# A Single Network Adaptive Critic-Based Redundancy Resolution Scheme for Robot Manipulators

Prem Kumar Patchaikani, *Student Member, IEEE*, Laxmidhar Behera, *Senior Member, IEEE*, and Girijesh Prasad, *Senior Member, IEEE*

***Abstract*—This paper proposes an adaptive critic-based real-time redundancy resolution scheme for kinematic control of a redundant manipulator. The kinematic control of the redundant manipulator is formulated as a discrete-time input affine system, and then, an optimal real-time redundancy resolution scheme is proposed. The optimal control law is derived in real time using an adaptive critic framework. The proposed single network adaptive critic-based methodology defines the additional task in terms of integral cost function which results in global optimal solution. With adaptive critic-based optimal control scheme, the optimal redundancy resolution is achieved in real time without the computation of inverse, which makes the method computationally efficient. The problem formulation as proposed in this paper is first of its kinds. Furthermore, the real-time optimal redundancy resolution using an integral cost function has been comprehensively solved, which is also a novel contribution. The proposed scheme is tested in both simulation and experiment on a seven-degree-of-freedom PowerCube manipulator from Amtec Robotics.**

***Index Terms*—Adaptive critic, approximate dynamic programming (ADP), inverse kinematic control, redundancy resolution, redundant manipulator.**

## I. Introduction

REAL-LIFE challenges necessitate the use of redundant manipulators guided through visual feedback to work in a dynamically changing environment. Redundant manipulators can handle the real-life constraints such as the presence of obstacles [1], joints limits [2] and torque constraints [3], while vision provides the accurate information about the object location and changes in the dynamic environment [4].

Manuscript received April 8, 2010; revised September 20, 2010 and January 1, 2011; accepted March 25, 2011. Date of publication April 19, 2011; date of current version March 30, 2012. This work was supported in part by the Defence Research and Development Organization, Government of India, under the project titled "Neural network based visual motor coordination of a 7 DOF redundant manipulator and FPGA implementation of neural control algorithms" and in part by the U.K.–India Education and Research Initiative under the grant "Innovations in Intelligent Assistive Robotics."

P. K. Patchaikani and L. Behera are with the Department of Electrical Engineering, Indian Institute of Technology, Kanpur 208016, India (e-mail: premkani@iitk.ac.in; lbehera@iitk.ac.in).

G. Prasad is with the School of Computing and Intelligent Systems, Faculty of Computing and Engineering, University of Ulster, BT48 7JL Londonderry, U.K. (e-mail: G.Prasad@ulster.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2011.2143372

Vision has been either used in eye-in-hand configuration with a camera mounted on the manipulator [5] or eye-to-hand configuration [6] with a fixed camera. Visual servo control can be broadly classified into image-based visual servoing [7] and position-based visual servoing [8]. Recently, hybrid strategies have been proposed [9] to exploit the benefits of both the approaches. The reader can refer to [10] and [11] for a detailed survey on vision-based manipulator control. In general, the Cartesian space velocity is estimated from the vision space with the previously discussed visual control schemes. The redundancy is then resolved for the chosen additional task while following the estimated Cartesian trajectory. In a similar manner, Chaumette and Marchand [12] utilized vision to compute the Cartesian space velocity, and then, an iterative approach has been used for kinematic limit avoidance.

The control of redundant manipulator is a challenging task since the inverse kinematic relationship from the Cartesian space to the joint space is a one-to-many relationship. The redundancy resolution schemes discuss about different methodologies to exploit the redundancy for performing the additional tasks to occur in real life. Majority of the existing redundancy resolution schemes minimize the instantaneous cost functions, which involves the computation of pseudoinverse. The reader may refer to [13] for various pseudoinverse-based redundancy resolution methods. In addition to the computation of pseudoinverse, the major drawback of instantaneous cost function is that only suboptimal solution is reached. In contrast, the global optimum can be reached with integral cost functions. The integral-cost-based optimization has been implemented for path planning with the redundant manipulator in [14]. The $2n$ first-order partial differential equations with boundary conditions have been derived for the optimal control problem in [15]. Both of the aforementioned methods are offline processes and require the pseudoinverse of Jacobian.

Existing pseudoinverse-based schemes require an exact model of the system and generate a local suboptimal solution which is acceptable from engineering point of view. It is highly preferred if the system could improve its performance with time and cope up with model inaccuracies in real-time environment. Model inaccuracies pose a major challenge for pseudoinverse computation since the algorithm may suffer numerical instability with parameter variations. The real-time optimal control of a general class of nonlinear system is difficult since the analytic solution for Hamilton–Jacobi–Bellman equation is not known

and the costate dynamics is integrated backward in time. In addition, model inaccuracies necessitate the recomputation of the optimal solution for the system.

Werbos [16] proposed approximate dynamic programming (ADP) to overcome the fundamental issues in the real-time implementation of the optimal control, which solved the dynamic programming in forward direction, and hence, the controller can be implemented online. ADP is robust to model inaccuracies since the control policy is in feedback form and the performance of the controller is improved with time. Adaptive critic is a popularly known ADP-based dual neural architecture scheme used to solve the dynamic programming forward in time.

The readers may refer to [17] and [18] to know about various architectures and learning algorithm for the adaptive critic-based optimal control. The convergence of adaptive critic to optimal value has been derived for linear systems in [19] and for a general class of nonlinear systems with unconstrained and constrained inputs in [20] and [21], respectively. Mohagheghi *et al.* [22] used action-dependent heuristic dynamic-programming-based dual-network architecture to optimally control the static compensator of power systems. Recently, Padhi *et al.* [23] proposed a single network adaptive critic (SNAC) for discrete-time systems. It is demonstrated to work for input affine systems for which the optimal control input can be explicitly expressed in terms of state and costate variables. The ability to improve the performance of the controller with the adaptive critic is more suitable for the optimal redundant manipulator control in real world.

The ADP-based controller has not been tested on the kinematic control of manipulator since the approach is not direct. The redundancy resolution scheme is to be formulated as an optimal control problem for such critic-based approaches. We propose adaptive critic-based redundancy resolution for kinematically controlled redundant manipulators with fixed stereo vision. The SNAC-based redundancy resolution scheme has been introduced, and preliminary simulation results for random positioning task with quadratic cost minimization has been presented in [24]. At first, the closed-loop positioning task is formulated as a discrete-time input affine system, and the additional task is modeled as an integral quadratic cost function. The optimal controller is then derived for the quadratic cost function with the adaptive critic. It is shown that the inverse kinematic control in adaptive critic framework does not require the computation of pseudoinverse, which makes the proposed scheme computationally efficient. The computational requirement for the critic-based method is analyzed, and it shows that the proposed method is more efficient than the generalized pseudoinverse-based redundancy resolution scheme.

