Proceedings of International Conference on Neural Networks
(ICNN'97), Houston, TX, USA, 1997, vol. 4, pp. 2273-2278.

Asymptotic Dynamic Programming: Preliminary Concepts and Results[1]

Richard E. Saeks, Chadwick J. Cox, Karl Mathia, and Alianna J. Maren

Accurate Automation Corporation

7001 Shallowford Road
Chattanooga, Tennessee 37421
Phone: 423-894-4646

Abstract

A formal setting for the development of adaptive critic techniques is established in a nonlinear dynamical systems setting and some preliminary stability and suboptimality theorems are developed. Two alternative versions of the theory are developed, one with a known plant and one with an initial unknown plant which must be identified on-line.

1. Introduction

Consider a nonlinear system of the form

$$x_{i+1} = f(x_i, u_i) \quad : \quad x_0 = \underset{\sim}{x} \qquad \textbf{(1)}$$

$x_i \in X \subseteq R^n$, $u_i \in U \subseteq R^m$ where X and U are appropriate state and input spaces, respectively, and Equation 1 has a "fixed point at zero", i.e, $f(0, 0) = 0$. We desire to find a stabilizing *feedback controller* $k \in K$ such that, $u_i = k(x_i)$, minimizes the performance measure

$$J = \sum_{i=0}^{\infty} l(x_i, u_i) \qquad \textbf{(2)}$$

where $l{:}X \times U \to R^+$ is greater than or equal to zero with equality if and only if $x = 0$ and $u = 0$. Here, K is an appropriate space of feedback controllers which includes any desired stability constraints on the closed loop system.

The well know dynamic programming solution to this problem is obtained via the solution of the Bellman equation[1,2,3],

$$V(x_i) = \min_{\kappa \varepsilon K} [l(x_i, \kappa(x_i)) + V(f(x_i, \kappa(x_i)))] \qquad \textbf{(3)}$$

Here, $V{:}X \to R^+$ is the *optimal cost function* for the problem, i.e., V($\underset{\sim}{x}$) is the minimal achievable J for the optimal control problem with initial state $\underset{\sim}{x}$; while the optimal feedback law, k, is the minimizing κ in the Bellman equation. Moreover, the resultant feedback system, $x_{i+1} = f(x_i, k(x_i))$, is asymptotically stable.

[1] This research supported in part by National Science Foundation NSF Contract DMI-932164.

0-7803-4122-8/97 $10.00©1997 IEEE

While atheistically pleasing, the dynamic programming solution to the optimal control program is computationally untenable due to the backwards numerical process required for its solution - the so-called "curse of dimensionality". Over the past decade a number of researchers have attempted to circumvent these difficulties by computing a sequence of "critics", V_i, forward in time, which approximate V in the limit. A corresponding sequence of feedback laws, k_i, which minimize

$$[l(x_i, k_i(x_i)) + V_i(f(x_i, k_i(x_i)))] \quad \textbf{(4)}$$

over the constraint set, K, is computed and used to define the feedback law, $u_i = k_i(x_i)$, and closed loop system, $x_{i+1} = f(x_i, k_i(x_i))$. Although these techniques, which are collectively termed *adaptive critic methods*, have often yielded excellent results, little progress has been made in verifying that they produce a stabilizing control law and are optimal or sub-optimal in some sense. The purpose of this note is to provide a formal background and partial results in support of this methodology. In the following, we consider two alternative forms of the asymptotic dynamic programming problem; with a known plant $f{:}XxU \to X$ as described above, and with an unknown plant which is identified on-line, in parallel with the asymptotic dynamic programming process.

For the case of a known plant, the asymptotic dynamic programming process is described by the diagram of Figure 1 and the following iterative process.

1. Initialization: Initialize the asymptotic dynamic programming process with x_o and k_o (and V_o if required by the algorithm used in step 3).

2. Run System: Input x_i and $u_i = k_i(x_i)$ into the system and run it one time step computing x_{i+1}.

3. Principle of Optimality: Choose V_{i+1} to minimize the error, ε_p, between $V_{i+1}(x_j)$ and

$$l(x_j, k_j(x_j)) + V_{i+1}f(x_j, k_j(x_j)) \quad \textbf{(5)}$$

$$j = 0 \ldots 1$$

4. Optimization: Choose k_{i+1} in K to minimize

$$\varepsilon_o = l(x_{i+1}, k_{i+1}(x_{i+1})) + V_{i+1}(f_i(x_{i+1}, k_{i+1}(x_{i+1}))) \quad \textbf{(6)}$$

5. Increment Time Step: Increment i and go to 2.

Note: we do not specify the numerical processes to be used in steps 3 and 4. Rather, our goal is simply to show that the asymptotic dynamic programming process yields a stable "asymptotically optimal" control if these processes converge sufficiently rapidly. Although neural network methods are most commonly applied in 3 and 4, any other appropriate algorithm can be applied, classical optimization, reinforcement learning, etc.

