Learning for Intelligent Mobile Robots

E. L. Hall, X. Liao, 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/

ABSTRACT

Unlike intelligent industrial robots which often work in a structured factory setting, intelligent mobile robots must often operate in an unstructured environment cluttered with obstacles and with many possible action paths. However, such machines have many potential applications in medicine, defense, industry and even the home that make their study important. Sensors such as vision are needed. However, in many applications some form of learning is also required. The purpose of this paper is to present a discussion of recent technical advances in learning for intelligent mobile robots.

During the past 20 years, the use of intelligent industrial robots that are equipped not only with motion control systems but also with sensors such as cameras, laser scanners, or tactile sensors that permit adaptation to a changing environment has increased dramatically. However, relatively little has been done concerning learning. Adaptive and robust control permits one to achieve point to point and controlled path operation in a changing environment. This problem can be solved with a learning control. In the unstructured environment, the terrain and consequently the load on the robot's motors are constantly changing. Learning the parameters of a proportional, integral and derivative controller (PID) and artificial neural network provides an adaptive and robust control. Learning may also be used for path following. Simulations that include learning may be conducted to see if a robot can learn its way through a cluttered array of obstacles. If a situation is performed repetitively, then learning can also be used in the actual application.

To reach an even higher degree of autonomous operation, a new level of learning is required. 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. Examples from a variety of intelligent mobile robot applications are also presented. The significance of this work is in providing a greater understanding of the applications of learning to mobile robots that could lead to many applications.

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

1. INTRODUCTION

Intelligent robots are an ideal, a vision. All one has to do to see the intelligent robot model is to look in a mirror. Ideally, all intelligent robots move dexterously, smoothly, precisely, using multiple degrees of coordinated motion and do something like a human but that a human now doesn't have to do. They have sensors that permit them to adapt to environmental changes. They learn from the environment or from humans without making mistakes. They mimic expert human responses. They perform automatically, tirelessly, and accurately. They can diagnose their own problems and repair themselves. They can reproduce, not biologically but by robots making robots. They can be used in industry for a variety of applications. A good intelligent robot solution to an important problem can start an industry and spin off a totally new technology. For example, imagine a robot that can fill your car with gas, mow your lawn, a car that can drive you to work in heavy traffic, and a machine that repairs itself when it breaks down, a physician assistant for microsurgery that reconnects 40,000 axons from a severed finger nerve.

Intelligent robots are also a reality. A robot is making measurements on Mars. Some hospitals have food delivery and physician surgery aid robots. Industrial robots are now commonly used. Many more intelligent prototypes have been built. Typical applications are: high speed spot welding robots, precise seam welding robots, spray painting robots moving around the contours of an automobile body, robots palletizing of variable size parcels, robots loading and unloading machines.

A brief introduction to intelligent robots is given in Section 2. The adaptive and robust control approach is described in Section 3. The creative control is descried in Section 4. Conclusions and recommendations are given in Section 5.

2. INTELLIGENT ROBOTS

2.1 Introduction to Intelligent Robots

The components of an intelligent robot are a manipulator, sensors and controls. However, it is the design architecture or the combination of these components, the paradigms programmed into the controller, the foresight and genius of the system designers, the practicality of the prototype builders, the professionalism and attention to quality of the manufacturing engineers and technicians that makes the machine intelligent.

Just where is the intelligence in an intelligent robot? Is it in the controller just as the intelligence of a human is in the neural connections of the brain? Is it in the sensors that permit the robot to adapt? Is it in the manipulator which actually does the work? Or is it some remarkable architectural combination of these components? When are intelligent robots needed? When a task is repetitive such as making a million parts per year, automation is needed. The most suitable automation may be an intelligent robot. In addition, when a task is hazardous for humans, automation is needed. The best solution may be an intelligent remote manipulator. Finally, when an industry needs to be internationally competitive in cost and quality, automation is needed. Again, the intelligent robot may play a significant part in the solution.

What are the benefits from using intelligent robots? Robots can do many tasks now. However, the tasks that cannot be easily done today are often characterized by a variable knowledge of the environment. Location, size, orientation, shape of the work piece as well as of the robot must be known accurately to perform a task. Obstacles in the motion path, unusual events, breakage of tools, also create environmental uncertainty. Greater use of sensors and more intelligence should lead to a reduction of this uncertainty and because the machines can work 24 hours a day, should also lead to higher productivity.

Hall and Hall[1] defined an intelligent robot as one that responds to changes to its environment through sensors connected to a controller. Dynamic Programming (DP) is perhaps the most general approach for solving optimal control problems. Adaptive Critics Design (ACD) offers a unified approach to deal with the controller's nonlinearity, robustness, and reconfiguration for a system whose dynamics can be modeled by a general ordinary differential equation. ANN and Back propagation (BP) made it possible for ACD implementation[2]. However, in order to develop "brain-like intelligent control"[3], it is not enough to just have the adaptive critic portion. Here we proposed a novel algorithm, called Creative Learning (CL) to fill this gap.

2.2 Computed-Torque Controller

The third section of this paper presents the development of a proportional-plus-derivative (PD) Computed-Torque (CT) and proportional-plus-integral-plus-derivative (PID) CT controllers[4]. A dynamic simulation, based on a framework developed by Lewis, et al.[5], was conducted and modified to suit the navigation of a wheeled mobile robot (WMR). The simulation software takes, as input, the desired robot path from the navigation algorithm described in a previous paper by the authors[6]. The simulation software produced the suitable control torques. This simulation was developed by using Matlab and C++.
Different controllers had been developed for the motion of robot manipulators, however, not until recently where there has been an interest in moving the robot itself, not only its manipulators, hence, WMR motion control is a new research area.