This paper uses the SNAC methodology [23] to kinematically control the redundant manipulator with integral quadratic cost functions for minimum joint movement and joint limit avoidance. The nonlinear costate dynamics is learned using Takagi–Sugeno (TS)-fuzzy-based critic network. The proposed control scheme is simulated on kinematic control of a seven-degree-of-freedom (7DOF) PowerCube manipulator. The experimentations are performed on the PowerCube manipulator seen through stereo vision in eye-to-hand configuration.

(a) (b)

Fig. 1. Experimental setup: (a) PowerCube manipulator. (b) Workspace with stereo vision.

TABLE I
PARAMETERS OF POWERCUBE

| link ($i$) | $\alpha_i$ | $a_i$ | $d_i$ | $\theta_i$ |
|---|---|---|---|---|
| 1 | $-90^0$ | 0 | $d_1$ | $\theta_1$ |
| 2 | $90^0$ | 0 | 0 | $\theta_2$ |
| 3 | $-90^0$ | 0 | $d_3$ | $\theta_3$ |
| 4 | $90^0$ | 0 | 0 | $\theta_4$ |
| 5 | $-90^0$ | 0 | $d_5$ | $\theta_5$ |
| 6 | $90^0$ | 0 | 0 | $\theta_6$ |
| 7 | $0^0$ | 0 | $d_7$ | $\theta_7$ |

TABLE II
KINEMATIC LIMITS OF THE MANIPULATOR

| Joint angle | Joint Velocity ($rad/sec$) |
|---|---|
| $-160^\circ \leq \theta_1 \leq 160^\circ$ | $2.618 \leq \dot{\theta}_1 \leq 1.7e-5$ |
| $-95^\circ \leq \theta_2 \leq 95^\circ$ | $2.618 \leq \dot{\theta}_2 \leq 1.7e-5$ |
| $-160^\circ \leq \theta_3 \leq 160^\circ$ | $2.618 \leq \dot{\theta}_3 \leq 1.7e-5$ |
| $-90^\circ \leq \theta_4 \leq 90^\circ$ | $2.618 \leq \dot{\theta}_4 \leq 1.7e-5$ |
| $-160^\circ \leq \theta_5 \leq 160^\circ$ | $2.618 \leq \dot{\theta}_5 \leq 1.7e-5$ |
| $-120^\circ \leq \theta_6 \leq 120^\circ$ | $4.189 \leq \dot{\theta}_6 \leq 1.7e-5$ |
| $-720^\circ \leq \theta_7 \leq 720^\circ$ | $6.283 \leq \dot{\theta}_7 \leq 1.7e-5$ |

The rest of this paper is organized as follows. A brief introduction of the experimental setup is given in the following section, and then, the robot positioning task is expressed as a discrete-time control problem in Section III. The SNAC with TS fuzzy system is discussed in Section IV. The redundancy resolution scheme with SNAC is proposed in Section V, and the computational requirements are analyzed. The simulation results and experimental results are later presented in Sections VI and VII, respectively. This paper is finally concluded in Section VIII.

## II. EXPERIMENTAL SETUP

The proposed control strategy is tested on a 7DOF PowerCube robot manipulator from Amtec Robotics [25], whose end effector is visually seen through two cameras fixed on the workspace. The manipulator and the workspace are shown in Fig. 1. The forward kinematic relationship of the manipulator is obtained from the $D-H$ parameters [26] given in Table I. The dimensions of robot links are as follows: $d_1 = 0.368$ m, $d_3 = 0.3815$ m, $d_5 = 0.3085$ m, and $d_7 = 0.2656$ m. The kinematic limits of the manipulator are tabulated in Table II.

## III. Discrete-Time Input Affine System Representation of Forward Kinematics

The position of the tip of the end effector in the Cartesian space is represented as $\mathbf{x}_c$ and that of the desired target as $\mathbf{x}_d$. The forward kinematic relationship from the joint space to the Cartesian space is given as

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\boldsymbol{\theta}} \tag{1}$$

where $\mathbf{J}$ represents the forward kinematic Jacobian from the joint space to the Cartesian space, $\mathbf{x} = [x\ y\ z]^{\mathrm{T}}$, and $\boldsymbol{\theta}$ is the 7-D vector given by $\boldsymbol{\theta} = [\theta_1\ \theta_2\ \theta_3\ \theta_4\ \theta_5\ \theta_6\ \theta_7]^{\mathrm{T}}$ for the PowerCube manipulator.

The forward kinematics is to be represented as a discrete-time dynamic system such that the redundancy resolution can be modeled as an optimal control problem. The dynamic system representation of the positioning task is obtained by considering the forward differential kinematics

$$\Delta\mathbf{x} = \mathbf{J}\Delta\boldsymbol{\theta} \tag{2}$$

where $\Delta\mathbf{x} = [\Delta x\ \ \Delta y\ \ \Delta z]^{\mathrm{T}}$ and $\Delta\boldsymbol{\theta} = [\Delta\theta_1\ \ \Delta\theta_2\ \ \Delta\theta_3\ \Delta\theta_4\ \Delta\theta_5\ \Delta\theta_6\ \Delta\theta_7]^{\mathrm{T}}$.

The forward differential kinematics can be expressed as a set of discrete motion of the end effector at different instants as follows:

$$\begin{aligned}\Delta\mathbf{x} &= \mathbf{J}\Delta\boldsymbol{\theta}\\ \mathbf{x}(k+1) - \mathbf{x}(k) &= \mathbf{J}\Delta\boldsymbol{\theta}(k)\end{aligned} \tag{3}$$

where $\mathbf{x}(k+1)$ and $\mathbf{x}(k)$ are the end-effector positions at the $(k+1)$th and $k$th instants, respectively, and $\Delta\boldsymbol{\theta}(k)$ is the change in the joint angle at the $k$th instant. The reader should note that the instant $k$ represents the number of steps taken by the manipulator to reach the position and it is not specifying a fixed time duration. The discrete motion can be expressed as a dynamic system as

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \mathbf{J}\Delta\boldsymbol{\theta}(k). \tag{4}$$

Equation (4) represents the positioning task as a discrete-time input affine system. The closed-loop error dynamics which move the end effector from the current position $\mathbf{x}_c$ to the desired position $\mathbf{x}_d$ are derived as

$$\begin{aligned}\mathbf{e}(k+1) &= \mathbf{e}(k) - \mathbf{J}\Delta\boldsymbol{\theta}(k)\\ &= \mathbf{A}\mathbf{e}(k) + \mathbf{B}\mathbf{U}(k)\end{aligned} \tag{5}$$

where $\mathbf{e}(k) = \mathbf{x}_d(k) - \mathbf{x}_c(k)$, $\mathbf{A} = \mathbf{I}$, $\mathbf{B} = -\mathbf{J}$, and $\mathbf{U}(k) = \Delta\boldsymbol{\theta}(k)$. It is assumed that $\mathbf{x}_d(k+1) = \mathbf{x}_d(k)$, i.e., the position of the object is fixed and the robot reaches the object at the desired position in a multistep movement. The number of steps taken depends on the required accuracy, training, and also the distance between the initial and final position of the end effector.

