A Self-Learning Adaptive Critic Approach for Call Admission Control in Wireless Cellular Networks

Derong Liu and Yi Zhang
Department of Electrical and Computer Engineering
University of Illinois, Chicago, IL 60607, U. S. A.
Email: dliu@ece.uic.edu, yzhang@cil.ece.uic.edu

***Abstract*— We apply in the present paper adaptive critic designs to the call admission control problem in CDMA cellular networks. A novel learning control architecture of adaptive critic designs is used in our study. The call admission controller performs learning in real-time as well as in off-line environments and the controller improves its performance through continuous leaning.**

I. Introduction

Adaptive critic designs (ACDs) were first introduced in the 1970s [19], [20]. ACDs are defined as designs that *approximate dynamic programming in the general case*. There are many problems in practice that can be formulated as cost maximization or minimization problems. Dynamic programming is a very useful tool in solving these problems. However, it is often computationally untenable to run dynamic programming due to the backward numerical process required for its solutions, i.e., due to the "curse of dimensionality." Over the years, progress has been made to circumvent the "curse of dimensionality" by approximating dynamic programming solutions where a function approximation structure such as neural networks is used to approximate the cost function. ACDs have found many applications in learning control problems (e.g., see [1], [2], [7], [8], [13], [17]).

Call admission control policy is one of the most critical design considerations in wireless networks [3], [12], [16]. On one hand, call admission control schemes provide the users with access to wireless networks for services. On the other hand, they are the decision making part of the network carriers with the objectives of providing services to users with guaranteed quality and at the same time, achieving as high as possible resource utilization.

In the DS-CDMA cellular network model used in this paper, we assume that separate frequency bands are used for the reverse link and the forward link, so that the mobiles only experience interference from the base stations and the base stations only experience interference from the mobiles. The power received at the base station from the nth user (i.e., from the nth mobile station) is denoted by $S_n,\ n = 1, \cdots, N$. The bit SIR (or the bit energy-to-interference ratio) for the nth user at the base station (in a cell) can be expressed in terms of the received powers of the various users as [5]

$$(E_b/N_0)_n = \frac{S_n W}{I_n R_{\sigma(n)}} \tag{1}$$

where S_n is the power level of the nth user received at the base station, W is the chip rate, and $R_{\sigma(n)}$ is the data rate of service class $\sigma(n)$. The function $\sigma(n)$ is defined as $\sigma\colon Z^+ \to \{1, \cdots, K\}$ to indicate the fact that the nth user is from the service class $\sigma(n)$, where Z^+ denotes the set of nonnegative integers. I_n indicates the total interference to the nth user's signal received at the base station and is given by

$$I_n = (1+f) \sum_{i=1, i\neq n}^{N} \nu_{\sigma(i)} S_i + \eta_n,$$

where $\nu_{\sigma(i)}$ is the traffic (e.g., voice) activity factor of the ith user, η_n is the background (or thermal) noise, N is the number of active users in the network, and f is called the intercell interference factor [18].

Assume that the power control algorithm converges and it requires the power received at base station from each user by $S_n^*, n = 0, 1, \cdots, N$, where the total number of users in the system is $N+1$ and user 0 is the newly admitted caller. Obviously, if $S_n^* > H_{\sigma(n)}$ or $S_n^* \le 0$, for some n, $0 \le n \le N$, the admission should not be granted. Only when $0 < S_n^* \le H_{\sigma(n)}$ for all n, $n = 0, 1, \cdots, N$, the admission decision can be considered as a correct decision.

Many works have chosen to use the following definition for the grade of service (GoS):

$$\text{GoS} = P(\text{call blocking}) + w \times P(\text{handoff failure}), \tag{2}$$

where $P(a)$ is the probability of event a and w is typically chosen as, e.g., 10. On the other hand, quality of service (QoS) is usually defined according to the bit error rate in digital transmission. For example, the quality of service requirement for voice users is usually expressed as a bit error rate less than 10^{-3} in order to guarantee the quality of communication which can be satisfied by the power control mechanism keeping E_b/N_0 at a required value of 7 dB or higher [5], [12], [16].

