NATURAL HEURISTIC DYNAMIC PROGRAMMING FOR DYNAMIC SYSTEMS CONTROL

K. Wendy Tang* and Jahangir Rastegar**

**Department of Electrical and Computer Engineering*
***Department of Mechanical Engineering*
State University of New York at Stony Brook
Stony Brook, NY 11794
wtang@ee.sunysb.edu and rastegar@motion.eng.sunysb.edu

Abstract: Heuristic Dynamic Programming (HDP) is the simplest kind of Adaptive Critic (Werbos, 1992). It can be used to maximize or minimize any utility function, such as total energy or trajectory error, of a system over time in a noisy environment. This article proposes a new version of HDP, called NHDP (Natural Heuristic Dynamic Programming). This new version incorporates basic HDP algorithm with the following features:(i) use of Trajectory Pattern Method to guarantee smoothness of trajectory and control signals; (ii) use multiple critic networks to localize effect of each parameter mimicking the natural biological model of human brain; and (iii) allow the controller to learn from slow to fast motion, analogous to the natural learning behavior of humans. A simple dynamic system is used to illustrate NHDP. *Copyright©1998 IFAC*

Keywords: Neural control, Dynamic systems, Learning control, Neural Networks, Backpropagation.

1. INTRODUCTION

Recently Adaptive Critic Design (ACDs) has received increasing attention as a powerful form of neuro-control. A good overview for a range of ACDs can be found in (Werbos, 1992; Prokhorov and Wunch, 1997). Basically, the main advantage of ACD is its ability to maximize or minimize a utility function such as total energy or trajectory error over time. Among the various ACDs, Heuristic Dynamic Programming (HDP) is the simplest kind.

In this paper, the objective is to illustrate how HDP can be effectively used to control dynamic systems when combined with trajectory pattern method. The trajectory pattern method was proposed by Rastegar and Fardanesh (Rastegar and Fardanesh, 1991). It allows motion of the dynamic system be synthesized to contain sinusoidals with a fundamental frequency and its harmonics. The task of the controller is essentially simplified to identifying the coefficients of these harmonics that constitute the inverse dynamic equations. Furthermore, by learning the motion from slow to fast through gradual increment of the fundamental frequency, the controller learns *naturally* the characteristics of the motion. The end result is a HDP controller that learns naturally to control a motion from slow to fast, hence the name *Natural HDP*.

This article is organized as follows: Section 2 is a description of *Natural Heuristic Dynamic Programming (NHDP)*. Section 3 illustrates how the algorithm is used for the control of a simple dynamic system. Finally, conclusions and a summary are included in Section 4.

2. NATURAL HEURISTIC DYNAMIC PROGRAMMING

In Natural Heuristic Dynamic Programming, the desired motion to be controlled is first synthesized as superpositions of sinusoidals through the Trajectory Pattern Method. Once the motion is synthesized, it can be shown that the desired control action and the motion are signals with only a fundamental frequency and its higher harmonics. The task of the controller is simplified to identifying these coefficients. Basic HDP algorithm is then deployed for this task. Furthermore, the controller first learns control of the system in slow motion corresponding to a small fundamental frequency. We then gradually tune the fundamental frequency to a higher value for faster motion, mimicking the natural learning strategy of human beings. In this section, the Trajectory Pattern Method is first used to synthesis motion. Then a description of NHDP for controlling the synthesized motion is described.

2.1 Trajectory Pattern Method

The Trajectory Pattern Method was first proposed by Rastegar and Fardanesh (Rastegar and Fardanesh, 1991). Readers interested in more detailed description of the method are referred to (Rastegar and Fardanesh, 1991; Fardanesh and Rastegar, 1992; Tu and Rastegar, 1993; Tu , *et al.*, 1994).

Basically, the method can be said to be a generalization of the computed torque method. The desired motion is synthesized by the selection of appropriate trajectory parameters to satisfy the desired end conditions and/or tracking requirement.

As a simple example, for a one dimensional system with the following dynamic equations:

$$u(t) = m\ddot{x}(t) + c\dot{x}(t) + kx(t) \qquad (1)$$

where *u(t)* is the control action at time *t*, x(t), $\dot{x}$(t), $\ddot{x}$(t) are the position, velocity and acceleration and *m,c,k* are constants.