The error dynamics (5) represents the closed-loop kinematic control of the manipulator as a discrete-time input affine system. The aforementioned formulation enables one to model the redundancy resolution scheme as an optimal control problem, if the additional cost is represented as an integral cost function. The existing optimal-control-based redundancy resolution schemes [14], [15] consider the robot kinematic relationship as constraints on the optimal controller. An explicit dynamic system is not considered in the optimal controller design. In contrast, this is the first time in the literature that the forward kinematics has been modeled as a discrete-time input affine system. It will be shown that such a formulation really simplifies the controller design and allows the improvement of the performance in real time.

## IV. SNAC

Padhi *et al.* [23] have proposed SNAC for a discrete-time dynamic system. This architecture is particularly suitable for the input affine system

$$\mathbf{X}(k+1) = \mathbf{f}\left(\mathbf{X}(k)\right) + \mathbf{g}\left(\mathbf{X}(k)\right)\mathbf{U}(k) \tag{6}$$

with a quadratic cost function

$$\begin{aligned}J_c &= \frac{1}{2}\sum_{k=0}^{\infty}\left(\mathbf{X}^{\mathrm{T}}(k)\mathbf{Q}\mathbf{X}(k) + \mathbf{U}^{\mathrm{T}}(k)\mathbf{R}\mathbf{U}(k)\right)\\ &= \frac{1}{2}\sum_{k=0}^{\infty}L\left(\mathbf{X}(k), \mathbf{U}(k)\right)\end{aligned} \tag{7}$$

where $\mathbf{X}(k) \in R^n$ is the state, $\mathbf{U}(k) \in R^m$ is the input of the discrete-time system, and $L(\mathbf{X}(k), \mathbf{U}(k))$ is the utility function. $\mathbf{Q}$ is a positive semidefinite matrix, and $\mathbf{R}$ is a positive definite matrix which penalizes the states and inputs, respectively, to optimize $J_c$. The optimal control input is given by

$$\mathbf{U}(k) = -\mathbf{R}^{-1}\mathbf{g}^{\mathrm{T}}\left(\mathbf{X}(k)\right)\boldsymbol{\lambda}^{\star}(k+1) \tag{8}$$

where $\boldsymbol{\lambda}^{\star}(k+1) = (\partial J_c^{\star}(k+1)/\partial\mathbf{X}(k+1))$ is the optimal costate vector of the system and $J_c^{\star}$ is the optimal cost.

In [23], SNAC has been proposed to learn the costate dynamics on the optimal path given by

$$\boldsymbol{\lambda}(k) = \frac{\partial L}{\partial\mathbf{X}(k)} + \left(\frac{\partial\mathbf{X}(k+1)}{\partial\mathbf{X}(k)}\right)^{\mathrm{T}}\boldsymbol{\lambda}(k+1). \tag{9}$$

In the case of linear time-invariant systems, a critic network of architecture $\hat{\boldsymbol{\lambda}}(k+1) = \mathbf{W}\mathbf{X}(k)$ has been considered. It is shown that the weights of the aforementioned critic network approach the optimum value for linear systems, which corresponds to

$$\mathbf{W} = \left(\mathbf{I} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}\right)^{-1}\mathbf{P}\mathbf{A} \tag{10}$$

where $\mathbf{P}$ is the solution of the algebraic Riccati equation.

The critic-based optimal control methodology allows one to implement the optimal controller in real time. It gives near-optimal solution which is acceptable for most of the practical applications and also guarantees that the optimal controller will be achieved with exhaustive learning. This real-time capability and the performance improvement with time make it

more attractive for real-time redundancy resolution where only suboptimal solutions have been achieved so far.

### A. TS-Fuzzy-Model-Based Critic Network

It is well known that a nonlinear system behaves like a linear system in a narrow operating zone, and hence, the nonlinear dynamics has been represented as clusters of linear systems. It is quite intuitive to design the optimal controller in terms of clusters of optimal controllers of linear systems since a linear controller effectively controls the nonlinear system in small operating zone. The TS fuzzy model is used in this paper to model the nonlinear costate dynamics in (9) as fuzzy clusters of the costate dynamics of local linear model.

The $i$th rule of the TS-fuzzy-based critic network for nonlinear system with state $\mathbf{X}$ is given by

$$\text{IF } X_1(k) \text{ is } F_1^i \text{ AND } \ldots \text{ AND } X_n(k) \text{ is } F_n^i \text{ THEN}$$
$$\hat{\boldsymbol{\lambda}}_i(k+1) = \mathbf{W}_i \mathbf{X}(k)$$

where $F_j^i$, $j = 1, 2, \ldots, n$, is the $j$th fuzzy set of the $i$th rule. Let

$$\mu_i = \prod_{j=1}^{n} \mu_i^j(\mathbf{X}) \tag{11}$$

where $\mu_i^j(\mathbf{X})$ is the membership function of the fuzzy set $F_j^i$, $i = 1, 2, \ldots, r$.

Given the current state vector $\mathbf{X}(k)$, the fuzzy model around the operating point is constructed as the weighted average of the local models and has the form

$$\hat{\boldsymbol{\lambda}}(k+1) = \frac{\sum_{i=0}^{r} \mu_i \mathbf{W}_i \mathbf{X}(k)}{\sum_{i=0}^{r} \mu_i}. \tag{12}$$

The reader should note that the costate dynamics in each fuzzy zone is learned with a network similar to that of the costate dynamics of the linear system. We know from [23] that the weights would converge to optimum value for linear systems. Hence, with the proposed TS-fuzzy-based critic network, it is expected that the network would converge to the optimal value, corresponding to local linear model of each fuzzy zone.

## V. SNAC-Based Optimal Redundancy Resolution

The discrete-time input affine formulation of the closed-loop positioning dynamics discussed in (5) is suitable for SNAC-based control if we could model the additional task in the form of quadratic cost functions, as discussed in Section IV. It will be shown in subsequent sections that such formulation is indeed possible.

