Intelligent robots that adapt, learn, and predict.

E. L. Hall, X. Liao , M. Ghaffari, , and S. M. Alhaj Ali*
Center for Robotics Research
University of Cincinnati
Cincinnati, OH 45221-0072 USA
Phone: 513-556-2730
Fax: 513-556-3390
Email: Ernie.Hall@uc.edu
Internet: http://www.robotics.uc.edu/
* The Hashemite Univ. (Jordan)

ABSTRACT

The purpose of this paper is to describe the concept and architecture for an intelligent robot system that can adapt, learn and predict the future. This evolutionary approach to the design of intelligent robots is the result of several years of study on the design of intelligent machines that could adapt using computer vision or other sensory inputs, learn using artificial neural networks or genetic algorithms, exhibit semiotic closure with a creative controller and perceive present situations by interpretation of visual and voice commands. This information processing would then permit the robot to predict the future and plan its actions accordingly. In this paper we show that the capability to adapt, and learn naturally leads to the ability to predict the future state of the environment which is just another form of semiotic closure. That is, predicting a future state without knowledge of the future is similar to making a present action without knowledge of the present state. The theory will be illustrated by considering the situation of guiding a mobile robot through an unstructured environment for a rescue operation. The significance of this work is in providing a greater understanding of the applications of learning to mobile robots.

Keywords: Intelligent robots, adaptive control, creative control, reinforcement learning, adaptive critic

1. INTRODUCTION

The purpose of this paper is to describe a theory of robust learning for intelligent machines and propose its application to the design of systems in uncertain environments. The proposed architecture for machine learning is also based on the perceptual creative controller for an intelligent robot that uses a multi- modal adaptive critic for performing learning in an unsupervised situation but can also be trained for tasks in another mode and then is permitted to operate autonomously. The robust nature will be derived from the automatic changing of modes based on internal measurements of error at appropriate locations in the controller.

The creative controller method is designed for unstructured environments. Creative learning architectures integrate a Task Control Center (TCC) and a dynamic database (DD) into adaptive critic learning algorithms to permit these solutions. Determining the task to be performed and the data base to be updated are the two new elements of the proposed research. These new decision processes encompass both decision and estimation theory and can be modeled by neural networks.

The main thrust of this paper is to present a theory of learning that can be used for developing control architectures for intelligent machines. The control architectures for neural network control of vehicles in which the kinematic and dynamic models are known but one or more parameters must be estimated is a simple task that has been demonstrated. The mathematical models for the kinematics and dynamics were developed and the main emphasis was to explore the use of neural network control and demonstrate the advantages of these learning methods. The results indicate the method of solution and its potential application to a large number of currently unsolved problems in unstructured environments. The adaptive critic neural network control is an important starting point for future learning theories that are applicable to robust control and learning situations.

The general goal of this research is to further develop a theory of learning that is based on human learning but applicable to machine learning and to demonstrate its application in the design of robust intelligent systems. To obtain broadly applicable results, a generalization of adaptive critic learning called Creative Control (CC) for intelligent robots in unstructured environments will be used. The creative control learning architecture integrates a Task Control Center (TCC) and a Dynamic Knowledge Database (DKD) with adaptive critic learning algorithms.

Recently learning theories such as the adaptive critic have been proposed. In this type of learning a critic provides a grade to the controller of an action module such as a robot. The creative control process is used that is "beyond the adaptive critic." A mathematical model of the creative control process is presented that illustrates the use for mobile robots.

Dynamic Programming

The intelligent robot in this paper is defined as a decision maker for a dynamic system that may make decisions in stages. The outcome of each decision may not be fully predictable but may be anticipated or estimated to some extent before the next decision is made. Furthermore, an objective or cost function can be defined. There may also be natural constraints. Generally, the goal is to minimize this cost function over some decision space subject to the constraints. With this definition, the intelligent robot can be considered as a set of problems in dynamic programming and optimal control as defined by Bertsekas[1].

Dynamic programming (DP) is the only approach for sequential optimization applicable to general nonlinear, stochastic environments. However, DP needs efficient approximate methods to overcome its dimensionality problems. It is only with the presence of artificial neural network (ANN) and the invention of back propagation that such a powerful and universal approximate method has become a reality. The essence of dynamic programming is Bellman's *Principle of Optimality*.[2]

The original Bellman equation of dynamic programming for adaptive critic algorithm may be written as shown in Eq (1):

$$J(R(t)) = \max_{u(t)}(U(R(t),u(t)) + < J(R(t+1)) >)/(1+r) - U_0 \qquad (1)$$

where J is the criteria or cost-to-go function at time t, r and U_0 are constants that are used only in infinite-time-horizon problems and then only sometimes, and where the angle brackets refer to expected value. Regarding the finite horizon problems, which we normally try to cope with, one can use Eq (2):

$$J(R(t)) = \max_{u(t)}(U(R(t),u(t)) + < J(R(t+1)) >)/(1+r) \qquad (2)$$

Dynamic programming gives the exact solution to the problem of how to maximize a utility function U(R(t)) over the future times, t, in a nonlinear stochastic environment, where the vector R(t) represents the state of the environment at time t. Dynamic programming converts a difficult long-term problem in optimization over time <U(R(t))>, the expected value of U(R(t)) over all the future times, into a much more straightforward problem in simple, short-term function maximization – after we know the function J. Thus, all of the approximate dynamic programming methods discussed here are forced to use some kind of general-purpose nonlinear approximate to the J function, the value function in the Bellman equation, or something closely related to J[3].

In most forms of adaptive critic design, we approximate J by using a neural network. Therefore, we approximate J(R) by some function $\hat{J}(R,W)$, where W is a set of weights or parameters, $\hat{J}$ is called a Critic network [4,5]

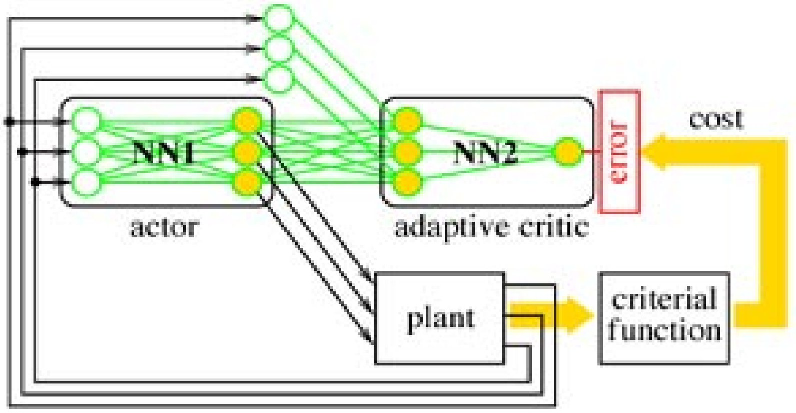
If the weights **W** are adapted or iteratively solved for, in real time learning or offline iteration, we call the Critic an Adaptive Critic[6].

