A Self-Learning Call Admission Control Scheme for CDMA Cellular Networks

Derong Liu, *Fellow, IEEE*, Yi Zhang, *Student Member, IEEE*, and Huaguang Zhang, *Senior Member, IEEE*

Abstract—In the present paper, a call admission control scheme that can learn from the network environment and user behavior is developed for code division multiple access (CDMA) cellular networks that handle both voice and data services. The idea is built upon a novel learning control architecture with only a single module instead of two or three modules in adaptive critic designs (ACDs). The use of adaptive critic approach for call admission control in wireless cellular networks is new. The call admission controller can perform learning in real-time as well as in offline environments and the controller improves its performance as it gains more experience. Another important contribution in the present work is the choice of utility function for the present self-learning control approach which makes the present learning process much more efficient than existing learning control methods. The performance of our algorithm will be shown through computer simulation and compared with existing algorithms.

Index Terms—Adaptive critic designs (ACDs), approximate dynamic programming, call admission control, code division multiple access (CDMA), cellular networks, neural dynamic programming, wireless networks.

I. INTRODUCTION

THE DESIGN of modern wireless networks is based on a cellular architecture that allows efficient use of the limited available spectrum. Call admission control policy is one of the most critical design considerations in wireless networks [6], [18], [27]. On one hand, call admission control schemes provide 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. To the network carriers, high resource utilization usually means high revenue.

Adaptive critic designs (ACDs) were first introduced in the 1970s [34], [35]. ACDs are defined as schemes that *approximate dynamic programming in the general case*, i.e., approximate optimal control over time in nonlinear environments. There are many problems in practice that can be formulated as cost maximization or minimization problems. Examples include error minimization, energy minimization,

profit maximization, and the like. 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 performance index. ACDs have found many applications in learning control problems [2], [5], [11], [13], [23], [31]. Other terms used in the literature for ACDs include neurodynamic programming [4], approximate dynamic programming [35], adaptive dynamic programming [22], and reinforcement learning [28]. Recently, neurodynamic programming and reinforcement learning have been applied to the call admission control problem in (wireline) communication networks [21], [29].

In the present paper, a self-learning call admission control algorithm is developed for signal-to-interference ratio (SIR)-based power-controlled direct-sequence (DS)-code division multiple access (CDMA) cellular networks that provide both voice and data services. The present paper is organized as follows. In Section II, the CDMA wireless system used in the present paper is briefly described. In Section III, the ACD scheme that is suitable for application to call admission control problems is introduced. In Section IV, our self-learning call admission control algorithm for SIR-based power-controlled DS-CDMA networks is developed. The present work will assume the use of artificial neural networks as a means for function approximation in the implementation of ACDs. In particular, multilayer feedforward neural networks are considered, even though other types of neural networks are also applicable in this case. In Section V, the performance of the present algorithm is studied through computer simulation and compared with existing call admission control algorithms. The simulation studies will show that the present self-learning call admission control algorithm outperforms existing call admission control algorithms. Finally, in Section VI, the present paper is concluded with a few pertinent remarks.

II. CDMA WIRELESS SYSTEM DESCRIPTION

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. We consider cellular networks that support both voice and data services. Assume that

there are K classes of services provided by the wireless networks under consideration, where $K \geq 1$ is an integer. We define a mapping $\sigma : Z^+ \rightarrow \{1, \ldots, K\}$ to indicate the fact that the nth connection is from the service class $\sigma(n)$, where Z^+ denotes the set of nonnegative integers. We assume that each connection in our network may be from a different service class that requires a different quality of service (QoS) target (e.g., in terms of different bit error rate for each service class). This includes the case when we allow each call to specify its own QoS requirements. We assume that traffic from the same service class has the same data rate, the same activity factor, the same desired SIR value, and the same maximum power limit that can be received at the base station.

Consider a base station currently with N active connections. The power received at the base station from the user (mobile station) of the nth connection is denoted by $S_n, n = 1, \ldots, N$. In an SIR-based power-controlled DS-CDMA network [1], [6], [15], [27], the desired value of S_n is a function of the number of active home connections and total other cell interference. If we assume that the maximum received power at a base station is limited to H_k for connections from service class $k = \sigma(n)$, then S_n is a random variable in the range of $(0, H_k]$. The maximum power limits $H_k, k = 1, \ldots, K$, are determined by the power limit of mobile transmitters, the cell size, the path loss information, and the user's service class. They have been used in several previous works on call admission control [8], [18], [27].

In CDMA networks, the instantaneous bit SIR (or the bit energy-to-interference ratio) for the nth connection at the base station (in a cell) can be expressed in terms of the received powers of the various connections as [8]

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

where S_n is the instantaneous power level of the nth connection received at the base station, W is the total spread bandwidth (or the chip rate), and $R_{\sigma(n)}$ is the data rate of service class $\sigma(n)$. I_n in (1) indicates the instantaneous total interference to the nth connection received at the base station and it 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 connection which is from the service class $\sigma(i)$, η_n is the background (or thermal) noise, N is the number of active connections in the cell, and f is called the intercell interference factor [33] with a typical value of 0.55. As shown previously, the value of f may not always be constant in a system. Its value can be calculated using existing measurements and can be updated periodically to reflect changes in traffic conditions and traffic distributions.

Assume that after the admission of a new call or a handoff call, the power control algorithm starts to evolve until convergence. Assume that the power control algorithm converges and it requires the power received at a base station from each connection in the system given by $S_n^*, n = 0, 1, \ldots, N$, where the

total number of connections in the system is $N + 1$ and connection 0 is the newly admitted caller. Obviously, if $S_n^* > H_{\sigma(n)}$ or $S_n^* \leq 0$, for some $n, 0 \leq n \leq N$, the admission should not be granted since it leads to an outage. Only when $0 < S_n^* \leq H_{\sigma(n)}$ for all $n, n = 0, 1, \ldots, N$, the admission decision can be considered as a correct decision. The goal of the present study is to develop a self-learning control algorithm that learns to achieve the correct admission decisions under various, possibly changing, environment conditions and user behaviors and to optimize the grade of service (GoS) measure.

