i(t) = A_i - BP_i(\mathbf{s}(t))$, then $\nabla s_n(\tau_k^+) = \nabla s_n(\tau_k^-)$ and $\nabla R_i(\tau_k^+) = \nabla R_i(\tau_k^-)$ for all $n = 1, \ldots, N$.
2. If an event at time τ_k is such that $\dot{R}_i(t)$ switches from $\dot{R}_i(t) = A_i - BP_i(\mathbf{s}(t))$ to $\dot{R}_i(t) = 0$ (i.e., $R_i(\tau_k)$ becomes zero), then $\nabla s_n(\tau_k^+) = \nabla s_n(\tau_k^-)$ and $\nabla R_i(\tau_k^+) = 0$.
3. If an event at time τ_k is such that $u_n(t)$ switches from ± 1 to 0, or from 0 to ± 1, we need the components of $\nabla s_n(\tau_k^+)$ in (2.26) and (2.27) which are obtained as follows. First, for $\frac{\partial s_n(\tau_k^+)}{\partial \theta_{n,\xi}}$, if an event at time τ_k is such that $u_n(t)$ switches from ± 1 to 0, then $\frac{\partial s_n}{\partial \theta_{n,\xi}}(\tau_k^+) = 1$ and

$$\frac{\partial s_n(\tau_k^+)}{\partial \theta_{n,j}} = \begin{cases} 0, & \text{if } j \neq \xi \\ 1, & \text{if } j = \xi \end{cases}, \quad j < \xi.$$

If on the other hand, $u_n(t)$ switches from 0 to ± 1, then $\frac{\partial \tau_k}{\partial \theta_{n,\xi}} = -sgn(u(\tau_k^+))$ and

$$\frac{\partial s_n(\tau_k^+)}{\partial \theta_{n,j}} = \begin{cases} \frac{\partial s_n}{\partial \theta_{n,j}}(\tau_k^-) + 2, & \text{if } u_n(\tau_k^+) = 1,\ j \text{ even}, \\ & \text{or } u_n(\tau_k^+) = -1,\ j \text{ odd} \\ \frac{\partial s_n}{\partial \theta_{n,j}}(\tau_k^-) - 2, & \text{if } u_n(\tau_k^+) = 1,\ j \text{ odd}, \\ & \text{or } u_n(\tau_k^+) = -1,\ j \text{ even} \end{cases}, \quad j < \xi,$$

Finally, for $\frac{\partial s_n(\tau_k^+)}{\partial w_{n,\xi}}$, we have

$$\frac{\partial s_n(\tau_k^+)}{\partial w_{n,j}} = \begin{cases} 0, & \text{if } u_n(\tau_k^-) = \pm 1,\ u_n(\tau_k^+) = 0 \\ \mp 1, & \text{if } u_n(\tau_k^-) = 0,\ u_n(\tau_k^+) = \pm 1 \end{cases}.$$

In summary, this provides an event-driven procedure for evaluating $\nabla J(\theta, w)$ and proceeding with a gradient-based algorithm as shown in Figure 2.1 to determine optimal agent trajectories online or at least improve on current ones.

Furthermore, let us return to the case of stochastic environmental uncertainties manifested through random processes $\{A_i(t)\}$ in (2.23). Observe that the evaluation of $\nabla R_i(t)$, hence $\nabla J(\theta, w)$, is *independent* of A_i, $i = 1, \ldots, M$; in particular, note that A_i does not appear in the function $G_{n,i}(t)$ in (2.25) or in any of the expressions for $\frac{\partial s_n(\tau_k^+)}{\partial \theta_{n,j}}$, $\frac{\partial s_n(\tau_k^+)}{\partial w_{n,j}}$. In fact, the dependence of $\nabla R_i(t)$ on A_i manifests itself through the event times τ_k, $k = 1, \ldots, K$, that do affect this evaluation, but they, unlike A_i which may be unknown, are directly observable during the gradient evaluation process. This, once again is an example of the IPA robustness property discussed in Section 4.2. Extensive numerical examples of how agent trajectories are adjusted online for the persistent monitoring problem may be found in [49].

