A Basic Formula for Online Policy Gradient Algorithms

Xi-Ren Cao

***Abstract*—This note presents a (new) basic formula for sample-path-based estimates for performance gradients for Markov systems (called policy gradients in reinforcement learning literature). With this basic formula, many policy-gradient algorithms, including those that have previously appeared in the literature, can be easily developed. The formula follows naturally from a sensitivity equation in perturbation analysis. New research direction is discussed.**

***Index Terms*—Markov decision processes, online estimation, perturbation analysis (PA), perturbation realization, Poisson equations, potentials, reinforcement learning.**

I. Introduction

The policy-gradient approach has recently attracted increasing attentions the optimization and reinforcement learning communities. In the terminology of perturbation analysis (PA) [18], [6], [5], [13], policy-gradient algorithms are called single-sample-path-based performance gradient algorithms. This note presents a basic formula for policy gradients, based on which many policy-gradient algorithms, including those that have previously appeared in literature (e.g., [1], [2], [12], [19], and [20]), can be developed. This basic formula follows naturally from a performance sensitivity equation derived by using perturbation analysis of Markov processes [7], [8]. Performance optimization algorithms for Markov systems can be developed by using this basic formula together with stochastic approximation methods.

The main contributions of this note are as follows. First, for the first time, we derive the basic formulas (7) and develop a general algorithm for performance gradients (8) and prove its convergence. Second, we show that various algorithms in the existing literature can be obtained as special cases of the general algorithm. Third, this general algorithm provides a direction for developing new performance-gradient algorithms, especially for problems with special structures.

II. Basic Formula

Consider an irreducible and aperiodic Markov chain $\mathbf{X} = \{X_n : n \geq 0\}$ on a finite state $\mathcal{S} = \{1, 2, \ldots, M\}$ with transition probability matrix $P = [p(i,j)] \in [0,1]^{M \times M}$. Let $\pi = (\pi(1), \ldots, \pi(M))$ be the (row) vector representing its steady-state probabilities, and $f = (f(1), f(2), \ldots, f(M))^T$ be the (column) performance vector, where "T" represents transpose. We have $Pe = e$, where $e = (1, 1, \ldots, 1)^T$ is an M-dimensional vector whose components all equal 1, and $\pi e = 1$. The steady-state probability flow balance equation is $\pi = \pi P$. The performance measure is the long-run average defined as

$$\eta = E_\pi(f) = \sum_{i=1}^{M} \pi(i) f(i) = \pi f = \lim_{L \to \infty} \frac{1}{L} \sum_{l=0}^{L-1} f(X_l), \qquad w.p.1.$$

We start with the *Poisson equation*

$$(I - P)g + e\eta = f. \tag{1}$$

Manuscript received November 25, 2003; revised August 23, 2004. Recommended by Associate Editor R. S. Srikant. This work was supported in part by a grant from the Hong Kong UGC.

The author is with the Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: eecao@ee.ust.hk).

Digital Object Identifier 10.1109/TAC.2005.847037

Its solution $g = (g(1), \ldots, g(M))^T$ is called a *performance potential* vector, and $g(i)$ is the potential at state i. (It is equivalent to the value function in dynamic programming, or the "differential" or "relative cost vector" [3], and "bias" [21].) The solution to (1) can only be obtained up to an additive constant, i.e., if g is a solution to (1), then so is $g + ce$. The difference of the potentials at two states is called a perturbation realization factor in PA literature and is denoted as $d(i,j) = g(j) - g(i)$, $i, j \in \mathcal{S}$ [7], [8].

Let P' be another irreducible and aperiodic transition probability matrix on the same state–space and π' be its steady-state probability. Let f' be the performance function for the system with P', $Q = P' - P = [q(i,j)]$ and $h = f' - f$. Then, $Qe = 0$. The steady-state performance corresponding to P' is $\eta' = \pi' f'$. Multiplying both sides of (1) with π', we can verify that