In the case where the plant, f, is initially unknown we add an extra step to the above algorithm, identifying f by approximating it with a sequence of models, $f_i{:}XxU \to X$ in parallel with the above described asymptotic dynamic programming process. This process is illustrated in Figure 2 and employs the following iterative process.

1. Initialization: Initialize the asymptotic dynamic programming process with x_o and k_o (and f_o and V_o if required by the algorithm used in steps 3 and 4).

2. Run System: Input x_i and $u_i = k_i(x_i)$ into the system and run it one time step computing x_{i+1}.

3. Identification: Choose f_i to minimize the error, ε_i, between x_{j+1} and $f_i(x_j, k_j(x_j); j=0 \ldots i$.

4. Principle of Optimality: Choose V_{i+1} to minimize the error, ε_p, between $V_{i+1}(x_j)$ and

$$l(x_j, k_j(x_j)) + V_{i+1}(f_i(x_j, k_j(x_j))){:}(j = 0) \quad \textbf{(7)}$$

5. Optimization: Choose k_{i+1} in K to minimize

$$\varepsilon_o = l(x_{i+1}, k_{i+1}(x_{i+1})) + V_{i+1}(f_i(x_{i+1}, k_{i+1}(x_{i+1}))) \quad \textbf{(8)}$$

6. Increment Time Step: Increment i and go to 2.

As above, the numerical processes to be used in steps 3, 4 and 5 are unspecified, and, our goal is simply to show that the algorithm yields a stable

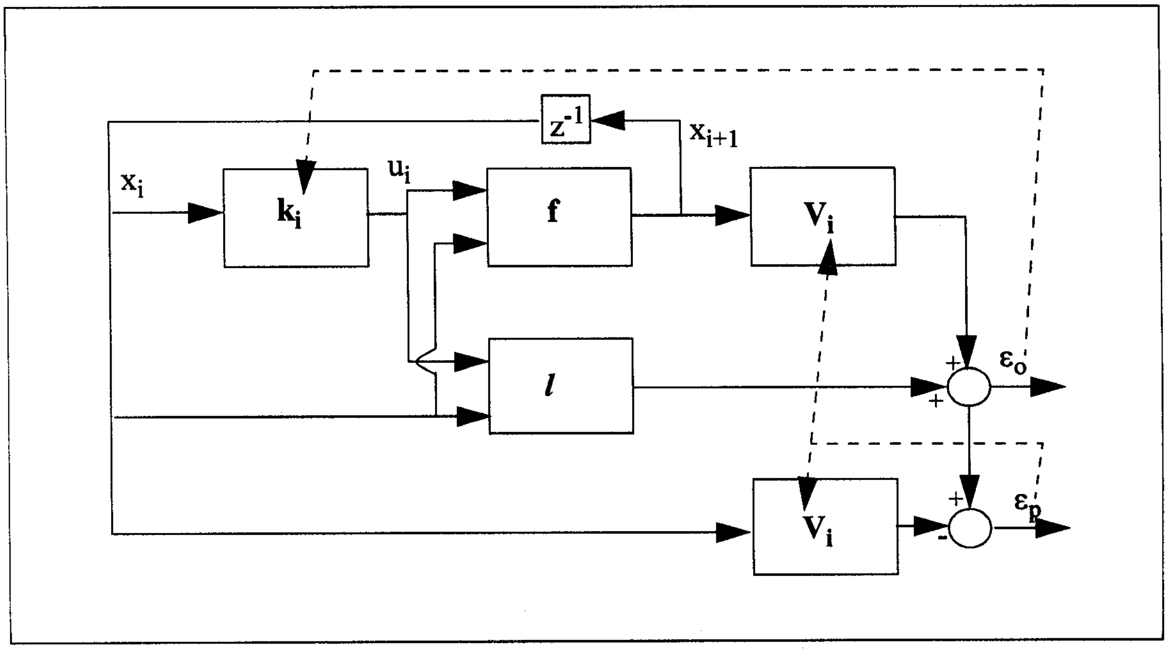

Figure 1: Asymptotic Dynamic Programming with Known Plant.