An adaptive critic design (ACD) is any system which includes an adapted critic component; a critic, in turn, is a neural net or other nonlinear function approximation which is trained to converge to the function **J(X).**

In adaptive critic learning or designs, the critic network learns to approximate the cost-to-go or strategic utility function J and uses the output of an action network as one of its' inputs, directly or indirectly. When the critic network learns, back propagation of error signals is possible along its input feedback to the action network. To the back propagation algorithm, this input feedback looks like another synaptic connection that needs weights adjustment. Thus, no desired control action information or trajectory is needed as supervised learning.

2. ADAPTIVE CRITIC AND CREATIVE CONTROL

Most advanced methods in neurocontrol are based on adaptive critic learning techniques consisting of an action network, adaptive critic network, and model or identification network as show in Figure 17. These methods are able to control processes in such a way, which is approximately optimal with respect to any given criteria taking into consideration of particular nonlinear environment. For instance, when searching for an optimal trajectory to the target position, the distance of the robot from this target position can be used as a criteria function. The algorithm will compute proper steering, acceleration signals for control of vehicle, and the resulting trajectory of vehicle will be close to optimal. During trials (the number depends on the problem and the algorithm used) the system will improve performance and the resulting trajectory will be close to optimal. The freedom of choice of the criteria function and the ability to derive a control strategy only from trial/error experience are very strong advantages of this method.



Structure of the adaptive critic controller based on artifical neural networks.

Figure 1 Structure of the adaptive critic controller [7]

What can the adaptive critic do and what can't it do? What can creative learning do that is beyond adaptive critic learning? As it is well-known, adaptive critic learning is a way to solve dynamic programming in a general nonlinear plant. It takes the approach of approximate the control processes or estimating the cost-to-go function J but does not relate easily to decision-making theory. For instance, what are the criteria or critics for the different sub-tasks, how does one choose the criteria function or utility function, how does one memorize the experience as human-like memories? All of these are concerns of novel learning techniques. In this paper, we present a creative learning structure with evolutionary learning strategies. Adaptive critic learning method is a part of the creative learning algorithm, however, creative learning with decision-making capabilities is beyond the adaptive critic learning. The most important characteristics of the creative learning structure are as shown in Figure 2:

(1) Brain-like decision-making task control center, entails the capability of human brain decision-making, a true intelligent center.
(2) Dynamic criteria knowledge database integrated into the critic-action framework, makes the adaptive critic controller reconfigurable and enables the flexibility of the network framework.
(3) Multiple criteria, multi-layered structure and increase of the degree of derivatives of J function.
(4) Modeled and forecasted critic modules, result in faster training networks.
(5) Also, a predictive action module can be realized according to Syam, et al [8].

The structure of creative learning system proposed above is discussed in the following sections.

Creative Learning Structure

It is assumed that we can use a kinematic model of a mobile robot to provide a simulated experience to construct a value function in the critic network and to design a kinematic based controller for the action network. A proposed diagram of creative learning algorithm is shown in Figure 2, 3 [9-11]. In this proposed diagram, there are six important components: task control center, dynamic knowledge database, critic network, action network, model-based action and utility funtion. Both the critic network and action network can be constructed by using any artificial neural networks with sigmoidal function or radial basis function (RBF). Furthermore, the kinematic model is also used to construct a model-based action in the framework of adaptive critic-action approach. In this algorithm, Dynamic databases are built to generalize the critic network and its training process and provide evironmental information for decision-making purpose. It is especially critical when the operation of mobile robots is in an unstructured environments. Furthermore, the dynamic databases can also used to store environmental parameters such as Global Position System (GPS) weight points, map information, etc. Another component in the diagram is the utility function for a tracking problem (error measurement). In the diagram, X_k, X_{kd}, X_{kd+1} are input and Y is the ouput and J(t), J(t+1) is the critic function at the time. The simulated results are under investigation of two-link robot manipulators tracking problem.