For a given set of parameters including traffic statistics and mobility characteristics, fixed call admission control schemes can sometimes yield optimal solutions [14] in terms of GoS. All such schemes (cf. [14], [15], [16]), however, by reserving a fixed part of capacity, cannot adapt to changes in the network conditions due to its static nature. We develop in the present work a self-learning call admission control algorithm for CDMA wireless networks that can automatically adapt to changes in the environment.

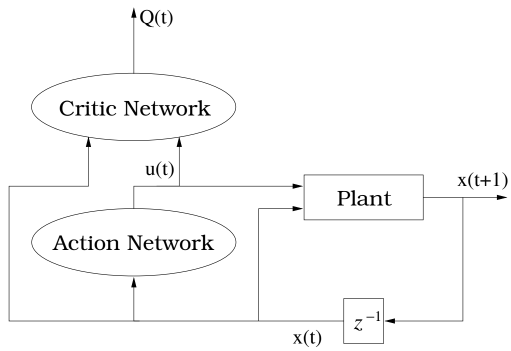

Fig. 1. A typical scheme of an action-dependent heuristic dynamic programming [11].

II. Architectures of Adaptive Critic Designs

Suppose that one is given a discrete-time nonlinear dynamical system

$$x(t+1) = F[x(t), u(t), t]$$

where $x \in R^n$ represents the state vector of the system and $u \in R^m$ denotes the control action. Suppose that one associates with this system the performance index (or cost)

$$J[x(i), i] = \sum_{k=i}^{\infty} \gamma^{k-i} U[x(k), u(k), k] \tag{3}$$

where U is called the utility function or local cost function and γ is the discount factor with $0 < \gamma \le 1$. The objective is to choose the control sequence $u(k)$, $k = i, i+1, \cdots$, so that the function J (i.e., the cost) in (3) is *minimized*.

Consider the action-dependent heuristic dynamic programming (ADHDP) shown in Figure 1 (cf. [11]). The critic network in this case will be trained by minimizing the following error measure over time,

$$\|E_q\| = \sum_t [Q(t-1) - U(t) - \gamma Q(t)]^2 \tag{4}$$

where $Q(t) = Q[x(t), u(t), t, W_C]$. When $\|E_q\| = 0$, (4) implies that

$$\begin{aligned} Q(t-1) &= U(t) + \gamma Q(t) \\ &= U(t) + \gamma[U(t+1) + \gamma Q(t+1)] \\ &= \cdots \\ &= \sum_{k=t}^{\infty} \gamma^{k-t} U(k). \end{aligned} \tag{5}$$

Comparing (5) to (3), we can see that when minimizing the error function in (4), we have a neural network trained so that its output becomes an estimate of the cost function defined in dynamic programming for $i = t+1$, i.e., the value of the cost function in the immediate future [11].

There are many problems in practice that have a control action space that is finite. When the application has only a finite action space, the decisions about which action signal to take is constrained to a limited number of choices, e.g., a binary choice in the case of call admission control problem. When a new call or a handoff call arriving at a base station requesting for admission, the decisions that a base station can make are constrained to two choices. Let us denote these two options by using $u(t) = +1$ for "accept" and $u(t) = -1$ for "reject." Because of this, the ACDs introduced earlier in Figure 1 can be further simplified so that only the critic network is needed. The adaptive critic approach in our self-learning call admission control in wireless cellular networks is illustrated in Figure 2. When a new call or a handoff call arriving at a base station requesting for admission, we can first ask the critic network to see whether $u(t) = +1$ (accept) or $u(t) = -1$ (reject) will give a smaller output value. We will then choose the control action from $u(t) = +1$ and $u(t) = -1$ that gives a smaller critic network output. As in the case of Figure 1, the critic network would also take the states of the system as inputs. We note that Figure 2 is only a schematic diagram that shows how the computation takes place while making a call admission control decision. The two blocks for the critic network in Figure 2 represent the same network or computer code in software. The block diagram in Figure 2 indicates that the critic network will be used twice in making an admission decision.