We have the system state $\mathbf{X}(k) = \mathbf{e}(k)$ and the input $\mathbf{U}(k)$ as $\Delta\boldsymbol{\theta}(k)$ for critic-based redundancy resolution scheme. The optimal redundancy resolution scheme with single network based adaptive critic is shown in Fig. 2. With adaptive critic-based redundancy resolution scheme, the joint angle input $\Delta\boldsymbol{\theta}$ is computed as

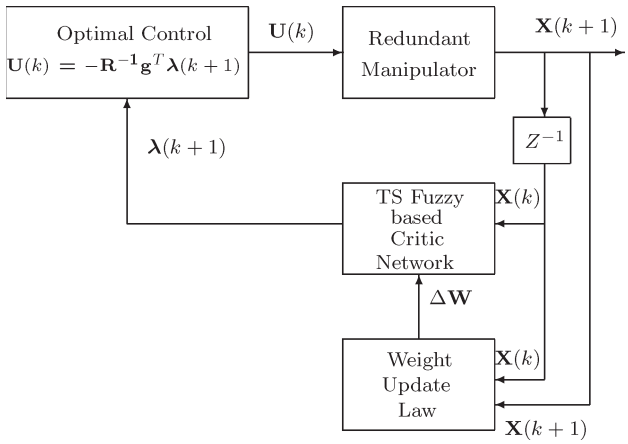


Fig. 2. Optimal redundancy resolution scheme with SNAC.

$$\Delta\boldsymbol{\theta}(k) = \mathbf{R}^{-1}\mathbf{J}^{\mathrm{T}}\hat{\boldsymbol{\lambda}}(k+1) \tag{13}$$

where $\hat{\boldsymbol{\lambda}}(k+1)$ is computed from the critic network, which optimizes the given cost. The advantage of such an approach is that the optimal closed-loop positioning of the redundant manipulator is achieved in real-time without the computation of pseudoinverse of forward Jacobian. Hence, we could obtain a computationally efficient global optimum solution with critic-based approach. This claim will be further confirmed with computational complexity analysis.

### A. TS-Fuzzy-Based Critic Network for Closed-Loop Positioning Task

A TS-fuzzy-based critic network is designed for the stabilization of the states, and the states are fuzzified to learn the costate dynamics, as discussed in Section IV-A. Hence, the critic network should be fuzzified with an end-effector position error $\mathbf{e}(k)$ for a closed-loop positioning task. It is always desired to achieve a global positioning where the end effector can be moved from any initial position to arbitrary desired position over the entire workspace. The global positioning depends on both the current and desired positions of the end effector. The performance of the critic network will be poor if we consider the error as an input to the fuzzifier, since error changes based on both the current and desired positions. The current position of the end effector specifies the kinematic state of the manipulator better than the positioning error since the forward Jacobian changes as the end effector moves along the trajectory.

To achieve a better global positioning, the current position of the end effector $\mathbf{x}_c$ is given as input to the fuzzifier instead of error $\mathbf{e}$. As the manipulator changes its joint angle configuration along the trajectory, the different fuzzy zones of critic will be activated, and the corresponding weights will be trained accordingly. The current position of the end effector has been suggested in [10] and [27] for better dynamic performance.

With the aforementioned modification, the $i$th rule of the TS-fuzzy-based critic network for robot end-effector positioning is given by

$$\text{IF } x(k) \text{ is } F_1^i \text{ AND } y(k) \text{ is } F_2^i \text{ AND } z(k) \text{ is } F_3^i \text{ THEN}$$
$$\hat{\boldsymbol{\lambda}}_i(k+1) = \mathbf{W}_i \boldsymbol{e}(k).$$

### B. Selection of Input Weight Matrix

The cost function has to be appropriately chosen to achieve the additional task. The redundancy can be resolved in real time using SNAC if the primary positioning task and the additional task are specified in the form of a quadratic cost. The state weight matrix $\mathbf{Q}$ specifies the primary task. In the current work, the state weight matrix $\mathbf{Q}$ is chosen as an identity matrix to ensure uniform convergence toward the desired position. The input weight matrix $\mathbf{R}$ is to be chosen to penalize the individual joints based on the additional task.

*1) Quadratic Cost Minimization:* Quadratic cost measure has been used to design the controller which minimizes the variation of the system states and input. $\mathbf{R}$ is chosen to control the variation in input. In this paper, a diagonal matrix with constant entry $R_{\text{gain}}$ for all the joints is chosen, which uniformly weighs the motion of all the joints. The speed of convergence to the desired position can be controlled by varying $R_{\text{gain}}$.

*2) Joint Limit Avoidance:* The joint limit avoidance is one of the key additional tasks expected from the redundant manipulator due to its physical limitation. In this paper, the performance criterion proposed by Zghal *et al.* [28] has been used to obtain the input weight matrix $\mathbf{R}(\boldsymbol{\theta})$ to constrain the joints within the kinematic limits. The performance criterion is given as

$$H(\boldsymbol{\theta}) = \sum_{i=1}^{N} \frac{1}{4} \frac{(\theta_{i_{\max}} - \theta_{i_{\min}})^2}{(\theta_{i_{\max}} - \theta_i)(\theta_i - \theta_{i_{\min}})} \tag{14}$$

where $N$ is the number of links of the redundant manipulator. The input weight matrix $R$ of the quadratic cost function is defined as

$$\mathbf{R}(\boldsymbol{\theta}) = \begin{bmatrix} R_1 & 0 & 0 & \dots & 0 \\ 0 & R_2 & 0 & \dots & 0 \\ & & & \ddots & \\ \dots & \dots & \dots & & \dots \\ 0 & 0 & 0 & \dots & R_7 \end{bmatrix} \tag{15}$$

where $\mathbf{R}(\boldsymbol{\theta})$ is a diagonal matrix with $i$th diagonal entry defined as

$$R_i(\boldsymbol{\theta}) = \begin{cases} 1 + \left| \frac{\partial H(\boldsymbol{\theta})}{\partial \theta_i} \right|, & \text{if } \Delta \left| \frac{\partial H(\boldsymbol{\theta})}{\partial \theta_i} \right| \geq 0 \\ 1, & \text{if } \Delta \left| \frac{\partial H(\boldsymbol{\theta})}{\partial \theta_i} \right| < 0. \end{cases} \tag{16}$$

The input weight $R_i(\boldsymbol{\theta})$ penalizes only the $i$th joint while the joint moves toward its kinematic limits. Since $\mathbf{R}(\boldsymbol{\theta})$ is defined as a diagonal matrix, its inverse can be computed with the reciprocal of the corresponding elements, which makes the chosen performance criterion computationally efficient.

The network has to be trained by penalizing the joint motion based on the additional task requirement. Hence, a smooth joint motion is necessary so that the links will be penalized from the initial position. To achieve such a smooth motion, the input to the critic network is given as $K\mathbf{e}(k)$, where $K$ is the feedback gain which is typically chosen as $K \leq 1$. The smaller gain value results in a slower motion, and the movement will not be jerky due to the large value of initial error.