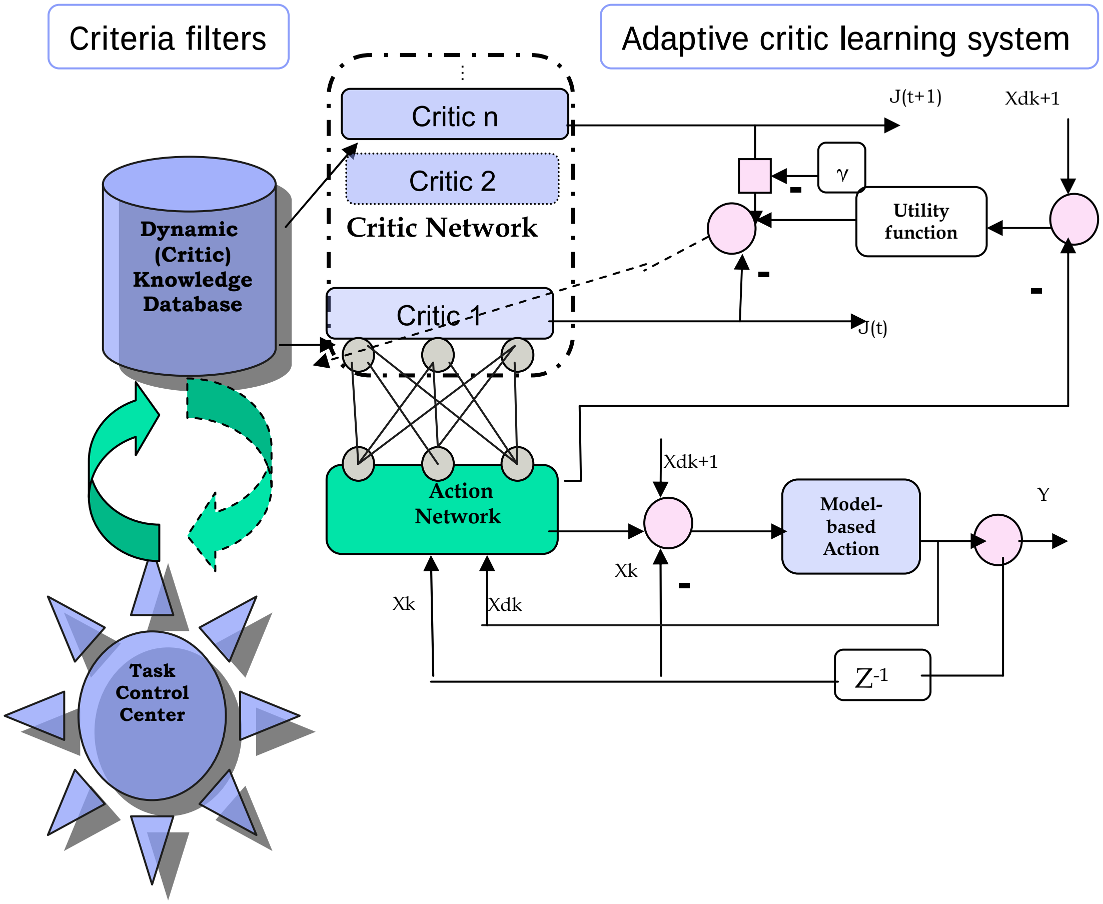


Figure 2 Proposed Creative Learning Algorithm Structure

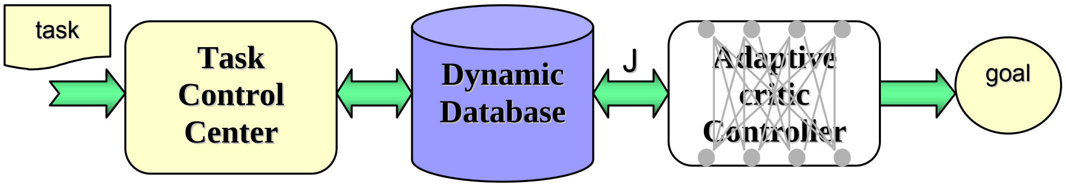


Figure 3 Decomposition of the creative learning structure

Dynamic Knowledge Database (DKD)

How to build the dynamic databases as domain knowledge and at the same time to learn itself. Dynamic knowledge databases defined as a "neurointerface" [12] is a dynamic filtering system based on neural networks (NNs) that serves as a "coupler" between a task control center and a nonlinear system or plant that is to be controlled or directed. The purpose of the coupler is to provide the criteria functions for the adaptive critic learning system and filter the task strategies commanded by the task control center. The proposed dynamic database contains a copy of the model (or identification). Action and critic networks are utilized to control the plant under nominal operation, as well as make copies of a set of HDP or DHP parameters (or scenario) previously adapted to deal with a plant in a known dynamic environment. It also stores copies of all the partial derivatives required when updating the neural networks using backpropagation through time[13] . The dynamic database can be expanded to meet the requirements of unstructured environment.

The data stored in the dynamic database can be uploaded to support offline or online training of the dynamic plant and provide a model for identification of nonlinear dynamic environment with its modeling function. Another function module of the database management is designed to analyze the data stored in the database including the sub-task optima, pre-existing models of the network and newly added models. The task program module is used to communicate with the task control center. The functional structure of the proposed database management system (DBMS) is shown in Figure 4. The DBMS can be customized from a open source object-relational database which is to be developed as a future research project.

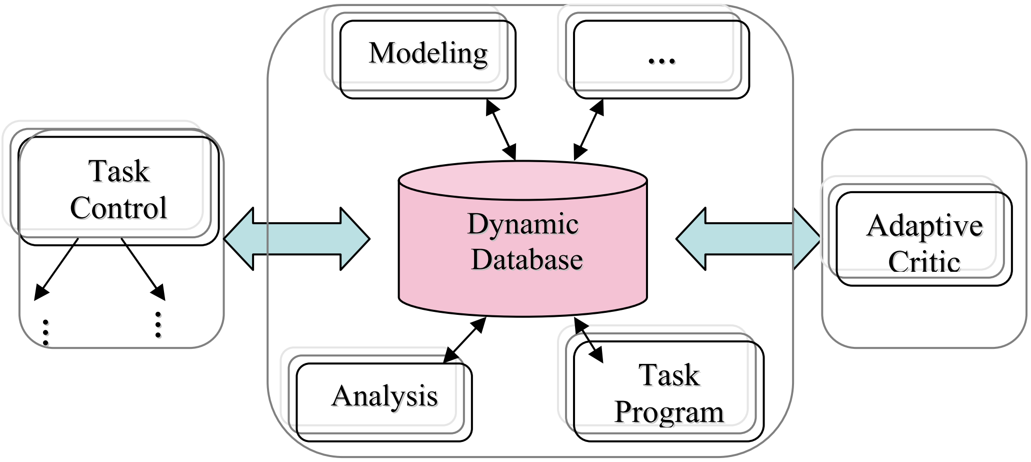


Figure 4 Functional structure of dynamic database

Task Control Center (TCC)

What is task control center? What does it do in creative learning system? How does the task control center embed into the adaptive critic learning networks? The task control center (TCC) can build task-level control systems for the creative learning system. By "task-level", we mean the integration and coordination of perception, planning and real-time control to achieve a given set of goals (tasks)15. TCC provides a general task control framework, and it is to be used to control a wide variety of tasks. Although TCC has no built-in control functions for particular tasks (such as robot path planning algorithms), it provides control functions, such as task decomposition, monitoring, and resource management, that are common to many applications. The particular task built-in rules or criteria or learning J functions are managed by the dynamic database controlled with TCC to handle the allocation of resources. The dynamic database matches the constraints on a particular control schemes or sub-tasks or environment allocated by TCC.

The task control center acts as a decision-making system. It integrates domain knowledge or criteria into the database of the adaptive learning system. According to Simmons14, task control architecture for mobile robots provides a variety of control constructs that are commonly needed in mobile robot applications, and other autonomous mobile systems. The goal of the architecture is to enable autonomous mobile robot system to easily specify hierarchical task-decomposition strategies, such as how to navigate to a particular location, or how to collect a desired sample, or follow a track in an unstructured environment. This can include temporal constraints between sub-goals, leading to a variety of sequential or concurrent behaviors. TCC schedules the execution of planned behaviors, based on those temporal constraints acting as a decision-making control center.

