is given as

$$\frac{\partial r_{yu}(\tau,\boldsymbol{\theta})}{\partial \theta_i} = \mathbf{c}_1^T \Big(\frac{\partial e^{\mathbf{A}(\boldsymbol{\theta})\tau}}{\partial \theta_i} \mathbf{P}(\boldsymbol{\theta}) + e^{\mathbf{A}(\boldsymbol{\theta})\tau} \frac{\partial \mathbf{P}(\boldsymbol{\theta})}{\partial \theta_i} \Big) \mathbf{c}_2 \tag{19}$$

where θ_i is the ith element of $\boldsymbol{\theta}$. To find $\partial e^{\mathbf{A}(\boldsymbol{\theta})\tau}/\partial \theta_i$, it is noted that the matrix exponential $e^{\mathbf{A}(\boldsymbol{\theta})\tau}$ has the spectral representation

$$e^{\mathbf{A}(\boldsymbol{\theta})\tau} = \sum_{j=1}^{\eta} \boldsymbol{\phi}_j \boldsymbol{\xi}_j^H e^{\lambda_j \tau} \tag{20}$$

where $\boldsymbol{\phi}_j$ and $\boldsymbol{\xi}_j$ are the right and left eigenvectors of $\mathbf{A}(\boldsymbol{\theta})$, respectively, normalized such that $\boldsymbol{\phi}_j^H \boldsymbol{\xi}_j = 1$, and where λ_j are the eigenvalues of $\mathbf{A}(\boldsymbol{\theta})$. Then, [9]

$$\frac{\partial e^{\mathbf{A}(\boldsymbol{\theta})\tau}}{\partial \theta_i} = \sum_{j=1}^{\eta} \sum_{\ell=1}^{\eta} \boldsymbol{\phi}_j \xi_j^H \frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_i} \boldsymbol{\phi}_\ell \xi_\ell^H g_{j\ell}(\tau) \tag{21}$$

where

$$\frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_i} = \begin{cases} -\mathbf{e}_i(\eta)\mathbf{e}_1^T(\eta), & i = 1,\ldots,n \\ \mathbf{e}_{i-n}(\eta)\mathbf{e}_{n+1}^T(\eta), & i = n+1,\ldots,2n \end{cases} \tag{22}$$

where $\mathbf{e}_k(j)$ is the kth column of the identity matrix of dimension j, and where

$$g_{j\ell}(\tau) = \begin{cases} \tau e^{\lambda_j \tau}, & \lambda_j = \lambda_\ell \\ \dfrac{e^{\lambda_\ell \tau} - e^{\lambda_j \tau}}{\lambda_\ell - \lambda_j}, & \lambda_j \neq \lambda_\ell. \end{cases} \tag{23}$$

Moreover, $\partial \mathbf{P}(\boldsymbol{\theta})/\partial \theta_i$ is given as the solution to the Lyapunov equation

$$\mathbf{A}(\boldsymbol{\theta}) \frac{\partial \mathbf{P}(\boldsymbol{\theta})}{\partial \theta_i} + \frac{\partial \mathbf{P}(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{A}^T(\boldsymbol{\theta}) + \frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{P}(\boldsymbol{\theta}) + \mathbf{P}(\boldsymbol{\theta}) \frac{\partial \mathbf{A}^T(\boldsymbol{\theta})}{\partial \theta_i} = \mathbf{0}. \tag{24}$$

REFERENCES

[1] T. Söderström, H. Fan, B. Carlsson, and S. Bigi, "Least squares parameter estimation of continuous-time ARX models from discrete-time data," *IEEE Trans. Automat. Control*, vol. 42, no. 5, pp. 659–673, May 1997.

[2] E. K. Larsson, M. Mossberg, and T. Söderström, "Identification of continuous-time ARX models from irregularly sampled data," *IEEE Trans. Automat. Control*, vol. 52, no. 3, pp. 417–427, Mar. 2007.

[3] M. Mossberg, "Identification of continuous-time ARX models using sample cross-covariances," in *Proc. Amer. Control Conf.*, Portland, OR, Jun. 8–10, 2005, pp. 4766–4771.

[4] E. K. Larsson, M. Mossberg, and T. Söderström, "Estimation of continuous-time stochastic system parameters," in *Continuous-Time from Sampled Data*, H. Garnier and L. Wang, Eds. New York: Springer-Verlag, 2008.

[5] K. J. Åström, *Introduction to Stochastic Control Theory*. Mineola, NY: Dover, 2006, (Republication of the edition published by Academic Press, New York, 1970).

[6] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder–Mead simplex method in low dimensions," *SIAM J. Optim.*, vol. 9, no. 1, pp. 112–147, 1998.

[7] L. Ljung, *System Identification*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

[8] E. K. Larsson and E. G. Larsson, "The CRB for parameter estimation in irregularly sampled continuous-time ARMA systems," *IEEE Signal Process. Lett.*, vol. 11, no. 2, pp. 197–200, Feb. 2004.

[9] J. W. Brewer, "The derivative of the exponential matrix with respect to a matrix," *IEEE Trans. Automat. Control*, vol. 22, no. 4, pp. 656–657, Aug. 1977.