$$\eta' - \eta = \pi'(Qg + h). \tag{2}$$

Now, suppose that P changes to $P(\delta) = P + \delta Q = \delta P' + (1-\delta)P$, and f changes to $f(\delta) = f + \delta h$, with $\delta \in (0, 1]$. Then the performance measure changes to $\eta(\delta) = \eta + \Delta\eta(\delta)$. The derivative of η in the direction of Q is defined as $d\eta/d\delta = \lim_{\delta \to 0} \Delta\eta(\delta)/\delta$. Taking $P(\delta)$ as the P' in (2), we have $\eta(\delta) - \eta = \pi(\delta)(\delta Qg + \delta h)$. Letting $\delta \to 0$, we get

$$\frac{d\eta}{d\delta} = \pi(Qg + h). \tag{3}$$

For references, see, e.g., [7] and [8]. Since $Qe = 0$, for any g satisfying (1) for any constant c, we have $Qg = Q(g + ce)$, thus both (3) and (2) still hold for $g' = g + ce$.

In (3), a linear structure $P(\delta) = P + \delta Q$ is assumed. In general, the transition probability matrix may depend on an arbitrary parameter θ, which is normalized to lie in [0,1]; i.e., $P(\theta) = P + Q(\theta)$ with $Q(\theta)e = 0$, $Q(0) = 0$, $P(0) = P$, and $Q(1) = P(1) - P = P' - P$. Similarly, we assume $f(\theta) = f + h(\theta)$. Thus, for $\theta \ll 1$, we have $P(\theta) = P + \{dQ/d\theta\}_{\theta=0}\theta$, and $f(\theta) = f + \{dh/d\theta\}_{\theta=0}\theta$. Replacing Q in (3) with $\{dQ/d\theta\}_{\theta=0}$ and h with $\{dh/d\theta\}_{\theta=0}$ and noting that $dP/d\theta = dQ/d\theta$ and $df/d\theta = dh/d\theta$ we get

$$\frac{d\eta}{d\theta}\,|_{\theta=0} = \pi \left\{ \left(\frac{dP}{d\theta} \right)_{\theta=0} g + \left(\frac{df}{d\theta} \right)_{\theta=0} \right\}. \tag{4}$$

Therefore, without loss of generality, we shall mainly discuss the linear case (3).

Both sensitivity equations (3) and (2) depend mainly on the same quantity: the performance potential, and both depend on only the potential g (not g'). These two equations form the basis for performance optimization of Markov systems. Two basic approaches can be developed from them. First, policy iteration algorithms can be developed using (2) (see, e.g., [10]). Next, performance gradients can be estimated using (3); this is called perturbation analysis in control literature and policy gradient in reinforcement learning. Combining the gradient estimation with stochastic approximation techniques leads to performance optimization algorithms.

Compared with policy iteration, the policy gradient method (or perturbation analysis) has some advantages: both π and g can be estimated based on a single sample path of the Markov chain with transition matrix P. Thus, for any given Q and h in (3) [or $(dP/d\theta)_{\theta=0}$ and $(df/d\theta)_{\theta=0}$ in (4)], we can estimate the gradient on a sample path with P. Furthermore, algorithms can be developed to estimate πQg directly without estimating each component of g.

There are a number of policy gradient algorithms in literature. In this note, we present a basic formula for online estimations of the performance gradient. This fundamental formula provides a clear picture for the existing policy gradient algorithms as well as points to the direction of the development of new algorithms.

Consider a stationary Markov chain $\mathbf{X} = (X_0, X_1, \ldots,)$. (This implies the initial probability distribution is the steady-state distribution π.) Let E denote the expectation on the probability space generated by $\mathbf{X}$. Denote a generic time instant as k. Because it is impossible for a sample path with transition matrix P to contain information about P' (or $Q = P' - P$), we need to use a standard technique in simulation called *importance sampling*. First, we make a standard assumption in importance sampling: for any $i, j \in \mathcal{S}$, if $q(i,j) \neq 0$, then $p(i,j) \neq 0$. Then, we have

$$\begin{aligned}\frac{d\eta}{d\delta} &= \pi(Qg + h) = \sum_{i\in\mathcal{S}}\sum_{j\in\mathcal{S}}[\pi(i)q(i,j)g(j)] + \sum_{i\in\mathcal{S}}[\pi(i)h(i)] \\ &= \sum_{i\in\mathcal{S}}\sum_{j\in\mathcal{S}}\left\{\pi(i)\left[p(i,j)\frac{q(i,j)}{p(i,j)}g(j) + h(i)\right]\right\} \\ &= E\left\{\frac{q(X_k, X_{k+1})}{p(X_k, X_{k+1})}g(X_{k+1}) + h(X_k)\right\}. \end{aligned} \tag{5}$$