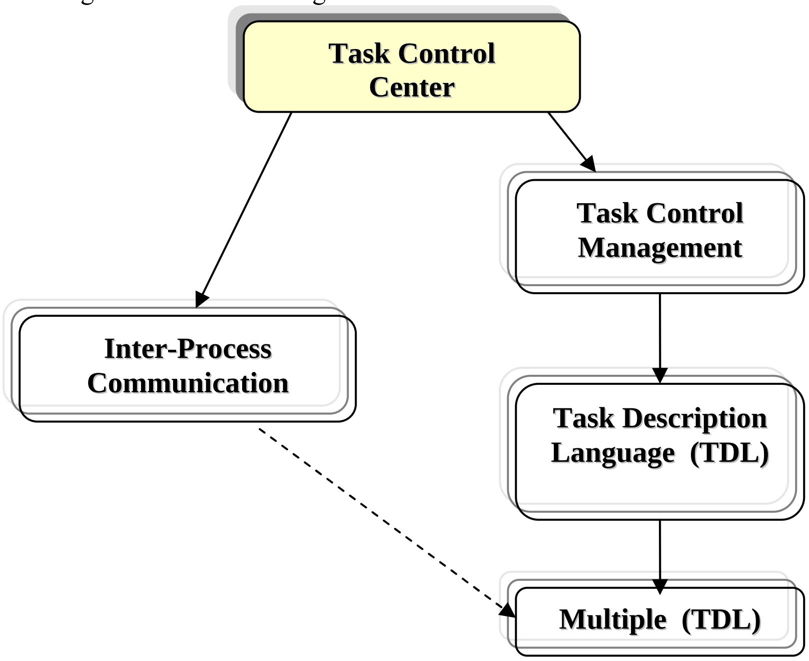


Figure 5 Decomposition of the structure of task control center

Integrating TCC with adaptive critic learning system and interacting with the dynamic database, the creative learning system could provide both task-level and real-time control or learning within a single architectural framework as shown in Figure 5. Through interaction with human beings to attain the input information for the system, the TCC could decompose the task strategies to match the dynamic database for the rules of sub-tasks by constructing a distributed system with flexible mechanisms, which automatically marshal and unmarshal data. TCC also provides orderly access to the resources of the dynamic database with built-in learning mechanisms according to a queue mechanism. This is the inter-process communication capability between the task control center and the dynamic database. The algorithm on how to link between the task control center and the dynamic database is proposed to be a future research project as well.

Creative learning controller for intelligent robot control

Creative learning is used to explore the unpredictable environment, permit the discovery of unknown problems, ones that are not yet recognized but may be critical to survival or success. By learning the domain knowledge, the system should be able to obtain the global optima and escape local optima. It generalizes the highest level of human learning – imagination. As a ANN robot controller, the block diagram of the creative controller can be presented in Figure 6. Experience with the guidance of a mobile robot has motivated this study to progress from simple line following to the more complex navigation and control in an unstructured environment. The purpose in this system is to better understand the adaptive critic learning theory and move forward to develop more human-intelligence-like components into the intelligent robot controller. Moreover, it should extend to other applications. Eventually, integrating a criteria knowledge database into the action module will develop a real imaginational adaptive critic learning module.

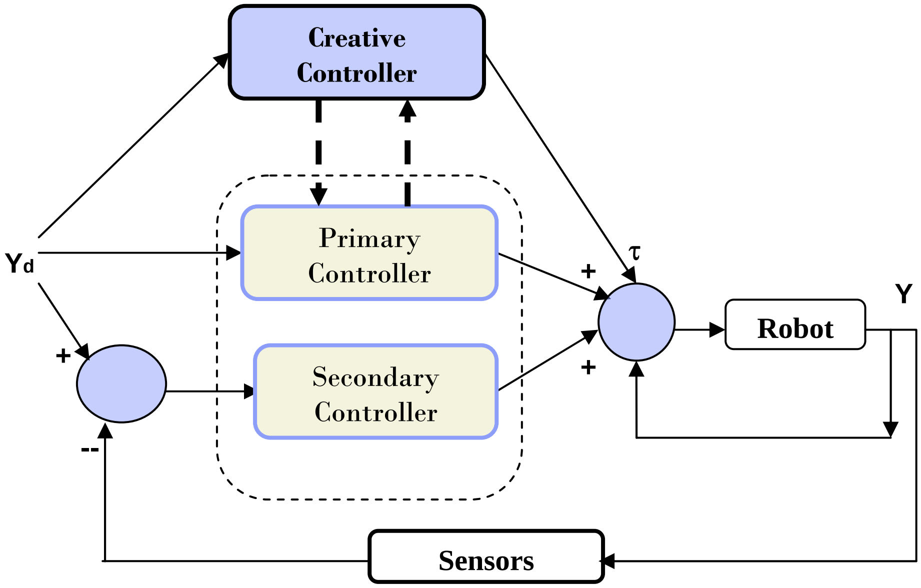


Figure 6 Block diagram of creative controller

A creative controller is designed to integrate domain knowledge or criteria database or task control center into the adaptive critic neural network controller. It’s needed to be well-defined structure according to the autonomous mobile robot application. We take intelligent mobile robot as the test-bed for the creative controller. The task control center of the creative learning system would be hierarchically learning the task as follows:

- Mission for robot – e.g. mobile robot
- Task for robot to follow – J : task control
- Track for robot to follow
- Learn non-linear system model- model discovery
- Learn unknown parameters such as kinematics, dynamics parameters

Adaptive Critic system Implementation

Adaptive Critic system and NN

In order to develop the creative learning algorithm addressed above, we take a bottom-up approach to implement adaptive critic controllers first using neural network on-line or off-line learning methods 16 Then the proposed dynamic knowledge database and task control center will be realized in future research projects.

Artificial neural network (ANN) made adaptive critic learning possible. Give x, a real vector, a one-layer feedforward neural network (NN) has a net output given by

$$y_i = \sum_{j=1}^{N_h} [w_{ij}\varphi(x)] + \theta_{wi} \quad ; i=1,\ldots,m \qquad (3)$$

Where φ(.) the activation functions and wij the output-layer weights. The θwi, i=1,2, … , are threshold offsets and Nh is the number of hidden layer neurons 17 . A three hidden-layer neural network shown in Figure 7.