Shim and Sung[7] proposed a WMR asymptotic control with driftless constraints based on empirical practice using the WMR kinematic equations. They showed that with the appropriate selection of the control parameters, the numerical performance of the asymptotic control could be effective. The trajectory control of a wheeled inverse pendulum type robot had been discussed by Yun-Su and Yuta[8], their control algorithm consists of balance and balance and velocity control, steering control and straight line tracking control for navigation in a real indoor environments.

Rajagopalan and Barakat[9] developed a computed torque control scheme for Cartesian velocity control of WMRs. Their control structure can be used to control any mobile robot if its inverse dynamic model exists. A discontinuous stabilizing controller for WMRs with nonholonomic constraints where the state of the robot asymptotically converges to the target configuration with a smooth trajectory was presented by Zhang and Hirschorn[10]. A path tracking problem was formulated by Koh and Cho[11] for a mobile robot to follow a virtual target vehicle that is move exactly along the path with specified velocity. The driving velocity control law was designed based on bang-bang control considering the acceleration bounds of driving wheels and the robot dynamic constraints in order to avoid wheel slippage or mechanical damage during navigation. Zhang et al.[12] employed a dynamic modeling to design a tracking controller for a differentially steered mobile robot that is subject to wheel slip and external loads.

A sliding mode control was used to develop a trajectory tracking control in the presence of bounded uncertainties[13]. A solution for the trajectory tracking problem for a WMR in the presence of disturbances that violate the nonholonomic constraint is proposed later by the same authors based on discrete-time sliding mode control[14-15].

An electromagnetic approach was investigated for path guidance for a mobile-robot-based automatic transport service system with a PD control algorithm was presented by Wu et al.[16]. Jiang et al.[17] developed a model-based control design strategy that deal with global stabilization and global tracking control for the kinematic model a nonholonomic WMR in the presence of input saturations. Adaptive robust controllers were also proposed for globally tracking problem for of the dynamic of the non-holonomic systems with unknown dynamics[18]. However, real time adaptive control is not common in practical applications due partly to the stability problems associated with it[19].

A fuzzy logic controller had also been tried for WMRs navigation. Montaner and Ramirez-Serrano[20] developed a fuzzy logic controller that can deal with the sensors inputs uncertainty and ambiguity for a direction and velocity maneuvers. A locomotion control structure was developed based the integration of an adaptive fuzzy-net torque controller with a kinematic controller to deal with unstructured unmodeled robot dynamics for a non-holonomic mobile robot cart[21]. Toda et al.[22] employed a sonar-based mapping of crop rows and fuzzy logic control-based steering for the navigation of a WMR in an agricultural environment. They constructed a crop row map from the sonar readings and transferred it to the fuzzy logic control system, which steers the robot along the crop row. A local guidance control method for WMR using fuzzy logic for guidance, obstacle avoidance and docking of a WMR was proposed by Vázquez and Garcia[23] the method provide a smooth but not necessary optimal solution.

2.3 Intelligent Control Theory and Neurocontroller

In order to design intelligent robot controllers, one must also provide the robot with a means of responding to problems in both temporal and spatial context. It is the goal of the robot researcher to design a neural learning controller to utilize the available data from the repetition in robot operations. The neural learning controller in Fig. 1, based on the recurrent network architecture, has the time-variant feature that once a trajectory is learned, it should learn a second one in a shorter time.

An artificial neural network (ANN) can be used to obtain the system model identification that can be used to design the appropriate intelligent robot controller. Once the real system model is available, they can also be used directly in design of the controller[24]. In Fig. 1, the time-variant, recurrent network will provide the learning block, or primary controller, for the inverse dynamic equations discussed above. The network compares the desired trajectories with continuous paired values for the multi-axis robot, at every instant in a sampling period. The new trajectory parameters are then combined with the error signal from the secondary controller (feedback controller) for actuating the robot manipulator arm. Neural networks can be applied either as a system identification model or as a control for the robot controller described in Fig. 1. ANNs can be used to obtain the system model identification that can be used to design the appropriate controller. Once the real system model is available, they can also be used directly in design of the controller. Neural network approaches to robot control are discussed in general by Psaltis *et al.* [25], and Yabuta and Yamada [26]. These approaches can be classified as: (1) Supervised control, a trainable controller that, unlike the old teaching pendant, allows responsiveness to sensory inputs; (2) Direct inverse control is trained for the inverse dynamic of the robot; (3) Neural adaptive control, neural nets combined with adaptive controllers result in greater robustness and the ability to handle nonlinearity; (4) Backpropagation of utility involves information flowing backward through time. (5) Adaptive critic method uses a critic evaluating robot performance during training. This is a very complex method that requires more testing.

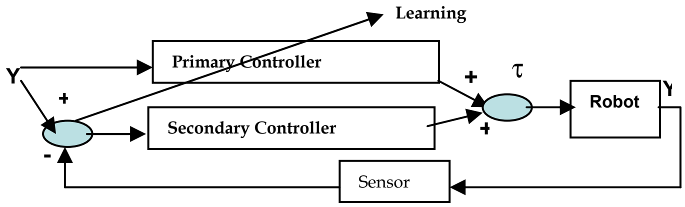

Figure 1. Recurrent neural learning controller

3. ADAPTIVE AND ROBUST CONTROL WITH PD AND PID CT CONTROLLERS

This section describes the development of proportional derivative (PD) and proportional, integral and derivative (PID), CT controllers for WMRs for autonomous navigation in outdoor environments[4]. To provide focus for the presented model, a particular configuration will be addressed: the Bearcat III robot, which has two driven fixed wheels and a caster. Bearcat III was developed at the Center for Robotics Research, at the University of Cincinnati. A simplified kinematic and dynamic model is derived from the Newton-Euler method by considering the velocity along the x and y axes and the angular velocity, with the robot center of mass as a reference point.

