Proceedings of the American Control Conference
Chicago, Illinois • June 2000

CONVERGENCE ANALYSIS OF ADAPTIVE CRITIC BASED OPTIMAL CONTROL

Xin Liu[1], S. N. Balakrishnan[2]
Department of Mechanical and Aerospace Engineering and Engineering Mechanics
University of Missouri-Rolla
Rolla, MO 65409-0050
email:bala@umr.edu, xliu@umr.edu

Abstract

Adaptive critic based neural networks have been found to be powerful tools in solving various optimal control problems. The adaptive critic approach consists of two neural networks which output the control values and the Lagrangian multipliers associated with optimal control. These networks are trained successively and when the outputs of the two networks are mutually consistent and satisfy the differential constraints, the controller network output produces optimal control. In this paper, we analyze the mechanics of convergence of the network solutions. We establish the necessary conditions for the network solutions to converge and show that the converged solution is optimal.

1 Introduction

The adaptive critic based optimal control methodology comprises of successive adaptations of two neural networks, namely "action" and "critic" neural network (which approximate the discrete Hamilton - Jacobi Bellman (HJB) equation associated with optimal control theory) until closed loop optimal control is achieved. In our previous study we have used this methodology to solve linear and even nonlinear problems [1][2][13][14], some other people have also contributed to this research area[11]. Although these papers have shown impressive results, so far there is no analysis on the mechanics of the method. In other words, the conditions for the convergence of this method (on which the success or the use of this method depends) has not been established. In this study, we establish the conditions for the convergence of infinite time (regulator) problems. Similar efforts has been done by S. J. Bradtke to find optimal policy based on Q-learning [6]. The conditions for the convergence of the individual networks during the iterative process are derived. The conditions for the successive training processes to reach optimal control is also derived.

[1]Graduate student
[2]Professor, Aerospace Engineering (contact person)

0-7803-5519-9/00 $10.00 © 2000 AACC

2 Background

2.1 Adaptive Critic Based Optimal Control

Optimal controller designs seek to accomplish some desired objectives by minimizing a pre-defined cost functional, and simultaneously, satisfying some boundary conditions and constraints. The cost or performance index is expressed by a mathematical expression in terms of the system variables and controls. For the problems discussed in this paper, the cost function is chosen to be in a quadratic form which is used in most applications. It is:

$$I(x_0, u) = \sum_{k=0}^{\infty} [x_k^T Q x_k + u_k^T R u_k] \qquad (1)$$

where, x is the state vector, u is the control vector and k is the time index. In Eq. (1), Q is a symmetric positive semi-definite matrix and R is a symmetric positive definite matrix. The choice of these matrices is a design decision to give different degrees of importance to the state trajectory and the control effort. The optimal control problem can now be stated as - minimize the cost function I given in Eq. (1) with the differential constraints of the state equation:

$$x_{k+1} = f(x_k, u_k) \qquad (2)$$

when the initial state $x(0)$ is given and final time $t_f \to \infty$. In our study this problem is solved through approximate dynamic programming formulation.

2.2 Approximate Dynamic Programming Method

Dynamic programming provides a computational technique to apply the principle of optimality to sequence of decisions which define an optimal control policy. A general mathematical description of the optimality conditions obtained as a direct convergence of the "Principle of Optimality" is the Hamilton-Jacobi Bellman (HJB) equation[10]. The HJB equation for a discrete-time system is given by:

$$J(x_k) = \min_{u(x_k)} \{U(x_k, u(x_k)) + < J(f(x_k, u(x_k))) >\} \qquad (3)$$

where, state at time step k is given by x_k and the control by $u(x_k)$, $J(x_k)$ represents the minimum cost asso-

ciated with going from k to final step N, $U(x_k, u(x_k))$ is the utility function denoting the cost incurred in going from k to $k+1$ using control $u(x_k)$ and $J(x_{k+1})$ is the minimum cost associated in going from state $k+1$ to final state N. Now a co-state (or Lagrangian multiplier) $\lambda(x_k)$ is defined as $\lambda(x_k) = \frac{\partial J(x_k)}{\partial x_k}$, then differentiate Eq.(3) with respect to x_k, (Note, we also use $u_k = u(x_k)$ and $\lambda_k = \lambda(x_k)$ in this paper):