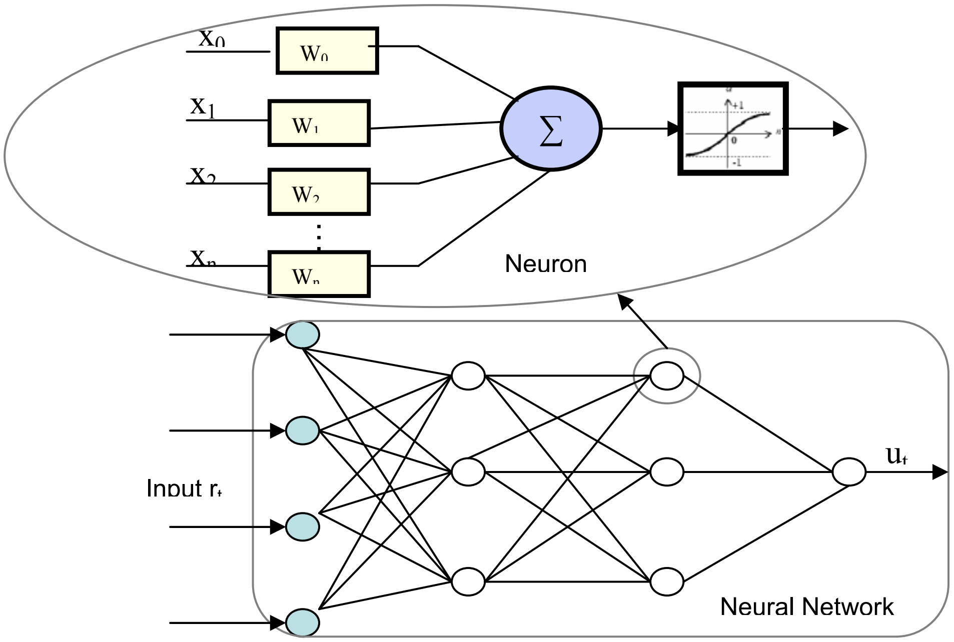


Figure 7 Three-layer neural network

An artificial neural network consists of a nonlinear mapping, denoted by NN, that performs a nonlinear transformation of q- dimensional input r, into a p-dimensional output, Z [18]

$$Z=NN(r) \qquad (4)$$

The network architecture and parameters characterize the nature of this transformation and can be determined based on input, output and derivative information pertaining to the function to be approximated.

As described in chapter 3, the adaptive critic learning structure, Dual Heuristic Programming (DHP), includes action network and critic network adaptation as shown in Figure 6 and.7. The action network approximates the optimal control law and the critic network evaluates the action network performance by approximating the derivative of the optimal value function with respect to the state:

$$U(t)= NN_A[r(t)]=Z_A(t) \qquad (5)$$

$$\lambda(t) = NN_C[r(t)]=Z_C(t) \qquad (6)$$

Where NN_A, NN_C denoted as action network and critic network nonlinear approximate function, respectively, Z_A is the output from action network, Z_C is the output from critic network. The input to both networks includes the dynamically significant auxiliary inputs a, i.e.

$$r(t)=[x(t)^T, a(t)^T]^T \qquad (7)$$

During each time interval $\Delta t=t_{k+1}-t_k$, the action and critic networks are adapted to more closely approximate the optimal control law and value function derivatives, respectively. The recurrence relation provides for adaptation criteria that, over time, guarantee convergence to the optimal solution.

A comparison of HDP, DHP

DHP is capable of generating smoother derivatives and has shown improved performance when compared to HDP. Those results were reported in 19 , where both were applied to a turbogenerator in a highly complex, nonlinear, fast-acting, multivariable system with dynamic characteristics that vary as operating conditions change. DHP has an important advantage over HDP since its critic network builds a representation for the derivatives of J function by being explicitly trained on them through $\partial U(t)/\partial R(t)$ and $\partial U(t)/\partial A(t)$.

Both HDP and DHP techniques were used to implement adaptive critic learning module. General training procedure is that suggested in 20-21and it is applicable to any adaptive critic design (ACD). It consists of two training cycle: that of the critic, and that of the action. The critic's adaptation is done initially with the action network offline trained to ensure the whole system with ACD and nonlinear plant stable. Then the action network is trained further while keeping the critic network weights fixed. This process of training the critic and the action alternatively until the acceptable performance is achieved. The model network is previously trained offline, not concurrently trained in the process of action and critic network. Critic network and action network weights: WC and WA are initialized to any reasonable values.

In critic network's training cycle, an incremental optimization is carried out using a suitable optimization technique (e.g. LMS). The following operations are repeated N1 times:

1. Initialize t=0 and y(0);
2. Compute output of the critic network at time t, J(t);
3. Compute output of the action network at time t, A(t);
4. Compute output of the model network at time t+1, Y(t+1);
5. Compute output of the critic network at time t+1, J(t+1);
6. Compute the critic network error at time t, E(t),
7. Update the critic network's weights using the backpropagation algorithm;
8. Repeat step 2 to 7.

In the action network's training cycle, an incremental learning is also carried out using backpropagation algorithm, as in the critic network's training cycle above. The list of operations for the action network's training cycle is almost the same as that for the critic network's cycle above. However, instead of using Equation (3.3) and/or (3.7) and ∂J/∂WC, ∂J/∂A and ∂A/∂WA are used for updating the action network's weights. The action network's training cycle is repeated N2 times while keeping the critic network's weights WC fixed. N1 and N2 are the lengths of the corresponding training cycles.

Tuning algorithm and stability analysis

For linear time invariant systems it is straightforward to examine stability by investigating the poles in the s-plane. However, stability of a nonlinear dynamic systems is much more complex, thus the stability criteria and tests are much more difficult to apply than those for linear time invariant systems[21]. For general nonlinear continuous time systems, the model is

$$\begin{aligned} \dot{x} &= f[x(t),u(t)] \\ y &= g[x(t),u(t)] \qquad (8) \end{aligned}$$

where the nonlinear differential equation is in state variable form, x(t) is the state vector and u(t) is the input and the second equation y(t) is the output of the system.

Creative controller and nonlinear dynamic system

For a creative controller, the task control center and the dynamic database are not time-variable system; therefore, the adaptive critic learning component determines stability of the creative controller. As it is discussed in the previous section, the adaptive critic learning is based on critic and action network designs, which are originated from artificial neural network (ANN), thus stability of the system is determined by stability of the neural networks (NN) or convergence of the critic network and action network training procedure.

The creative controller proposed in this thesis is a nonlinear system as its types. It is not realistic to explore all the possibilities of the nonlinear systems and prove that the controller is in a stable state. We use robot arm manipulators to explain a large class of problems known as tracking in this study. The objective of tracking is to follow a reference trajectory as closely as possible. This may also be called optimal control since we optimize the tracking error over time.

The adaptive critic controller architecture shown in Figure 8 [16] is a combination of an action network that produces the control input for the system, and a critic network that provides an adaptive-learning signal, and a fixed gain controller in the performance measure loop which uses an error based on the given reference trajectory. The further discussion of stability of the adaptive critic control is based on Lewis's adaptive critic feedback controller. Here we interpret Lewis's proof on stability of the adaptive critic learning structure[22.]