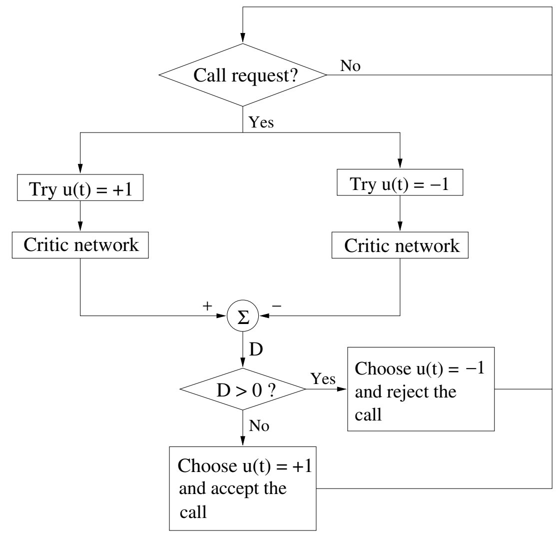

Fig. 2. The block diagram of the present adaptive critic approach.

The above description assumes that the critic network has been learned/trained successfully. Once the critic network training is done, it can be applied as in Figure 2. To guarantee that the overall system will achieve optimal performance now and in the future environments which may be significantly different from what it is now, we will allow the critic network to perform continuous learning while it is being used.

III. Adaptive Critic Approach for Call Admission Control in CDMA Cellular Networks

We use a utility function as reward or penalty to the action made by the call admission control scheme. When the call admission control scheme makes a decision about accepting a call, it will lead to two distinct results. The first is that the decision of accepting a call is indeed the right decision due to the guarantee of quality of service during the entire call

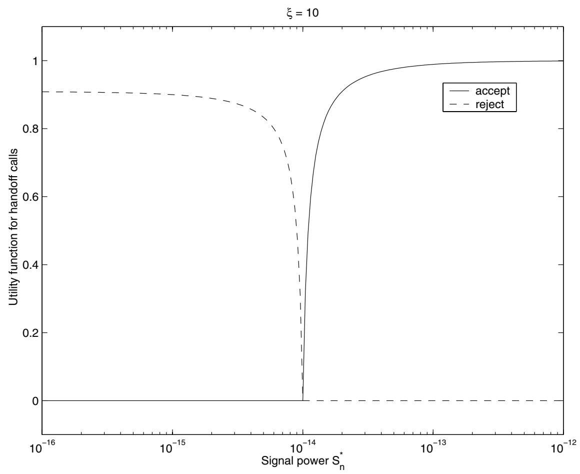

Fig. 3. Utility function for handoff calls.

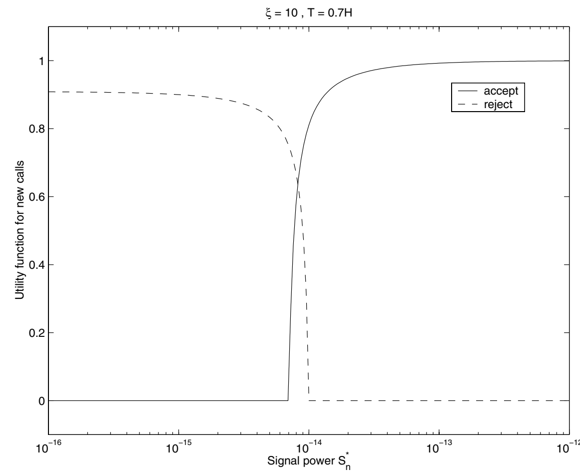

Fig. 4. Utility function for new calls.

duration. In this case we should give a reward to the decision of accepting a call. Otherwise a penalty is assigned to this decision. On the other hand, if rejecting a call would have been the right decision due to call dropouts or system outage after the call acceptance, we will also give a reward to the decision of rejecting a call. Otherwise a penalty is assigned to this decision.

It is generally accepted in practice that handoff calls will be given higher priority than new calls [14], [15], [16]. This is accomplished in our call admission control scheme by using different thresholds for new calls and handoff calls. A handoff call can be admitted if $0 < S_n^* \leq H_{\sigma(n)}$ for all $n = 0, 1, \cdots, N$. A new call can only be admitted if $0 < S_0^* \leq T_{\sigma(0)}$ and $0 < S_n^* \leq H_{\sigma(n)}$ for $n = 1, 2, \cdots, N$, where $T_{\sigma(0)} < H_{\sigma(0)}$ is the threshold for new calls.