The controller's task is to identify the control signals *u(t)*for a given initial and end position, velocity and acceleration. The dynamic equation (Equation 1) is either assumed to be unknown or having inaccurate parameters. In the Trajectory Pattern Method, we synthesize the desired motion as composed of superposition of sinusoidals with a fundamental frequency and its higher harmonics. Furthermore, the parameters of the trajectory are synthesized such that the initial and end conditions as specified by the problem are met.

For this simple example, given that the initial (*t=0*) and final (t= t_f) position, velocity and acceleration are:

$$\begin{aligned} x(0) &= \dot{x}(0) = \ddot{x}(0) = 0 \\ x(t_f) &= 1.0, \dot{x}(t_f) = \ddot{x}(t_f) = 0 \end{aligned} \qquad (2)$$

We can synthesize the desired trajectory as:

$$\begin{aligned} x(t) &= k_0 + k_1(\cos\omega t - \cos 3\omega t / 9) \\ \dot{x}(t) &= -k_1\omega(\sin\omega t - \sin 3\omega t / 3) \\ \ddot{x}(t) &= -k_1\omega^2(\cos\omega t - \cos 3\omega t) \end{aligned} \qquad (3)$$

where k_0 =*0.5* and k_1= *-9/16* are chosen to meet the initial and end conditions (Equation 2) requirements with the condition that $\omega t_f = \pi$. That is, when the fundamental frequency is small, t_f is large and the motion is slow. Since the desired motion is synthesized to compose of sinusoidals with a fundamental frequency ω and its third harmonics 3ω, the desired control signal is of the form:

$$u(t) = A\cos\omega t + B\sin\omega t + C\cos 3\omega t + D\sin 3\omega t + E \qquad (4)$$

where *A, B, C, D, E* are constants analogous to the Fourier coefficients of the control signals. Once this observation is made, the controller's task is now simplified to identifying these constant coefficients. Because of this simplification, HDP, the simple form of ACDs can now be effectively utilized. Another advantage of the Trajectory Pattern Method is that the desired motion is always composed of superpositions of sinusoidal signals with a fundamental frequency and its higher harmonics. These motions are smooth functions similar to that of natural motions. Furthermore, when the fundamental frequency ω is small, the motion is slow. By gradually increasing ω, the motion becomes faster.

Once the Trajectory Pattern Method has been incorporated into the control problem, in the next section, we discuss how the original HDP algorithm can be used to learn the coefficients *A,* *B, C, D, E* first for a slow motion and then for a faster one.

It should be noted also that in the above example, we synthesized the desired motion to contain the fundamental frequency and its third harmonics ($\omega, 3\omega$). In general, however, the synthesized motion may contain the fundamental frequency ω and any of its higher harmonics. Readers interested in more advanced Trajectory Pattern Method are referred to (Rastegar and Fardanesh, 1991; Fardanesh and Rastegar, 1992; Tu and Rastegar, 1993; Tu , *et al.*, 1994).

2.2 *Natural Heuristic Dynamic Programming (NHDP) Components*

The key element of the HDP design is the "Critic" network. The function of the critic network is to learn the cost-to-go *J* function in the Bellman equation of dynamic programming (Werbos, 1992; Prokhorov and Wunsch, 1997) :

$$J(t) = \sum_{k=0}^{\infty} \gamma^k U(t+k) \qquad (5)$$

where γ is a discount factor and *U(.)* is the utility function. The basic idea is that the current action should be optimized to minimize or maximize not just current utility but future utility. This goal is achieved through minimization or maximization of *J*, since it is a sum of current and future utility. The multilayered backpropagation neural network is a convenient tool for such optimization as the gradient of *J* can be obtained through backward propagation. However, during our practice with HDP, we found that the algorithm does not perform well for problems involving multiple variables. This perhaps is the major reason that ACD practitioners generally opted for the more advanced design such as DHP.

From a seminar in neurobiology, we recently learned that in the human brain, various functions are highly localized (Damasio, 1997). Each subset of neurons is only responsible for a specific local task. This combined with the approach on Trajectory Pattern Method, gives rise to a new form of HDP that mimics the natural learning behavior of humans, hence the name Natural HDP.

From the discussions in Trajectory Pattern Method, the original control problem is transformed to the problem of identifying the coefficients, *A, B, C, D, E* as in the example of Section 2.1. To localize the effect of each parameter, we use four critic networks, each learning how changes in each coefficient affects the cost-to-go *J* function (the constant *E* does not change with ω). The end result is schematically depicted in Figure 4. From Figure 4, the neuro-controller contains four simple, critic networks, each can perform backpropagation in parallel. Such a neuro-architecture is consistent with the biological model that a subset of neurons is responsible for a local task, learning *J* as a function of a specific coefficient. Such learning is accomplished in parallel, taking the full advantages of the Neural Network properties.