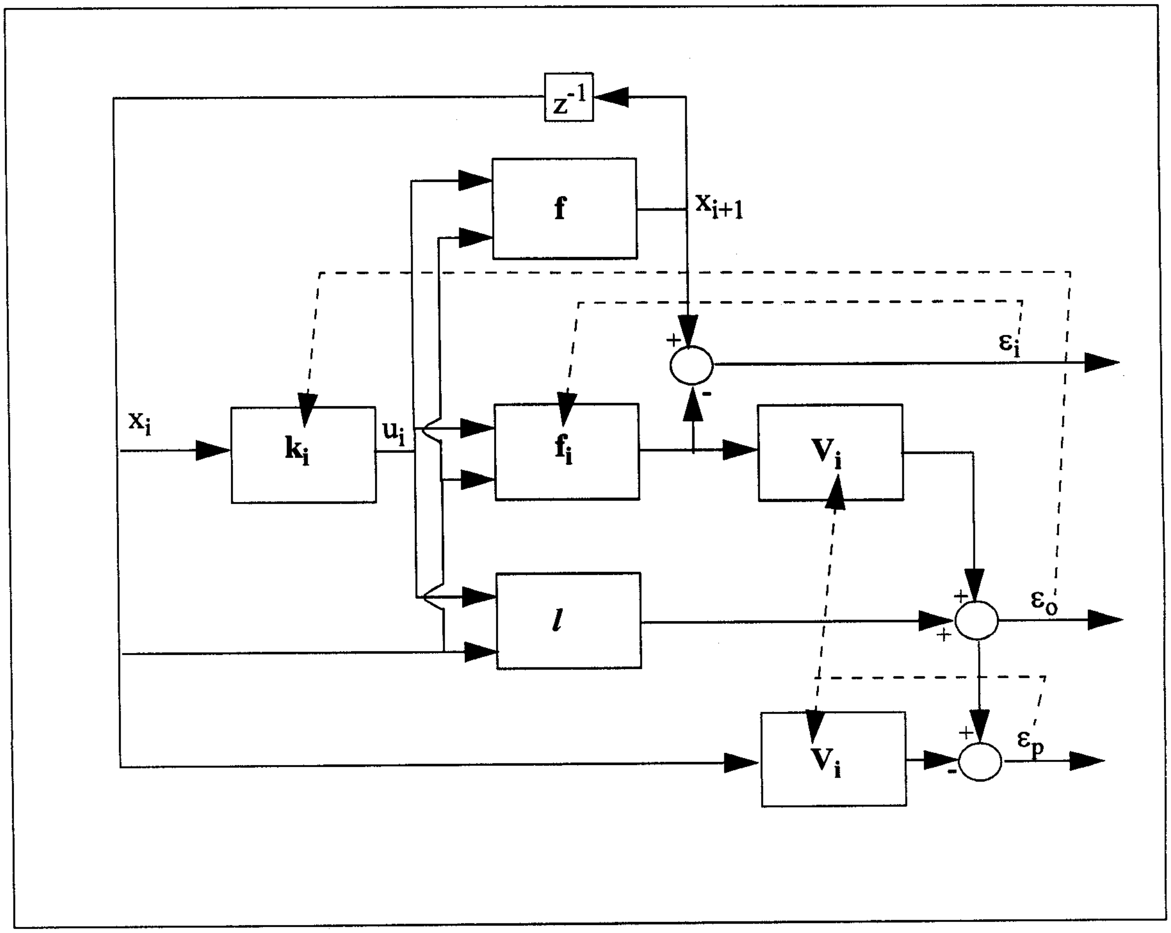

Figure 2: Asymptotic Dynamic Programming with Unknown Plant.

"asymptotically optimal" control if these processes converge sufficiently rapidly.

Since the Asymptotic Dynamic Programming algorithm is designed for on-line implementation it is essential to guarantee that the algorithm always produces a stabilizing control **even if it fails to converge** to an optimal or suboptimal solution. To achieve this goal, we adopt a much more stringent stability criterion than would otherwise be required. In particular, rather than requiring that the system be stable relative to *some Lyapunov function,* we require stability relative to a *prescribed Lyapunov function of norm type.* I.e., we select a norm, $| \ |$, and say that the dynamical system, $x_{k+1} = h(x_k)$, is stable if there exists an N, such that for $k > N$, $|x_{k+1}| \leq |x_k|$. Similarly, we say that it is asymptotically stable if $|x_{k+1}| < |x_k|$ and exponentially stable if $|x_{k+1}| \leq \gamma \, |x_k|$, $\gamma < 1$.

We require that the feedback law constructed at each time step in our asymptotic dynamic programming algorithm be *exponentially stable with respect to some (arbitrarily) prescribed norm,* $| \ |$ and a fixed γ. To emphasize that we require exponential stability with respect to a prescribed Lyapunov function of norm type, we denote the feedback law constraint set by $K_{| \ |}$ instead of K in the remainder of the paper.