For handoff calls, when $0 < S_n^* \leq H_{\sigma(n)}$ for $n = 0, 1, \cdots, N$, accepting a handoff call will be given a reward and rejecting a handoff call will be given a penalty. On the other hand, when $S_n^* > H_{\sigma(n)}$ for some n, $0 \leq n \leq N$, accepting a handoff call (i.e., the 0th caller) will be given a penalty and rejecting a handoff call will be given a reward. Obviously, when $S_n^* \leq 0$ for some n, $0 \leq n \leq N$, the call should be rejected. In this case, the action of rejection is given a reward and the action of acceptance is given a penalty. We define the cost function for handoff calls as follows ($S_n^* \geq 0$ for $n = 0, 1, \cdots, N$):

$$E_n = \xi * \max\left\{ u(t) * \left[\frac{S_n^*}{H_{\sigma(n)}} - 1 \right], 0 \right\} \tag{6}$$

where ξ is a coefficient and $u(t) = 1$ represents accepting a call and $u(t) = -1$ represents rejecting a call. We note that the conditions, $0 \leq S_n^* \leq H_{\sigma(n)}$ for $n = 0, 1, \cdots, N$, must hold for the entire duration of all calls in order for the system to give reward to the action of accepting a handoff call.

For new calls, when $0 < S_0^* \leq T_{\sigma(0)}$ and $0 < S_n^* \leq H_{\sigma(n)}$ for $n = 1, 2, \cdots, N$, we give a reward to the action of accepting a new call, and we give a penalty to the action of rejecting a new call. When $S_0^* > T_{\sigma(0)}$, or $S_n^* > H_{\sigma(n)}$ for some n, $n = 1, 2, \cdots, N$, or $S_n^* \leq 0$ for some n, $n = 0, 1, \cdots, N$, we give penalty for accepting a new call and we give a reward for rejecting a new call. The cost function for new calls is defined as ($S_n^* \geq 0$ for $n = 0, 1, \cdots, N$) :

$$E_n = \begin{cases} \xi * \max\left\{ \left[\dfrac{S_n^*}{T_{\sigma(n)}} - 1 \right], 0 \right\}, & \text{when } u(t) = 1 \\ \xi * \max\left\{ \left[1 - \dfrac{S_n^*}{H_{\sigma(n)}} \right], 0 \right\}, & \text{when } u(t) = -1 \end{cases} \tag{7}$$

where $T_{\sigma(n)} < H_{\sigma(n)}, n = 0, 1, \cdots, N$. We note that the conditions, $0 < S_n^* \leq H_{\sigma(n)}$ for $n = 0, 1, \cdots, N$, must again hold for the entire duration of all calls in order for the system to give reward to the action of accepting a handoff call.

Functions defined above satisfy the condition that $E_n \geq 0$ for all n, $n = 0, 1, \cdots, N$. From equations (6) and (7), we can see that when the action is "accept," if the value of the utility function of any user is larger than 0, this action should be penalized. Also, when the action is "reject," if the value of the utility function of any user is zero, this action should be rewarded. Therefore, from the system's point of view, the cost function can be chosen as

$$E = \begin{cases} \max\limits_{0 \leq n \leq N} (E_n), & \text{if } u(t) = 1 \\ \min\limits_{0 \leq n \leq N} (E_n), & \text{if } u(t) = -1. \end{cases} \tag{8}$$

The cost function defined in (8) indicates that the goal of our call admission control algorithm is to minimize the value of function E, i.e., to reach its minimum value of zero and to avoid its positive values. The utility function U (used in (3)) in our present work is chosen as

$$U(u) = \frac{E}{1 + E}. \tag{9}$$

Figures 3 and 4 show plots of this utility function for handoff calls and new calls, respectively, when $\xi = 10$. From Figures 3 and 4 we see that the choice of the present utility functions in (9) clearly shows minimum points (the flat area) that our call admission control scheme tries to reach and the points with high penalty that our scheme should avoid.

The present self-learning call admission control scheme involves the following four steps.