$$\begin{aligned}\lambda^*(x_k) &= \frac{\partial U(x_k, u(x_k))}{\partial x_k} + \frac{\partial J(f(x_k, u(x_k)))}{\partial x_k} \\ &= \frac{\partial U(x_k, u(x_k))}{\partial x_k} + \left(\frac{\partial u(x_k)}{\partial x_k}\right)^T \frac{\partial U(x_k, u(x_k))}{\partial x_k} \\ &\quad + \left(\frac{\partial x_{k+1}}{\partial x_k} + \frac{\partial x_{k+1}}{\partial u_k}\frac{\partial u(x_k)}{\partial x_k}\right)^T \lambda(x_{k+1}) \quad (4)\end{aligned}$$

It can be seen that the co-state equation develops backwards in time. The Bellman's optimality equation is given by:

$$0 = \frac{\partial J(x_k)}{\partial u_k} = \frac{\partial U(x_k, u_k)}{\partial u_k} + \frac{\partial J(x_{k+1})}{\partial u_k} \quad (5)$$

Dynamic programming requires the system model and its derivatives in (4) and (5). These equations are used iteratively to solve for a control policy.

2.3 Adaptive Critic and General Training Procedure

Adaptive critic methodology has been proposed by Werbos[9] as a new optimization tool combining together concepts of reinforcement learning and approximate dynamic programming. Adaptive critic approach consists of two neural networks: one outputs the control u_k and the other outputs Lagrangian multiplier λ_k. Input to both are the states, x_k, at time k. The adaptive critic technique finds the control which minimizes the cost in (1) by solving Eq. (2) and Eq. (4) with the use of the optimality Eq. (5) and the known initial state.

The training procedure consists of two training cycles: *critic's* and *action's*. The action neural network outputs the control $u(x_k)$ for the state input x_k. The output of the plant x_k serve as input to the critic neural network which is trained to estimate the cost function J, or the *Hamiltonian* H, or their derivatives (we use derivatives in this paper) and so on. Thus the critic neural network contains information about the function to be minimized. The optimal control can be obtained by training action neural network and critic neural network successively.

3 Mathematical Analysis of the Convergence Conditions

There are a few papers in the literature that show the effectiveness of this approach to solve different optimal control problems[1][2][11][13][14]. However, they have not dealt with the operational mechanics of the networks. Do the action neural network and critic neural network converge in each training cycle? If they do, what are the conditions? Does such a training procedure eventually find the optimal control sequence? In order to find the answers to these questions, mathematical analysis of the convergence conditions is necessary. We develop such a procedure here with respect to a linear problem. The linear time-invariant, discrete, multi-variable system is given by the state equations:

$$x_{k+1} = f(x_k, u_k) = Ax_k + Bu_k \quad (6)$$

Where initial state $x(0) = x_0$, $x_k \in \Omega \subset \Re^p$, $u_k \in \Re^m$, $k = 1, 2, 3, \cdots$, $A \in \Re^{p\times p}$, $B \in \Re^{p\times m}$. The cost functional is defined to be functional (1). Utility in Eq. (3) is defined to be:

$$U(x_k, u(x_k)) = \frac{1}{2}(x_k^T Q x_k + u(x_k)^T R u(x_k)) \quad (7)$$

3.1 Convergence of λ_k Iterations

If we define $\lambda_k^* = \frac{\partial J(x_k)}{\partial x_k}$ (Lagrangian multiplier), then with the definition of the system model and utility co-state Eq. (4) can be rewritten as:

$$\lambda_k^* = Qx_k + \left(\frac{\partial u(x_k)}{\partial x_k}\right)^T Ru_k + \left(A + B\frac{\partial u(x_k)}{\partial x_k}\right)^T \lambda_{k+1} \quad (8)$$