The requirement that the Lyapunov function be of "norm type" is minor. The standard norm derived from the Lyapunov equation for a linear system is always of norm type, as are most of the Lyapunov functions used in nonlinear analysis. Reference 4 gives a constructive technique for computing a Lyapunov function of norm type, which is widely applicable to both discrete time and continuous time nonlinear dynamical systems.

The requirement that the Lyapunov function be fixed a-priori, is, however, non-trivial, and represents a constraint on the stable dynamics of the resultant feedback system. For instance, if one takes $| \ |$ to be the classical euclidean norm, the closed loop trajectories of the system are required to "decrease towards zero" are each time step as illustrated in Figure 3a while the system shown in Figure 3b, whose trajectories periodically diverge from zero, is stable with respect to the Tchebyshev norm but not the euclidean norm.

Once a stability concept has been specified via the norm, $| \ |$, we use this norm for all of our computations on the state space $X \subseteq R^n$ and employ the norms induced by $| \ |$ on the space of functions mapping the state space to itself

$$|g| = \sup_{x \neq 0} \frac{|g(x)|}{|x|} \qquad (9)$$

while we define two induced norms on the space of functionals mapping the state space to the reals, the gain norm,

$$|u| = \sup_{x \neq 0} \frac{|u(x)|}{|x|} \qquad (10)$$

and the Lipschitz norm,

$$|u|^{L} = \sup_{x \neq y} \frac{|u(x-y)|}{|x-y|} \qquad (11)$$

Clearly, $|g(x)| \leq |g| \, |x|$ and $|u(x)| \leq |u| \, |x|$ for all $x \in X$ and $|u(x-y)| \leq |u|^{L} |x-y|$ for all $x \neq y \in X$. With the exception of the Lipschitz norm, we use the same notation for all of the above norms distinguishing between them by context.

2. Preliminary Results

In the following, we formulate our basic asymptotic dynamic programming results for both the case of a known plant and an unknown plant. In both cases, stability results are obtained. With a known plant, the algorithm is guaranteed to produce a stable control even if it does not converge. If the Principle of Optimality and Identification approximations converge sufficiently rapidly the resultant control is asymptotically optimal in an appropriate sense.

To simplify the notation in the following analysis, we define the mapping, B, from the space of cost functionals on X to itself by

$$B(U)(.) = \min_{\kappa \varepsilon K_{| \, |}} \qquad (12)$$

$$[(l(. , \kappa(.)) + U(f(. , \kappa(.))))]$$

In our asymptotic dynamic programming algorithm, $B(V_i)$ is computed by the optimization step in the algorithm while the error between V_i and $B(V_i)$ is

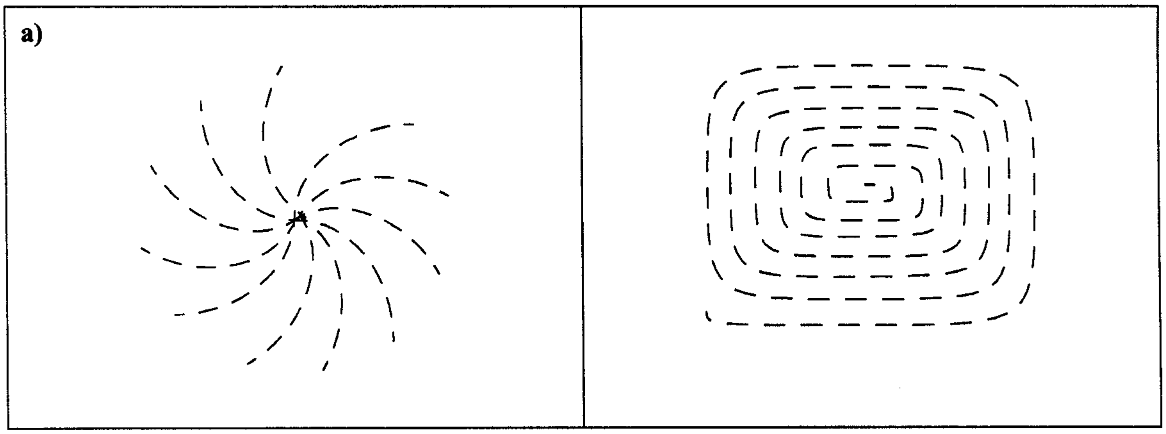

Figure 3: Systems which are stable with respect to a) euclidean and b) Tchebyshev norms.

minimized by the Principle of Optimality step. Moreover, a functional, V, is a solution to the Bellman equation if and only if $B(V) = V$.