1) Collecting data: During this phase, when a call comes, we can accept or reject the call with any scheme and calculate the utility function for the system as presented above. In the present paper, we simply accept and reject calls randomly with the same probability of 0.5. At the same time, we collect the states corresponding to each action. The states (environment) collected for each action include total interference, call type (new call or handoff call), call class (voice or data), etc.
2) Training critic network: Using the data collected to train the critic network as mentioned in the previous section.
3) Apply critic network: The trained critic network is then applied as shown in Figure 2.
4) Continuously updating critic network: The critic network will be updated as needed while it is used in application to accommodate environment changes. Data collection has to be performed again and the training of critic network as well. In this case, the above three steps will be repeated.

IV. Simulation Results

We conduct simulation studies for a network with single class of service (e.g., voice). The network parameters used in the present simulation are taken similarly as the parameters used in [9], [16].

The arrival rate consists of the new call attempt rate λ_c and the handoff call attempt rate λ_h. λ_c depends on the expected number of subscribers per cell. λ_h depends on such network parameters as traffic load, user velocity, and cell coverage areas [6]. In our simulation, we assume that $\lambda_c : \lambda_h = 5 : 1$. A channel is released by call completion or handoff to a neighboring cell. The channel occupancy time is assumed to be exponentially distributed [6] with the same mean value of $1/\mu = 3$ minutes.

In the following, we conduct comparison studies between the present self-learning call admission control algorithm and that of static algorithm developed in [10] with fixed thresholds for new calls given by $T = H$, $T = 0.8H$, and $T = 0.5H$, respectively. The arrival rate in all neighboring cells is fixed at 18 calls/minutes. The training data is collected as mentioned in the previous section. We choose $T_{\sigma(n)} = 0.5H_{\sigma(n)}$ and $\xi = 10$ in equation (7). The critic network has three inputs. They are the total interference received at the base station, the action (1 for accepting, -1 for rejecting), and the call type (1 for new calls, -1 for handoff calls). Figure 5 shows the simulation results. We see from the figure that the performance of the self-learning algorithm is similar to the case of static algorithm with $T = 0.5H$, because we choose $T_{\sigma(n)} = 0.5H_{\sigma(n)}$ in (7) for our learning control algorithm. When the call arrival rate is low, the self-learning algorithm is not so good because it reserves too much for handoff calls and it rejects too many new calls. That is why the GoS is worse than the other two cases of static algorithms ($T = 1.0H$ and $T = 0.8H$). In order to improve the GoS when the call arrival rate is low, we modify the utility function for new calls as follows:

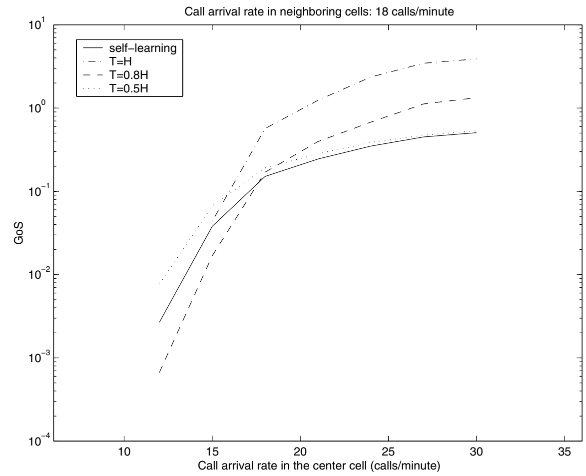

Fig. 5. Comparison study using utility function defined in (7).

$$E_n = \begin{cases} \xi * \max\left\{\left[\dfrac{S_n^*}{H_{\sigma(n)}} - 1\right], 0\right\}, & u(t) = 1,\ n_a \le N_h \\ \xi * \max\left\{\left[\dfrac{S_n^*}{T_{\sigma(n)}} - 1\right], 0\right\}, & u(t) = 1,\ n_a > N_h \\ \xi * \max\left\{\left[1 - \dfrac{S_n^*}{H_{\sigma(n)}}\right], 0\right\}, & u(t) = -1 \end{cases} \tag{10}$$

where n_a is the number of active calls handed off to the cell, N_h is a fixed parameter indicating the threshold for low traffic load. We choose $N_h = 15$ in the our simulation. Using this new utility function we collect the training data using one of the static algorithm with fixed threshold or the previous critic network. Then we train the new critic network with the newly collected data. This time the new critic network has four inputs. Three of them are the same as in the previous critic network. The new input is equal to 1 when $n_a \le N_h$ and otherwise it is equal to -1. Figure 6 shows the result of applying the new critic network to the same traffic pattern as in Figure 5. From the figure we see that the self-learning algorithm using the new critic network has the best GoS. By simply changing the cost function from (9) to (10), the self-learning algorithm can significantly improve its performance to outperform static admission control algorithms.