The GoS in cellular networks is mainly determined by the new call blocking probability and the handoff blocking probability. The first determines the fraction of new calls that are blocked, while the second is closely related to the fraction of already admitted calls that cannot maintain their required QoS (bit error rate) and are dropped. For example, many works have chosen to use the following definition for the GoS [36]:

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

where $P(a)$ is the probability of event a and w is typically chosen as, e.g., 10. In our simulation studies, we fix $w = 10$. The GoS defined in (2) provides a tradeoff between the new call blocking rate and the handoff call blocking rate. The parameter w is a weighting factor that decides how much emphasis is placed on handoff calls. Keeping the GoS defined in (2) under a desired target level would require to give much higher priority to handoff calls than to new calls when $w = 10$. On the other hand, QoS is usually defined according to the bit error rate in digital transmission. For example, the QoS 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 [8], [18], [27].

For a given set of parameters including traffic statistics and mobility characteristics, fixed call admission control schemes can sometimes yield optimal solutions [24] in terms of GoS. All such schemes [12], [24], [25], [27], however, by reserving a fixed part of capacity, cannot adapt to changes in the network conditions due to their static nature. Therefore, we develop in the present work a self-learning call admission control algorithm for CDMA wireless networks. The present call admission control algorithm based on ACDs has the capability to learn from the environment and the user behavior so that the performance of the algorithm will be improved through further learning.

III. ACDs FOR PROBLEMS WITH FINITE ACTION SPACE

We provide a brief introduction to ACDs in this section [19]. 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. In the present paper, the function F denotes a stochastic transition from the state $x(t)$ to the next state $x(t+1)$ under the given control action $u(t)$ at time

t. Suppose that one associates with this system the performance index

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

where U is called the utility function and γ is the discount factor with $0 < \gamma \le 1$. Note that J is dependent on the initial time i and the initial state $x(i)$, and it is referred to as the cost-to-go of state $x(i)$. The objective is to choose the control sequence $u(k)$, $k = i, i+1, \ldots$, so that the performance index J in (3) is minimized.

The class of ACDs considered in the present paper is called action-dependent heuristic dynamic programming (ADHDP), which is shown in Fig. 1 [17]. The critic network in this case will be trained by minimizing the following error measure over time:

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

where $Q(t)$ is the critic network output at time t. When $E_q(t) = 0$ for all t, (4) implies that

$$Q(t-1) = U(t) + \gamma Q(t)$$
$$= U(t) + \gamma[U(t+1) + \gamma Q(t+1)]$$
$$= \cdots$$
$$= \sum_{k=t}^{\infty} \gamma^{k-t} U(k). \quad (5)$$

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 performance index defined in dynamic programming for $i = t + 1$, i.e., the value of the performance index in the immediate future.

The input–output relationship of the critic network in Fig. 1 is given by

$$Q(t) = Q\left[x(t), u(t), t, W_C^{(p)}\right]$$

where $W_C^{(p)}$ represents the weights of the critic network after the pth weight update. There are two methods to train the critic network according to the error function (4) in the present case which are described in [17]. We will use the so-called forward-in-time method.

We can train the critic network at time $t - 1$, with the output target given by $U(t) + \gamma Q(t)$. The training of the critic network is to realize the mapping given by

$$C_f: \begin{Bmatrix} x(t-1) \\ u(t-1) \end{Bmatrix} \rightarrow \{U(t) + \gamma Q(t)\}. \quad (6)$$

In this case, the output from the network to be trained is $Q(t-1)$ and the input to the network to be trained is composed of $x(t-1)$ and $u(t-1)$. The target output value for the critic network training is calculated using its output at time t as indicated in (6). The goal of learning the function given by (6) is to have the critic network output satisfy

$$Q(t-1) \approx U(t) + \gamma Q(t) \quad \text{for all } t$$
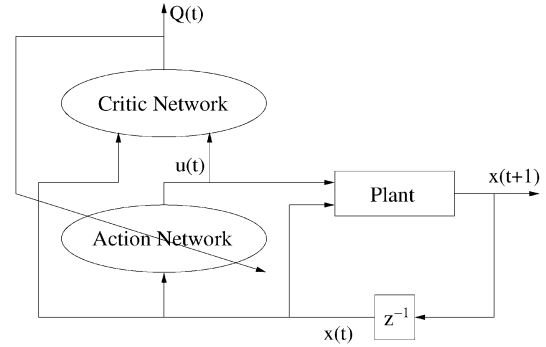

Fig. 1.   Typical scheme of an ADHDP [17].

which is required by (5) for approximating dynamic programming solutions.

The critic network training procedure is described in the following steps using the strategy of [16]:

Step 1)   initialize two critic networks: cnet1 = cnet2;
Step 2)   use cnet2 to get $Q(t)$, and then train cnet1 for 50 epochs using the Levenberg–Marquardt algorithm [10];
Step 3)   copy cnet1 to cnet2, i.e., let cnet2 = cnet1;
Step 4)   repeat Steps 2) and 3), e.g., four times;
Step 5)   repeat Steps 1)–4), e.g., ten times (start from different initial weights);
Step 6)   pick the best cnet1 obtained as the trained critic network.

After the critic network's training is finished, the action network's training starts with the objective of minimizing the critic network output $Q(t)$. In this case, we can choose the target of the action network training as zero, i.e., we will update the action network's weights so that the output of the critic network becomes as small as possible. In general, a good critic network should not output negative values if $U(t)$ is nonnegative. This is particularly true when $U(t)$ is chosen as the square error function in tracking control problems [32]. The desired mapping which will be used for the training of the action network in the Fig. 1 is given by

$$A: \{x(t)\} \rightarrow \{0(t)\} \quad (7)$$

where $0(t)$ indicates the target values of zero. We note that during the training of action network, it will be connected to the critic network as shown in Fig. 1. The target in (7) is for the output of the *whole network*, i.e., the output of *the critic network after it is connected to the action network* as shown in Fig. 1.