### C. Training Algorithm

The network should be trained such that the weight would converge to the optimal weights of local linear models. To achieve the smooth network convergence, the network is learned from a selected fuzzy zone $\mathbf{x}_{f0}$ to the entire workspace such that the weights would converge to optimal values in each zone. We define $S_i = \{\mathbf{x}_d(k) : \|\mathbf{x}_d(k) - \mathbf{x}_{f0}\| < C_i,\ i = 1, 2, \dots, I\}$, where $C_i$ is a positive constant. $C_i$ is chosen such that $C_i < C_{i+1}$. Initially, $C_1$ is chosen a small value so that the network would learn the optimal weights corresponding to the selected zone. Then, the operating zone is gradually increased. With such training, the network weights would vary smoothly from one zone of operation to another.

The critic network is learned as follows.

1) Generate $N_D$ random desired positions for each $S_i$. Initialize $i = 1$.
2) Set $k = 0$ and $n = 0$. Choose a random initial point $\mathbf{x}(k)$.
3) Compute the initial state $\mathbf{e}(k) = \mathbf{x}_d(n) - \mathbf{x}(k)$. Give $\mathbf{e}(k)$ as input to the critic network and $\mathbf{x}(k)$ as input to the fuzzifier, and then, compute $\hat{\boldsymbol{\lambda}}(k+1)$.
4) Compute the input $\Delta\boldsymbol{\theta}(k)$, from (8), using $\hat{\boldsymbol{\lambda}}(k+1)$.
5) Give the input to the plant dynamics (6), and compute the next instant end-effector position $\mathbf{x}(k+1)$ and positioning error $\mathbf{e}(k+1)$.
6) Compute $\hat{\boldsymbol{\lambda}}(k+2)$ from the critic network using $\mathbf{e}(k+1)$.
7) Compute $\boldsymbol{\lambda}_d(k+1)$ using $\hat{\boldsymbol{\lambda}}(k+2)$, $\mathbf{e}(k+1)$ in costate dynamics (9). Consider $\boldsymbol{\lambda}_d(k+1)$ as the desired costate vector, and update the network weights to minimize $\|\boldsymbol{\lambda}_d(k+1) - \hat{\boldsymbol{\lambda}}(k+1)\|$.
8) Increment $k$ and check $\|\mathbf{e}(k+1)\| < \epsilon_e$. If $\|\mathbf{e}(k+1)\| > \epsilon_e$, then repeat from Step 3) with $k = k+1$ until $N_{\max}$ iterative steps.
9) If the desired position is reached with chosen accuracy, set $n = n+1$ repeat from Step 2) for $N_D$ random points.
10) Check $\|\boldsymbol{\lambda}_d(k+1) - \hat{\boldsymbol{\lambda}}(k+1)\| < \epsilon_\lambda$ for $N_D$ points in $S_i$. If no, repeat Steps 2)–9). Otherwise, set $i = i+1$ and repeat Steps 2)–9) until $i = I$.

The optimal weights in each zone will vary smoothly from the optimal value of the fuzzy zone $\mathbf{x}_{f0}$, since the system dynamics gradually deviates from the linear behavior as the zone of operation increases. Considering this fact, the weights of the critic network are always initialized with the optimal value of the fuzzy zone $\mathbf{x}_{f0}$, which can be computed using the algebraic Riccati equation with the linearized model.

### D. Computational Complexity

The optimal costate vector is learned during the training, and the trained network is used to compute the input in real

time. It will be shown that the computational requirement of the TS fuzzy network is low since the nonlinear costate vector is approximated with clusters of local linear models.

The local costate dynamics is represented by $n \times n$ matrix, where $n = 3$ is the dimension of workspace. The computation of the local costate vector requires $n^2$ order flops. If $N_r$ rules are firing at each operating point, then the local costate vectors are computed with $N_r n^2$ order flops, and furthermore, $N_r n$ order flops are required for computing the overall costate vector. It is clearly evident that the computation of the costate vector is independent of the DOF of the manipulator and, in general, $N_r \ll r$.

The computation of input from the costate vector requires $Nn$ order flops for computing $\mathbf{J}^{\mathrm{T}}\hat{\boldsymbol{\lambda}}$ and $N$ order flops for $\mathbf{R}^{-1}\mathbf{J}^{\mathrm{T}}\hat{\boldsymbol{\lambda}}$, since $\mathbf{R}$ is a diagonal matrix. The proposed adaptive critic-based redundancy resolution scheme requires a total of $N_r(n^2+n)+N(n+1)$ order flops for computing the input which is linear with the DOFs of the manipulator.

The pseudoinverse-based redundancy resolution scheme [2] involves computation of the minimum norm motion for the primary task and the self-motion for accomplishing the additional task. The redundancy resolution requires computation of Moore–Penrose pseudoinverse. The computation of Moore–Penrose pseudoinverse [29] involves singular value decomposition (SVD) and matrix multiplications for inverse computation. The SVD computation is of order $O(Nn^2)$ and requires $2Nn^2+n^3$ flops approximately. The multiplications involve $Nn^2+N^2n$ order flops, and hence, the pseudoinverse computation requires a total of $3Nn^2+n^3+nN^2$ order flops. The minimum norm input computation involves $nN$ order flops. The self-motion is computed with $N(N-1)n/2+N^2$ order flops, where $N(N-1)n/2$ is required for computing $(I-J^{+}J)$ and $N^2$ is for $(I-J^{+}J)K$. The total computational cost of the pseudoinverse-based technique is $(3Nn^2+n^3+3/2N^2n-Nn/2)$ order flops, which is more than the adaptive critic-based method.

The computational requirement increases in the order $O(N^2)$ for the pseudoinverse-based technique while it is linear with degrees of freedom for critic-based approach, which makes it a better approach for real-time implementation. In addition to low computational requirements, critic-based methodology guarantees optimal solution, while optimality is not ensured with the pseudoinverse-based technique.

## VI. Simulation Results

The critic network is trained within a cubic volume of workspace with diagonal vertices (0.2, −0.25, 0.0) and (0.7, 0.25, 0.3). The Gaussian function is chosen as a fuzzy membership function, and the width of the fuzzy membership function is selected such that the effect of the two fuzzy zones will be predominant at each operating point and the effect of neighboring fuzzy zones at the fuzzy center will be less than 5%. Hence, the number of fuzzy zones firing at each operating point for the critic network becomes $N_r = 2^3 = 8$. The cubic volume is fuzzified into five fuzzy zones in each coordinate direction of the Cartesian space. The critic network totally comprises 125 fuzzy zones with a linear weight matrix $\mathbf{W}_i$ to model the local costate dynamics. The critic network is trained with 500 000 random points with a learning rate of 0.01, $\epsilon_e = 0.5$ mm, and $N_{\max} = 50$.