The traffic load in telephony systems is typically time varying. Figure 7 shows a pattern concerning call arrivals during a typical 24 hour business day, beginning at midnight [4]. It can be seen that the peak hours occur around 11:00 a.m. and 4:00 p.m. Next, we use our newly trained critic network above to this traffic pattern. Figure 8 gives the simulation results under the assumption that the traffic load is spatially uniformly distributed among cells, and follows the time-varying pattern given in Figure 7. Figure 8 compares the four call admission control algorithm and shows that the self-learning algorithm has the best GoS among all the algorithms tested.

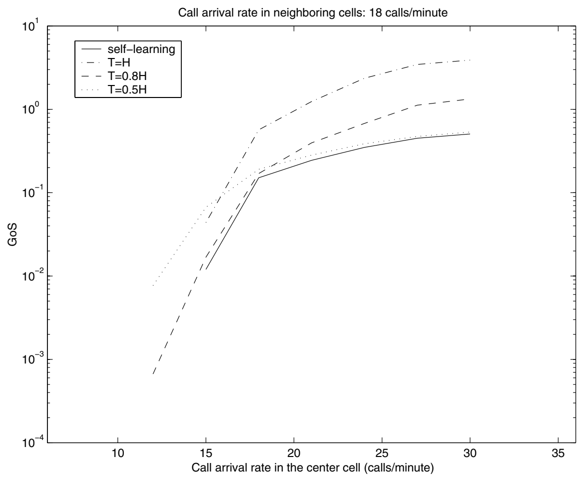

Fig. 6. Comparison study using utility function defined in (10).

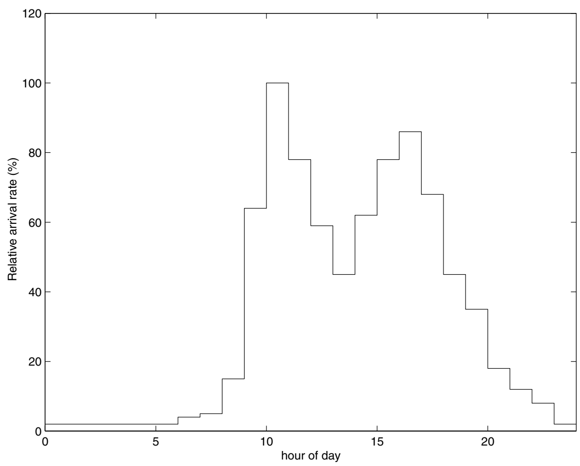

Fig. 7. A traffic pattern of a typical business day.

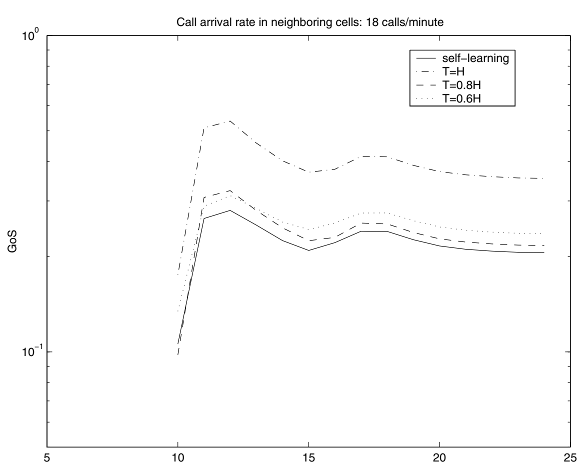

Fig. 8. Comparison study for time-varying traffic.

ACKNOWLEDGMENT

This work was supported by the NSF under Grant ECS-9996428 and Grant ANI-0203063. The authors would like to thank Dr. A. Molchanov for helpful discussions.

REFERENCES

[1] S. N. Balakrishnan and V. Biega, "Adaptive-critic-based neural networks for aircraft optimal control," *J. Guidance*, *Contr.*, *Dynamics*, vol. 19, no. 4, pp. 893–898, July-Aug. 1996.