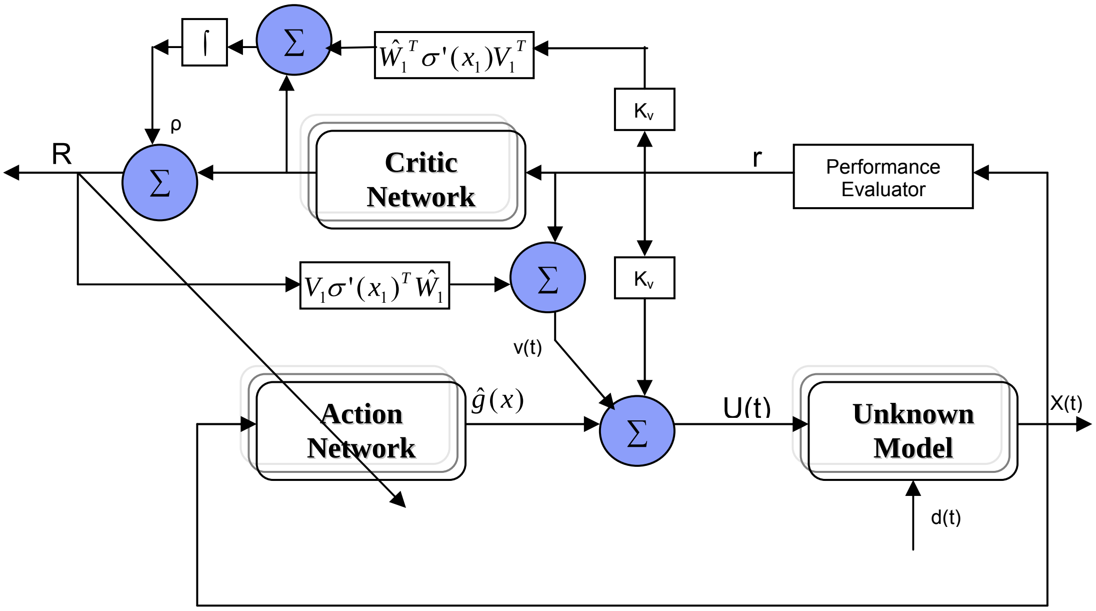


Figure 8 Adaptive critic feedback controller - control schema [10]

Critic and Action NN Weights Tuning Algorithm

In adaptive critic learning controller, both the critic network and action network use multilayer NN. Multilayer NN are nonlinear in the weights V and so weight tuning algorithms that yield guaranteed stability and bounded weights in closed-loop feedback systems have been difficult to discover until a few years ago.

Here, we interpret Lewis's results on stability of the adaptive critic control scheme as shown in Figure 8, [15]. Consider a mn-th order multi-input and multi-output system given by the Brunovsky form

$$\begin{aligned} &\dot{x}_1 = x_2 \\ &\vdots \\ &\dot{x}_{n-1} = x_n \qquad\qquad (9) \\ &\dot{x}_n = g(x) + u(t) + d(t) \\ &y = x_1 \end{aligned}$$

with state $x = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T$, with u(t) the control input to the plant, d(t) denotes the unknown disturbance with a known upper bound b_d, g(x):R^n→Rm unknown smooth functions and output tracking y.

Given a desired trajectory and its derivatives values

$$x_d(t) = \begin{bmatrix} x_d & \dot{x}_d & \cdots & x_d^{n-1} \end{bmatrix}, \quad (10)$$

define the tracking error as

$$e(t) = x(t) - x_d(t) \qquad (11)$$

and the filtered tracking error r(t) by

$$r = \dot{e} + \Lambda e \qquad (12)$$

with Λ>0 a positive definite design parameter matrix.

A choice of a critic signal R is

$$R = \hat{W}_1^T \sigma(x_1) + \rho\,, \qquad (13)$$

where ρ is an adaptive term detailed later and the first term is the output of the critic NN. The actual weights are denoted $\hat{W}_1$.

The dynamics of the performance measure signal can be written

$$\dot{r} = g(e, x_d^{(n-1)}) + u(t) + d(t) \qquad (14),$$

where $g(e, x_d^{(n-1)})$ is a complex nonlinear function of error vector e and the (n-1)th derivative of the trajectory x_d. According to the approximation properties of NN, the continuous nonlinear function can be expressed as

$$g(e, x_d^{(n-1)}) = W_2^T \sigma(x_2) + \varepsilon(x_2) \qquad (15)$$

where the NN reconstruction error $\varepsilon(x_2)$ is bounded by a known constant ε_N. The ideal weight W_2 for g(.) are unknown. The functional estimate for $g(e, x_d^{(n-1)})$ can be given by a second NN as

$$\hat{g}(e, x_d^{(n-1)}) = \hat{W}_2^T \sigma(x_2) \qquad (16)$$

From the adaptive critic learning architecture shown in Figure 4.8, the control input u(t) is given by

$$u(t) = -K_v r - \hat{g}(e, x_d^{(n-1)}) + v(t), \qquad (17)$$

where K_v is a gain matrix, generally chosen diagonal; v(t) is a robustifying signal to compensate for unmodelled unstructured disturbances d(t) and offset the NN functional reconstruction error ε(x).

$$\dot{r} = -K_v r + \tilde{g}(e, x_d^{(n-1)}) + d(t) + v(t) \qquad (18)$$

where the functional estimation error is defined as

$$\tilde{g}(e, x_d^{(n-1)}) = g(e, x_d^{(n-1)}) - \hat{g}(e, x_d^{(n-1)}). \qquad (19)$$

Using (16), (17) and (18), the dynamics for the performance measure can be expressed as

$$\dot{r} = -K_v r + \widetilde{W}_2^T \sigma(x_2) + \varepsilon(x_2) + d(t) + v(t) \qquad (20)$$

with the weight estimation error $\widetilde{W}_2 = W_2 - \hat{W}_2$.

The main result of Lewis's paper is to show how to adjust the weights of both critic NN and action NN to guarantee closed-up stability. Let the control action u(t) be provided by (17) and the robustifying term be given by

$$v(t) = -k_z . \frac{V_1 \sigma'(x_1) \hat{W}_1 R + r}{\left\| V_1 \sigma'(x_1) \hat{W}_1 R + r \right\|}. \qquad (21)$$

with $k_z > b_d$. Let the critic signal be provided by

$$R = \hat{W}_1^T \sigma(x_1) + \rho \qquad (22)$$

Let the weight tuning for the critic NN and the action NN be

$$\dot{\hat{W}}_1 = -\sigma(x_1) R^T - \hat{W}_1 \quad (23)$$

$$\dot{\hat{W}}_2 = \Gamma \sigma(x_2).(r + V_1 \sigma'(x_1)^T \hat{W}_1 R)^T - \Gamma \hat{W}_2 \qquad (24)$$

with $\Gamma = \Gamma^T > 0$. Finally let the auxiliary adaptive term ρ be tuned by the following

$$\dot{\rho} = \hat{W}_1^T [2\sigma(x_1) + \sigma'(x_1) V_1^T K_v r] \qquad (25)$$

Then the errors r, $\widetilde{W}_1, \widetilde{W}_2$ are *Uniformly Ultimately Bounded* (*UUB*). Moreover, the performance measure r(t) can be arbitrary small by increasing the fixed control gain K_v.