Furthermore, due to the gradient decent nature of backpropagation, the initial values of the coefficients *A, B, …, E* is important. To have an intelligent guess of the initial coefficients, we first use Backpropagation of Utility (Werbos, 1990; Tang and Pingle, 1996) to find the coefficients for a very slow motion with a small value for the fundamental frequency ω and large value of t_f.

Once a good initial guess is obtained, Natural Heuristic Dynamic Programming with multiple critic networks is used to control the system for a faster motion (larger value for the fundamental frequency ω). Learning is achieved in two phases. In phase I, each critic network learns the cost-to-go function *J* associated with each coefficient through standard backpropagation. Then in Phase II, through backpropagation, the gradient of *J* with respect to the coefficients is computed and a new coefficient is obtained through gradient decent. As an example, for the first critic network, we obtain $\partial J_A / \partial A$ through backward propagation. New values of the coefficient *A* is then computed as:

$$\text{New } A = \text{Old } A - \alpha\, \partial J_A / \partial A \qquad (6)$$

where $0 < \alpha < 1$ is a constant learning rate. Once a new coefficient is obtained the two phases repeated for another iteration until the current utility is less than or bigger than a prescribed value if we are minimizing or maximizing the utility function.

3. A SIMPLE DYNAMIC SYSTEM

To illustrate Natural HDP, we used the simple example outlined in Section 2.1. The dynamic equation is described by Equation 1 with the constants assuming the following values: $m=10$, $c=2$, $k=5.5$. The desired initial and final position, velocity, and acceleration are specified by Equation 2. First, we use backpropagation of utility (Werbos, 1990; Tang and Pingle, 1996) to learn a very slow motion, $\omega=0.01\ \pi$, $t_f=100$. The desired coefficients are:

$$A = -3.088,\ B=3.534\text{e-}2,\ C=3.382\text{e-}1,\ D=-1.178\text{e-}2,\ E=2.75 \tag{7}$$

Using Backpropagation of Utility, the coefficients are identified as:

$$A = -3.097,\ B = 4.080\text{e-}3,\ C = 3.456\text{e-}1,\ D = -1.017\text{e-}2,\ E = 2.774 \tag{8}$$

Next we use these as initial values to control a faster motion, $\omega=0.1\pi$, $t_f=10$. Since the coefficients, with the exception of E are functions of the fundamental frequency, ω. The desired coefficients for this faster motion are:

$$A = -2.539,\ B = 3.534\text{e-}1,\ C = -2.114\text{e-}1,\ D = -1.178\text{e-}1,\ E = 2.75 \tag{9}$$

For each of the coefficient, with the exception of E, we created a critic network that inputs a coefficient and outputs its corresponding cost-to-go function J as defined in Equation 5. The discount factor and the learning rate in Equations 5 and 6 are $\gamma=\alpha=0.1$ and the Utility function is:

$$U = \int_0^{t_f} (x(t) - x_d(t))^2\, dt \tag{10}$$

where $x(t)$ is the actual position as a result the action signals $u(t)$ and $x_d(t)$ is the desired position as described in the synthesized motion (Equation 3). Obviously, the neuro-controller's task is to find the coefficients that can minimize this utility function over time.

Once the coefficients are determined by multiple critic networks, the control signals are obtained by Equation 4. Figure 2 is a comparison of the control signal $u(t)$ at various iterations versus time ($t=0, \ldots, t_f=10$). The result at $Iter=0$ corresponds to the initial coefficients (Equation 8 with $t_f=100$). As the iteration progresses, the control signal approaches that of the desired value. Figure 3 plots the absolute error between the obtained and the desired control signals.

Figure 4 is a comparison for the position trajectory at various iterations. Again, the position of the dynamic system gradually approaches that of the desired values. Note that at all iterations, the trajectory is a smooth function. This is a direct consequence of the Trajectory Pattern Method. The trajectory profile for velocity and acceleration depicted similar behavior and is not shown here. Figure 5 plots the absolute trajectory error between the actual and desired position, velocity and acceleration after 5,000 iterations. In all cases, the absolute error is less than $3.0\text{e-}2$, which means that the trajectory generated by the control signal is very close to their desired values.