There are many problems in practice that have a control action space that is finite. Typical examples include bang-bang control applications where the control signal only takes a few (finite) extreme values or vectors. When the application has only a finite action space, the decisions that can be made are 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 arrives at a base station requesting for admission, the decisions that a base station can make are constrained to two choices, i.e., to accept the call or to reject the call. Let us denote the two options by using $u(t) = +1$ for "accept" and $u(t) = -1$ for "reject." It is important to realize that in the present case the control actions are limited to a binary choice, or to only two possible
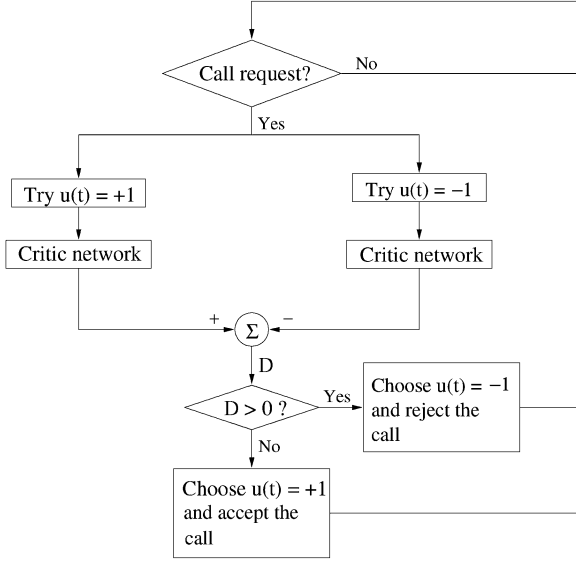
Fig. 2. Block diagram of the present adaptive critic approach.

options. Because of this, the ACDs introduced in Fig. 1 can be further simplified so that only the critic network is needed. Our self-learning call admission control scheme for wireless cellular networks using ACDs is illustrated in Fig. 2. When a new call or a handoff call arrives 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 Fig. 1, the critic network would also take the states of the system as inputs. We note that Fig. 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 Fig. 2 represent the same network or computer code in software. The block diagram in Fig. 2 indicates that the critic network will be used twice in calculations (with different values of $u(t)$) to make a decision about whether or not to accept a call.

The previous description assumes that the critic network has been trained successfully. Once the critic network training is done, it can be applied as in Fig. 2. To guarantee that the overall system will achieve optimal performance now and in future environments which may be significantly different from what they are now, we will allow the critic network to perform further learning when needed in the future. In the next section, we describe how the critic network learning is performed. In particular, we describe how the training data is collected at each time step and how the utility function $U(t)$ is defined. We note that once the training data is collected, the training of the critic network can use, e.g., forward-in-time method, described in this section. We also note that the description here for the critic network training applies to both the initial training of the critic network and further training of the critic network when needed in the future.

IV. Self-Learning Call Admission Control for 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 ad-

mission 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 QoS during the entire call 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 dropping 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 [12], [24], [25], [27]. 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 $n = 0, 1, \ldots, N$. A new call can only be admitted if $0 < S_0^* \leq T_{\sigma(0)}$ and $0 < S_n^* \leq H_{\sigma(n)}$ for $n = 1, 2, \ldots, N$, where $T_{\sigma(0)} < H_{\sigma(0)}$ is the threshold for new calls (where connection 0 is the new caller).

For handoff calls, when $0 < S_n^* \leq H_{\sigma(n)}$ for $n = 0, 1, \ldots, N$, 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. We note that if the power control algorithm leads to either $S_n^* > H_{\sigma(n)}$ or $S_n^* \leq 0$ for some n, the network will enter an outage, i.e., some calls will have to be terminated prematurely since they cannot maintain required QoS (bit error rate). In this case, the action of rejection is given a reward and the action of acceptance is given a penalty. Note that S_n^*, $n = 0, 1, \ldots, N$, are power levels for all connections after the call admission decision and the power control algorithm convergence.

We first define the cost function for handoff calls as follows ($S_n^* \geq 0$ for $n = 0, 1, \ldots, N$):

$$E_n = \xi \max\left\{u(t)\left[\frac{S_n^*}{H_{\sigma(n)}} - 1\right], 0\right\} \tag{8}$$

where $\xi > 0$ is a coefficient and $u(t) = 1$ represents accepting a call and $u(t) = -1$ represents rejecting a call. We emphasize that the conditions, $0 \leq S_n^* \leq H_{\sigma(n)}$ for $n = 0, 1, \ldots, N$, 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 $n = 1, 2, \ldots, N$, 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, $n = 1, 2, \ldots, N$, or $S_n^* \leq 0$ for some $n, n = 0, 1, \ldots, N$, 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 $n = 0, 1, \ldots, N$)

$$E_n = \begin{cases} \xi \max\left\{\left[\frac{S_n^*}{T_{\sigma(n)}} - 1\right], 0\right\}, & \text{when } u(t) = 1 \\ \xi \max\left\{\left[1 - \frac{S_n^*}{H_{\sigma(n)}}\right], 0\right\}, & \text{when } u(t) = -1 \end{cases} \tag{9}$$

where $T_{\sigma(n)} < H_{\sigma(n)}, n = 0, 1, \ldots, N$. We note again that the conditions, $0 < S_n^* \leq H_{\sigma(n)}$ for $n = 0, 1, \ldots, N$, must hold for the entire duration of all calls in order for the system to give reward to the action of accepting a new call, even though
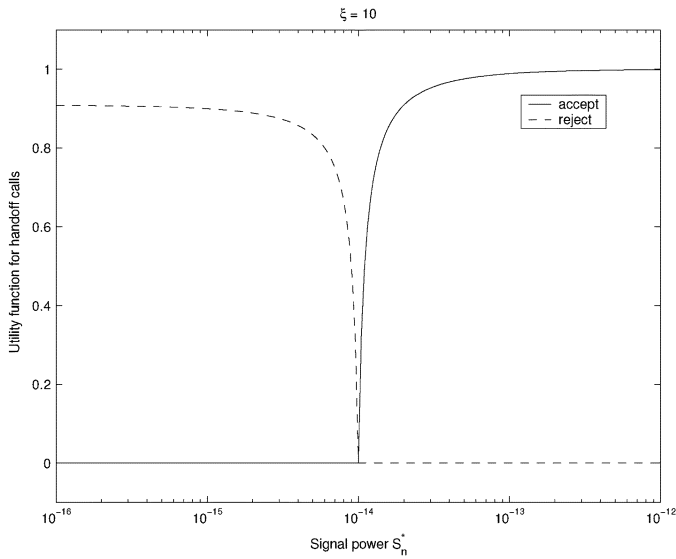
Fig. 3. Utility function for handoff calls.