Event-Based Optimization of Markov Systems

Xi-Ren Cao and Junyu Zhang

***Abstract*—Recent research indicates that Markov decision processes (MDPs) and perturbation analysis (PA) based optimization can be derived easily from two fundamental performance sensitivity formulas. With this sensitivity point of view, an event-based optimization approach, including event-based sensitivity analysis and event-based policy iteration, was proposed via an example by X. R. Cao (*Discrete Event Dyn. Syst.: Theory Appl.*, vol. 15, pp. 169–197, 2005). This approach utilizes the special feature of a system and illustrates how the potentials can be aggregated using the special feature. The approach applies to many practical problems that do not fit well the standard MDP formulation. This note provides a mathematical formulation and proves the main results for this approach.**

***Index Terms*—Markov decision processes (MDPs), performance potentials, perturbation analysis (PA), policy gradients, policy iteration.**

I. INTRODUCTION

It is shown in [3] and [5] that performance optimization of Markov systems is based on two fundamental sensitivity formulas: one for performance difference and the other for performance derivative. Policy iteration in Markov decision processes (MDPs), in which the policies are updated between discrete points in the policy space, can be developed easily from the performance difference formula [5], and gradient-based optimization with perturbation analysis (PA), in which system parameters are updated by a small amount in each step, is based on the performance derivative formula [4]. Sample-path-based algorithms have been developed for estimating potentials or performance derivatives. With these potential estimates, sample-path-based policy iteration algorithms and policy gradient algorithms have been developed [2], [7]–[9]. The sensitivity point of view provides a new perspective that allows us to explore alternative approaches for performance optimization of Markov systems with some special features. With this perspective, we can develop an event-based optimization approach; the basic idea is illustrated by an example in [3]; and in this note, we provide a mathematical formulation and a formal study for this event-based optimization approach.

A system is modeled as a Markov chain; for simplicity, we only consider the discrete-time model. A physical event that occurs at a particular time instant can be characterized by the state transition at that instant, e.g., if a customer enters a network at a particular instant, then the population of the network increases by one at that instant. Thus, the event corresponding to a customer arrival corresponds to the set of state transitions with the population increasing by one. In general, an event is defined as a set of state transitions that share some common properties.

Events can be classified into three types: the observable events, the controllable events, and the natural transition events. The physical meaning of these events can be explained clearly in real systems. These three events occur simultaneously in the Markov model; there is, however, a logical order in timing among them. Associated with the

Manuscript received August 29, 2006; revised September 10, 2007. Recommended by Associate Editor Y. Paschalidis. This work was supported by the Hong Kong University Grant Committee (UGC).

X.-R. Cao is with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong (e-mail: eecao@ust.edu.hk).

J. Zhang is with the School of Mathematics and Computational Science, Sun Yat-sen University, Guangzhou 510275, China (e-mail: mcszhjy@mail.sysu.edu.cn).

Digital Object Identifier 10.1109/TAC.2008.919557

observable event (in the first phase) is some information about the system; an action is chosen based on the information observed to control the probabilities of the controllable events that follow.

In many practical problems, control actions can be taken only when an event occurs. The actions depend on the information contained in the observable events. A mapping from the observable event space to the action space is called an event-based policy. The goal is to find an event-based policy that attains the best performance in some sense. The main ideas for the solutions to this problem are motivated by the sensitivity view of optimization: first, we derive/construct the performance difference and derivative formulas based on performance potentials; then we derive policy iteration (under some conditions) and gradient-based optimization in the event-based policy space with these sensitivity formulas.

There are a number of advantages of the event-based optimization. First, the approach applies to many practical problems where actions depend on events, not states. With an event-based policy, the same action may be taken when the same event is observed, which may correspond to many different states. Such problems do not fit the standard MDP, or partially observable MDP (POMDP), formulation well. Second, performance potentials can be aggregated by exploiting the event-based system structure, and sample-path-based estimation algorithms can be developed for the aggregated potentials. This may reduce the number of potentials to be estimated in the learning process and significantly save computation. Despite the fact that the number of states usually grows exponentially with respect to the system size, the number of aggregated potentials depends on the number of controllable events, which may scale to the system size. Third, the performance sensitivity formulas can be expressed in terms of structural parameters rather than transition probabilities of the underlying Markov chain. This provides structural insights and overcomes the difficulties associated with determining the transition probability matrix in a large state space. Finally, this approach may be applied to a number of subjects such as multilevel (hierarchical) control, state and time aggregations, options [1], singular perturbation, and POMDPs, etc., by formulating different events to capture the different features of these problems.

In Section III, we introduce the concepts of event and event space. In Section IV, we classify three types of events. In Sections V-B and Section V-D, we derive two fundamental sensitivity formulas. They have a similar structure as those with the standard MDP formulation, except: 1) steady-state probabilities of observable events (instead of states) are used, 2) actions depend on observable events (instead of states), and 3) potentials are generally aggregated. With these two formulas, the event-based optimization (gradient-based in general and policy iteration in some special cases) is developed in Section V-E. The introductory background material is given in Section II. Conclusions are drawn in Section VI.

II. Background and Motivation

We now review the results in MDPs with a sensitivity point of view; they motivate the study of this note. Consider a discrete-time MDP with a finite state space $\mathcal{S} = \{1, 2, \ldots, S\}$. Let $\mathcal{A}$ be the finite action space consisting of all available actions and $\mathcal{A}_i \subseteq \mathcal{A}$ be the set of all actions that are available in state i. If the system is in state i and action $\alpha \in \mathcal{A}_i$ is taken, the transition probability is $p_\alpha(i, j)$, and a finite reward $f(i, \alpha)$ is received.

Denote the set of all stationary deterministic Markovian policies as D and we use d to denote such a policy. If policy d is adopted, the state transition probability matrix is denoted as $P_d = [p_{d(i)}(i, j)]_{i,j=1}^{S}$. Let $\pi_d = (\pi_d(1), \ldots, \pi_d(S))$ be the (row) vector representing its steady-state probabilities. We have $\pi_d = \pi_d P_d$, $P_d e = e$, and $\pi_d e = 1$, with $e = (1, \ldots, 1)^T$ being an S-dimensional column vector whose components are all equal to 1, where the superscript "T" denotes transpose. The reward function becomes $f(i, d(i))$, $i \in \mathcal{S}$. Let $f_d = (f(1, d(1)), f(2, d(2)), \ldots, f(S, d(S)))^T$ be the (column) reward (or performance) vector.