4. SUMMARY

In this paper, a new version of HDP, the simplest kind of ACDs, is presented. The main features of this new version include: (i) use of Trajectory Pattern Method to guarantee smoothness of trajectory and control signals; and simplification of the control problem; (ii) use multiple critic networks to localize effect of each parameter mimicking the natural biological model of human brain; and (iii) allow the controller to learn from slow to fast motion, analogous to the natural learning behavior of humans. In other words, this new version of the HDP algorithm uses the natural model of human learning as a reference, hence the name Natural HDP.

As a preliminary result, we applied the algorithm for a simple, one-dimensional, linear dynamic system. The results are encouraging. The NHDP controller is able to control the system from slow to fast motion. It is also important to note that the strength of NHDP algorithm will be more apparent for complex, non-linear system as the Trajectory Pattern Method provides a greater degree of simplifications for these systems. We are currently working on the application of NHDP to a nonlinear dynamic system.

REFERENCE

Damasio, A. (1997). The ghost in the Machine: Exploration on the Minded Brain, Neurobiology seminar, SUNY Stony Brook.

Fardanesh, B. and J. Rastegar. (1992). A New Model Based Tracking Controller for Robot Manipulators Using the Trajectory Pattern Inverse Dynamics. *IEEE Transactions on Robotics and Automation*, **8(2)**:279--285.

Prokhorov, D.V. and D.C. Wunsch II. (1997). Adaptive Critic Design. *IEEE Transactions on Neural Networks*, **8(5)**:1997—1007.

Rastegar, J. and B. Fardanesh. (1991). Inverse Dynamic Models of Robot Manipulator Using Trajectory Patterns - With Application to Learning Controllers. *In Proceedings of the 8th World Congress on the Theory of Machines and Mechanisms*, Czechoslovakia.

Tang K.W., and G. Pingle. (1996). Neuro-Remodeling via Backpropagation of Utility.' *Journal of Mathematical Modeling and Scientific Computing*, Accepted for Publication.

Tu, Q. , J. Rastegar, and R.J. Singh. (1994). Trajectory Synthesis and Inverse Dynamic Model Formulation and Control of Tip Motion of a High Performance Flexible Positioning System. *Mechanism and Machine Theory*, **29(7)**:959--968.

Tu, Q. and J. Rastegar. (1993). Manipulator Trajectory Synthesis for Minimal Vibrational Excitation Due to Payload. *Transactions of Canadian Society of Mechanical Engineers*, **17(4)**:557--566.

Werbos, P.J. (1990). Backpropagation Through Time: What It Does and How to Do It. *Proceedings of the IEEE*, **78(10)**:1550--1560.

Werbos, P.J. (1992). Approximate Dynamic Programming for Real-Time Control and Neural Modeling. In *Handbook of Intelligent Control*, (White, D.A. and Sofge, D.A. (Ed)), pages 493--525. Van Nostrand Reinhold, New York.

ACKNOWLEDGEMENT

The authors acknowledge and appreciate discussions with Paul Werbos. This research was supported by the National Science Foundation under Grant ECS-9626655.