Next, let $\hat{g}(X_{k+1}, X_{k+2}, \ldots)$ be an unbiased estimate of $g(x_{k+1})$, i.e.,

$$g(i) = E\{\hat{g}(X_{k+1}, X_{k+2}, \ldots) \mid X_{k+1} = i\}, \qquad i \in \mathcal{S}. \tag{6}$$

With (6), we have

$$\begin{aligned} &E\left\{\frac{q(X_k, X_{k+1})}{p(X_k, X_{k+1})}\hat{g}(X_{k+1}, X_{k+2}, \ldots) + h(X_k)\right\} \\ &= E\left\{E\left[\frac{q(X_k, X_{k+1})}{p(X_k, X_{k+1})}\hat{g}(X_{k+1}, X_{k+2}, \ldots)\right.\right. \\ &\quad \left.\left. + h(X_k) \mid X_k, X_{k+1}\right]\right\} \\ &= E\left\{\frac{q(X_k, X_{k+1})}{p(X_k, X_{k+1})}E\left[\hat{g}(X_{k+1}, X_{k+2}, \ldots) \mid X_k, X_{k+1}\right]\right. \\ &\quad \left. + h(X_k)\right\} \\ &= E\left\{\frac{q(X_k, X_{k+1})}{p(X_k, X_{k+1})}g(X_{k+1}) + h(X_k)\right\}. \end{aligned}$$

Therefore

$$\frac{d\eta}{d\delta} = E\left\{\frac{q(X_k, X_{k+1})}{p(X_k, X_{k+1})}\hat{g}(X_{k+1}, X_{k+2}, \ldots) + h(X_k)\right\}. \tag{7}$$

Next, we develop single-sample-path-based algorithms for estimating the gradients. It is natural to consider

$$\frac{1}{K}\sum_{k=0}^{K-1}\left\{\left[\frac{q(X_k, X_{k+1})}{p(X_k, X_{k+1})}\right]\hat{g}(X_{k+1}, X_{k+2}, \ldots) + h(X_k)\right\}. \tag{8}$$

Because $\hat{g}(X_{k+1}, X_{k+2}, \ldots)$, $k = 0, 1, \ldots$, are not independent, so the law of large numbers for $K \to \infty$ does not apply directly. Fortunately, a theorem on ergodicity can be used to prove the convergence of (8).

Theorem 1: For an ergodic Markov chain $\mathbf{X} = \{X_0, X_1, \ldots, \}$, let $\hat{g}(X_{k+1}, X_{k+2}, \ldots)$ be an unbiased estimate of $g(X_{k+1})$ satisfying (6). Suppose for any $i, j \in \mathcal{S}$, if $q(i,j) \neq 0$ then $p(i,j) \neq 0$. Then, for the performance gradient defined in (3), we have (9), as shown at the bottom of the page.

Proof: The proof is based on a fundamental theorem on ergodicity [4]: If $\mathbf{X} = \{X_k, k \geq 0\}$ is an ergodic process on state $\mathcal{S}$, let $\phi(x_1, x_2, \ldots)$ be a measurable function on $\mathcal{S}$, then the process $Z = \{Z_k, k \geq 0\}$ with $Z_k = \phi(X_k, X_{k+1}, \ldots)$ is also ergodic. In our case, we define $Z_k = (q(X_k, X_{k+1})/p(X_k, X_{k+1}))\hat{g}(X_{k+1}, X_{k+1}, \ldots) + h(X_k)$; then $\mathbf{Z} = \{Z_k, k \geq 0\}$ is ergodic. Thus, (9) converges w.p. 1. to the steady-state mean in $\mathbf{Z}$, which is (7). The theorem is thus proved. □

The ergodic theorem in [4] is very useful in provong many similar results. It was first used to prove a special case of (9) ([12]; see Algorithm 1) and later in [1] and [2] for similar results.

III. Online Policy Gradient Algorithms

In this section, we show how (9) can be used to derive specific policy-gradient algorithms and present several such algorithms.