Stability Analysis

The proof of the theorem above is given in the following by Lewis [15]. From equations in (A.6), $\dot{L}$ is negative outside a compact set. According to a standard Lyapunov theorem extension, it can be concluded that the tracking error r(t) and the NN weights estimates $\dot{L} \leq 0$, $\dot{L} \leq 0$ are *Global Uniformly Ultimately Bounded (GUUB).*

3. CREATIVE CONTROL MOBILE ROBOT SCENEARIOS

Suppose a mobile robot is used for urban rescue as shown in Figure 9 [10]. It waits at a start location until a call is received from a command center. Then it must go rescue a person. Since it is in an urban environment, it must use the established roadways. Along the roadways, it can follow pathways. However, at intersections, it can choose various paths to go to the next block. Therefore, it must use different criteria at the corners. The overall goal is to arrive at the rescue site with minimum distance or time. To clarify the situations consider the following steps.

1. Start location – the robot waits at this location until it receives a task command to go to a certain location.
2. Along the path, the robot follows a road marked by lanes. It can use a minimum mean square error between its location and the lane location during this travel.
3. At intersections, the lanes disappear but a database gives a GPS waypoint and the location of the rescue goal.

This example requires the use of both continuous and discrete tracking, a database of known information and multiple criteria optimization. It is necessary to add a large number of real-world issues including position estimation, perception, obstacles avoidance, communication, etc.

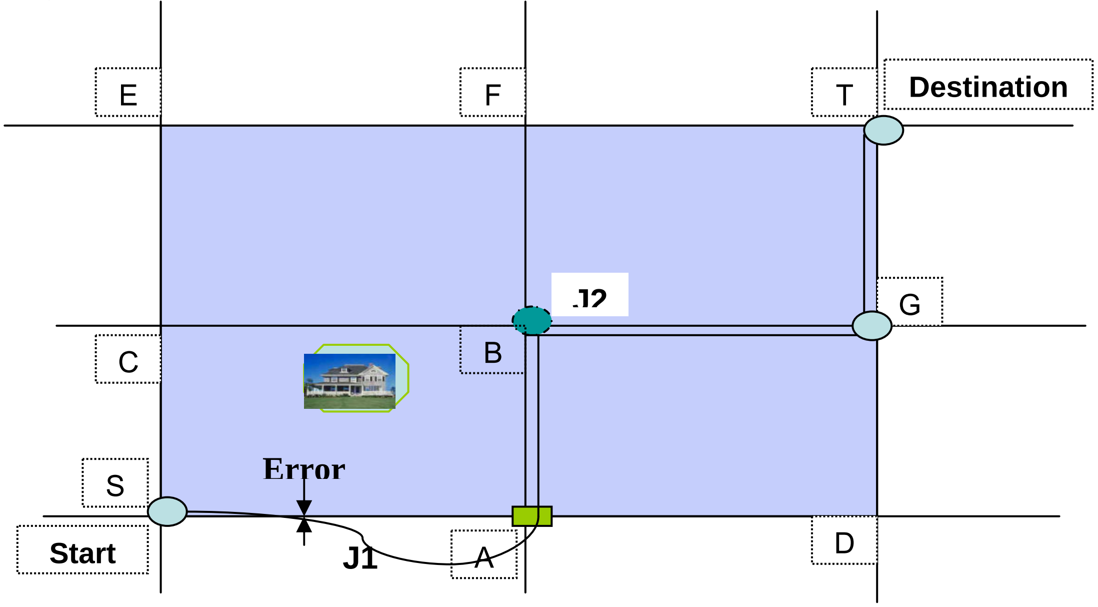


Figure 9 Simple urban rescue site

Scenarios

In an unstructured environment as shown in Figure 9, we assume that information collected about different potions of the environment could be available to the mobile robot, improving its overall knowledge. As any robot moving autonomously in this environment must have some mechanism for identifying the terrain and estimating the safety of the movement between regions (blocks), it is appropriate for a coordination system to assume that both local obstacle avoidance and a map-building module are available for the robot which is to be controlled. The most important module in this system is the adaptive system to learn about the environment and direct robot action, and then it has the necessary capabilities to allow good behaviors [23]

Using Global Position System (GPS) to measure the robot position and the distance from the current site to the destination and provide part of information for the controller to make decision on what to do at next move. GPS system also provides the coordinates of the obstacles for the learning module to learn the map, and then try to avoid the obstacles when navigating through the intersections A, B or G, D to destination T.

Task control center

The task control center (TCC) acts a decision-making command center. It takes perception information from sensors and other inputs to the creative controller and derives the criteria functions. We can decompose the robot mission at the urban rescue site shown as Figure 9 into sub-tasks as shown in Figure 10. Moving the robot between the intersections, making decisions is based on control-center-specified criteria functions to minimize the cost of mission. It’s appropriate to assume that J1 and J2 are the criteria functions that the task control center will transfer to the learning system at the beginning of the mission from the Start point to Destination (T). J1 is a function of t related to tracking error. J2 is to minimize the distance of the robot from A to T since the cost is directly related to the distance the robot travels.

- From Star (S) t to intersection A: robot follow the track SA with the J1 as objective function

- From intersection A to B or D: which one will be the next intersection, the control center takes both J1 and J2 as objective functions.

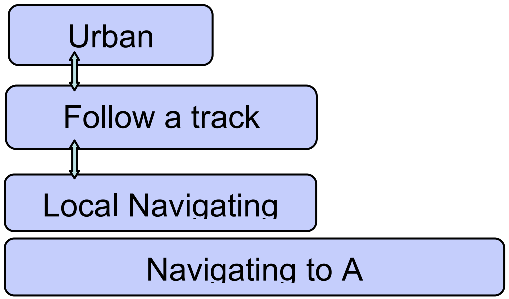


Figure 10 Mission decomposition diagram

Dynamic databases

Dynamic databases could store task-oriented environment knowledge, adaptive critic learning parameters and other related information on accomplishing the mission. In this scenario, a robot is commanded to reach a dangerous site to conduct a rescue task. The dynamic databases saved a copy of the GPS weight points S, A, B, C, D, E, F, G and T. The map for direction and possible obstacle information is also stored in the dynamic databases. A copy of the model parameters can be saved in the dynamic database as shown in the simplified database Figure 11. The action model will be updated in the dynamic database if the current training results are significantly superior to the previous model stored in the database.