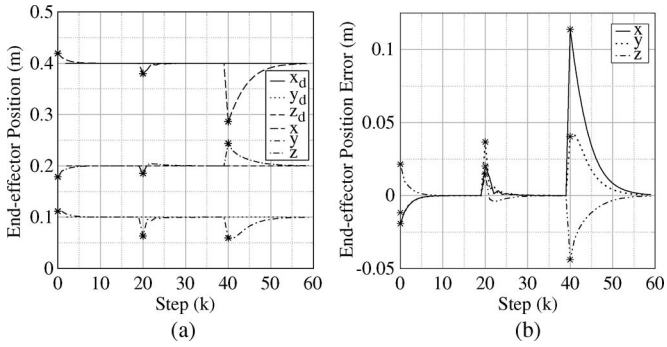


Fig. 3. Closed-loop positioning at the desired position (0.4, 0.1, 0.2)$m$. (a) Position. (b) Error.

### A. Quadratic Cost Minimization

The control task is to compute the input $\boldsymbol{\Delta\theta}(\mathbf{k})$ for the closed-loop positioning task, which minimizes the cost function

$$J_c = \frac{1}{2}\sum_{k=0}^{\infty}\left(\mathbf{e}^{\mathrm{T}}(k)\mathbf{Q}\mathbf{e}(k)+\Delta\boldsymbol{\theta}^{\mathrm{T}}(k)\mathbf{R}\Delta\boldsymbol{\theta}(k)\right). \tag{17}$$

The state weight matrix $\mathbf{Q}$ and the input weight matrix $\mathbf{R}$ are chosen as identity matrices, as discussed in Section V-B. $R_{\mathrm{gain}}$ is chosen as 1, and the feedback gain $K$ is chosen as 0.5 in the experiment.

After critic training, the network is analyzed in two stages. In the first stage, the desired position of the end effector is fixed as $\mathbf{x}_d = (0.4, 0.1, 0.2)$m, and the robot is started from random initial positions. The position of the end effector at successive instants and the corresponding error are shown in Fig. 3. The initial random points are explicitly shown with $\star$ in the figures. It is observed in simulation that the end effector reaches the desired position with an accuracy of 0.1 mm in 20 iterative steps, indicating that the network is trained adequately. An instant in the figures corresponds to one iterative step taken by the manipulator to move from the current position toward the desired position.

The corresponding joint angle movement is shown in Fig. 4. The initial joint angle corresponding to the random initial positions that are represented as $\star$ shows that the manipulator starts from the random initial joint angle and reaches the desired position with smooth joint movements. The seventh joint angle is not shown since it is ideally zero over the entire trajectory. The seventh link creates end-effector roll and does not contribute to any position change, and hence, it will not be shown in further experiments too.

Since we are interested in global positioning from random initial position to the arbitrary desired position, the controller performance is further analyzed for various desired positions over the entire workspace. The successive position of the end effector and the corresponding error are shown in Fig. 5. The instant at which the desired position changes is shown with $\star$. The end effector reaches the various desired positions

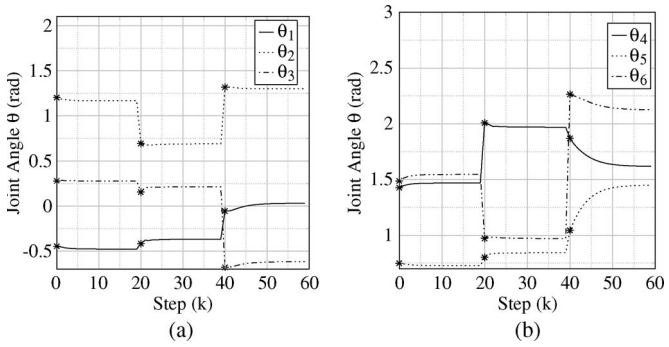


Fig. 4. Joint angle $\boldsymbol{\theta}$ during positioning at the desired position (0.4, 0.1, 0.2)$m$. (a) $(\theta_1, \theta_2, \theta_3)$. (b) $(\theta_4, \theta_5, \theta_6)$.

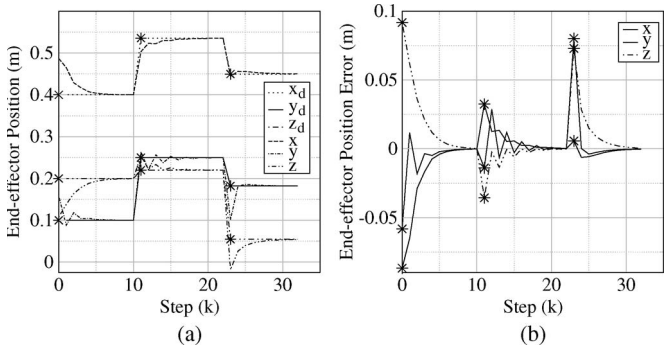


Fig. 5. Closed-loop positioning at the arbitrary desired position from random initial position. (a) Position. (b) Error.

with an average accuracy of 0.1 mm in 20 iterative steps and stays within an error of 1 mm after 10 iterative steps. The reader should note that the positioning accuracy varies over the workspace since the uniform convergence is possible only with exhaustive training. As discussed in Section IV, the adaptive critic-based approaches guarantee the performance improvement with time, and the controller approaches the optimal value with continuous weight update. Hence, better positioning accuracy and optimality are feasible with further weight update, which makes the critic-based approach superior to the existing suboptimal approaches.

Fig. 5(a) shows that the closed-loop positioning exhibits an oscillatory behavior for the chosen $\mathbf{Q}$ and $\mathbf{R}$. It is well known that a nonoscillatory performance can be achieved in optimal-control-based strategies by choosing appropriate $\mathbf{Q}$ and $\mathbf{R}$, which is an iterative process.

The corresponding joint angle movement for random positioning task is shown in Fig. 6. It is clear from the $\star$ mark in the figures that, initially, there is a large change in the joint angle due to huge error and the movement decreases with decreasing error.