Let $\lambda_k = g(x_k)$, $\lambda_k^* = g^*(x_k)$ and consider $\lambda_{k+1} = g(x_{k+1})$ in infinite horizon, Eq. (8) can be written as:

$$\begin{aligned}g^*(x_k) &= Qx_k + \left(\frac{\partial u(x_k)}{\partial x_k}\right)^T Ru_k \\ &\quad + \left(A + B\frac{\partial u(x_k)}{\partial x_k}\right)^T g(x_{k+1}) \quad (9)\end{aligned}$$

Now assume $g(x_k) = g_n(x_k)$, $g^*(x_k) = g_{n+1}(x_k)$ to get a more explicit iterative form of Eq. (9):

$$\begin{aligned}g_{n+1}(x_k) &= Qx_k + \left(\frac{\partial u(x_k)}{\partial x_k}\right)^T Ru_k \\ &\quad + \left(A + B\frac{\partial u(x_k)}{\partial x_k}\right)^T g_n(x_{k+1}) \quad (10)\end{aligned}$$

Where $x_{k+1} = Ax_k + Bu_k$

Claim: The convergent condition for the sequence $g_n(x_k)$ is:

$$\left\| A + B\frac{\partial u(x_k)}{\partial x_k} \right\| < 1, \quad \forall x_k \in \Omega \subset \Re^p \quad (11)$$

Proof: From Eq. (10), we can obtain (with $n = n-1$):

$$\begin{aligned}g_n(x_k) &= Qx_k + \left(\frac{\partial u(x_k)}{\partial x_k}\right)^T Ru_k \\ &\quad + \left(A + B\frac{\partial u(x_k)}{\partial x_k}\right)^T g_{n-1}(x_{k+1}) \quad (12)\end{aligned}$$

The subtraction of Eq. (12) from Eq. (10) leads to:

$$g_{n+1}(x_k) - g_n(x_k) = \left(A + B\frac{\partial u(x_k)}{\partial x_k}\right)^T \times (g_n(x_{k+1}) - g_{n-1}(x_{k+1})) \quad (13)$$

We can go on expanding Eq. (13) in this way, with $n = n-1, n-2, \cdots, 1$ to get:

$$g_{n+1}(x_k) - g_n(x_k) = \left(A + B\frac{\partial u(x_k)}{\partial x_k}\right)^T \times \left(A + B\frac{\partial u(x_{k+1})}{\partial x_{k+1}}\right)^T \cdots \left(A + B\frac{\partial u(x_{k+n-1})}{\partial x_{k+n-1}}\right)^T \times (g_1(x_{k+n-1}) - g_0(x_{k+n-1})) \quad (14)$$

Let the maximum norm of matrix $A + B\frac{\partial u(x_j)}{\partial x_j}$ $(j = k, k+1, \cdots, k+n-1)$ be M then:

$$\|g_{n+1}(x_k) - g_n(x_k)\| \leq M^n \|g_1(x_{k+n-1}) - g_0(x_{k+n-1})\| \quad (15)$$

Hence, if $M < 1$ and both $g_1(x_{k+n-1})$ and $g_0(x_{k+n-1})$ are bounded, then sequence $g_n(x_k)$ will converge. Since the condition that both $g_1(x_{k+n-1})$ and $g_0(x_{k+n-1})$ are bounded can always be guaranteed from the initial guess, the condition for the convergence is thus $\|A + B\frac{\partial u(x_j)}{\partial x_j}\| < 1$. This completes the proof.