Fig. 4. Utility function for new calls.

the admission decision is according to the condition $0 < S_0^* \leq T_{\sigma(0)}$ for the new call.

The functions defined previously satisfy the condition that $E_n \geq 0$ for all $n, n = 0, 1, \ldots, N$. From (8) and (9), we can see that when the action is "accept," if the value of the utility function of any connection is larger than 0, this action should be penalized. Also, when the action is "reject," if the value of the utility function of any connection is zero, this action should be rewarded. Therefore, from the system's point of view, the cost function should be chosen as

$$E = \begin{cases} \max_{0 \leq n \leq N}(E_n), & \text{if } u(t) = 1 \\ \min_{0 \leq n \leq N}(E_n), & \text{if } u(t) = -1. \end{cases} \quad (10)$$

The cost function defined in (10) 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

$$U(u) = \frac{E}{1 + E}. \quad (11)$$

Figs. 3 and 4 show plots of this utility function for handoff calls and new calls, respectively, when $\xi = 10$. The parameter ξ is used to obtain a desired shape of the utility function. Since our algorithm will search for the optimal performance that corresponds to small values of the utility function, the shape of the utility function will have some effects on the optimization process. When $\xi = 10$, we can see that the utility function becomes more selective than $\xi = 1$ for any condition indicated by signal power. From Figs. 3 and 4, we see that the choice of the present utility functions in (11) clearly shows minimum points (the flat areas) that our call admission control scheme tries to reach and the points with high penalty that our scheme should avoid. In addition, the conversion from E to U guarantees the convergence of the performance index of dynamic programming, which is defined as in (3). With the present utility function given by (10) and (11), we have
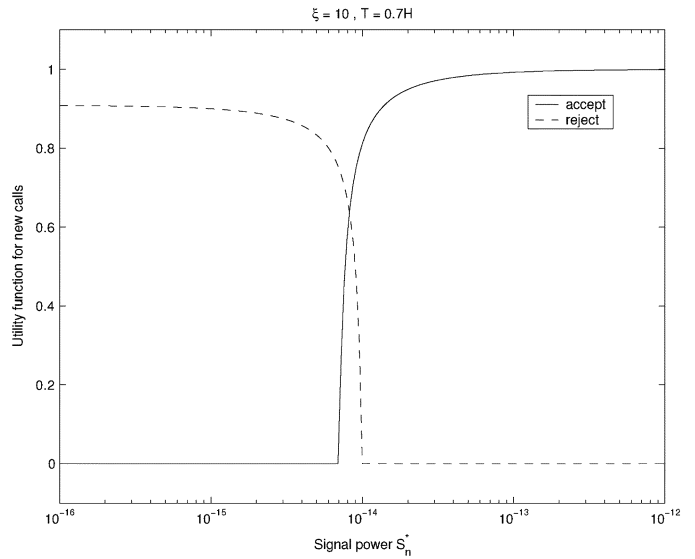
$$0 < J(t) < \frac{1}{1 - \gamma}$$

since U in (11) satisfies $0 \leq U < 1$. Without the conversions in (11), there is no guarantee that the infinite summation in (3) will be bounded. We note that the present critic network produces an output that approximates the performance index $J(t)$ in (3), and the admission action chosen each time will minimize the critic network output, to achieve approximate optimal control.

In stationary environment, where user traffic statistics (patterns) remain unchanged, a simple static call admission control algorithm [20] will be able to achieve the admission objective described previously. However, traffic patterns including user arrival rate, call holding times, user mobility patterns, etc., may show significant changes from time to time. To deal with changing environments, static call admission control algorithm would not be appropriate. The present call admission control algorithm based on ACDs will be able to deal with environment changes through further learning in the future. Another benefit of the present self-learning call admission control algorithm is its ability to improve performance through further learning as the controller gains more and more experience.

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

Step 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 previously. 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.

Step 2) Training critic network: Using the data collected to train the critic network as mentioned in the previous section. Examples of input variables chosen for the critic network will be given in our simulation examples.

Step 3) Applying critic network: The trained critic network is then applied as shown in Fig. 2.

Step 4) Further updating critic network: The critic network will be updated as needed while it is used in application to accommodate environment changes, for example, user pattern and behavior changes or new requirements for the system. Data collection has to be performed again and the training of critic network as well. In this case, the previous three steps will be repeated.

The critic network will be updated when there are changes in call admission requirements or if the already trained ACD scheme does not satisfy new requirements. In fact, ACD is going to learn the rules imposed by the utility function of the system. Therefore, rule changes can be accommodated by modifying the utility function to accommodate new requirements. For example, to satisfy certain requirements, we can modify the cost function in (9) to become

$$
E_n = \begin{cases} \xi \max\left\{\left[\frac{S_n^*}{H_{\sigma(n)}} - 1\right], 0\right\}, & \text{when } u(t) = 1 \\ & \text{and } n_a \leq N_h \\ \xi \max\left\{\left[\frac{S_n^*}{T_{\sigma(n)}} - 1\right], 0\right\}, & \text{when } u(t) = 1 \\ & \text{and } n_a > N_h \\ \xi \max\left\{\left[1 - \frac{S_n^*}{H_{\sigma(n)}}\right], 0\right\} & \text{when } u(t) = -1 \end{cases} \quad (12)
$$

where n_a is the number of active handoff calls in the cell and N_h is a fixed parameter indicating the threshold for low traffic load.

When a call arrives at the base station, the admission decision would be either "accept" or "reject." If the decision is to accept a call, there will be two kinds of data collected. One is that the decision is incorrect due to call dropping and the other one is that the decision is correct since $0 < S_n^* \leq H_{\sigma(n)}, n = 0, 1, \ldots, N$, is maintained for the entire duration of all calls. In the former case, a penalty is recorded and in the latter case, a reward is recorded. If the decision is to reject a call, there will be also be two kinds of data collected that correspond to a reward and a penalty. Note that in the case of a "reject" decision, the value of the utility function is determined as in (8)–(11) where the values of $S_n^*, n = 0, 1, \ldots, N$, are calculated according to [3], [20], and [30].