Database fields	
Field	**Description**
MODEL_ID	Action model ID
MODEL_NAME	Action model name
UTILITY_FUN	Utility function
CRITERIA_FUN	Criteria function
…	…
Adaptive Critic Training Parameters	
INPUT_CRITIC	Input to critic network
DELT_J	J(t+1)-J(t)
…	…

Figure 11 Semantic dynamic database structure

Robot Learning Module

Initial plan such as road tracking and robot navigating based on known and assumed information, and then incrementally revises the plan as new information is discovered about the environment. The control center will create criteria functions according to the revised information of the world through the user interface. These criteria functions along with other model information of the environment will be input to the learning system. There is a data transfer module from the control center to the learning system as well as a module from learning system to the dynamic database. New knowledge is to explore and learn, training according to the knowledge database information and then decide which to store in the dynamic database and how to switch the criteria. The simplest style in the adaptive critic family is heuristic dynamic programming (HDP) is shown in Figure 8. This is NN on-line adaptive critic learning. There is one critic network, one action network and one model network in the learning structure. U(t) is the utility function. R is the critic signal as J (criteria function). The learning structure and the parameters are saved a copy in the dynamic database for the system model searching and updating. The system learning will be speeded tremendously by time and iterations.

4. CONCLUSIONS AND RECOMMENDATIONS

The creative learning system is proposed structurally and established on adaptive critic learning system acted as a component of the diagram. The creative learning structure is also composed of task control center and dynamic knowledge databases. The task control center is a decision-making command center for the intelligent creative learning system. Dynamic knowledge database integrates task control center and adaptive critic learning algorithm into one system and makes adaptive critic learning adaptable. It also provides a knowledge domain for task command center to perform decision-making. Furthermore, creative learning is used to explore the unpredictable environment, permit the discovery of unknown problems. By learning the domain knowledge, the system should be able to obtain the global optima and escape local optima. It generalizes the highest level of human learning – imagination.

REFERENCES

1. D. P. Bertsekas, Dynamic Programming and Optimal Control, Vol. I, Second Edition, Athena Scientific, Belmont, MA, 2000, pp. 2, 364.
2. D. White and D. Sofge, Handbook of Intelligent Control, Van Nostrand, 1992
3. P.J. Werbos, "Tutorial on Neurocontrol, Control Theory and Related Techniques: From Backpropagation to Brain-Like Intelligent Systems," *the Twelth International Conference on Mathematical and Computer Modelling and Scientific Computing (12th ICMCM & SC)*, http://www.iamcm.org/pwerbos/, 1999.
4. B. Widrow, N. Gupta, and S. Maitra, "Punish/reward: Learning with a Critic in Adaptive Threshold Systems," *IEEE Trans. Systems, Man, Cybemetics*, v.5 pp. 455-465, 1973.
5. X. Pang, J. Werbos, "Generalized Maze Navigation: SRN Critics Solve What Feedforward or Hebbian Nets Cannot", *Systems, Man, and Cybernetics, IEEE International Conference on*, pp.1764 -1769, v.3, 1996.
6. P. Werbos, "Backpropagation and Neurocontrol: a Review and Prospectus," *IJCNN Int Jt Conf Neural Network*, pp.209-216,1989.
7. Jaksa, R. and P. Sinc 簪, L*arge Adaptive Critics and Mobile Robotics.* July 2000.
8. Syam, R., et al. Control of Nonholonomic Mobile Robot by an Adaptive Actor-Critic Method with Simulated Experience Based Value-Functions. in Proc. of the 2002 IEEE International Conference on Robotics and Automation. 2002.
9. Liao, X. and E. Hall. Beyond Adaptive Critic - Creative Learning for Intelligent Autonomous Mobile Robots. in Intelligent Engineering Systems Through Artificial Neural Networks, ANNIE, in Cooperation with the IEEE Neural Network Council. 2002. St. Louis - Missouri.
10. Liao, X., et al. Creative Control for Intelligent Autonomous Mobile Robots. in Intelligent Engineering Systems Through Artificial Neural Networks, ANNIE. 2003.
11. Ghaffari, M., Liao, X., Hall, E. A Model for the Natural Language Perception-based Creative Control of Unmanned Ground Vehicles. in SPIE Conference Proceedings. 2004.
12. Widrow, B. and M.M. Lamego, N*eurointerfaces.* Control Systems Technology, IEEE Transactions on, 2002. 1**0(**2): p. 221 -228.
13. Yen, G.G. and P.G. Lima. Dynamic Database Approach for Fault Tolerant Control Using Dual Heuristic Programming. in Proceedings of the American Control Conference. May 2002.
14. Simmons, R., T*ask Control Architecture.* http://www.cs.cmu.edu/afs/cs/project/ TCA/www/TCA-history.html, 2002.
15. Lewis, F.L., S. Jagannathan, and A. Yesildirek, N*eural Network Control of Robot manipulators and Nonlinear Systems.* 1999, Philadelphia: Taylor and Francis.
16. Campos, J. and F.L. Lewis. Adaptive Critic Neural Network for Feedforward Compensation. in American Control Conference, 1999. Proceedings of the 1999. 1999.
17. Ferrari, S., Algebraic and Adaptive Learning in Neural Control System. Nov. 2002, Princeton University.
18. Venayagamoorthy, G.K., R.G. Harley, and D.C. Wunsch, Comparison of Heuristic Dynamic Programming and Dual Heuristic Programming Adaptive Critics for Neurocontrol of a Turbogenerator. IEEE Transactions on Neural Networks, May 2002. 1**3(**3): p. 764-773.
19. Lendaris, G.G., C. Paintz, and T. Shannon. More on Training Strategies for Critic and Action Neural Networks in Dual Heuristic Programming Method. in Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, 1997 IEEE International Conference on. 1997.
20. Lendaris, G.G. and C. Paintz. Training Strategies for Critic and Action Neural Networks in Dual Heuristic Programming Method. in Neural Networks, International Conference on. 1997.
21. Stubberud, A.R. and S.C. Stubberud, S*tability,* in H*andbook of Industrial Automation,* R.L. Shell and E.L. Hall, Editors. 2000, MARCEL DEKKER, INC.: New York.

22. Lewis, F.L., D.M. Dawson, and C.T. Abdallah, R*obot Manipulator Control: Theory and Practice*. 2nd Rev&Ex edition ed. 2003: Marcel Dekker (December 1, 2003). 430.
23. Brumitt, B.L., A Mission Planning System for Multiple Mobile Robots in Unknown, Unstructured, and Changing Environments. 1998, Carnegie Mellon University.