3.2 Convergence of u_k Iterations

Considering the optimality condition, we can rewrite Eq. (5) as following:

$$0 = \frac{\partial U(x_k, u_k)}{\partial u_k} + \frac{\partial J(x_{k+1})}{\partial u_k} = Ru_k^* + B^T g(x_{k+1}) \quad (16)$$

Hence, we can get u_k and $\frac{\partial u(x_k)}{\partial x_k}$ (we will use its expression (18) later) if we can solve the following two equations:

$$u_k^* = -R^{-1}B^T g(x_{k+1}),\ x_{k+1} = Ax_k + Bu_k \quad (17)$$

$$\frac{\partial u^*(x_k)}{\partial x_k} = -R^{-1}B^T\left(\frac{\partial g(x_{k+1})}{\partial x_{k+1}}\right)\left(A + B\frac{\partial u(x_k)}{\partial x_k}\right) \quad (18)$$

It is easy to see that both Eq. (17) and Eq. (18) should also be solved iteratively. If we define $f_n(x_k) = u(x_k)$, $f_{n+1}(x_k) = u^*(x_k)$, $f_n'(x_k) = \frac{\partial u(x_k)}{\partial x_k}$ and $f_{n+1}'(x_k) = \frac{\partial u^*(x_k)}{\partial x_k}$, then explicit iterative forms of Eq. (17) and Eq. (18) are:

$$f_{n+1}(x_k) = -R^{-1}B^T g(Ax_k + Bf_n(x_k)) \quad (19)$$

$$f_{n+1}'(x_k) = -R^{-1}B^T g'(x_{k+1})(A + Bf_n'(x_k)) \quad (20)$$

Let's consider Eq. (19) first.

Claim: The convergence condition for the sequence $f_n(x_k)$ is:

$$\|R^{-1}B^T g'(x_{k+1})B\| < 1 \quad (21)$$

Proof: From Eq. (19):

$$f_n(x_k) = -R^{-1}B^T g(Ax_k + Bf_{n-1}(x_k)) \quad (22)$$

Subtracting Eq. (22) from Eq. (19), we get:

$$f_{n+1}(x_k) - f_n(x_k) = -R^{-1}B^T(g(Ax_k + Bf_n(x_k))) - g(Ax_k + Bf_{n-1}(x_k)) \quad (23)$$

By using mean value theorem:

$$g(Ax_k + Bf_n(x_k)) - g(Ax_k + Bf_{n-1}(x_k)) = g'(\xi)B(f_n(x_k) - f_{n-1}(x_k)) \quad (24)$$

Where $\xi \in [x_{n-1}(k+1), x_n(k+1)]$. Now, we have:

$$f_{n+1}(x_k) - f_n(x_k) = -R^{-1}B^T g'(\xi)B(f_n(x_k) - f_{n-1}(x_k)) \quad (25)$$

The convergence condition for Eq.(25) is $\|R^{-1}B^T g'(\xi)B\| < 1$. Since $\xi \in [x_{n-1}(k+1), x_n(k+1)]$, we can use $g'(x_{k+1})$ to represent $g'(\xi)$, i.e. the convergence condition for sequence $f_n(x_k)$ is $\|R^{-1}B^T g'(x_{k+1})B\| < 1$. This completes the proof.

3.3 Discussion on Relaxation of Convergence Condition

The inequality in Eq. (21) maybe very restrictive, because in many cases we can not guarantee $g'(x_k)$ to be small enough to satisfy this condition. Therefore, we need to find some way to relax this condition without affecting the convergence of the algorithm. For this purpose, consider Eq. (3), the gradient of $J(x_k)$ over u_k can be explained as:

$$\nabla_{u_k} J(x_k) = \nabla_{u_k} U(x_k, u_k) + \nabla_{u_k} J(x_{k+1}) = Ru_k + B^T\lambda_{k+1} \quad (26)$$

Multiply both sides of Eq. (26) by the inverse of matrix R, we get:

$$R^{-1}\nabla_{u_k} J(x_k) = u_k + R^{-1}B^T\lambda_{k+1} \quad (27)$$

and

$$R^{-1}B^T\lambda_{k+1} = -u_k + R^{-1}\nabla_{u_k} J(x_k) \quad (28)$$

Eq. (28) can be used to substitute the right hand side of Eq. (17) to obtain:

$$u_k^* = u_k - R^{-1}\nabla_{u_k} J(x_k) \quad (29)$$

We can assume $u_k^* = u_{n+1}$, $u_k = u_n$ in the above equation to get an explicit iterative form. For our problem, a learning rate α can be added to the last term in Eq. (29):