V. SIMULATION RESULTS

We first 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 [15] and [27] (see Table I).

The arrival rate consists of the new call attempt rate λ_c and the handoff call attempt rate λ_h. The new call attempt rate λ_c depends on the expected number of subscribers per cell. The handoff call attempt rate λ_h depends on such network parameters as traffic load, user velocity, and cell coverage areas [9], [12]. In our simulation, we assume that $\lambda_c : \lambda_h = 5 : 1$ [12]. A channel is released by call completion or handoff to a neighboring cell. The channel occupancy time is assumed to be exponentially distributed [9], [12] with the same mean value of

TABLE I
NETWORK PARAMETERS

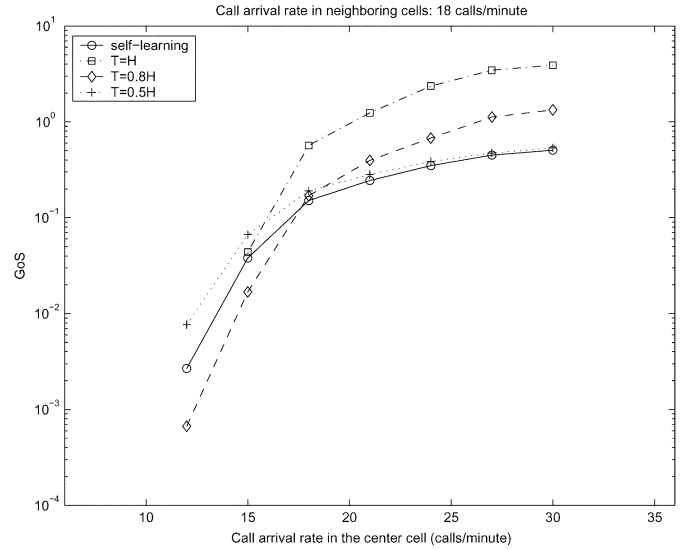| Parameters	Values	Parameters	Values
W	1.2288 Mcps	R	9.6 kbps
η	1×10^{-14} W	H	1×10^{-14} W
E_b/N_0	7 dB	v	3/8

Fig. 5. Comparison study using utility function defined in (9).

$1/\mu = 3$ min. For each neural network training in our simulation studies, we generated 35 000 data points according to the data collection procedure in the previous section.

In the following, we conduct comparison studies between the present self-learning call admission control algorithm and the algorithm developed in [20] 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/min. The training data is collected as mentioned in the previous section. We choose $T_{\sigma(n)} = 0.5H_{\sigma(n)}$ and $\xi = 10$ in (9). The critic network has three inputs. The first is the total interference received at the base station, the second is the action (1 for accepting, -1 for rejecting), and the third is the call type (1 for new calls, -1 for handoff calls). The critic network is a multilayer feedforward neural network with 3-6-1 structure, i.e., three neurons at the input layer, six neurons at the hidden layer, and one neuron at the output layer. Both the hidden and output layers use the hyperbolic tangent function as the activation function. Fig. 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 (9) 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 as a result 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 this case, the self-learning algorithm is trained to learn a call admission control scheme that gives higher priority to handoff calls.
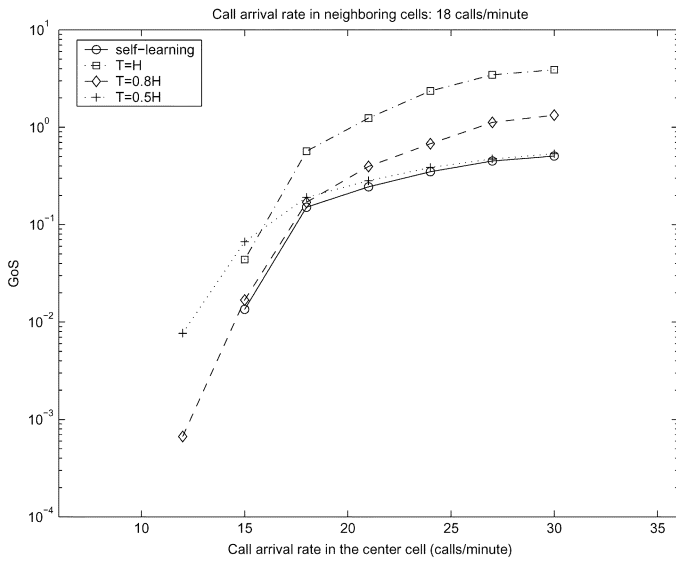
Fig. 6. Comparison study using utility function defined in (12).

Fig. 7. Traffic pattern of a typical business day.

Fig. 8. Comparison study for varying traffic.

In order to improve the GoS when the call arrival rate is low, we use the modified cost function for new calls as in (12), where we choose $N_h = 15$ in our simulation. Using this new utility function we collect the training data using one of the static algorithms with fixed threshold or the previous critic network. Then we train a new critic network with the newly collected data. This time the 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 $n_a \leq N_h$ and otherwise it is equal to -1. The critic network in this case has a structure given by 4-8-1. Fig. 6 shows the result of applying the new critic network to the same traffic pattern as in Fig. 5. From Fig. 6, we see that the self-learning algorithm using the new critic network has the best GoS. We can see that by simply changing the cost function from (11) to (12), the self-learning algorithm can significantly improve its performance to outperform static admission control algorithms. One of the benefits of self-learning call admission control algorithm is that we can easily and efficiently design call admission control algorithms by modifying the cost function (equivalently, the utility function) to satisfy the requirements or to accommodate new environment changes.