It is easy to see that, under a (stationary) policy $d \in D$, a discrete-time Markov decision process is a Markov chain. In this note, we assume that the Markov chain under any policy $d \in D$ is ergodic. Let $\boldsymbol{X}^d = \{X_l^d, l = 0, 1, \ldots\}$ denote the Markov chain under policy $d \in D$, with X_l^d denoting its state at time l. The long-run *average performance* is defined as

$$\eta_d = \sum_{i=1}^{S} \pi_d(i) f(i, d(i)) = \pi_d f_d$$

$$= \lim_{L \to \infty} \frac{1}{L} \sum_{l=0}^{L-1} f(X_l^d, d(X_l^d)), \qquad \text{with probability 1.}$$

We start with the *Poisson equation* [4]

$$(I - P_d) g_d + e \eta_d = f_d. \tag{1}$$

Its solution $g_d = (g_d(1), \ldots, g_d(S))^T$ is called a *performance potential* vector, and $g_d(i)$ is the potential of state i under policy d. The solution to (1) is only up to an additive constant, i.e., if g_d is a solution to (1), then so is $g_d + ce$, where c is any constant.

Let policy $h \in D$ be another policy on the same state space with the transition probability matrix P_h, and π_h, f_h, and η_h be the steady-state probability, the reward function, and the long-run average performance for the system under policy h, respectively. Then, $\eta_h = \pi_h f_h$. Premultiplying both sides of (1) by π_h, and using $\pi_h = \pi_h P_h$ and $\pi_h e = 1$, we can verify

$$\eta_h - \eta_d = \pi_h [(P_h - P_d) g_d + f_h - f_d]. \tag{2}$$

Now, suppose that P_d changes to $P_{d,h}(\delta) := P_d + \delta(P_h - P_d) = \delta P_h + (1 - \delta) P_d$ and f_d changes to $f_{d,h}(\delta) := f_d + \delta(f_h - f_d) = \delta f_h + (1 - \delta) f_d$, with $\delta \in [0, 1]$. Then, the average performance changes to $\eta_{d,h}(\delta)$. The derivative of $\eta_{d,h}(\delta)$ in the direction of $(P_h - P_d)$ is denoted as $d\eta_{d,h}(\delta)/d\delta|_{\delta=0}$. Taking $P_{d,h}(\delta)$ as P_h in (2) and letting $\delta \to 0$, we get [4]

$$\left. \frac{d\eta_{d,h}(\delta)}{d\delta} \right|_{\delta=0} = \pi_d [(P_h - P_d) g_d + f_h - f_d]. \tag{3}$$

Policy iteration algorithms can be developed from (2) and gradient-based approaches are based on (3).

III. Events Associated With Markov Systems

In many problems, the special features related to the changes in system parameters and structures can be characterized by "events." In a real-world system, the system behavior is modeled as a Markov chain, and an event is defined as a set of state transitions that satisfy some common properties. We first formally define the events.

Definition 1: A single event, denoted as $\langle i, j \rangle$, is a state transition from i to j, $i, j \in \mathcal{S}$. The space of all the single events is denoted as $\mathcal{E} = \{\emptyset, \langle i, j \rangle : i, j \in \mathcal{S}\}$, with $\emptyset$ being a null event. A set of single events is called an *event*.

By convention, we say that a Markov chain $\boldsymbol{X} = \{X_l, l \geq 0\}$ makes a transition at time l from X_{l-1} to X_l, $l \geq 1$. Thus, a single event $\langle i, j \rangle$ occurs at time l if $X_{l-1} = i$ and $X_l = j$. The null event $\emptyset$ is defined purely for logical purpose and is different from any real event. An event

a is a subset of $\mathcal{E}$: $a \subseteq \mathcal{E}$. Thus, all the set operations apply to events. The single-event space is $\mathcal{S} \times \mathcal{S}$. There are $2^{S \times S}$ possible events.

Definition 2: The input set of event a is $I(a) := \{\text{all } i \in \mathcal{S} : \langle i, j\rangle \in a \text{ for some } j\}$. The output set of event a is $O(a) = \{\text{all } j \in \mathcal{S} : \langle i, j\rangle \in a \text{ for some } i\}$. The input set of state j in event a is $I_j(a) = \{\text{all } i \in \mathcal{S} : \langle i, j\rangle \in a\}$. The output set of state i in event a is $O_i(a) = \{\text{all } j \in \mathcal{S} : \langle i, j\rangle \in a\}$.

IV. Classification of Three Types of Events

Consider an ergodic Markov chain [6], which can be viewed as a model of a real-world system under a given (stationary) policy, either state dependent or event dependent, as described in Section V.

We first study the logical relation among different events. In many problems, actions are taken only after some events occur. These events are observable and contain some information about the system. They are called the observable events. Based on the information contained in the observable events, we may take actions that control the probabilities of the subevents that the state transitions belong to. These subevents are called the controllable events. Finally, the nature completes the transition. These are called the natural transition events. The three types of events, the observable, controllable, and natural transition events, occur at the same time in the Markov model, and they together determine a state transition in the Markov model, but they have a logical timing order. See [3] for an example.

We provide a general formulation of this structure in the event space $\mathcal{E}$. In the approach, we mainly deal with events; the system state is only a hidden concept that helps the analysis.

The first type of event is the *observable event*. An observable event has two features: 1) we can tell whether the event occurs at any time instant from the system behavior and 2) the event contains some information about the system, which can be used to determine the control actions. Because the information carried in different observable events is different, the event space $\mathcal{E}$ can be decomposed into exclusive observable events:

$$\mathcal{E} = \cup_{k=1}^{k_o} e_o(k), \qquad e_o(k) \cap e_o(k') = \emptyset$$
$$k \neq k', \qquad k, k' \in \{1, 2, \ldots, k_o\}$$

where $e_o(k)$, $k = 1, 2, \ldots, k_o$, are the observable events and k_o is the number of observable events. Denote $\mathcal{E}_o = \{e_o(1), e_o(2), \ldots, e_o(k_o)\}$ as the set of all observable events.