$$u_{n+1} = u_n - \alpha R^{-1}\nabla_{u_n} J(x_k) \quad (30)$$

Substitute $\nabla_{u_n} J(x_k)$ in Eq. (30) with Eq. (26):

$$\begin{aligned} u_{n+1} &= u_n - \alpha R^{-1}(Ru_n + B^T\lambda_{k+1}) \\ &= (1-\alpha)u_n - \alpha R^{-1}B^T\lambda_{k+1} \\ &= (1-\alpha)u_n - \alpha u^*,\ u_n = u_n(x_k) \end{aligned} \quad (31)$$

If we differentiate both sides of Eq. (31), we get:

$$f'_{n+1}(x_k) = -\alpha R^{-1}B^T g'(x_{k+1})A - ((1-\alpha)I - \alpha R^{-1}B^T g'(x_{k+1})B)f'_n(x_k) \quad (32)$$

Now, the condition for series $f'_n(x_k)$ to converge is:

$$\|(1-\alpha)I - \alpha R^{-1}B^T g'(x_{k+1})B\| < 1 \quad (33)$$

where $I \in \Re^{m\times m}$ is an identity matrix. This condition also guarantees the convergence of sequence $f_n(x_k)$. Hence, if α is properly chosen, the convergence condition can be relaxed.

3.4 Guaranteed Convergence of the Successive Iteration Towards Optimality

We have derived the convergence conditions for λ_k and u_k. Now we will consider the convergence of the successive iteration procedure. To solve both Eq. (10) and Eq. (19), we need to differentiate Eq. (10) first:

$$g'_{n+1}(x_k) = Q + \left(\frac{\partial^2 u(x_k)}{\partial x_k^2}\right)^T Ru_k + \left(\frac{\partial u(x_k)}{\partial x_k}\right)^T R\frac{\partial u(x_k)}{\partial x_k} + \left(A + B\frac{\partial u(x_k)}{\partial x_k}\right)^T g'_n(x_{k+1})\left(A + B\frac{\partial u(x_k)}{\partial x_k}\right) + \left(\frac{\partial^2 u(x_k)}{\partial x_k^2}\right)^T B^T g_n(x_{k+1}) \quad (34)$$

For linear problem, we can always assume an initial control which is a linear function of x_k, then the second derivative of u_k in Eq. (34) is zero for this given initial condition, and Eq. (34) reduces to:

$$g'_{n+1}(x_k) = Q + \left(\frac{\partial u(x_k)}{\partial x_k}\right)^T R\left(\frac{\partial u(x_k)}{\partial x_k}\right) + \left(A + B\frac{\partial u(x_k)}{\partial x_k}\right)^T g'_n(x_{k+1}) \times \left(A + B\frac{\partial u(x_k)}{\partial x_k}\right) \quad (35)$$

Eq. (35) can be solved iteratively, given the initial guess of the controller $u(x_k)$ and the initial condition of co-state function $g(x_k)$.

Claim: The convergence of the Eq. (35) is guaranteed given condition (11).

Proof: According to the given initial condition, $\frac{\partial u(x_k)}{\partial x_k}$ is a constant. Then we can substitute $\frac{\partial u(x_k)}{\partial x_k}$ in Eq. (35) with a constant C (scalar or vector, depends on the system formulation). Let $A + BC = S_c$, Then Eq. (35) can be reduced to:

$$g'_{n+1}(x_k) = Q + C^T RC + S_c^T g'_n(x_{k+1})S_c \quad (36)$$

The equation for the previous time step is:

$$g'_n(x_{k+1}) = Q + C^T RC + S_c^T g'_{n-1}(x_{k+2})S_c \quad (37)$$

Substitute $g'_n(x_{k+1})$ in Eq. (36) with Eq. (37), we get:

$$\begin{aligned} g'_{n+1}(x_k) &= (Q + C^T RC) + S_c^T[Q + C^T RC \\ &\quad + S_c^T g'_n(x_{k+1})S_c]S_c \\ &= (Q + C^T RC) + S_c^T(Q + C^T RC)S_c \\ &\quad + \left(S_c^T\right)^2 g'_{n-1}(x_{k+2})S_c^2 \end{aligned} \quad (38)$$