The performance of the controller is tested with a closed elliptical trajectory. A complex closed trajectory is particularly chosen to test the performance of the controller over the entire workspace. The critic-based methodology is devised for closed-loop positioning task with the assumption that $\mathbf{x}_d(k+1) = \mathbf{x}_d(k)$, i.e., the manipulator is moving to reach a stationary object. The aforementioned assumption is valid at each sampling instant while tracking a continuous trajectory such as ellipse, as explained in the following steps. Let the position of

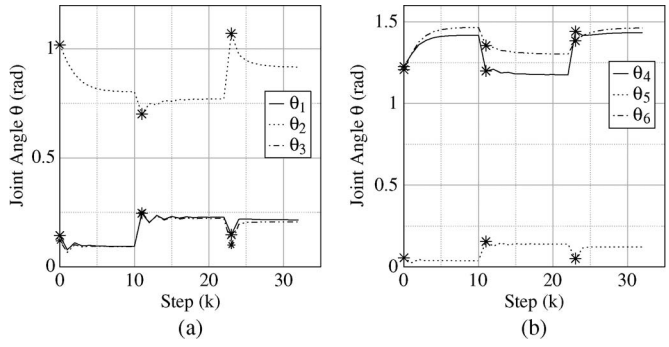


Fig. 6. Joint angle $\boldsymbol{\theta}$ during positioning at the arbitrary desired position from the random initial position. (a) $(\theta_1, \theta_2, \theta_3)$. (b) $(\theta_4, \theta_5, \theta_6)$.

the end effector at $t_k^{th}$ time instant be $\mathbf{x}(t_k)$ with a joint angle configuration $\boldsymbol{\theta}(t_k)$ and the desired position at the $(t_k+1)$th time instant be $\mathbf{x}_d(t_k+1)$.

1) The desired position to estimate the control law is fixed as $\mathbf{x}_{\text{desired}} = \mathbf{x}_d(t_k+1)$. The initial position and joint angle are taken as $\mathbf{x}(0) = \mathbf{x}(t_k)$ and $\boldsymbol{\theta}(0) = \boldsymbol{\theta}(t_k)$, respectively.
2) The adaptive critic is iterated with the kinematic model (1) until the end effector reaches the position $\hat{\mathbf{x}}(k)$ with the joint angle $\boldsymbol{\theta}(k)$ such that the positioning error $\|\mathbf{x}_{\text{desired}} - \hat{\mathbf{x}}(k)\| < \epsilon_{\max}$, where $\epsilon_{\max}$ is the desired positioning error tolerance and $k$ is the number of steps, as discussed in Section III. The end-effector position $\hat{\mathbf{x}}(k)$ is predicted using the forward kinematic model with $\boldsymbol{\theta}(k)$.
3) The control input is calculated as $\Delta\boldsymbol{\theta}(t_k) = \boldsymbol{\theta}(k) - \boldsymbol{\theta}(0)$ and is applied to the manipulator.
4) The end-effector moves to the position $\mathbf{x}(t_k+1)$ with the given input $\Delta\boldsymbol{\theta}(t_k)$.
5) The critic is then presented with the next desired position to move along the trajectory.

To analyze the controller performance, the number of steps taken by the critic at each instant to reach within the chosen accuracy will be computed in the following simulations.

The desired elliptical trajectory is taken as

$$\begin{aligned} x &= 0.45 + 0.15\cos(0.05t_k) \\ y &= 0.15\sin(0.05t_k) \\ z &= 0.15 \end{aligned} \tag{18}$$

where $t_k$ is the sampling instant and the rms positioning error tolerance $\epsilon_{\max}$ is chosen as 1.0 cm to iterate the critic-based optimal control law before applying to the robot. The experiment is performed in real time with a sampling time of 200 ms. Hence, the simulation results will be shown with the same sampling time for easier comparison with the real-time results.

The end-effector trajectory and its corresponding robot configuration are shown in Fig. 7. It is clear from the figure that the closed-loop system tracks the trajectory with smooth joint motion. The positioning error shown in Fig. 8 indicates that the manipulator tracks the trajectory with an accuracy of 7 mm. The corresponding joint angle input and the joint angle configurations are shown in Figs. 9 and 10, respectively, to demonstrate the smoothness of the learned solution. It is observed that

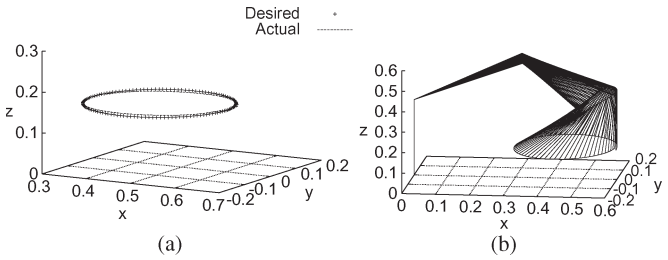


Fig. 7. End-effector motion while tracking the elliptical trajectory. (a) Trajectory (in meter). (b) Robot configuration (in meter).

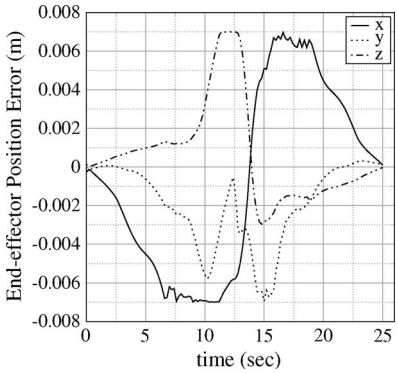


Fig. 8. End-effector position error while tracking the ellipse.

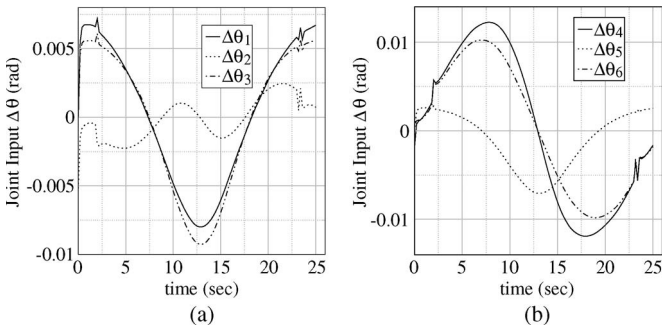


Fig. 9. Joint angle input $\Delta\boldsymbol{\theta}$ while tracking the ellipse. (a) $(\Delta\theta_1, \Delta\theta_2, \Delta\theta_3)$. (b) $(\Delta\theta_4, \Delta\theta_5, \Delta\theta_6)$.

the manipulator tracks the ellipse after 1.03 iterative steps on average. The number of iterations can be reduced, and also, the speed of the tracking can be increased by increasing feedback gain $K$ but it may result in an oscillatory behavior. All the aforementioned simulations corroborate our claim that the nonlinear costate dynamics can be represented as fuzzy cluster of linear costate dynamics of local model.