$$M(q)\ddot{q} + J(q,\dot{q})\dot{q} + F = \tau \qquad (1)$$

where:

$$q = \begin{bmatrix} x_c \\ y_e \\ \theta \end{bmatrix}$$

$$F = \begin{bmatrix} \dfrac{-f_n er}{d} \\ \dfrac{-f_n er}{d} \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}$$

$$M(q) = \begin{bmatrix} \dfrac{(mr^2\cos\theta + 2J_0\cos\theta)}{2r} & \dfrac{(mr^2\sin\theta + 2J_0\sin\theta)}{2r} & \dfrac{(mr^2 ed\sin^2\theta - mr^2 ed\sin\theta\cos\theta + J_c r^2 + 2J_0 d^2)}{2rd} \\ \dfrac{(mr^2\cos\theta + 2J_0\cos\theta)}{2r} & \dfrac{(mr^2\sin\theta + 2J_0\sin\theta)}{2r} & \dfrac{(mr^2 ed\sin^2\theta - mr^2 ed\sin\theta\cos\theta - J_c r^2 - 2J_0 d^2)}{2rd} \end{bmatrix}$$

$$J(q,\dot{q}) = \begin{bmatrix} \dfrac{-J_0\dot{\theta}\sin\theta}{r} & \dfrac{J_0\dot{\theta}\cos\theta}{r} & \dfrac{-mre\dot{\theta}\cos\theta(\sin\theta + \cos\theta)}{2} \\ \dfrac{-J_0\dot{\theta}\sin\theta}{r} & \dfrac{J_0\dot{\theta}\cos\theta}{r} & \dfrac{-mre\dot{\theta}\cos\theta(\sin\theta + \cos\theta)}{2} \end{bmatrix}$$

where

x_c : is the x- coordinate of the robot's center of mass.

y_e : is the y- coordinate of the center of the wheel axle.

θ : is the orientation angle of the robot.

f_n : is the resultant normal force.

e : is the distance between the robot's center of mass and the center of the wheel axle.

r : is the radius of the fixed wheel.

d : is half the width of the robot.

τ_r : is the right motor torque, acting on the right wheel.

τ_l : is the left motor torque, acting on the left wheel.

m : is the robot's mass, excluding the wheels.

$J_0 : J_0 = J_w + m_w r^2$, where J_w is the inertia of the wheel, and m_w is the mass of the wheel.
J_c: is the robot's inertia, excluding the wheels.
For more details, please refer to Alhaj Ali's dissertation[4].

3.1 Computed-Torque (CT) Controllers

Eq. 1 can be written as follows[5]:

$$M(q)\ddot{q} + N(q,\dot{q}) = \tau \tag{2}$$

where $N(q,\dot{q})$ represent the nonlinear terms

The objective of a motion controller is to move the robot or the robot manipulator according to a desired motion trajectory $q_d(t)$, while the actual motion trajectory is defined as $q(t)$, the tracking error in this case can be defined as[5]:

$$e(t) = q_d(t) - q(t) \tag{3}$$

The Brunovsky canonical form can be developed by differentiating $e(t)$ twice and write it in terms of the state x[5]:

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u \tag{4}$$

Where:

$$u \equiv \ddot{q}_d + M^{-1}(q)(N(q,\dot{q}) - \tau)$$

$$x = \begin{bmatrix} e^T \\ \dot{e}^T \end{bmatrix}$$

To develop the CT controller, a linear system design procedure will be used in order to select a feedback control $u(t)$ that stabilizes the tracking error equation, then compute the torques needed for the motors using the inverse of the dynamic equation for the WMR or the two-link manipulator[5]:

$$\tau = M(q)(\ddot{q}_d - u) + N(q,\dot{q}) \tag{5}$$

Two types of CT controllers will be developed here, namely, the PD CT controller and the PID CT controller.

PD CT controller

A PD feedback for $u(t)$ with a derivative gain matrix K_v and a proportional gain matrix K_p produced the PD CT controller where the motor torques equal to[5]:

$$\tau = M(q)(\ddot{q}_d + K_v\dot{e} + K_p e) + N(q,\dot{q}) \tag{6}$$

Which has the tracking error dynamics $\ddot{e} = -K_v\dot{e} - K_p e$

The gain matrices need to be selected positive definite in order to keep the tracking error dynamics stable.

PID CT controller

The PD CT controller described before can be adjusted easily to a PID CT controller by adding an integrator gain matrix K_i to $u(t)$ as follows[5]:

$$\tau = M(q)(\ddot{q}_d + K_v\dot{e} + K_p e + K_i(\int e)) + N(q,\dot{q}) \tag{7}$$

Which has the tracking error dynamics $\ddot{e} = -K_v\dot{e} - K_p e$

The integrator gain needs to be chosen not too large in order to keep the tracking error stable. The PD CT and PID CT controllers are illustrated in Fig. 2, where $K = [K_p \quad K_v]$ in case of a PD CT controller and $K = [K_p \quad K_v \quad K_i]$ for a PID CT controller. As shown in the figure it has an outer tracking loop and an inner feedback linearization loop consist of the nonlinear component $N(q, \dot{q})$.

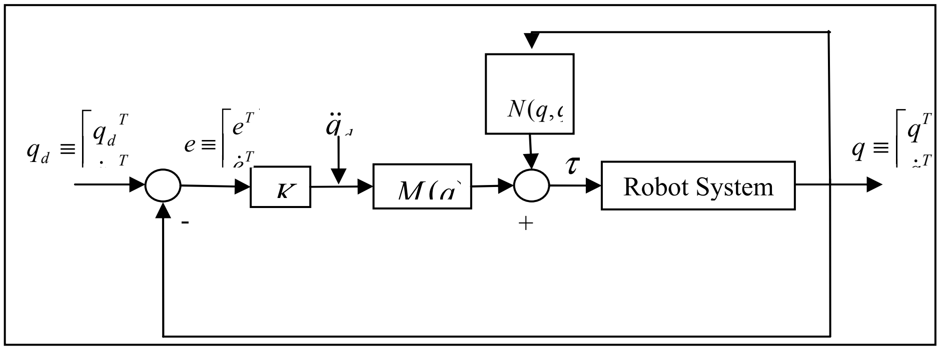

Figure 2. PD and PID CT controllers block diagram[5].