Extending this analysis from one-dimensional to two-dimensional mission spaces no longer yields optimal trajectories which are parametric in nature, as shown in [50]. However, one can represent an agent trajectory in terms of general function families characterized by a set of parameters that may be optimized based on an objective function such as (2.24) extended to two-dimensional environments. In particular, we may view each agent's trajectory as represented by parametric equations

$$s_n^x(t) = f(\Upsilon_n, \rho_n(t)), \;\; s_n^y(t) = g(\Upsilon_n, \rho_n(t)), \qquad (2.28)$$

for all agents $n = 1, \ldots, N$. Here, $\Upsilon_n = [\Upsilon_n^1, \Upsilon_n^2, \ldots, \Upsilon_n^\Gamma]^T$ is the vector of parameters through which we control the shapes and locations of the nth agent trajectory, and Γ is this vector's dimension. The agent position over time is controlled by a function $\rho_n(t)$ dependent on the agent dynamics. We can then formulate a problem such as

$$\min_{\Upsilon_n,\, n=1,\ldots,N} \; J = \int_0^T \sum_{i=1}^M R_i(\Upsilon_1, \ldots, \Upsilon_N, t)dt,$$

which involves optimization over the controllable parameter vectors $\Upsilon_n, n = 1, \ldots, N$, characterizing each agent trajectory and placing once again the problem in the general framework of Figure 2.1.

2.6 Conclusions

Glancing into the future of systems and control theory, the main challenges one sees involve larger and ever more distributed wirelessly networked structures in application areas spanning cooperative multi-agent systems, energy allocation and management, and transportation, among many others. Barring any unexpected dramatic developments in battery technology, limited energy resources in wireless settings will have to largely dictate how control strategies are designed and implemented so as to carefully optimize this limitation. Taking this point of view, the event-driven paradigm offers an alternative to the time-driven paradigm for modeling, sampling, estimation, control, and optimization, not to supplant it but rather complement it. In hybrid systems, this approach is consistent with the event-driven nature of IPA which offers a general-purpose process for evaluating or estimating (in the case of stochastic systems) gradients of performance metrics. Such information can then be used on line so as to maintain a desirable system performance and, under appropriate conditions, lead to the solution of optimization problems in applications ranging from multi-agent systems to resource allocation in manufacturing, computer networks, and transportation systems.

Acknowledgment

The author's work was supported in part by the National Science Foundation (NSF) under grants CNS-12339021 and IIP-1430145, by the Air Force Office of Scientific Research (AFOSR) under grant FA9550-12-1-0113, by the Office of Naval Research (ONR) under grant N00014-09-1-1051, and by the Army Research Office (ARO) under grant W911NF-11-1-0227.

Bibliography

[1] C. G. Cassandras, S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed., Springer, New York 2008.

[2] Y. C. Ho, C. G. Cassandras, A new approach to the analysis of discrete event dynamic systems, *Automatica*, vol. 19, pp. 149–167, 1983.

[3] P. J. Ramadge, W. M. Wonham, The control of discrete event systems, *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, 1989.

[4] F. Baccelli, G. Cohen, G. J. Olsder, J. P. Quadrat, *Synchronization and Linearity*. Wiley, New York 1992.

[5] J. O. Moody, P. J. Antsaklis, *Supervisory Control of Discrete Event Systems Using Petri Nets*. Kluwer Academic, Dordrecht, the Netherlands 1998.

[6] M. S. Branicky, V. S. Borkar, S. K. Mitter, A unified framework for hybrid control: Model and optimal control theory, *IEEE Trans. Autom. Control*, vol. 43, no. 1, pp. 31–45, 1998.

[7] P. Antsaklis, W. Kohn, M. Lemmon, A. Nerode, and S. Sastry (Eds.), *Hybrid Systems*. Springer-Verlag, New York 1998.

[8] C. G. Cassandras, D. L. Pepyne, Y. Wardi, Optimal control of a class of hybrid systems, *IEEE Trans. Autom. Control*, vol. 46, no. 3, pp. 398–415, 2001.

[9] P. Zhang, C. G. Cassandras, An improved forward algorithm for optimal control of a class of hybrid systems, *IEEE Trans. Autom. Control*, vol. 47, no. 10, pp. 1735–1739, 2002.