The second type of event is the *controllable event*. A controllable event is an event in which we can control the probability of its occurrence by taking actions based on the information obtained from an observable event that has just occurred. More precisely, we have

$$\mathcal{E} = \cup_{k=1}^{k_c} e_c(k), \qquad e_c(k) \cap e_c(k') = \emptyset$$
$$k \neq k', \qquad k, k' \in \{1, 2, \ldots, k_c\}$$

where $e_c(k)$, $k = 1, 2, \ldots, k_c$, are the controllable events and k_c is the number of the controllable events. Suppose $e_o(k_1)$ is the event we observed (i.e., the observable event); then we have $e_o(k_1) = \cup_{k_2=1}^{k_c} \{e_o(k_1) \cap e_c(k_2)\}$. With this form, we can take actions to assign probabilities to those controllable events $e_c(k_2)$ for which $e_o(k_1) \cap e_c(k_2) \neq \emptyset$, $k_2 = 1, 2, \ldots, k_c$.

In particular, if, for an observable event $e_o(k_1)$, there is only one controllable event $e_c(k_2)$ such that $e_o(k_1) \cap e_c(k_2)$ is nonnull, then at $e_o(k_1)$, there is only one choice of controllable events, $e_c(k_2)$. That is, at such an observable event $e_o(k_1)$, we can take only one action. In most cases, this unique action corresponds to "do nothing," and therefore, at such an observable event, the system is customarily said to be not controllable.

The third type of event is the *natural transition event*. A natural transition event is an event whose corresponding transitions are governed by nature; thus, the probability of the occurrence of a natural transition event cannot be controlled. Generally, we have

$$\mathcal{E} = \cup_{k=1}^{k_t} e_t(k), \qquad e_t(k) \cap e_t(k') = \emptyset$$
$$k \neq k', \qquad k, k' \in \{1, 2, \ldots, k_t\}$$

where $e_t(k)$, $k = 1, \ldots, k_t$, are the natural transition events and k_t is the number of such events.

As explained before, there is a logical timing order (causality) among the different types of events: at any instant, an observable event occurs first, and when it occurs, the exact state transition is not determined. One needs to take an action that determines the probabilities of the controllable events that is followed by a natural transition event. These three types of events occur in a logical sequence simultaneously; together they determine the exact transition from a state. Sometimes the nature may have only one choice, i.e., a controllable event will be followed by a unique natural transition event. In this special case, the observable and controllable events together may determine the exact state transition.

We assume that the classification of three types of events described in this section does not depend on any policy. In other words, the classification of events is determined only by the system. Many real systems possess such a property. This logical and structural property represented by events, however, is lost in the standard MDP formulation.

Because the decompositions are mutually exclusive, for any single event (a state transition) $\langle i, j\rangle \in \mathcal{E}$, there is a unique set of integers k_1, k_2, and k_3 such that

$$\langle i, j\rangle \in e_o(k_1) \cap e_c(k_2) \cap e_t(k_3) =: e(k_1, k_2, k_3) \tag{4}$$

with $k_1 \in \{1, \ldots, k_o\}, k_2 \in \{1, \ldots, k_c\}, k_3 \in \{1, \ldots, k_t\}$.

The three events $e_o(k_1)$, $e_c(k_2)$, and $e_t(k_3)$ in (4) may not specify the single event, i.e., $e_o(k_1) \cap e_c(k_2) \cap e_t(k_3)$ may not be a singleton. However, we hope that starting from any state i, if a single event $\langle i, j\rangle \in e_o(k_1) \cap e_c(k_2) \cap e_t(k_3)$, then j is uniquely determined. Therefore, we give the following definition.

Definition 3: An event a is said to be deterministic if, for every $i \in I(a)$, the output set $O_i(a)$ contains only one state.

Therefore, if a is deterministic and $i \in I(a)$, $\langle i, j\rangle \in a$, then j is determined uniquely. In other words, in a deterministic event a, a state cannot move to more than one state. We write $j = O_i(a)$ for convenience. Before the event-based optimization, we need the following assumption.

Assumption 1: Every nonnull $e_o(k_1) \cap e_c(k_2) \cap e_t(k_3)$ is deterministic, $k_1 = 1, 2, \ldots, k_o$, $k_2 = 1, 2, \ldots, k_c$, and $k_3 = 1, 2, \ldots, k_t$.

This assumption does not impose any restriction to the system, because we can always make the natural transition event decomposition "fine" enough to make sure that the final transition is uniquely determined (assuming we know the natural transition probabilities). That is, $e(k_1, k_2, k_3)$ in (4) is deterministic and we can denote

$$j = O_i[e_o(k_1) \cap e_c(k_2) \cap e_t(k_3)].$$

V. Event-Based Optimization

In this section, we give a mathematical model of the event-based optimization and describe the system evolution with this model. For the event-based optimization discussed in this note, we only consider the stationary policies that depend on the current observable events. Such a policy is a mapping from the set of observable events $\mathcal{E}_o$ to the action set $\mathcal{A} = \cup_{k_1=1}^{k_o} \mathcal{A}_{k_1}$, denoted as $d : \mathcal{E}_o \to \mathcal{A}$, which specifies the action $d[e_o(k_1)] \in \mathcal{A}_{k_1}$ taken when the observable event $e_o(k_1)$

is observed, where $\mathcal{A}_{k_1}$ is the set of actions that are applicable when $e_o(k_1)$ is observed. Denote the set of all the stationary policies that depend only on the current observable events as D_e.