THEOREM: For the Asymptotic Dynamic Programming algorithm with known plant (described in Steps 1 through 5, and Figure 1.)

a). The feedback control law, $u_i = k_i(x_i)$, exponentially stabilizes the system (with respect to the Lyapunov function, $| \; |$ and the coefficient γ).

b). If $|V_i - B(V_i)| \rightarrow 0$ then $|V_i - V| \rightarrow 0$ where V is the solution to the Bellman equation.

c). If $|V_i - B(V_i)| \rightarrow 0$ exponentially then the feedback control law, $u_i = k_i(x_i)$ is asymptotically optimal in the sense that for any ε there exists an N such that

$$\sum_{i=M}^{\infty} l(x_i, k_i(x_i)) - V(x_M) < \varepsilon \qquad \textbf{(13)}$$

for all $M \geq N$.

The numerical techniques used to implement the asymptotic dynamic programming process are not specified in the theorem. Rather, our purpose is simply to characterize the behavior of the control produced by the asymptotic dynamic programming process if the numerical processes employed converge sufficiently rapidly. One can use classical numerical methods, neural network techniques, reinforcement learning techniques, etc.

For the case of an unknown plant, we assume that f takes the form

$$f(x_i, u_i) = a(x_i) + b(x_i)u_i \qquad \textbf{(14)}$$

and the approximating sequence, f_i, takes the form

$$f_i(x_i, u_i) = a_i(x_i) + b(x_i)u_i \qquad \textbf{(15)}$$

Although representing a non-trivial restriction of the system dynamics, this case is indicative of the power of the asymptotic dynamic programming concept for the case of an unknown plant.

To simplify the following analysis, we adopt the (admittedly abusive) notation $B(V_i)$ for the functional defined by

$$B(V_i)(.) = \underset{\kappa \varepsilon K_{||}}{min} \qquad \textbf{(16)}$$
$$[(l(.\,, \kappa(.)) + V_i(f_i(.\,, \kappa(.))))]$$

THEOREM: For the Asymptotic Dynamic Programming algorithm with unknown plant (described in Steps 1 through 6, and Figure 2.) assume that the solution of the Bellman equation, V, has a finite Lipschitz constant, $| V|^L$.

a). If $\left|a_i - a\right| \to 0$, the feedback control law, $u_i = k_i(x_i)$, exponentially stabilizes the system (with respect to the Lyapunov function, $\|\;\|$ and the coefficient γ.

b). If $\left|a_i - a\right| \to 0$ and $\left|V_i - B(V_i)\right| \to 0$, then $\left|V_i - V\right| \to 0$ where V is the solution to the Bellman equation.

c). If $\left|a_i - a\right| \to 0$ exponentially and $\left|V_i - B(V_i)\right| \to 0$ exponentially then the feedback control law, $u_i = k_i(x_i)$ is asymptotically optimal in the sense that for any ε there exists an N such that

$$\sum_{i=M}^{\infty} l(x_i, k_i(x_i)) - V(x_M) < \varepsilon \qquad \textbf{(17)}$$

for all $M \geq N$.

Unlike the previous theorem, convergence of the identification process is required for stability. If one does not know the plant, nor can it be identified on-line, there is little hope of guaranteeing stability.

As before, the numerical techniques used to implement the asymptotic dynamic programming process with unknown plant are not specified in the theorem. Rather, our purpose is simply to characterize the behavior of the control produced by the asymptotic dynamic programming process if the numerical processes employed converge sufficiently rapidly.

3. Caveat

Although we have developed rigorous proofs of the above theorems, they represent only the preliminary first steps in formulation a meaningful theory of Asymptotic Dynamic Programming. Specifically,

- they make no reference to the learning technique employed or
- its computational complexity and
- they assume a highly restrictive stability concept.

These results represent a first step toward developing a rigorous theory of Asymptotic Dynamic Programming in support of the various on-going research activities in the adaptive critic area.

4. References

[1]: Bellman, R.E., (1957). *Dynamic Programming*, Princeton University Press, Princeton.

[2]: Bertsekas, D.P., (1987), *Dynamic Programming: Deterministic and Stochastic Models*, Prentice-Hall, Englewood Cliffs.

[3]: Luenberger, D.G., (1979). *Introduction to Dynamic Systems: Theory, Models, and Applications*, John Wiley and Sons, New York.

[4]: Brayton, R.K. and C.H. Tong, (1979). Stability of Dynamical Systems: A Constructive Approach. IEEE Trans. Circ. and Sys, vol. CAS-26, no. 4, April.