The traffic load in telephony systems is typically time varying. Fig. 7 shows a pattern concerning call arrivals during a typical 24 hour business day, beginning at midnight [7]. It can be seen that the peak hours occur around 11:00 am and 4:00 pm. Next, we use our newly trained critic network above to this traffic pattern. Fig. 8 gives the simulation results under the assumption that the traffic load was spatially uniformly distributed among cells, but followed the time-varying pattern given in Fig. 7. Fig. 8 compares the four call admission control algorithms and shows that the self-learning algorithm has the best GoS among all the algorithms tested. We note that the self-learning call admission control algorithm was not retrained from the previous case, i.e., we used the same critic network in the simulation results of Fig. 8 as in Fig. 6.

In the following, we conduct comparison studies between the present self-learning call admission control algorithm and that of [27]. 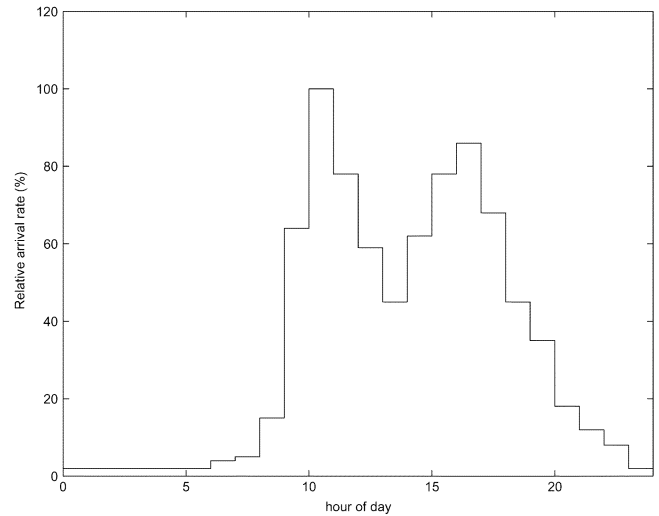Using the algorithm in [27], the base station controller reads the current interference from the power strength measurer. It then estimates the current interference margin (CIM) and handoff interference margin (HIM), where CIM < HIM. A total interference margin (TIM) is set according to the QoS target. If CIM > TIM, reject the call admission request. If HIM < TIM, accept the call request. If CIM < TIM < HIM, then only handoff calls will be accepted. Fig. 9 compares the present self-learning call admission control algorithm with the algorithm in [27] that reserves 1, 2, 3 channels for handoff calls, respectively. The arrival rate in all neighboring cells is fixed at 18 calls/min. We assume the use of hexagonal cell structure. From Fig. 9, we see that the present algorithm has the best GoS. That is because the algorithm in [27] is a kind of guard channel algorithm used in CDMA systems. Therefore, when the load is low, GC = 1 performs the best, and when the load is high, GC = 3 performs the best. However, our algorithm can adapt to varying traffic load conditions. It has the best overall performance under various traffic loads. Again, we used the same critic network in the simulation results of Fig. 9 as in Fig. 6.
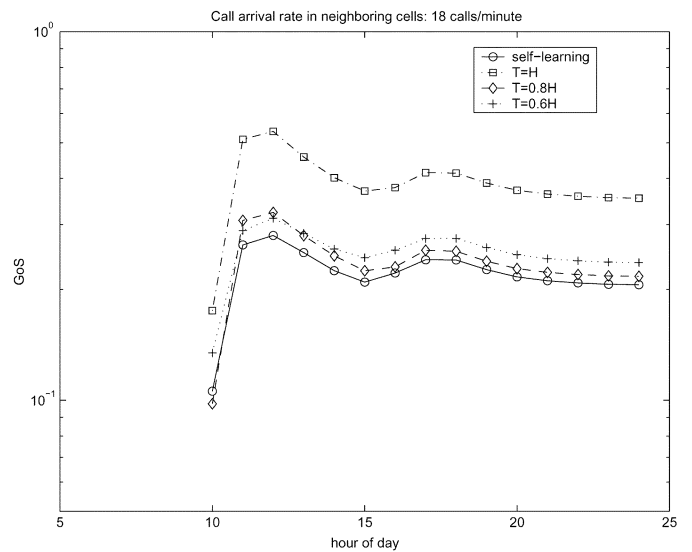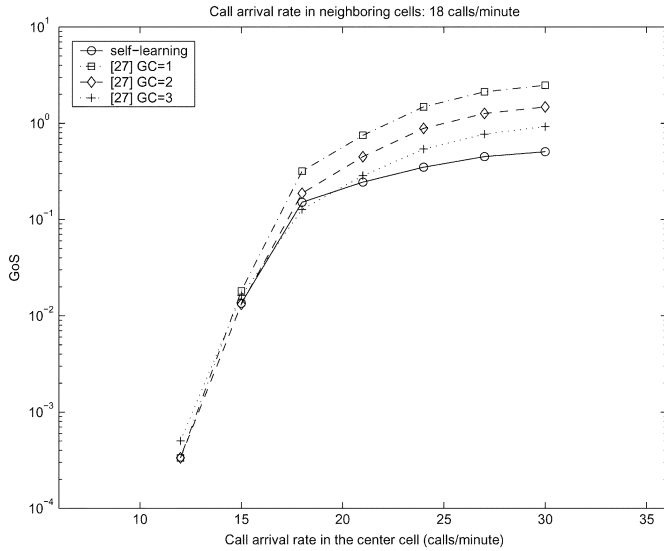
Fig. 9.   Comparison studies with the algorithm in [27].

Fig. 10.   GoS for voice calls.