A. Problem Formulation

The mechanism of the event-based optimization is as follows. The system is in state i. However, i is not observed and instead we observe an observable event $e_o(k_1) \subseteq \mathcal{E}$, with a probability distribution $\mu(e_o(k_1)|i)$, $k_1 = 1, 2, \ldots, k_o$. In addition to some knowledge about the current state, the observable event also contains some information about the next state after the transition; it, however, does not completely specify the transition. Based on the information contained in the observable event $e_o(k_1)$, we take an action $\alpha \in \mathcal{A}_{k_1}$. Once this action is taken, a controllable event $e_c(k_2)$ follows with probability $p_\alpha[e_c(k_2)|e_o(k_1)]$, $k_2 = 1, 2, \ldots, k_c$. After the controllable event $e_c(k_2)$ occurs, the nature chooses a natural transition event $e_t(k_3)$, $k_3 = 1, 2, \ldots, k_t$, which, together with $e_o(k_1)$ and $e_c(k_2)$, finally determines the state transition at this time instant. The reward function $f(i, \alpha)$, where $\alpha = d[e_o(k_1)]$, depends on both i and α, $i \in \mathcal{S}$, $\alpha \in \mathcal{A}_{k_1}$.

More precisely, let us denote the transition at some time as $\langle i, j\rangle$. Because we observe the event $e_o(k_1)$, $k_1 = 1, 2, \ldots, k_o$, we have $\langle i, j\rangle \in e_o(k_1)$, but both i and j may not be known. If action $\alpha \in \mathcal{A}_{k_1}$ is taken, then the conditional probability of $\langle i, j\rangle \in e_c(k_2)$, $k_2 = 1, 2, \ldots, k_c$, given that $\langle i, j\rangle \in e_o(k_1)$ is controlled by α and can be denoted as

$$p_\alpha[\langle i, j\rangle \in e_c(k_2)|\langle i, j\rangle \in e_o(k_1)] \quad k_1 = 1, 2, \ldots, k_o, \qquad k_2 = 1, 2, \ldots, k_c. \tag{5}$$

By convention, if $e_c(k_2) \cap e_o(k_1) = \emptyset$, we have $p_\alpha[\langle i, j\rangle \in e_c(k_2)|\langle i, j\rangle \in e_o(k_1)] = 0$ for all $\alpha \in \mathcal{A}_{k_1}$. We make the following assumption.

Assumption 2: The conditional probability in (5) depends only on $e_o(k_1)$ and $e_c(k_2)$, i.e., it is the same for all $i \in I[e_o(k_1)]$.

Assumption 2 is a restriction on the effect of control actions, and not on the system structure. It is reasonable because we may not be able to observe i. Under Assumption 2, we may denote (5) as

$$p_\alpha[e_c(k_2)|e_o(k_1)] := p_\alpha[\langle i, j\rangle \in e_c(k_2)|\langle i, j\rangle \in e_o(k_1)].$$

The natural transition probability given a pair of $e_o(k_1)$ and $e_c(k_2)$ is denoted as

$$p[e_t(k_3)|e_c(k_2), e_o(k_1)] := p[\langle i, j\rangle \in e_t(k_3)|\langle i, j\rangle \in e_c(k_2) \cap e_o(k_1)] \quad k_1 = 1, 2, \ldots, k_o, \qquad k_2 = 1, 2, \ldots, k_c, \qquad k_3 = 1, 2, \ldots, k_t.$$

They are determined by the nature (do not depend on actions). As shown later, for our analysis, we do not require this probability to be independent of i. The three events $e_o(k_1)$, $e_c(k_2)$, and $e_t(k_3)$ uniquely determine an output state $j = O_i[e_o(k_1) \cap e_c(k_2) \cap e_t(k_3)]$ according to Assumption 1.

From Section IV and Assumption 1, for any transition $\langle i, j\rangle$, $i, j \in \mathcal{S}$, there exists a unique set of integers, k_1, k_2, and k_3, $k_1 \in \{1, \ldots, k_o\}$, $k_2 \in \{1, \ldots, k_c\}$, and $k_3 \in \{1, \ldots, k_t\}$, such that

$$\langle i, j\rangle \in e_o(k_1) \cap e_c(k_2) \cap e_t(k_3) \tag{6}$$

and $j = O_i[e_o(k_1) \cap e_c(k_2) \cap e_t(k_3)]$. From (6) and the mathematical model of the event-based optimization, the state transition probabilities from state i under event-based policy d is

$$p_d(i, j) = \mu(e_o(k_1)|i)p_{d[e_o(k_1)]}[e_c(k_2)|e_o(k_1)]\, p[e_t(k_3)|e_c(k_2), e_o(k_1)] \tag{7}$$

where $j = O_i[e_o(k_1) \cap e_c(k_2) \cap e_t(k_3)]$. We denote the state transition probability matrix under event-based policy d as $P_d = [p_d(i, j)]_{i,j\in\mathcal{S}}$. As assumed, the probability distribution $\mu(e_o(k_1)|i)$ is independent of policy d. From (7), the process $\{X_l, l = 0, 1, \ldots\}$ under event-based policy d is indeed a time-homogenous Markov chain.

From (7), in event-based optimization, we decompose the state transition probability into the controllable part $p_{d[e_o(k_1)]}[e_c(k_2)|e_o(k_1)])$ and two uncontrollable parts $\mu(e_o(k_1)|i)$ and $p[e_t(k_3)|e_c(k_2), e_o(k_1)]$; each of them has a clear physical meaning. The decomposition utilizes the special features of a problem. With this formulation, actions depend on observable events, and therefore, the same action can be taken for different states. Furthermore, only the controllable part in the transition probability contains important parameters, and as we shall see later, the other parts may be "aggregated."

Generally, the long-run average performance of event-based policy d is defined as

$$\eta_d(i, \alpha) := \limsup_{L\to\infty} \frac{1}{L} \sum_{l=0}^{L-1} E\{f(X_l, A_l)|X_0 = i, A_0 = \alpha\} \tag{8}$$

where A_l is the action taken at time l according to policy d, $l = 0, 1, \ldots$. The goal is to find an event-based policy $d \in D_e$ that maximizes this performance or other performance criteria (e.g., discounted performance).

In this note, we study the case in which the Markov chain with the state transition probability matrix P_d is ergodic. In this case, for any policy $d \in D_e$, there always exists a steady-state probability, denoted as $\pi_d = (\pi_d(1), \pi_d(2), \ldots, \pi_d(S))$.