3.2 Computed-Torque (CT) Controllers Simulation: Simulation of the PD CT Controller for the WMR Motion

The function of this controller is to select the suitable motor torques so that the WMR will follow the desired motion trajectory $q_d(t)$. In case of a two-link manipulator, this is already available from the description of the task that the manipulator needs to perform. However, for a WMR this information is not available, $q_d(t)$ that will be used here is the desired path that has been produced from the navigation system described in part 1 of this paper[6].

Starting from the dynamic model described in Eq. 1, this can be written as illustrated in Eq. 5 in terms of a linear and a nonlinear components:

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau \tag{8}$$

Where $N(q, \dot{q})$ represent the nonlinear terms: $N(q, \dot{q}) = J(q, \dot{q})\dot{q} + F$

The simulation program developed has the following main components:

- The first component computes the desired WMR trajectory based on the input from the navigation system, the desired trajectory $q_d(t)$ is given as follows:

$$q_d = \begin{bmatrix} x_{cd} \\ y_{cd} \\ \theta_d \end{bmatrix} \tag{9}$$

Where:

x_{cd} : is the x-axis component of the desired position of the center of gravity of the WMR.

y_{cd} : is the y-axis component of the desired position of the center of gravity of the WMR.

θ_d : is the desired orientation of the WMR.

- The second component calculates the controller input by first calculating the tracking error between the desired trajectory $q_d(t)$ and the actual trajectory $q(t)$ where $q(t)$ is defined as follows:

$$q = \begin{bmatrix} x_c \\ y_c \\ \theta \end{bmatrix} \tag{10}$$

Where:

x_c : is the x-axis component of the actual position of the center of gravity of the WMR.

y_c : is the y-axis component of the actual position of the center of gravity of the WMR.

θ : is the actual orientation of the WMR.

Then computing the inertia term $M(q)$ and the nonlinear term $N(q,\dot{q})$ based on the WMR dynamic model described in Eq. 1. Finally, the motor torques is calculated using Eq. 6.

- The third component calculates the new position of the WMR using the state-space equation $\dot{x} = f(x,u)$ where the state-space position/velocity form is used[5]:

$$x \equiv \begin{bmatrix} q^T \\ \dot{q}^T \end{bmatrix} \tag{11}$$

$$\dot{x} = \begin{bmatrix} \dot{q} \\ -M^{-1}(q)N(q,\dot{q}) \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}(q) \end{bmatrix} \tau$$

This is used to update for the WMR actual position, as shown in Eq. 1, $M(q)$ is not square, hence, the Moore-Penrose generalized inverse is used refer to Alhaj Ali's dissertation[4].

The inputs to the PD CT controller simulation program are:

1. The desired motion trajectory $q_d(t)$.
2. Robot parameters.
3. Controller parameters k_p and k_v.

The outputs of the PD CT controller simulation program are:

4. The motor torques $\tau(t)$.
5. The actual path $q(t)$.

Several experiments had been conducted on the PD CT simulation software; the robot parameters were from Bearcat III. Various controller parameters were tested by sinusoidal desired motion trajectories. Experiments show that k_p and k_v need to be different for each component of the motion trajectory, q. Small positive values are required to obtain good results. The gain parameters, k_{p1}=2, k_{v1}=1, k_{p2}=0, k_{v2}=10, k_{p3}=2, and k_{v3}=1, give good results, as shown in the following figures. The tracking errors are in the range 0.00-0.04, as shown in Fig. 3. The desired versus the actual motion trajectory is shown in Fig. 4. As shown in the figures, the tracking errors are zero for both x and θ, with the actual motion trajectory following the desired motion trajectory for these two components. For y, the error oscillates around 0.02.

The selection process of proper parameters for the controller is a challenging task, since six parameters must be changed in each trial. Although x, y, and θ are independent of each other, they are, nevertheless, used to update the new values of the actual motion trajectories. Hence, they have an impact upon each other. Therefore, keeping the same controller parameters for a particular motion trajectory component does not imply that the actual motion trajectory will remain the same.

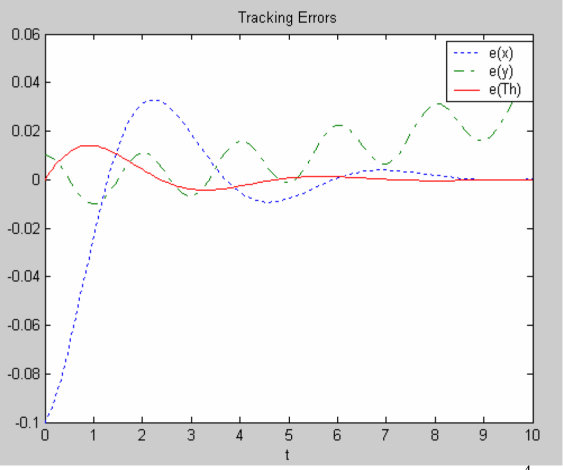

Figure 3: PD CT controller tracking errors versus time[4].

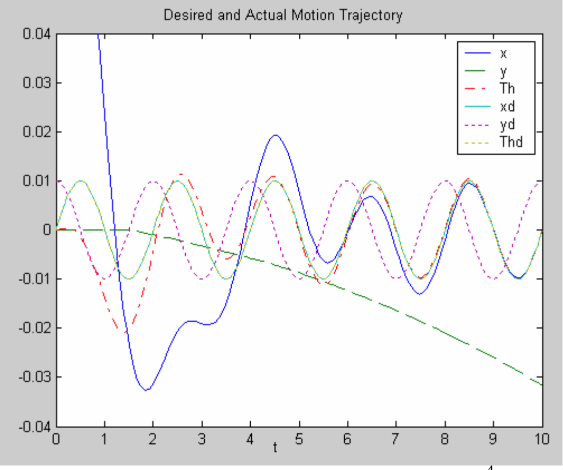

Figure 4: Desired versus actual motion trajectories[4].