[10] M. Lemmon, K. X. He, I. Markovsky, Supervisory hybrid systems, *IEEE Control Systems Magazine*, vol. 19, no. 4, pp. 42–55, 1999.

[11] H. J. Sussmann, A maximum principle for hybrid optimal control problems, in *Proceedings of 38th IEEE Conf. on Decision and Control*, December, 1999, pp. 425–430.

[12] A. Alur, T. A. Henzinger, E. D. Sontag (Eds.), *Hybrid Systems*. Springer-Verlag, New York 1996.

[13] Special issue on goals and challenges in cyber physical systems research, *IEEE Trans. Autom. Control*, vol. 59, no. 12, pp. 3117–3379, 2014.

[14] K. E. Arzen, A simple event based PID controller, in *Proceedings of 14th IFAC World Congress*, 2002, pp. 423–428.

[15] W. P. Heemels, J. H. Sandee, P. P. Bosch, Analysis of event-driven controllers for linear systems, *Int. J. Control*, vol. 81, no. 4, p. 571, 2008.

[16] J. Lunze, D. Lehmann, A state-feedback approach to event-based control, *Automatica*, vol. 46, no. 1, pp. 211–215, 2010.

[17] P. Tabuada, Event-triggered real-time scheduling of stabilizing control tasks, *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1680–1685, 2007.

[18] A. Anta, P. Tabuada, To sample or not to sample: Self-triggered control for nonlinear systems, *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2030–2042, 2010.

[19] V. S. Dolk, D. P. Borgers, W. P. M. H. Heemels, Dynamic event-triggered control: Tradeoffs between transmission intervals and performance, in *Proceedings 35th IEEE Conf. on Decision and Control*, 2014, pp. 2765–2770.

[20] K. J. Astrom, B. M. Bernhardsson, Comparison of Riemann and Lebesgue sampling for first order stochastic systems, in *Proceedings of 41st IEEE Conf. on Decision and Control*, 2002, pp. 2011–2016.

[21] T. Shima, S. Rasmussen, P. Chandler, UAV Team decision and control using efficient collaborative estimation, *ASME J. Dyn. Syst. Meas. Cont.* vol. 129, no. 5, pp. 609–619, 2007.

[22] X. Wang, M. Lemmon, Event-triggering in distributed networked control systems, *IEEE Trans. Autom. Control*, vol. 56, no. 3, pp. 586–601, 2011.

[23] M. Zhong, C. G. Cassandras, Asynchronous distributed optimization with event-driven communication, *IEEE Trans. Autom. Control*, vol. 55, no. 12, pp. 2735–2750, 2010.

[24] W. P. M. H. Heemels, M. C. F. Donkers, A. R. Teel, Periodic event-triggered control for linear systems, *IEEE Trans. Autom. Control*, vol. 58, no. 4, pp. 847–861, April 2013.

[25] S. Trimpe, R. D'Andrea, Event-based state estimation with variance-based triggering, *IEEE Trans. Autom. Control*, vol. 49, no. 12, pp. 3266–3281, 2014.

[26] E. Garcia, P. J. Antsaklis, Event-triggered output feedback stabilization of networked systems with external disturbance, in *Proceedings of 53rd IEEE Conf. on Decision and Control*, 2014, pp. 3572–3577.

[27] T. Liu, M. Cao, D. J. Hill, Distributed event-triggered control for output synchronization of dynamical networks with non-identical nodes, in *Proceedings of 53rd IEEE Conf. on Decision and Control*, 2014, pp. 3554–3559.

[28] Y. Shoukry, P. Tabuada, Event-triggered projected Luenberger observer for linear systems under sparse sensor attacks, in *Proceedings of 53rd IEEE Conf. on Decision and Control*, 2014, pp. 3548–3553.

[29] S. Trimpe, Stability analysis of distributed event-based state estimation, in *Proceedings of 53rd IEEE Conf. on Decision and Control*, 2014, pp. 2013–2018.

[30] Y. Khazaeni, C. G. Cassandras, A new event-driven cooperative receding horizon controller for multi-agent systems in uncertain environments, in *Proceedings of 53rd IEEE Conf. on Decision and Control*, 2014, pp. 2770–2775.