We can go on expanding in this way in order to obtain an expression as:

$$g'_{n+1}(x_k) = (Q + C^T RC) + S_c^T(Q + C^T RC)S_c + ... + \left(S_c^T\right)^{n+1} g'_0(x_{k+n+1})S_c^{n+1} \quad (39)$$

The last term in Eq. (39) goes to zero as $n \to \infty$ under the condition $\|S_c\| < 1$. Therefore, $g'_{n+1}(x_k)$ will converge to $g'(x_k)$ eventually, and it is easy to prove that $g'(x_k)$ is determined by equation :

$$g'(x_k) = \left(A + B\frac{\partial u(x_k)}{\partial x_k}\right)^T g'(x_k)\left(A + B\frac{\partial u(x_k)}{\partial x_k}\right) + Q + \left(\frac{\partial u(x_k)}{\partial x_k}\right)^T R\left(\frac{\partial u(x_k)}{\partial x_k}\right) \quad (40)$$

Where $C = \frac{\partial u(x_k)}{\partial x_k}$. This completes the proof.

It is easy to see that Eq. (40) is a discrete algebraic Lyapunov equation and can be solved with different methods [7]. Therefore, $g'(x_k)$ doesn't change with x_k, i.e. $g(x_k)$ is a linear function of x_k and the next control calculated from Eq. (17) is also a linear function of x_k, since R and B are constants. Hence, the second derivative of u_k in Eq. (34) is zero during the whole training procedure and Eq. (35) holds all the time.

Now, we can go back to discuss Eq. (20). Expand Eq. (20) to get the following form:

$$f'_{n+1}(x_k) = -R^{-1}B^T g'(x_{k+1})A - R^{-1}B^T g'(x_{k+1})Bf'_n(x_k) \quad (41)$$

Then if $g'(x_{k+1})$ is known, the sequence of $f'_n(x_k)$ will converge to $f'(x_k)$ under the condition that $\|R^{-1}B^T g'(x_{k+1})B\| < 1$. Let $f'_m(x_k)$ be the output of the m^{th} iteration of Eq. (41) and $g'_{m-1}(x_{k+1})$ be the output of the $(m-1)^{th}$ iteration of Eq. (35), then

$$f'_m(x_k) = -(R + B^T g'_{m-1}(x_{k+1})B)^{-1}B^T g'_{m-1}(x_{k+1})A \quad (42)$$

Rewrite Eq. (40) in term of $f'_m(x_k)$:

$$g'_m(x_k) = (A + Bf'_m(x_k))^T g'_m(x_k)(A + Bf'_m(x_k)) + Q + f'_m(x_k)^T Rf'_m(x_k) \quad (43)$$

(Note: index m here represents different iterative procedure from that of index n, it represents the iteration between (41) and (35), not inside each of Eq. (41) and Eq. (35)) Now we can combine Eq. (42) and Eq. (43) together and solve them iteratively to find $g'(x_k)$ and $f'(x_k)$. Since $g'(x_k)$ and $f'(x_k)$ do not change with x_k and $x_k = 0$ is the equilibrium point of the system model, we can find u_k^* and λ_k^* from Eq. (8) and Eq. (17).

Now the question here is whether the iteration algorithm (defined by Eq. (42) and Eq. (43)) converges to the optimal solution we want or not. The answer is yes. In fact this algorithm is nothing else but an

iterative method to solve the discrete algebraic Riccati equation:

$$P = A^T PA - A^T PB(R + B^T PB)^{-1} B^T PA + Q \quad (44)$$

and the algorithm presented here converges in a neighborhood of the steady state at a rate that is quadratic. The convergence of this algorithm under the condition that $\left\| A + B\frac{\partial u_0(x_k)}{\partial x_k} \right\| < 1$, i.e. choose the initial condition such that the feedback control system is stable, is proved by Gary A. Hewer [3] and David L. Kleinman [4]. Gary A. Hewer also showed in his paper[3] that the control derived from every iteration of the algorithm satisfies the condition $\left\| A + B\frac{\partial u_m(x_k)}{\partial x_k} \right\| < 1$, $m > 0$, given a stable initial control. Hence, The condition for the convergence of Eq. (10) is automatically satisfied if a stable initial control is set, and the only condition which need to be satisfied so as for Eq. (19) and for the iterative algorithm to converge is condition (33). In practice, we can obtain the optimal control in the form of:

$$u_k^* = -R^{-1} B^T g^*(x_{k+1}) \quad (45)$$

4 Conclusions

By analyzing the performance of the critic neural network and action neural network, we find that the training of critic neural network (which outputs the Lagrangian multiplier) is to follow an iterative procedure which is bound to converge based on certain condition that can be easily satisfied. Similarly, we have established the condition for the convergence of action (controller) neural network. We have provided a reformulation to relax convergence condition for action (controller) neural network too, this is more useful in practice. Furthermore, we have proved that the successive adaptation of these neural networks will converge to produce optimal control.

Acknowledgment: This research was supported by NSF grant No. ECS9976588.

References

[1] S. N. Balakrishnan, V. Biega, "Adaptive critic based neural networks for aircraft optimal control," Journal of Guidance, Control, and Dynamics, Vol. 19, No. 4, July-August 1996, pp. 893-898

[2] G. Saini, S. N. Balakrishnan, "Adaptive critic based neurocontroller for autolanding of aircrafts, " Proceedings of the 1997 American Control Conference. June 2-4, 1997 Albequerqul, NM. vol.2, pp.1081−1085

[3] G. A. Hewer, "An Iterative Technique for the Computation of the Steady State Gains for the Discrete Optimal Regulator", IEEE Trans. Automat. Contr. Vol AC-16, pp. 382-384, Aug. 1997.

[4] David L. Kleinman, "Stabilizing a Discrete, Constant, Linear System with Application to Iterative Methods for Solving the Riccati Equation ", IEEE Trans. Automat. Contr. Vol. AC−19, no.3, June 1974, pp.252−254.

[5] David L. Kleinman, "On an Iterative Technique for Riccati Equation Computations", IEEE Trans. Automat. Contr. (Corresp.), Feb. 1968, pp.114-115.

[6] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, "Adaptive Linear Quadratic Control Using Policy Iteration", Proc. Am. Contr. Conf., Baltimore, MD, June 1994, pp. 3475-3479

[7] Zoran Gajic, Muhammad Tahir Javed Qureshi, *Lyapunov Matrix Equation in System Stability and Control*, Vol. 195 in Mathematics in Science and Engineering, 1995

[8] Ian R. Petersen, Christopher V. Hollot, "A Riccati Equation Approach to the Stabilization of Uncertain Linear Systems", Automatica, Vol. 22, No. 4, pp. 397−411, 1986

[9] D.A. White and D.A.Sofge, Eds., *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*, New York, NY, Van Nostrand Reinhold, 1992

[10] R. Boudarel, J.Delmas, P.Guichet, *Dynamic Programming and Its Application to Optimal Control*, New York, NY, Academic Press, 1971

[11] Danil Prokhorov, Don Wunsch, "Adaptive critic Designs", IEEE Trans. on Neural Networks, Sept. 1997, pp. 997−1007

[12] A. E. Bryson, Y. Ho, *Applied Optimal Control*, Hemisphere Publishing Co., 1975

[13] Han, D., S. N. Balakrishnan, "Adaptive critic based neural networks for agile missile control, " AIAA Guidance, Navigation, and Control Convergence and Exhibit, Boston, Massachusetts. August 10-12, 1998, pp. 1803-1812.

[14] Han, D., S. N. Balakrishnan, "Robust adaptive critic based neural networks for speed-constrained agile missile control, " AIAA Guidance, Navigation, and Control Convergence and Exhibit, Portland, Oregon. August 8-10, 1999