3.3 Simulation of the PID CT Controller for WMR Navigation

The PID CT controller simulation program for the WMR motion developed in the same way the PD CT controller simulation program was developed except for the equations for calculating the motor torques; in this case, Eq. 7 was used for the calculations[4]. The inputs to the PID CT controller simulation program are:

1. The desired motion trajectory $q_d(t)$.
2. Robot parameters.
3. Controller parameters k_p, k_v and k_i.

The outputs of the PID CT controller simulation program are:

4. The motor torques $\tau(t)$.
5. The actual path $q(t)$.

Several experiments had been conducted on the PID CT simulation software, and different trajectories and controller parameters were tried. The results of the experiments show that the controller parameters need to be small positive numbers to obtain good results. It is also noteworthy that increasing k_p and fixing k_v and k_i, or increasing k_v and fixing k_p and k_i, reduces the tracking errors of θ and x, while it increases the tracking error of y. However, there is a limit to this increase, which is about 10.

Using very large, or zero, values for k_p, k_v, or k_i is not recommended. Additionally, the value of k_i must not be too large, as a condition for having a stable tracking error; k_p=2, k_v=1, and k_i=1 give very reasonable results, as shown in the following figures. The tracking error for θ is zero. For x, it oscillates around zero. For y, it starts at zero and increases to 0.35, as shown in Fig. 5. The desired versus the actual motion trajectories are shown in Fig. 6.

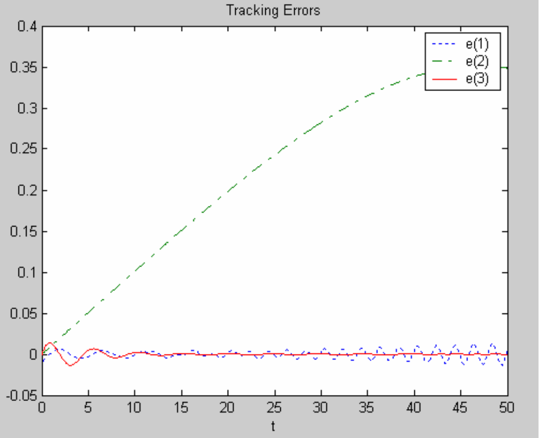

Figure 5: PID CT controller tracking errors versus time[4].

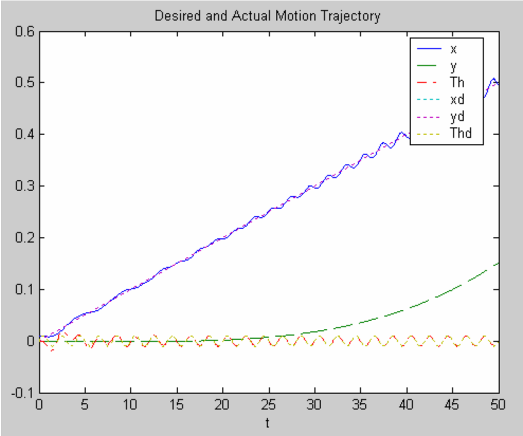

Figure 6: Desired versus actual motion trajectories[4].

4. CREATIVE LEARNING AND CONTROL

“Creative learning” provides a model, for understanding the kind of intelligence that exists in biological brains. A creative control architecture as shown in Fig. 7 is proposed in this paper according to the creative learning theory [27]. In this proposed diagram, there are three important components: task control center, criteria (critic) knowledge database, and learning system. 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: (1) Brain-like decision-making task control center, entails the capability of human brain decision-making; (2) Dynamic criteria 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; (4) Modeled and forecasted critic modules result in faster training network.

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. Furthermore, the kinematic model is also used to construct a model-based action in the framework of adaptive critic-action approach. In this algorithm, we build a criteria (critic) database to generalize the critic network and its training process. It is especially critical when the operation of mobile robots is in an unstructured environments. Another component in the diagram is the utility function for a tracking problem (error measurement). A creative controller is designed to integrate the domain knowledge and task control center into the adaptive critic controller. It needs to be a well-defined structure such as in the autonomous mobile robot application as the test-bed for the creative controller.

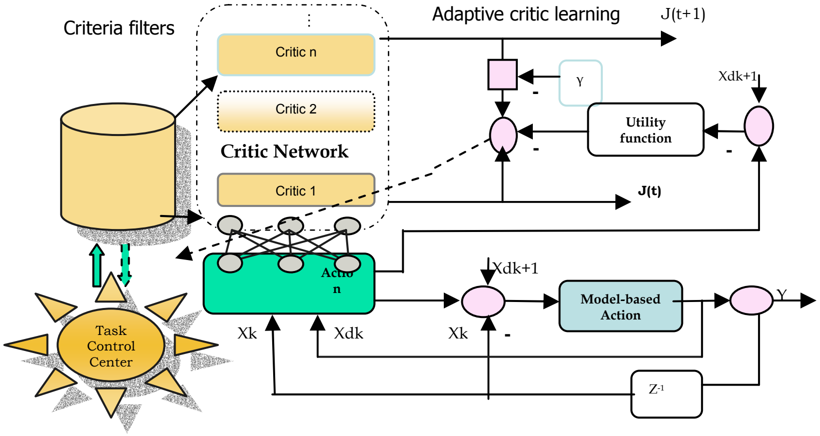

Figure 7. Proposed CL Algorithm Architecture

4.1 Adaptive Critic Control

Adaptive critic (AC) control theory is a component of creative learning theory. Werbos summarized recent accomplishments in neurocontrol as a "brain-like" intelligent system. It should contain at least three major general-purpose adaptive components: (1) an Action or Motor system, (2) an "Emotional" or "Evaluation" system or "Critic" and (3) an "Expectations" or "System Identification" component [28].