TABLE II
NETWORK PARAMETERS

Voice users		Data users	
Parameters	Values	Parameters	Values
W_v	4.9152 Mcps	W_d	4.9152 Mcps
R_v	9.6 kbps	R_d	38.4 kbps
H_v	1×10^{-14} W	H_d	1×10^{-13} W
$(E_b/N_0)_v$	7 dB	$(E_b/N_0)_d$	9 dB
ν_v	3/8	ν_d	1

Finally, we conduct simulation studies for cellular networks with two classes of services. One class is voice service and the other is data service. Network parameters in our simulations are chosen in reference to the parameters used in [14] and [26] (see Table II). In our simulation, the data traffic is similar to that in [14], i.e., low-resolution video or interactive data. In this case, the data traffic can be specified by a constant transmission rate. The background noise in this case is chosen the same as in Table I. The utility function is defined for voice and data calls as in (9) and (11). In (12), we choose $T_{\sigma(n)} = 0.6H_{\sigma(n)}$ and $\xi = 10$ for both voice calls and data calls. N_h is chosen as 20 and 4 for voice calls and data calls, respectively. The critic network now has five inputs. The newly added input is the call class which is 1 for voice calls and -1 for data calls. The critic network structure is chosen as 5-10-1. Figs. 10 and 11 compare between our self-learning call admission control algorithm and the static algorithm [20] with fixed thresholds given by $T = H$ and $T = 0.8H$, respectively. The arrival rates of voice users and data users in all neighboring cells are fixed at 20 calls/min and 3 calls/min, respectively. From Figs. 10 and 11, we see that the present self-learning algorithm has the best GoS for almost all call arrival rates tested. We can conclude that the present self-learning algorithm performs better than the fixed algorithms due to the fact that the self-learning algorithm can adapt to varying traffic conditions and environment changes.
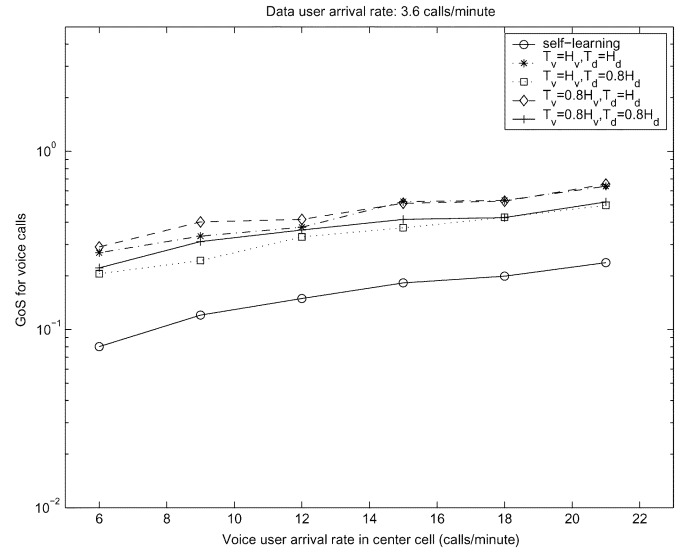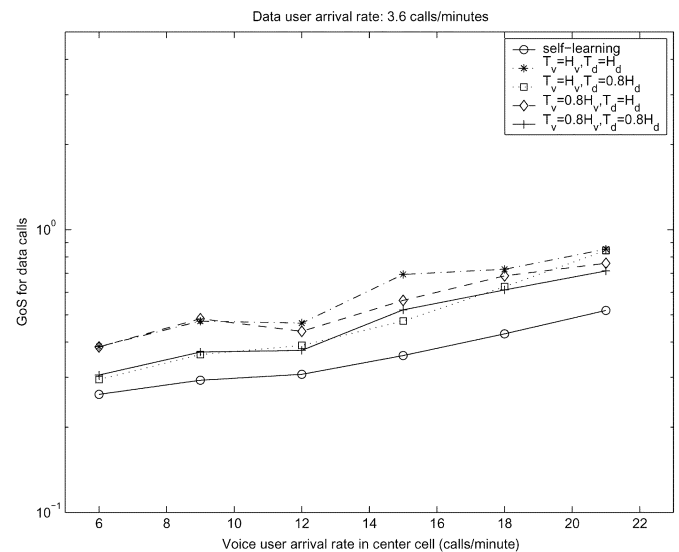

Fig. 11.   GoS for data calls.

VI. CONCLUSION

In this paper, we developed a self-learning call admission control algorithm based on ACDs for multiclass traffic in SIR-based power-controlled DS-CDMA cellular networks. The most important benefit of our self-learning call admission control algorithm is that we can easily and efficiently design call admission control algorithms to satisfy the system requirement or to accommodate new environments. We note that changes in traffic conditions are inevitable in reality. Thus, fixed call admission control policies are less preferable in applications. Our simulation results showed that when traffic condition changes, self-learning call admission control algorithm can adapt to changes in the environment, while fixed admission policy will suffer either from higher new call blocking rate, higher handoff call blocking rate, or higher interference than the tolerance.

REFERENCES

[1] S. Ariyavisitakul, "Signal and interference statistics of a CDMA system with feedback power control—Part II," *IEEE Trans. Commun.*, vol. 42, pp. 597–605, Feb.-Apr. 1994.

[2] S. N. Balakrishnan and V. Biega, "Adaptive-critic-based neural networks for aircraft optimal control," *J. Guid. Control*, vol. 19, no. 4, pp. 893–898, Jul.–Aug. 1996.

[3] N. Bambos, S. C. Chen, and G. J. Pottie, "Channel access algorithms with active link protection for wireless communication networks with power control," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 583–597, Oct. 2000.

[4] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

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

[6] Z. Dziong, M. Jia, and P. Mermelstein, "Adaptive traffic admission for integrated services in CDMA wireless-access networks," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 9, pp. 1737–1747, Dec. 1996.

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

[8] 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, no. 2, pp. 303–312, May 1991.

[9] R. A. Guerin, "Channel occupancy time distribution in a cellular radio system," *IEEE Trans. Veh. Technol.*, vol. VT-35, no. 3, pp. 89–99, Aug. 1987.

[10] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994.

[11] D. Han and S. N. Balakrishnan, "State-constrained agile missile control with adaptive-critic-based neural networks," *IEEE Trans. Contr. Syst. Technol.*, vol. 10, no. 4, pp. 481–489, Jul. 2002.

[12] D. Hong and S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures," *IEEE Trans. Veh. Technol.*, vol. VT-35, no. 3, pp. 77–92, Aug. 1986.

[13] M. S. Iyer and D. C. Wunsch II, "Dynamic re-optimization of a fed-batch fermentor using adaptive critic designs," *IEEE Trans. Neural Netw.*, vol. 12, no. 6, pp. 1433–1444, Nov. 2001.

[14] Y. W. Kim, D. K. Kim, J. H. Kim, S. M. Shin, and D. K. Sung, "Radio resource management in multiple-chip-rate DS/CDMA systems supporting multiclass services," *IEEE Trans. Veh. Technol.*, vol. 50, no. 3, pp. 723–736, May 2001.