If we set $Y_l = (X_l, X_{l+1})$, $l = 0, 1, \ldots$, then the augmented chain $\{Y_l, l = 0, 1 \ldots\}$ is also a Markov chain. In addition, we define a reward function $h(Y_l) = f(X_l, d[e_o(k_1)])$, with $\langle X_l, X_{l+1}\rangle \in e_o(k_1)$. Then, the long-run average performance of $\{Y_l, l = 0, 1, \ldots\}$ with reward function $h(Y_l)$ is the same as (8). By the ergodic property, the sample-path average converges with probability 1 and is independent of the initial condition; therefore, (8) becomes

$$\eta_d = E_d[f(X_l, A_l)] = E_d\{E_d[f(X_l, A_l)|X_l]\} = E_d[\bar{f}_d(X_l)]$$

where E_d denotes the steady-state mean and

$$\bar{f}_d(i) = E_d[f(i, \alpha)|X_l = i] = \sum_{k_1=1}^{k_o} \mu(e_o(k_1)|i) f(i, d[e_o(k_1)]) \quad i \in \mathcal{S}. \tag{9}$$

Let $\bar{f}_d := (\bar{f}_d(1), \ldots, \bar{f}_d(S))^T$ be the vector of the equivalent reward. Then, the average performance $\eta_d = \pi_d \bar{f}_d$. The performance potential $\bar{g}_d$ is given by the Poisson equation (1), in which the performance vector f_d is replaced by $\bar{f}_d$. That is,

$$(I - P_d)\bar{g}_d + e\eta_d = \bar{f}_d.$$

B. Performance Difference Formulas for Event-Based Policies

Let $\pi_d(e_o(k_1))$ be the steady-state probability of observable event $e_o(k_1)$ under policy $d \in D_e$. We have

$$\pi_d(e_o(k_1)) = \sum_{i\in I[e_o(k_1)]} \pi_d(i)\mu(e_o(k_1)|i), \qquad k_1 \in \{1, \ldots, k_o\}.$$

In addition, we can write

$$\pi_d(i) = \sum_{k_1=1}^{k_o} \pi_d(e_o(k_1))\pi_d(i|e_o(k_1)), \qquad i \in \mathcal{S}$$

where the steady-state conditional probability

$$\pi_d(i|e_o(k_1)) = \frac{\pi_d(i)\mu(e_o(k_1)|i)}{\pi_d(e_o(k_1))} = \frac{\pi_d(i)\mu(e_o(k_1)|i)}{\sum_{j\in I[e_o(k_1)]}\pi_d(j)\mu(e_o(k_1)|j)} \tag{10}$$

and $\sum_{i\in I[e_o(k_1)]}\pi_d(i|e_o(k_1)) = 1$.

By the MDP performance difference formula, for two event-based policies d and h, we have

$$\eta_h - \eta_d = \pi_h[(P_h - P_d)\bar{g}_d + \bar{f}_h - \bar{f}_d]. \tag{11}$$

Then by (10), (9), and (11), we have

$$\begin{aligned}
\eta_h - \eta_d &= \pi_h[(P_h - P_d)\bar{g}_d + \bar{f}_h - \bar{f}_d] \\
&= \sum_{i=1}^{S}\pi_h(i)\sum_{j=1}^{S}[p_h(i,j) - p_d(i,j)]\bar{g}_d(j) \\
&\quad + \sum_{i=1}^{S}\pi_h(i)[\bar{f}_h(i) - \bar{f}_d(i)] \\
&= \sum_{k_1=1}^{k_o}\sum_{k_2=1}^{k_c}\sum_{k_3=1}^{k_t}\sum_{i=1}^{S}\pi_h(i)\mu(e_o(k_1)|i) \\
&\quad \left\{p_{h[e_o(k_1)]}[e_c(k_2)|e_o(k_1)] - p_{d[e_o(k_1)]}[e_c(k_2)|e_o(k_1)]\right\} \\
&\quad p[e_t(k_3)|e_c(k_2), e_o(k_1)]\bar{g}_d(O_i[e_o(k_1)\cap e_c(k_2)\cap e_t(k_3)]) \\
&\quad + \sum_{i=1}^{S}\pi_h(i)\sum_{k_1=1}^{k_o}\mu(e_o(k_1)|i)\{f(i,h[e_o(k_1)]) \\
&\quad - f(i,d[e_o(k_1)])\} \\
&= \sum_{k_1=1}^{k_o}\pi_h(e_o(k_1))\left\{\sum_{k_2=1}^{k_c}\left\{p_{h[e_o(k_1)]}[e_c(k_2)|e_o(k_1)]\right.\right. \\
&\quad \left.\left. - p_{d[e_o(k_1)]}[e_c(k_2)|e_o(k_1)]\right\}\bar{g}_{d,h}(k_1,k_2) + B_{d,h}(k_1)\right\}
\end{aligned} \tag{12}$$

where

$$B_{d,h}(k_1) := \sum_{i\in I[e_o(k_1)]}\pi_h(i|e_o(k_1)) \{f(i,h[e_o(k_1)]) - f(i,d[e_o(k_1)])\} \tag{13}$$

$$\bar{g}_{d,h}(k_1,k_2) = \sum_{i\in I[e_o(k_1)]}\sum_{k_3=1}^{k_t} \{\pi_h(i|e_o(k_1))p[e_t(k_3)|e_c(k_2), e_o(k_1)]\bar{g}_d(j)\} \tag{14}$$

with $j = O_i[e_o(k_1)\cap e_c(k_2)\cap e_t(k_3)]$, is the aggregated potential depending on both policies d and h. Equation (12) is the average performance difference formula for the event-based policies.