"Critic" serves as a model of the external environment to be controlled; solving an optimal control problem over time may be classified as adaptive critic designs (ACD). ACD is a large family of designs which learn to perform utility maximization over time. In dynamic programming, normally the user provides the function U($\underline{X}$(t), $\underline{u}$(t)) , an interest rate r**,** and a stochastic model. Then the analyst tries to solve for another function J($\underline{X}$(t)), so as to satisfy some form of the Bellman equation shown in Eq. (12) that underlies dynamic programming [3]:

$$J(\underline{X}(t)) = \max_{\underline{u}(t)}(U(\underline{X}(t),\underline{u}(t)) + < J(\underline{X}(t+1)) > /(1+r)) \qquad (12)$$

where "<>" denotes expected value.

In principle, any problem in decision or control theory can be classified as an optimization problem. Many ACDs solve the problem by approximating the function J. The most popular methods to estimate J in ACDs are heuristic dynamic programming (HDP), Dual Heuristic Programming (DHP) and Globalized DHP (GDHP) [28 29]. HDP and its ACD form have a critic network that estimates the function J (cost-to-go or strategic utility function) in the Bellman equation of dynamic programming, presented as follows:

$$J(t) = \sum_{k=0}^{\infty} \gamma^k U(t+k) \qquad (13)$$

where γ is a discount factor ($0<\gamma<1$), and U(.) is the utility function or local cost. An alternative approach referred to as Dual Heuristic Programming (DHP) has been proposed. Here, the critic network approximates the derivatives of the future cost with respect to the state variable. It has been proved that DHP is capable of generating smoother derivatives and has shown improved performance when compared to HDP [30 31]. Werbos first proposed the idea how to do GDHP [32]. Training the critic network in GDHP utilizes an error measure which is a combination of the error measures of HDP and DHP.

4.2 Task Control Center (TCC)

The task control center (TCC) can build task-level control systems for the creative learning system as shown in Fig. 8. [27] By "task-level", we mean the integration and coordination of perception, planning and real-time control to achieve a given set of goals (tasks). TCC provides a general task control framework, and it is intended to be used to control a wide variety of tasks and permits responsive actions based on mission commands, on interactions with other robots. 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 in the dynamic database matches the constraints on 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 Simmons[33], 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. 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.

4.3 Dynamic Knowledge Database (DKD)

A dynamic knowledge database serves as a "coupler" between a task control center and a nonlinear learning system. 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 scenarios) 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 back propagation through time[34] .

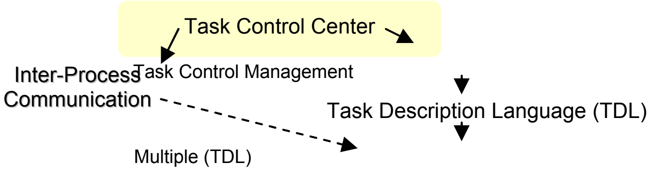

Figure 8. The structure of task control center

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 dynamic database is shown in Fig. 9 and 10.

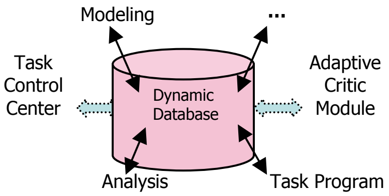

Figure 9. Functional structure of dynamic database

Database fields	
MODEL_ID	Action model ID
CRITERIA_FUN	Criteria function
Adaptive Critic Training Parameters	
INPUT_CRITIC	Input to critic network
DELT_J	J(t+1)-J(t)
…	…

Figure 10. Semantic database structure

4.4 Creative Control Mobile Robot Example

Suppose a mobile robot is used for urban rescue as shown in Fig. 11. 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 road ways 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.

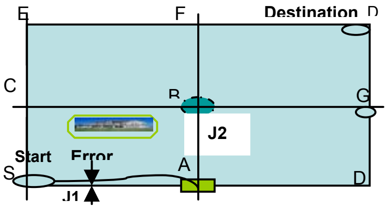

Figure 11. Simple urban rescue sites

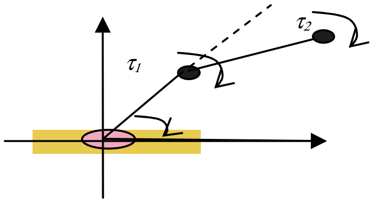

Figure 12. Two-link robot arm manipulator

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. Moving the robot between the intersections, decision making is based on the control-center-specified criteria functions to minimize the cost of the mission. It is 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 (S) to Destination (D). 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) to intersection A the robot should follow the track SA with the J1 as the objective function. From intersection A to B or D, the robot must decide which one will be the next intersection. The control center takes both J1 and J2 as objective functions.

The initial plan such as road tracking and robot navigating are based on known and assumed information, and then the plan is incrementally revised 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) as shown in Fig.13 . 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 improved tremendously by time and iterations.

The robot system designed to operate successfully in the unstructured environment must be able to learn the environment frequently. Since there will always be some delay between the acquisition of data information and the incorporation of that information into the control system, the creative control structure will facilitate faster learning and planning of the mission. More discussion is presented in Liao's dissertation[27] .