Algorithm 1. (Approximation by Truncation): It is well known that (see, e.g., [7] and [8])

$$g(i) = E\left\{\sum_{k=0}^{\infty}[f(X_k) - \eta] \mid X_0 = i\right\}.$$

To avoid the difficulty in computation caused by the infinite sum, we use an approximation by truncation

$$g(i) \approx E\left\{\sum_{k=0}^{L-1}[f(X_k) - \eta] \mid X_0 = i\right\}.$$

Because the potentials are defined only up to an additive constant, we may ignoring the constant term $L\eta$ and obtain

$$g(i) \approx E\left\{\sum_{k=0}^{L-1} f(X_k) \mid X_0 = i\right\}.$$

Therefore, from (6) we may choose

$$\hat{g}(X_{k+1}, X_{k+2}, \ldots) \approx \sum_{l=0}^{L-1} f(X_{k+l+1}).$$

Using this $\hat{g}$ in (9) (for simplicity, assume $h(X_k) \equiv 0$), we get

$$\frac{d\eta}{d\delta} \approx \lim_{K\to\infty}\frac{1}{K}\left\{\sum_{k=0}^{K-1}\left\{\frac{q(X_k, X_{k+1})}{p(X_k, X_{k+1})}\right\}\left[\sum_{l=0}^{L-1} f(X_{k+l+1})\right]\right\}, \qquad w.p.1. \tag{10}$$

This is equivalent to

$$\frac{d\eta}{d\delta} \approx \lim_{K\to\infty}\frac{1}{K}\sum_{k=0}^{K-1}\left\{f(X_{k+L})\sum_{l=0}^{L-1}\left[\frac{q(X_{k+l}, X_{k+l+1})}{p(X_{k+l}, X_{k+l+1})}\right]\right\}, \qquad w.p.1. \tag{11}$$

This algorithm and similar ones for Markov processes and queueing networks are presented in [12].

$$\frac{d\eta}{d\delta} = \lim_{K\to\infty}\frac{1}{K}\sum_{k=0}^{K-1}\left\{\left[\frac{q(X_k, X_{k+1})}{p(X_k, X_{k+1})}\right]\hat{g}(X_{k+1}, X_{k+2}, \ldots) + h(X_k)\right\}, \qquad w.p.1 \tag{9}$$

Algorithm 2. (Approximation by Discount Factors): Potential g can be approximated by the α-potential g_α, satisfying the following discounted Poisson equation [9]:

$$(I - \alpha P + \alpha e\pi)g_\alpha = f$$

with $0 < \alpha < 1$ being a discount factor. It is shown [9] that

$$\lim_{\alpha \to 1} g_\alpha = g.$$

Ignoring the constant term, we have [9]

$$g_\alpha(i) = E\left[\sum_{l=0}^{\infty} \alpha^l f(X_l) \mid X_0 = i\right].$$

Therefore, we can choose

$$\hat{g}(X_{k+1}, X_{k+2}, \ldots) \approx \sum_{l=0}^{\infty} \alpha^l f(X_{k+l+1}).$$

Using this as the $\hat{g}$ in (7), we get

$$\frac{d\eta}{d\delta} \approx \lim_{K\to\infty} \frac{1}{K}\left\{\sum_{k=0}^{K-1}\left\{\frac{q(X_k, X_{k+1})}{p(X_k, X_{k+1})}\right\}\left[\sum_{l=0}^{\infty}\alpha^l f(X_{k+l+1})\right]\right\}, \qquad w.p.1. \tag{12}$$

If we can exchange the order of $f(X_k)$ and

$$q(X_k, X_{k+1})/p(X_k, X_{k+1})$$

in the aforementioned double sum, then we have

$$\frac{d\eta}{d\delta} \approx \lim_{K\to\infty} \frac{1}{K}\sum_{k=0}^{K-1}\left\{f(X_k)\sum_{l=0}^{k-1}\left[\alpha^{k-l-1}\frac{q(X_l, X_{l+1})}{p(X_l, X_{l+1})}\right]\right\}, \qquad w.p.1. \tag{13}$$

This is the policy-gradient algorithm developed in [1], [2]. [1] also contains a proof for the fact that the order of the double sum in (12) is indeed exchangable.