[15] 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, no. 4, pp. 1094–1100, Jul. 2000.

[16] G. G. Lendaris and C. Paintz, "Training strategies for critic and action neural networks in dual heuristic programming method," in *Proc. 1997 Int. Conf. Neural Networks*, Houston, TX, Jun. 1997, pp. 712–717.

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

[18] Z. Liu and M. E. Zarki, "SIR-based call admission control for DS-CDMA cellular systems," *IEEE J. Sel. Areas Commun.*, vol. 12, no. 4, pp. 638–644, May 1994.

[19] D. Liu and Y. Zhang, "A new learning control approach suitable for problems with finite action space," in *Proc. Int. Conf. Control and Automation*, Xiamen, China, Jun. 2002, pp. 1669–1673.

[20] D. Liu, Y. Zhang, and S. Hu, "Call admission policies based on calculated power control setpoints in SIR-based power-controlled DS-CDMA cellular networks," *Wireless Netw.*, vol. 10, no. 4, pp. 473–483, Jul. 2004.

[21] P. Marbach, O. Mihatsch, and J. N. Tsitsiklis, "Call admission control and routing in integrated services networks using neuro-dynamic programming," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 2, pp. 197–208, Feb. 2000.

[22] J. J. Murray, C. J. Cox, G. G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 32, no. 2, pp. 140–153, May 2002.

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

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

[25] 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, no. 3, pp. 443–458, Aug. 1996.

[26] A. Sampath and J. M. Holtzman, "Access control of data in integrated voice/data CDMA systems: Benefits and tradeoffs," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 8, pp. 1511–1526, Oct. 1997.

[27] 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, no. 1, pp. 233–239, Jan. 1999.

[28] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[29] H. Tong and T. X. Brown, "Adaptive call admission control under quality of service constraints: A reinforcement learning solution," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 2, pp. 209–221, Feb. 2000.

[30] V. V. Veeravalli and A. Sendonaris, "The coverage-capacity tradeoff in cellular CDMA systems," *IEEE Trans. Veh. Technol.*, vol. 48, no. 5, pp. 1443–1450, Sep. 1999.

[31] 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 Netw.*, vol. 13, no. 3, pp. 764–773, May 2002.

[32] N. A. Visnevski and D. V. Prokhorov, "Control of a nonlinear multivariable system with adaptive critic desings," in *Proc. Conf. Artificial Neural Networks in Engineering*, St. Louis, MO, Nov. 1996, pp. 559–565.

[33] 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.-Apr. 1994.

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

[35] *Handbook of Intelligent Control: Neural, Fuzzy, and Approaches*, D. A. White and D. A. Sofge, Eds., Van Nostrand, New York, 1992. P. J. Werbos, Approximate dynamic programming for real-time control and neural modeling.

[36] *A Guide to DECT features that influence the traffic capacity and the maintenance of high radio link transmission quality, including the results of simulations*, http://www.etsi.org, Jul. 1992.

Derong Liu (S'91–M'94–SM'96–F'05) received the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, in 1994, the M.S. degree in electrical engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1987, and the B.S. degree in mechanical engineering from the East China Institute of Technology (now Nanjing University of Science and Technology), Nanjing, China, in 1982.

From 1982 to 1984, he was a Product Design Engineer at China North Industries Corporation, Jilin, China. From 1987 to 1990, he was an Instructor in the Graduate School of the Chinese Academy of Sciences, Beijing, China. From 1993 to 1995, he was a Staff Fellow at General Motors Research and Development Center, Warren, MI. From 1995 to 1999, he was an Assistant Professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ. He joined the University of Illinois at Chicago in 1999 as an Assistant Professor of Electrical Engineering and Computer Science, where he is now an Associate Professor of Electrical and Computer Engineering, of Bioengineering, and of Computer Science. He is a coauthor (with A. N. Michel) of the books *Dynamical Systems with Saturation Nonlinearities: Analysis and Design* (New York: Springer-Verlag, 1994) and *Qualitative Analysis and Synthesis of Recurrent Neural Networks* (New York: Marcel Dekker, 2002). He is a coeditor (with P. J. Antsaklis) of the book *Stability and Control of Dynamical Systems with Applications* (Boston, MA: Birkhauser, 2003).

Dr. Liu was a Member of the Conference Editorial Board of the IEEE Control Systems Society (1995–2000), served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-I: FUNDAMENTAL THEORY AND APPLICATIONS (1997–1999), and the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2001–2003). Since 2004, he has been an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS. He is the Program Chair for the 21st IEEE International Symposium on Intelligent Control (2006) and 2006 International Conference on Networking, Sensing and Control, and he has served and is serving as a member of the organizing committee and the program committee of several international conferences. He was recipient of the Michael J. Birck Fellowship from the University of Notre Dame (1990), the Harvey N. Davis Distinguished Teaching Award from Stevens Institute of Technology (1997), and the Faculty Early Career Development (CAREER) award from the National Science Foundation (1999). He is a Member of Eta Kappa Nu.

Yi Zhang (S'04) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Chicago in 2004 and the M.S. degree in electrical engineering form Zhejiang University, Hangzhou, China, in 1999.

He is currently with the Department of Electrical and Computer Engineering, University of Illinois at Chicago. His current research interests include wireless communication, signal processing, and wireless circuit design.

Huaguang Zhang (SM'04) was born in Jilin, China, in 1959. He received the B.S. and M.S. degrees in control engineering from Northeastern Electric Power University of China, Jilin, in 1982 and 1985, respectively, and the Ph.D. degree in thermal power engineering and automation from Southeast University of China, Nanjing, in 1991.

He entered the Automatic Control Department, Northeastern University, Shenyang, China, in 1992 as a Postdoctoral Fellow. Since 1994, he has been a Professor and Head of the Electric Automation Institute, Northeastern University. His main research interests are fuzzy control, chaos control, neural networks based control, nonlinear control, signal processing, and their applications.

Dr. Zhang received the "Distinguished Young Scientist Award" from the National Natural Science Foundation of China in 2003.