[2] C. Cox, S. Stepniewski, C. Jorgensen, R. Saeks, and C. Lewis, "On the design of a neural network autolander," *Int. J. Robust Nonlinear Contr.*, vol. 9, no. 14, pp. 1071–1096, Dec. 1999.

[3] Z. Dziong, M. Jia, and P. Mermelstein, "Adaptive traffic admission for integrated services in CDMA wireless-access networks," *IEEE J. Select. Areas Commn.*, vol. 14, pp. 1737–1747, Dec. 1996.

[4] R. L. Freeman, *Telecommunication System Engineering,* 3rd Ed. New York: Wiley, 1996.

[5] K. S. Gilhousen, I. M. Jacobs, R. Padovani, A. J. Viterbi, L. A. Weaver, and C. E. Wheatley III, "On the capacity of a cellular CDMA system," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 303–312, May 1991.

[6] R. A. Guerin, "Channel occupancy time distribution in a cellular radio system," *IEEE Trans. Veh. Technol.*, vol. VT-35, pp. 89–99, Aug. 1987.

[7] D. Han and S. N. Balakrishnan, "State-constrained agile missile control with adaptive-critic-based neural networks," *IEEE Trans. Contr. Syst. Technology*, vol. 10, pp. 481–489, July 2002.

[8] M. S. Iyer and D. C. Wunsch II, "Dynamic re-optimization of a fed-batch fermentor using adaptive critic designs," *IEEE Trans. Neural Networks*, vol. 12, pp. 1433–1444, Nov. 2001.

[9] D. K. Kim and D. K. Sung, "Capacity estimation for an SIR-based power-controlled CDMA system supporting ON-OFF traffic," *IEEE Trans. Veh. Technol.*, vol. 49, pp. 1094–1100, July 2000.

[10] D. Liu and Y. Zhang, "An adaptive call admission control algorithm for SIR-based power-controlled integrated services CDMA cellular networks," *Proceedings of the International Conference on Telecommunications,* Beijing, China, June 2002, pp. 610–638.

[11] D. Liu, X. Xiong, and Y. Zhang, "Action-dependent adaptive critic designs," *Proc. INNS-IEEE Int. Joint Conf. Neural Networks*, Washington, DC, July 2001, pp. 990–995.

[12] Z. Liu and M. E. Zarki, "SIR-based call admission control for DS-CDMA cellular systems," *IEEE J. Select. Areas Commn.*, vol. 12, pp. 638–644, May 1994.

[13] D. V. Prokhorov, R. A. Santiago, and D. C. Wunsch, "Adaptive critic designs: A case study for neurocontrol," *Neural Networks*, vol. 8, no. 9, pp. 1367–1372, 1995.

[14] R. Ramjee, D. Towsley, and R. Nagarajan, "On optimal call admission control in cellular networks," *Wireless Networks*, vol. 3, no. 1, pp. 29–41, 1997.

[15] S. S. Rappaport and C. Purzynski, "Prioritized resource assignment for mobile cellular communication systems with mixed services and platform types," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 443–458, Aug. 1996.

[16] S. M. Shin, C.-H. Cho, and D. K. Sung, "Interference-based channel assignment for DS-CDMA cellular systems," *IEEE Trans. Veh. Technol.*, vol. 48, pp. 233–239, Jan. 1999.

[17] G. K. Venayagamoorthy, R. G. Harley, and D. C. Wunsch II, "Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbogenerator," *IEEE Trans. Neural Networks*, vol. 13, pp. 764–773, May 2002.

[18] A. J. Viterbi, A. M. Viterbi, and E. Zehavi, "Other-cell interference in cellular power-controlled CDMA," *IEEE Trans. Commun.*, vol. 42, pp. 1501–1504, Feb./Mar./Apr. 1994.

[19] P. J. Werbos, "Advanced forecasting methods for global crisis warning and models of intelligence," *General Systems Yearbook*, vol. 22, pp. 25–38, 1977.

[20] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control: Neural, Fuzzy, and Approaches* (Chapter 13), Edited by D. A. White and D. A. Sofge, New York, NY: Van Nostrand Reinhold, 1992.