It is easy to estimate

$$z_k := \sum_{l=0}^{k-1}\left[\alpha^{k-l-1}(q(X_l, X_{l+1})/p(X_l, X_{l+1}))\right]$$

recursively

$$z_{k+1} = \alpha z_k + \frac{q(X_k, X_{k+1})}{p(X_k, X_{k+1})}.$$

On the other hand, to estimate

$$\sum_{l=0}^{L-1}[q(X_{k+l}, X_{k+l+1})/p(X_{k+l}, X_{k+l+1})]$$

one has to store L values.

Finally, the discount factor approximation is also used in [20] to reduce the variance in estimating the performance gradients.

Algorithm 3 (Based on Perturbation Realization Factors): It is sometimes easier to estimate the differences between the potentials of two states, called *perturbation realization factors* in perturbation analysis [7], [10], [8], which is defined as

$$d(i, j) = g(j) - g(i), \qquad i, j \in \mathcal{S}.$$

The matrix $D = [d(i, j)]$ is called a *realization matrix*. We have $D^T = -D$ and $D = eg^T - ge^T$. D satisfies the Lyapunov equation

$$D - PDP^T = F$$

with $F = ef^T - fe^T$.

Now, we consider a Markov chain $\mathbf{X} = \{X_k, k \geq 0\}$ with initial state $X_0 = i$, we define $L_i(j) = \min\{n : n \geq 0, X_n = j\}$, i.e., at $n = L_i(j)$, the Markov chain reaches state j for the first time. We have $E[L_i(j) \mid X_0 = i] < \infty$ [14], and from [7], [8]

$$d(j, i) = E\left\{\sum_{k=0}^{L_i(j)-1}[f(X_k) - \eta] \mid X_0 = i\right\}. \tag{14}$$

To develop an algorithm, we first use (14) to obtain a $\hat{g}$. To this end, we choose any regenerative state i^*. For convenience, we set $X_0 = i^*$ and define $u_0 = 0$, and $u_{m+1} = \min\{n : n > u_m, X_m = i\}$ be the sequence of regenerative points. Set $g(i^*) = 0$. From (14), for any $X_n = i \neq i^*$ and $u_m \leq n < u_{m+1}$ we have

$$g(i) = d(i^*, i) = E\left\{\sum_{l=n}^{u_{m+1}-1}[f(X_l) - \eta] \mid X_n = i\right\}.$$

Because the lengths $u_{m+1} - n$ are not the same for different i, we cannot ignore the term η in the above equation. Choose $\hat{g}(X_{k+1}, \ldots) = \sum_{l=k+1}^{u_{m+1}-1}[f(X_l) - \eta]$. Then, by (9), we have

$$\frac{d\eta}{d\delta} = \lim_{K\to\infty} \frac{1}{K}\sum_{k=0}^{K-1}\left\{\left[\frac{q(X_k, X_{k+1})}{p(X_k, X_{k+1})}\right]\sum_{l=k+1}^{u_{m+1}-1}[f(X_l) - \eta]\right\}, \qquad w.p.1$$

where u_{m+1} is the first time after X_{k+1} that the Markov chain reaches state i^*. (If $k + 1 = u_{m+1}$, then $g(X_{k+1}) = 0$, and the term in the above sum is zero.) The optimization scheme proposed in [19] is essentially a result of combining the previous algorithm with stochastic approximation techniques.

Algorithm 4. (Partially Observable Markov Decision Processes): The algorithms can be easily extended to the partially observable Markov decision processes (POMDP) (see, e.g., [1], [2]). The POMDP model in [1] and [2] is described as follows. In addition to the state–space $\mathcal{S} = \{1, \ldots, M\}$, there is a control space denoted as $\mathcal{U} = \{1, \ldots, N\}$ consisting of N controls and an observation space $\mathcal{Y} = \{1, \ldots, L\}$ consisting of L observations. Each $u \in \mathcal{U}$ determines a transition probability matrix P^u, which does not depend on the parameter θ. When the Markov chain is at state $i \in \mathcal{S}$, an observation $y \in \mathcal{Y}$ is obtained according to a probability distribution $\nu_i(y)$. For any observation y, we may choose a random policy $\mu_y(u)$, which is a probability distribution over the control space $\mathcal{U}$. It is assumed that the distribution depends on a parameter θ and, therefore, is denoted as $\mu_y(\theta, u)$.