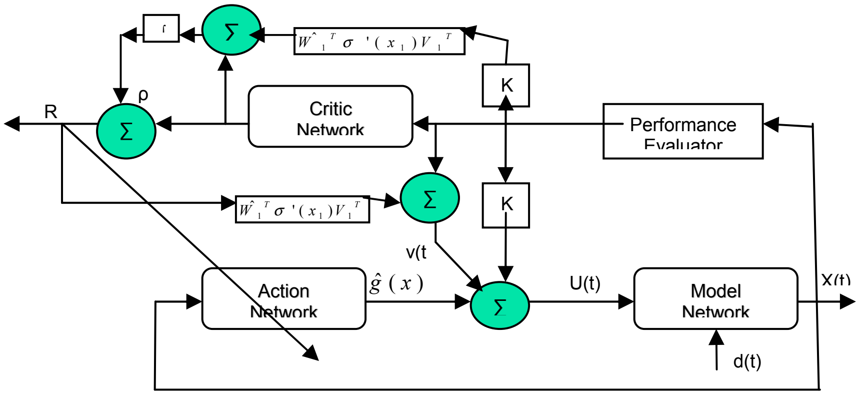

Figure 13. Adaptive critic learning structure [35]

4.5 Creative Controller and Simulation Results

As an ANN robot controller, the block diagram of the creative controller can be presented as shown in Fig. 14. A creative controller is integrated into the intelligent robot learning system. As an expermental study, we used a two-link robot arm maniputlator shown in Fig. 12 as one simulation example for adaptive critic learning. The dynamics of the two-link robot arm manipulater is shown as Eq. 14. The HDP in ACD family is used for learning portion as shown in Fig. 13 . The simulation results are shown in Fig. 15. When training with the adaptive crtic system, it is more stable and the tracking errors are smaller. The furthur results present a comparison with the adaptive critic learning techiques and more case studies for the system shown in Eq. 9 [27].

$$M(q)\ddot{q} + V(q,\dot{q}) + F(q,\dot{q}) + G(q) + \tau_d = \tau \qquad (14)$$

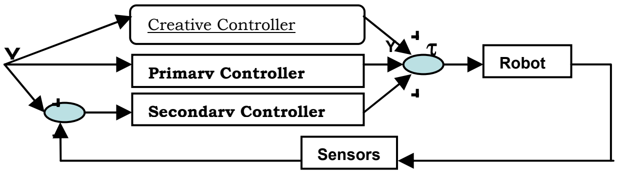

Figure 14. Creative controller structure

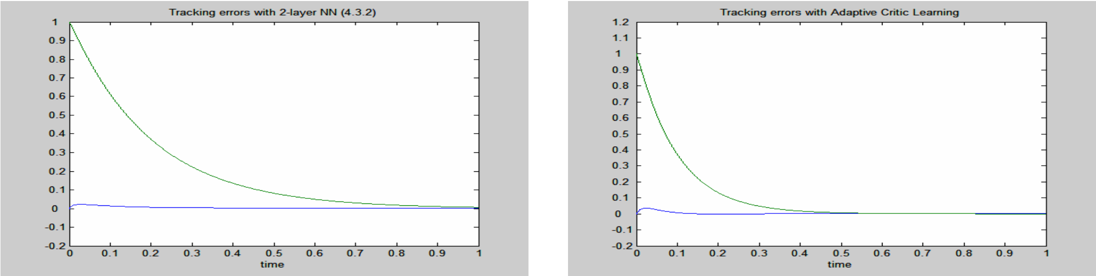

Figure 15 (a) Tracking error with 2-layer neural network vs. (b) Tracking error with adaptive critic learning (t=1sec)

5. CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH

The design of intelligent mobile robots is a challenging task. However, several learning methods were described in Section 4 that provides solutions for the point to point and controlled path motions even in an unstructured environment. The controllers select suitable control torques, so that the WMR will follow the desired path from a navigation algorithm described in a previous paper. The controllers are tested under various control parameters and motion trajectories. Simulation results show that, for a PD CT controller, the gain matrices need to be different for each component of the motion trajectory; small positive values are needed to get good results. Simulation results for a PID CT controller show that the gain matrices also need to be small positive numbers. k_p=2, k_v=1, and k_i=1 produce reasonable results. Selecting the proper parameters for the controller is a challenging task, since there are too many parameters involved.

When a task must also be selected automatically, the creative control approach described in Section 5 may be used. A creative learning theory proposed in this paper includes all the components in the adaptive critic family, which generalizes adaptive critic learning by modifying the learning rates and utilizing multiple critics and providing a model-based action database. A creative controller can be used to explore an unpredictable environment, and 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. The creative controller for intelligent robots like the adaptive critic controller has information stored in a dynamic database, plus a dynamic task control center that functions as a command center to decompose tasks into sub-tasks with different dynamic models and criteria functions. This is again a learning approach. The intelligent mobile robot is capable of learning; however, the goal of building an intelligent robot is a challenge and further research is needed to realize the full potential of safe and useful machines.

REFERENCES

1. Hall, E.L., and Hall, B. C., *Robotics: A User-Friendly Introduction*, pp. 1-8, Holt, Rinehart and Winston, New York, 1985.
2. Werbos, P. *The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting*, Wiley and Sons, 1994.
3. Werbos, P.J., Tutorial on Neurocontrol, Control Theory and Related Techniques: From Backpropagation to Brain-Like Intelligent Systems, *Proc. of the 12th Int. Conf. on Mathematical and Computer Modeling and Scientific Computing*, 1999.
4. Alhaj Ali, S. M., *Technologies for autonomous navigation in unstructured outdoor environments*, Chapter 3, Ph. D. dissertation, University of Cincinnati, Cincinnati, OH, 2003.
5. Lewis, F. L., Jagannathan, S., and Yesildirek, A., *Neural Network Control of Robot Manipulators and Nonlinear Systems*, pp. 147-167, Taylor and Francis Ltd, T. J. International Ltd, Padstow, UK, 1999.
6. Alhaj Ali, S. M., and Hall, E. L., "Technologies for autonomous operation in unstructured outdoor environments, Part I: Navigation," *Proc. of the Artificial Neural Networks in Eng. Conf.,* Vol. 12, pp. 57-62, 2002.
7. Shim, H.-S. , and Sung, Y.-G., "Asymptotic control for wheeled mobile robots with driftless constraints," *Robotics and Autonomous Systems*, **43(1)**, pp. 29-37, 2003.
8. Yun-Su, H., and Yuta, S, "Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot," *Robotics and Autonomous Systems*, **17(1-2)**, pp. 65-80, 1996.
9. Rajagopalan, R., and Barakat, N., "Velocity control of wheeled mobile robots using computed torque control and its performance for a differentially driven robot*," Journal of Robotic Systems*, **14(4)**, pp. 325 – 340, 1997.
10. Zhang, M., and Hirschorn, R. M., "Discontinuous feedback stabilization of nonholonomic wheeled mobile robots," *Dynamics and Control*, **7(2)**, pp. 155-169, 1997.
11. Koh, K. C., and Cho, H. S., "A smooth path tracking algorithm for wheeled mobile robots with dynamic constraints," *Journal of Intelligent and Robotic Systems*, **24(4)**, pp. 367-385, 1999.