[31] Y. C. Ho, X. Cao, *Perturbation Analysis of Discrete Event Dynamic Systems*. Kluwer Academic, Dordrecht, the Netherlands 1991.

[32] C. G. Cassandras, Y. Wardi, C. G. Panayiotou, C. Yao, Perturbation analysis and optimization of stochastic hybrid systems, *Eur. J. Control*, vol. 16, no. 6, pp. 642–664, 2010.

[33] C. G. Cassandras, J. Lygeros (Eds.), *Stochastic Hybrid Systems*. CRC Press, Boca Raton, FL 2007.

[34] W. B. Powell, *Approximate Dynamic Programming* (2nd ed.). John Wiley and Sons, New York.

[35] C. G. Cassandras, Y. Wardi, B. Melamed, G. Sun, C. G. Panayiotou, Perturbation analysis for on-line control and optimization of stochastic fluid models, *IEEE Trans. Autom. Control*, vol. 47, no. 8, pp. 1234–1248, 2002.

[36] A. Kebarighotbi, C. G. Cassandras, A general framework for modeling and online optimization

of stochastic hybrid systems, in *Proceedings of 4th IFAC Conference on Analysis and Design of Hybrid Systems*, 2012.

[37] Y. Wardi, R. Adams, B. Melamed, A unified approach to infinitesimal perturbation analysis in stochastic flow models: The single-stage case, *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 89–103, 2010.

[38] Y. Wardi, A. Giua, C. Seatzu, IPA for continuous stochastic marked graphs, *Automatica*, vol. 49, no. 5, pp. 1204–1215, 2013.

[39] H. J. Kushner, G. G. Yin, *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, New York 1997.

[40] R. Rubinstein, *Monte Carlo Optimization, Simulation and Sensitivity of Queueing Networks*. John Wiley and Sons, New York 1986.

[41] C. Yao, C. G. Cassandras, Perturbation analysis and optimization of multiclass multiobjective stochastic flow models, *J. Discrete Event Dyn. Syst.*, vol. 21, no. 2, pp. 219–256, 2011.

[42] C. Yao, C. G. Cassandras, Perturbation analysis of stochastic hybrid systems and applications to resource contention games, *Frontiers of Electrical and Electronic Engineering in China*, vol. 6, no. 3, pp. 453–467, 2011.

[43] H. Yu, C. G. Cassandras, Perturbation analysis for production control and optimization of manufacturing systems, *Automatica*, vol. 40, pp. 945–956, 2004.

[44] C. G. Cassandras, Stochastic flow systems: Modeling and sensitivity analysis, in *Stochastic Hybrid Systems* (C. G. Cassandras, J. Lygeros, Eds.). Taylor and Francis, 2006, pp. 139–167.

[45] G. Sun, C. G. Cassandras, C. G. Panayiotou, Perturbation analysis of multiclass stochastic fluid models, *J. Discrete Event Dyn. Syst.: Theory Appl.*, vol. 14, no. 3, pp. 267–307, 2004.

[46] I. Rekleitis, V. Lee-Shue, A. New, H. Choset, Limited communication, multi-robot team based coverage, in *IEEE Int. Conf. on Robotics and Automation*, vol. 4, 2004, pp. 3462–3468.

[47] N. Nigam, I. Kroo, Persistent surveillance using multiple unmanned air vehicles, in *IEEE Aerospace Conf.*, 2008, pp. 1–14.

[48] S. L. Smith, M. Schwager, D. Rus, Persistent monitoring of changing environments using a robot with limited range sensing, in *Proc. of IEEE Conf. on Robotics and Automation*, pp. 5448–5455, 2011.

[49] C. G. Cassandras, X. Lin, X. C. Ding, An optimal control approach to the multi-agent persistent monitoring problem, *IEEE Trans. on Automatic Control*, vol. 58, no. 4, pp. 947–961, 2013.

[50] X. Lin, C. G. Cassandras, Trajectory optimization for multi-agent persistent monitoring in two-dimensional spaces, in *Proc. of 53rd IEEE Conf. Decision and Control*, 2014, pp. 3719–3724.