Furthermore, in (14), we may allow the natural transition probabilities to depend on state i, which is denoted as $p_i[e_t(k_3)|e_c(k_2), e_o(k_1)]$. In this case, (12) remains the same and (14) becomes

$$\bar{g}_{d,h}(k_1,k_2) = \sum_{i\in I[e_o(k_1)]}\sum_{k_3=1}^{k_t} \{\pi_h(i|e_o(k_1))p_i[e_t(k_3)|e_c(k_2), e_o(k_1)]\bar{g}_d(j)\} \tag{15}$$

with $j = O_i[e_o(k_1)\cap e_c(k_2)\cap e_t(k_3)]$.

C. Aggregated Potentials

The aggregated potential (14) [or (15)] and (13) depend on both policies d and h. The difference formulas with aggregated potentials in such a form are generally not useful in performance optimization because one cannot explore every pair of policies d and h. To develop policy-iteration-based algorithms, we need to find the conditions under which the aggregated potential and (13) depend only on policy d.

For some systems, the following equation holds for the conditional probability of i:

$$\pi_h(i|e_o(k_1)) \equiv \pi(i|e_o(k_1)) \quad \forall i\in I[e_o(k_1)],\quad \forall k_1\in\{1,2,\ldots,k_o\},\quad \text{and}\quad \forall h\in D_e. \tag{16}$$

In such cases, the aggregated potential (15) becomes

$$\bar{g}_{d,h}(k_1,k_2) = \sum_{i\in I[e_o(k_1)]}\sum_{k_3=1}^{k_t} \{\pi(i|e_o(k_1))p_i[e_t(k_3)|e_c(k_2), e_o(k_1)]\bar{g}_d(j)\} =: \bar{g}_d(k_1,k_2) \tag{17}$$

which depends only on policy d. It is the expected potential under policy d given that events $e_c(k_2)$ and $e_o(k_1)$ occur. Also, (13) becomes

$$B_{d,h}(k_1) = \sum_{i\in I[e_o(k_1)]}\pi(i|e_o(k_1)) \{f(i,h[e_o(k_1)]) - f(i,d[e_o(k_1)])\} =: B_d(k_1).$$

The aggregated potential $\bar{g}_d(k_1,k_2)$ and $B_d(k_1)$ depend only on policy d. Consequently, they can be directly estimated from a sample path of the system under policy d without explicitly knowing $\pi(i|e_o(k_1))$, $p_i[e_t(k_3)|e_c(k_2), e_o(k_1)]$, and $\bar{g}_d(j)$; see [3] for an example. Furthermore, under condition (16), the average performance difference formula (12) becomes

$$\begin{aligned}
&\eta_h - \eta_d \\
&= \sum_{k_1=1}^{k_o}\pi_h(e_o(k_1))\left\{\sum_{k_2=1}^{k_c}\left\{p_{h[e_o(k_1)]}[e_c(k_2)|e_o(k_1)]\right.\right. \\
&\quad \left.\left. - p_{d[e_o(k_1)]}[e_c(k_2)|e_o(k_1)]\right\}\bar{g}_d(k_1,k_2) + B_d(k_1)\right\}.
\end{aligned}$$

The second condition is

$$\begin{aligned}
&\text{both } p_i[e_t(k_3)|e_c(k_2), e_o(k_1)] \text{ and} \\
&j = O_i[e_o(k_1)\cap e_c(k_2)\cap e_t(k_3)] \text{ do not depend on } i; \\
&\text{and } f(i,\alpha) = f(i) \qquad \text{for all } \alpha\in\mathcal{A}.
\end{aligned}$$

In this case, from (15) and $\sum_{i\in I[e_o(k_1)]}\pi_h(i|e_o(k_1))=1$, the performance difference formula (12) becomes

$$\eta_h-\eta_d=\sum_{k_1=1}^{k_o}\pi_h(e_o(k_1))\sum_{k_2=1}^{k_c}\{p_h[e_c(k_2)|e_o(k_1)]-p_d[e_c(k_2)|e_o(k_1)]\}\bar{g}_d(k_1,k_2)$$

where

$$\bar{g}_d(k_1,k_2)=\sum_{k_3=1}^{k_t}\{p[e_t(k_3)|e_c(k_2),e_o(k_1)]\bar{g}_d(j)\} \tag{18}$$

which depends only on policy d and can be estimated on a sample path under d.

Under the two conditions mentioned before, we can develop event-based policy iteration algorithms to get the optimal event-based policies.

D. Performance Derivative Formulas for Event-Based Policies

To study the average performance gradients, we assume that the conditional transition probabilities (i.e., the policy) depend on a continuous parameter $\theta\in\Theta\subseteq\mathcal{R}$ and are denoted as $p_\theta[e_c(k_2)|e_o(k_1)]$. Taking $p_{\theta_2}[e_c(k_2)|e_o(k_1)]$ as $p_h[e_c(k_2)|e_o(k_1)]$ and $p_{\theta_1}[e_c(k_2)|e_o(k_1)]$ as $p_d[e_c(k_2)|e_o(k_1)]$ in (12), where $\theta_1,\theta_2\in\Theta$, and letting $\theta_2\to\theta_1$ and assuming the derivatives exist, we obtain

$$\frac{d\eta(\theta)}{d\theta}\Big|_{\theta=\theta_1}=\sum_{k_1=1}^{k_o}\left\{\pi_{\theta_1}(e_o(k_1))\left[\sum_{k_2=1}^{k_c}\frac{d}{d\theta}p_\theta[e_c(k_2)|e_o(k_1]\Big|_{\theta=\theta_1}\bar{g}_{\theta_1}(k_1,k_2)\right]+\sum_{i\in I[e_o(k_1)]}\pi_{\theta_1}(i|e_o(k_1))\frac{d}{d\theta}f(i,\theta)\Big|_{\theta=\theta_1}\right\} \tag{19}$$

and

$$\bar{g}_{\theta_1}(k_1,k_2)=\sum_{i\in I[e_o(k_1)]}\sum_{k_3=1}^{k_t}\{\pi_{\theta_1}(i|e_o(k_1))p_i[e_t(k_3)|e_c(k_2),e_o(k_1)]\bar{g}_{\theta_1}(j)\}. \tag{20}$$

All the terms in (19) and (20) depend only on θ_1. With (19), gradient-based optimization algorithms can be developed.