Given an observation distribution $\nu_i(y)$ and a random policy $\mu_y(\theta, u)$, the corresponding transition probabilities are

$$p_\theta(i, j) = \sum_{u,y}\{\nu_i(y)\mu_y(\theta, u)p^u(i, j)\}.$$

Therefore

$$\frac{d}{d\theta}p_\theta(i, j) = \sum_{u,y}\left\{\nu_i(y)p^u(i, j)\frac{d}{d\theta}\mu_y(\theta, u)\right\}. \tag{15}$$

In POMDP, we assume that although the state X_n, $n = 0, 1, \ldots$, is not completely observable, the cost $f(X_n)$ is known. Thus, algorithms can be developed by replacing $q(i, j)$ in (7) and (9) with $(d/d\theta)p_\theta(i, j)$ of (15). We have

$$\frac{d\eta_\delta}{d\delta} = E\left\{\frac{\frac{d}{d\theta}p_\theta(X_k, X_{k+1})}{p(X_k, X_{k+1})}\left\{\sum_{l=0}^{\infty}\alpha^l f(X_{k+l+1})\right\}\right\}$$

in which $f(X_k)$ is assumed to be observable. Based on this equation a recursive algorithm called GPOMDP is presented in [1].

If the recurrent state i^* is observable, we have the following algorithm (cf. Algorithm 3):

$$\frac{d\eta_\delta}{d\delta} = E\left\{\frac{\frac{d}{d\theta}p_\theta(X_k, X_{k+1})}{p(X_k, X_{k+1})}\left\{\sum_{l=k+1}^{u_{m+1}-1}[f(X_l)-\eta]\right\}\right\}.$$

IV. Remarks and Discussions

Early work on sample-path-based performance gradient estimation include the PA [18], [6], [13] and the likelihood ratio (LR) [also called the score function (SF)] methods [15], [16], [22], [23]. PA was first developed for queueing networks; efficient algorithms have been developed [17]. The main idea of PA, perturbation realization, was later extended to performance gradients of Markov systems [7], [8].

Policy gradient [1], [2] is a terminology used in recent years in RL community for sample-path-based performance gradient estimate of PA. However, there is a slight difference in their emphases. Most policy gradient papers focus on developing simulation/online algorithms for estimating performance gradients. PA, on the other hand, emphasizes two aspects: deriving performance gradient formulas (those similar to (2)), and developing estimation algorithms. With the concept of perturbation realization factors, we can flexibly derive sensitivity formulas for many problems; these formulas are otherwise difficult to conceive [11]. Sample-path-based algorithms can be developed/designed only after these performance gradient formulas are derived. The readers can find some examples of the performance gradient formulas for systems with special structures in [11]. The basic formula (7) and the general algorithm (8) presented in this note provide a direction for developing performance gradient algorithms using the performance gradient formulas.

References

[1] J. Baxter and P. L. Bartlett, "Infinite-horizon policy-gradient estimation," *J. Art. Intell. Res.*, vol. 15, pp. 319–350, 2001.

[2] J. Baxter, P. L. Bartlett, and L. Weaver, "Experiments with infinite-horizon policy-gradient estimation," *J. Art. Intell. Res.*, vol. 15, pp. 351–381, 2001.

[3] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 1995, vol. I and II.

[4] L. Breiman, *Probability*. Reading, MA: Addison-Wesley, 1968.

[5] X. R. Cao, "Convergence of parameter sensitivity estimates in a stochastic experiment," *IEEE Trans. Autom. Control*, vol. AC- 30, no. 9, pp. 845–853, Sep. 1985.

[6] ——, *Realization Probabilities: The Dynamics of Queueing Systems*. New York: Springer-Verlag, 1994.

[7] X. R. Cao, X. M. Yuan, and L. Qiu, "A single sample path-based performance sensitivity formula for Markov chains," *IEEE Trans. Autom. Control*, vol. 41, no. 12, pp. 1814–1817, Dec. 1996.