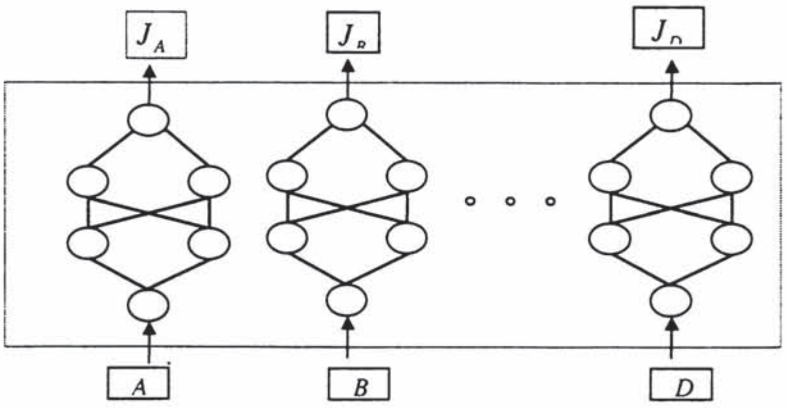

Figure 1: Schematic Representation of NHDP Architecture

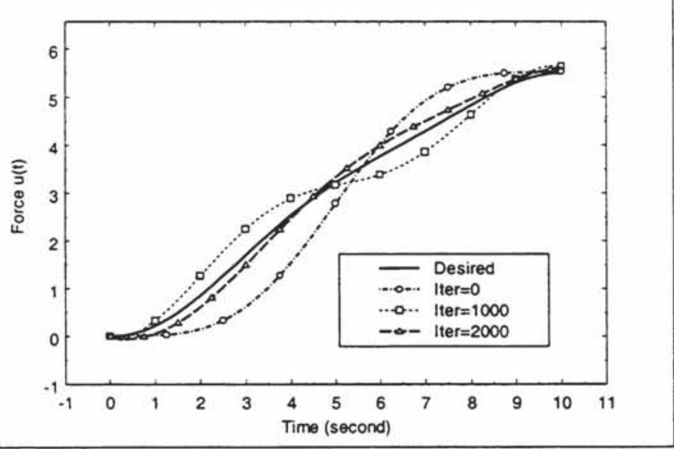

Figure 2: Comparison of Control Signals at Various Iterations

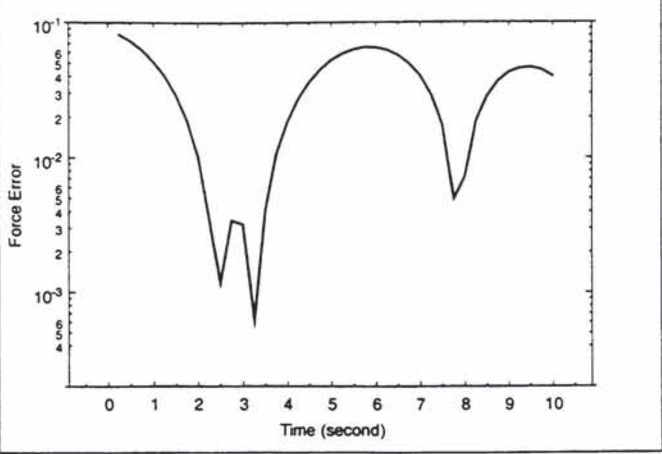

Figure 3: Force Error After 5,000 Iterations

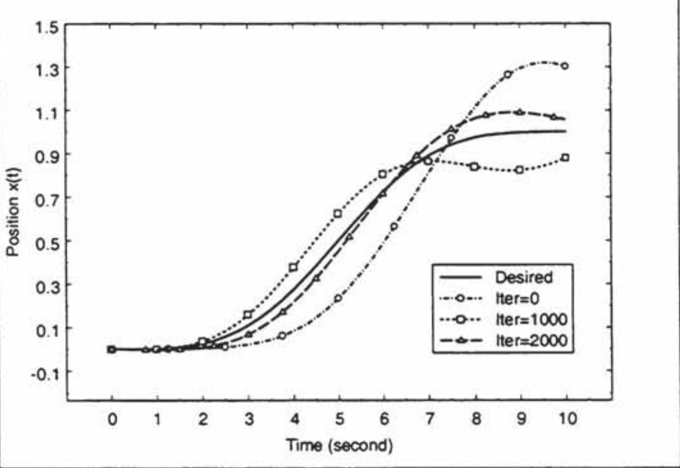

Figure 4: Comparison of Trajectory at Various Iterations

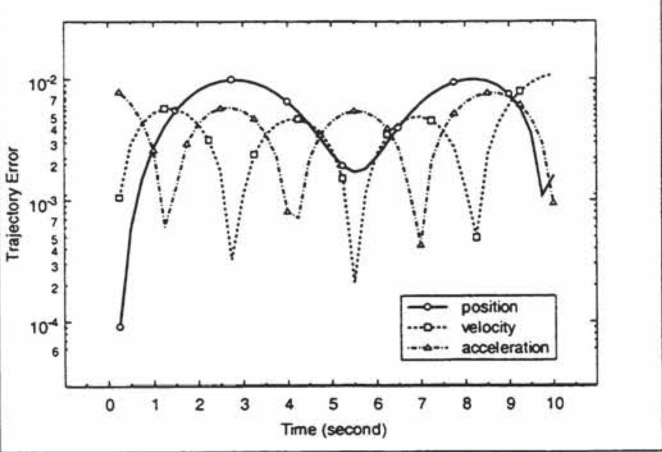

Figure 5: Trajectory Error After 5,000 Iterations