12. Zhang, Y., Chung, J. H., and Velinsky, S. A., "Variable structure control of a differentially steered wheeled mobile robot*," Journal of Intelligent and Robotic Systems*, **36(3)**, pp. 301-314, 2003.
13. Corradini, M. L., and Orlando, G., "Robust tracking control of mobile robots in the presence of uncertainties in the dynamical model," *Journal of Robotic Systems*, **18(6)**, pp. 317 – 323, 2001.
14. Corradini, M. L., and Orlando, G., "Control of mobile robots with uncertainties in the dynamical model: a discrete time sliding mode approach with experimental results," *Control Engineering Practice*, **10(1)**, pp. 23-34, 2002.
15. Corradini, M. L., Leo, T., and Orlando, G., "Experimental testing of a discrete-time sliding mode controller for trajectory tracking of a wheeled mobile robot in the presence of skidding effects," *Journal of Robotic Systems*, **19(4)**, pp. 177 – 188, 2002.
16. Wu, S.-F., Mei, J.-S., and Niu, P.-Y., "Path guidance and control of a guided wheeled mobile robot," *Control Engineering Practice*, **9(1)**, pp. 97-105, 2001.
17. Jiang, Z.-P., Lefeber, E., and Nijmeijer, H., "Saturated stabilization and tracking of a nonholonomic mobile robot," *Systems and Control Letters*, **42(5)**, pp. 327-332, 2001.
18. Dong, W., Liang Xu, W., and Huo, W., "Trajectory tracking control of dynamic non-holonomic systems with unknown dynamics*," Int. J. of Robust and Nonlinear Control*, **9(13)**, pp. 905–922, 1999.
19. Werbos, P.J., "New designs for universal stability in classical adaptive control and reinforcement learning," *Proc. of the Int. Joint Conf. on Neural Networks IJCNN*, Vol. 4, pp. 2292-2295, 1999.
20. Montaner, M. B., and Ramirez-Serrano, A., "Fuzzy knowledge-based controller design for autonomous robot navigation," Expert Systems with Applications, **14(1-2)**, pp. 179-186, 1998.
21. Topalov, A. V., Kim, J.-H., and Proychev, T. P., "Fuzzy-net control of non-holonomic mobile robot using evolutionary feedback-error-learning," *Robotics and Autonomous Systems*, **23(3)**, pp. 187-200, 1998.
22. Toda, M., Kitani, O., Okamoto, T., and Torii, T., "Navigation method for a mobile robot via sonar-based crop row mapping and fuzzy logic control," *Journal of Agricultural Engineering Research*, **72(4)**, pp. 299 – 309, 1999.
23. Vázquez, F., and Garcia, E., "A local guidance method for low-cost mobile robots using fuzzy logic," Annual Review in Automatic Programming, **19**, pp. 203-207, 1994.
24. Narendra, K.S., and Parthasarathy, K., (1990) Identification and control of dynamical systems using Neural networks, *IEEE Transactions on Neural Networks*, **1(1),** pp. 4-27.
25. Psaltis, D., Sideris, A., and Yamamura, A. A., A multilayered neural network controller, *IEEE Control Systems Magazine*, **8(2)**, pp. 17-21, 1998
26. Yabuta, T., and Yamada, T., Neural network controller characteristics with regard to adaptive control, *IEEE Transactions on System, Man, and Cybernetics*, **22(1)**, pp. 170-176, 1992.
27. Liao, X., Creative Learning for Intelligent Autonomous Mobile Robots, PhD Dissertation, University of Cincinnati, 2003.
28. Werbos, P., Optimal neurocontrol: practical benefits, new results and biological evidence, *IEEE Westcon Conference Record*, p.580-585, 1995.
29. Prokhorov, D., and Wunsch, D., Adaptive Critic Designs, Neural Networks*, IEEE Trans. on Neural Networks*, **8(5)**, pp. 997-1007, 1997.
30. Lendaris, G.G., Shannon, T.T. and Rustan, A., A Comparison of Training Algorithms for DHP Adaptive Critic Neurocontrol, Neural Networks, *Proc. Int. Joint Conf.*, pp: 2265 -2270, v. 4, 1999.
31. Venayagamoorthy, G. K., Harley, R. G. and Wunsch, D. C., Comparison of Heuristic Dynamic Programming and Dual Heuristic Programming Adaptive Critics for Neurocontrol of a Turbogenerator, *IEEE Transactions on Neural Networks*, **13(3)**, pp764-773, 2002.
32. Werbos, P.J., Approximate Dynamic Programming for Real-Time Control and Neural Modeling, In White D, A and D.A. Sofge, (Eds.), in *Handbook of Intelligent Control*, pages 493--525. Van Nostrand Reinhold, 1992.
33. Simmons, R., Task Control Architecture, Available: <http://www.cs.cmu.edu/afs/cs/project/ TCA/www/TCA-history.html.
34. Yen, G.G., and Lima, P. G., Dynamic Database Approach for Fault Tolerant Control Using Dual Heuristic Programming, *Proc. of the American Control Conference*, pp. 5080-5085, 2002.
35. Campos, J., and Lewis, F.L., Adaptive Critic Neural Network for Feedforward Compensation, *Proc. American Control Conference*, **4**, pp. 2813–2818, 1999.