E. Event-Based Optimization

Both the performance difference and derivative formulas (12) and (19) have a similar form as the ones for the standard MDPs (2) and (3). Therefore, gradient-based and policy-iteration optimization approaches may be developed based on these two formulas. In this section, we provide a brief discussion.

1) Gradient-Based Optimization: The aggregated potential $\bar{g}_{\theta_1}(k_1,k_2)$ in (20) can be estimated on a sample path of the system under parameter (policy) θ_1. Efficient sample-path-based algorithms to estimate $\bar{g}_{\theta_1}(k_1,k_2)$'s are to be developed (see [3] for an algorithm for similar items). There are $k_o\times k_c$ aggregated potentials $\bar{g}_{\theta_1}(k_1,k_2)$ in (20) (compared with S potentials in the standard MDPs). The number of aggregated potentials is usually smaller than the number of states. Once these aggregated potentials are estimated, the performance gradients with respect to any parameter can be obtained by (19).

Developing efficient algorithms for performance derivatives with event-based policies is a future research topic.

2) Policy Iteration: The aggregated potential (14) [or (15)] contains items for both policies: $\pi_h(i|e_o(k_1))$ for policy h and $\bar{g}_d(j)$ for policy d. Such a quantity cannot be used in policy iteration. When the aggregated potentials depend only on policy d, as shown in (17) and (18), they can be either calculated analytically by studying the system under policy d, or estimated from a sample path of the system under policy d. Event-based policy iteration algorithms can be developed from the performance difference formula (12) following the same idea as the standard MDPs. In essence, at each iteration, one chooses the action α^* among $\mathcal{A}_{k_1}$ [the available action set for the observable event $e_o(k_1)$] that leads to the largest value of the average aggregated potential given the observable event $e_o(k_1)$, i.e., one chooses

$$\alpha^*=\arg\max_{\alpha\in\mathcal{A}_{k_1}}\left\{\sum_{k_2=1}^{k_c}\{p_\alpha[e_c(k_2)|e_o(k_1)]-p_{d[e_o(k_1)]}[e_c(k_2)|e_o(k_1)]\}\bar{g}_d(k_1,k_2)\right\}$$

where $p_{d[e_o(k_1)]}[e_c(k_2)|e_o(k_1)]$, $k_2=1,\ldots,k_c$, are the transition probabilities under the current event-based policy d. By the same principle as policy iteration in the standard MDPs, we know that the policy iteration procedure eventually leads to the optimal policy among the event-based policy space D_e.

VI. Conclusion

In this note, we presented a mathematical formulation for the event-based optimization approach proposed in [3] and provided rigorous proofs for its main results. This approach utilizes the special feature of a system and illustrates how the potentials can be aggregated based on the special feature. The aggregated potentials can be used to build performance sensitivity formulas that lead to gradient-based optimization, and with some conditions, event-based policy iteration. The approach applies to many practical problems that do not fit well the standard MDP formulation. Many subjects, including the multilevel control problem, time aggregation and options, state aggregation, singular perturbation, queueing applications, and POMDPs, fit the event-based framework if the corresponding events are defined properly. This opens up many research problems.

The limitation of the approach is that the aggregated potentials in the performance difference formula may depend on two policies under comparison. This prevents the aggregated potentials from being used in policy iteration. It is shown in Section V-E that, under some special conditions, the aggregated potentials depend only on one policy and can be estimated on a sample path. In such cases, event-based policy iteration algorithms can be developed. In this regard, the approach clearly indicates whether the potentials can be aggregated in performance difference formulas, and if not, why. It is clear, however, in performance derivative analysis, that potentials can always be aggregated with the event-based structure. Therefore, performance-gradient-based optimization (based on events) is more applicable than the event-based policy iteration.

References

[1] A. Barto and S. Mahadevan, "Recent advances in hierarchical reinforcement learning, special issue on reinforcement learning," *Discrete Event Dyn. Syst.: Theory Appl.*, vol. 13, pp. 41–77, 2003.

[2] J. Baxter and P. L. Bartlett, "Infinite-horizon policy-gradient estimation," *J. Artif. Intell. Res.*, vol. 15, pp. 319–350, 2001.
[3] X. R. Cao, "Basic ideas for event-based optimization of Markov systems," *Discrete Event Dyn. Syst.: Theory Appl.*, vol. 15, pp. 169–197, 2005.
[4] X. R. Cao and H. F. Chen, "Perturbation realization, potentials and sensitivity analysis of Markov processes," *IEEE Trans. Autom. Control*, vol. 42, no. 10, pp. 1382–1393, Oct. 1997.
[5] X. R. Cao and J. Y. Zhang, "The nth-order bias optimality for multi-chain Markov decision processes," *IEEE Trans. Autom. Control*, vol. 53, no. 2, pp. 496–508, Mar. 2008.
[6] E. Çinlar, *Introduction to Stochastic Processes*. Upper Saddle River, NJ: Prentice-Hall, 1995.
[7] W. L. Cooper, S. G. Henderson, and M. E. Lewis, "Convergence of simulation-based policy iteration," *Probab. Eng. Inf. Sci.*, vol. 17, pp. 213–234, 2003.
[8] H. T. Fang and X. R. Cao, "Potential-based on-line policy iteration algorithms for Markov decision processes," *IEEE Trans. Autom. Control*, vol. 49, no. 4, pp. 493–505, Apr. 2004.
[9] P. Marbach and T. N. Tsitsiklis, "Simulation-based optimization of Markov reward processes," *IEEE Trans. Autom. Control*, vol. 46, no. 2, pp. 191–209, Feb. 2001.