[8] 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.

[9] X. R. Cao, "A unified approach to Markov decision problems and performance sensitivity analysis," *Automatica*, vol. 36, pp. 771–774, 2000.

[10] ——, "From perturbation analysis to Markov decision processes and reinforcement learning," *Discrete Event Dyna. Syst.: Theory Appl.*, vol. 13, pp. 9–39, 2003.

[11] ——, "The potential structure of sample paths and performance sensitivities of Markov systems," *IEEE Trans. Autom. Control*, vol. 49, no. 12, pp. 2129–2142, Dec. 2004.

[12] X. R. Cao and Y. W. Wan, "Algorithms for sensitivity analysis of Markov systems through potentials and perturbation realization," *IEEE Trans. Control Syst. Technol.*, vol. 6, no. 4, pp. 482–494, Jul. 1998.

[13] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Norwell, MA: Kluwer, 1999.

[14] E. Çinlar, *Introduction to Stochastic Processes*. Upper Saddle River, NJ: Prentice-Hall, 1975.

[15] P. W. Glynn, "Likelihood ratio gradient estimation: An overview," in *Proc. Winter Simulation Conf.*, 1987, pp. 366–375.

[16] ——, "Optimization of stochastic systems via simulation," in *Proc. Winter Simulation Conf.*, 1989, pp. 90–105.

[17] Y. C. Ho and X. R. Cao, "Perturbation analysis and optimization of queueing networks," *J. Optim. Theory Appl.*, vol. 40, no. 4, pp. 559–582, 1983.

[18] ——, *Perturbation Analysis of Discrete-Event Dynamic Systems*. Norwell, MA: Kluwer, 1991.

[19] 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.

[20] ——, "Approximate gradient methods in policy-space optimization of Markov reward processes," J. Discrete Event Dyna. Syst., 2002, to be published.

[21] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York: Wiley, 1994.

[22] M. I. Reiman and A. Weiss, "Sensitivity analysis via likelihood ratio," *Oper. Res.*, vol. 37, pp. 830–844, 1989.

[23] R. V. Rubinstein, *Monte Carlo Optimization, Simulation, and Sensitivity Analysis of Queueing Networks*. New York: Wiley, 1986.

A Min-Plus Derivation of the Fundamental Car-Traffic Law

Pablo A. Lotito, Elina M. Mancinelli, and Jean-Pierre Quadrat

***Abstract*—We give deterministic and stochastic models of the traffic on a circular road without overtaking. From this model the mean speed is derived as an eigenvalue of the min-plus matrix describing the dynamics of the system in the deterministic case and as the Lyapunov exponent of a min-plus stochastic matrix in the stochastic case. The eigenvalue and the Lyapunov exponent are computed explicitly. From these formulas, we derive the fundamental law that links the flow to the density of vehicles on the road. Numerical experiments using the MaxPlus toolbox of Scilab confirm the theoretical results obtained.**

***Index Terms*—Cellular automata, fundamental diagram, Lyapunov exponent, max-plus algebra.**

I. Introduction

For simple traffic models a well known relation exists between the flow and the density of vehicles called *fundamental traffic law*. This law has been studied empirically and theoretically using exclusion processes (see, for example, [5]–[7], [3], [12], and [8]) and cellular automata (see [1]).

In this note, we analyze the simplest deterministic and stochastic traffic models using the so called *min-plus algebra*. Within this algebra the equations of the dynamics become linear and the eigenvalue or the Lyapunov exponent of the corresponding min-plus matrix gives the mean speed from which we easily derive the density-flow relation.

Manuscript received August 20, 2003; revised April 15, 2004 and July 6, 2004. Recommended by Associate Editor A. Giua.

P. A. Lotito is with the GRETIA-INRETS, 94114 Arccueil, France (e-mail: pablo.lotito@inria.fr).

E. M. Mancinelli is with the INRIA, 78153 Le Chesnay, Cedex, France, and also with the CONICET, Argentina (e-mail: elina.mancinelli@inria.fr).

J.-P. Quadrat is with the INRIA, 78153 Le Chesnay, Cedex, France (e-mail: jean-pierre.quadrat@inria.fr).

Digital Object Identifier 10.1109/TAC.2005.848336