The computation time requirement for the critic-based approach is further analyzed during the experiments. The computation time for drawing the elliptical trajectory for 100 cycles is measured. The complete cycle of the elliptical trajectory (18) is represented by 126 points, and hence, the end effector moves in between 12 600 operating points over the experiment. The simulation is run on a general-purpose computer with Intel Core 2 Duo E7300 CPU with 2.66-GHz clock and 4-GB RAM. The computer is operated with Debian 4.02 operating system running in multiuser graphics mode with all the services enabled. The simulations are performed at different times of the day with various CPU load conditions. The average and

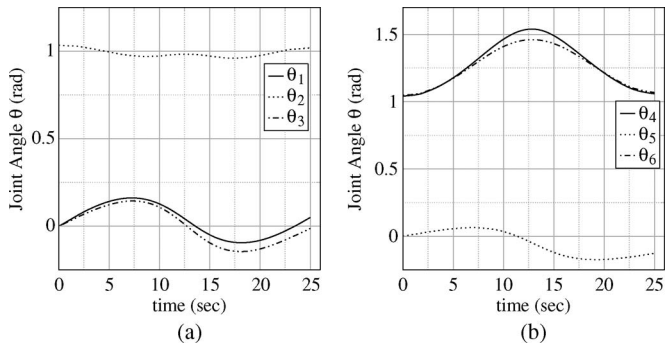


Fig. 10. Joint angle $\boldsymbol{\theta}$ while tracking the elliptical trajectory. (a) $(\theta_1, \theta_2, \theta_3)$. (b) $(\theta_4, \theta_5, \theta_6)$.

TABLE III
COMPUTATION TIME REQUIREMENT FOR MINIMUM NORM MOTION

| critic ($\mu.sec$) | | Pseudo-inverse ($\mu.sec$) | |
|---|---|---|---|
| Average | Std. Deviation | Average | Std. Deviation |
| 32.54 | 16.25 | 41.92 | 20.27 |
| 32.14 | 16.55 | 40.91 | 21.06 |
| 32.14 | 15.57 | 40.04 | 20.20 |
| 31.35 | 17.11 | 40.91 | 19.81 |
| 32.54 | 15.25 | 40.91 | 21.06 |

standard deviation of the computation time at each operating point over various runs are presented in Table III. The computation time is compared with the minimum norm motion from the pseudoinverse-based technique, and it is clear from the table that the proposed approach takes approximately 75% time that of pseudoinverse for 7DOF robot. The critic-based approach takes approximately 33 $\mu$s at each operating point to compute the input, which is negligible compared to the time constant of the redundant manipulator. This confirms our claim that the adaptive critic-based method ensures optimal redundancy resolution in real time. The reader should note that the pseudoinverse technique moves along the negative gradient of the instantaneous cost, and hence, optimality is not guaranteed.

### B. Joint Limit Avoidance

The control task considered is to compute the input $\mathbf{U}(k) = \Delta\boldsymbol{\theta}(k)$ to position the end effector of the redundant manipulator at the desired position, which minimizes the cost function

$$J_c = \frac{1}{2}\sum_{k=0}^{\infty}\left(\mathbf{e}^{\mathrm{T}}(k)\mathbf{Q}\mathbf{e}(k) + \Delta\boldsymbol{\theta}^{\mathrm{T}}(k)\mathbf{R}(\boldsymbol{\theta})\Delta\boldsymbol{\theta}(k)\right) \quad (19)$$

where $\mathbf{R}(\boldsymbol{\theta})$ is computed, as discussed in Section V-B2. In this simulation, the kinematic constraint is applied to the fourth joint with the limit given as

$$-1.25 < \theta_4 < 1.25 \text{ rad}. \quad (20)$$

The aforementioned operating range is chosen based on the observations from the quadratic cost minimization. The joint angle of the fourth link varies in the range of (1.04, 1.53) rad while tracking the elliptical trajectory (18). A simple quadratic cost minimization violates the aforementioned kinematic limit. The critic network is trained with feedback gain $K = 0.1$ to accurately learn the cost penalization.

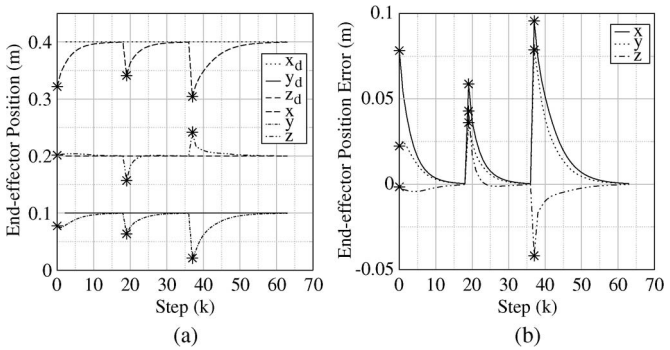


Fig. 11. Closed-loop positioning at the desired position (0.4, 0.1, 0.2)$m$ with kinematic limit avoidance. (a) Position. (b) Error.

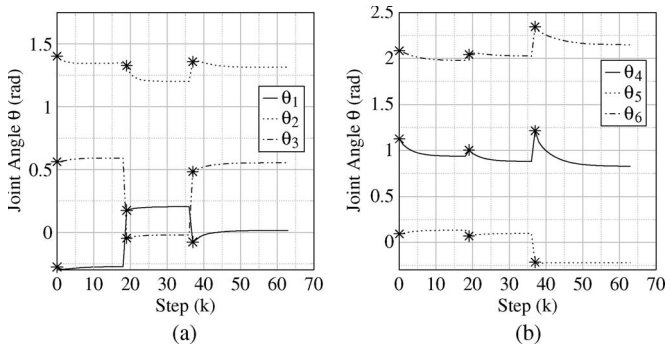


Fig. 12. Joint angle $\boldsymbol{\theta}$ during positioning at the desired position (0.4, 0.1, 0.2)$m$ with kinematic limit avoidance. (a) $(\theta_1, \theta_2, \theta_3)$. (b) $(\theta_4, \theta_5, \theta_6)$.

After the initial training, the network is analyzed in two stages, as discussed in Section VI-A. Initially, the desired position of the robot is chosen as the same as that of the previous experiment, and then, the robot is started from different initial positions. The corresponding end-effector position at successive instants and the positioning error are plotted in Fig. 11, and the corresponding joint angles are shown in Fig. 12. Then, the controller is further checked for arbitrary positioning from various initial positions over the entire workspace. The corresponding positioning results are shown in Fig. 13 with joint angles in Fig. 14. The initial random point is explicitly shown with $\star$ in the figures. The results are similar to that of the quadratic cost minimization with two major differences. It is observed that the desired position is reached with an accuracy of 0.1 mm in 50 iterative steps in contrast to 20 steps, and the joint angle of the fourth link is within the chosen limits. More iterative steps are taken since the feedback gain is chosen as $K = 0.1$, so that joint will be penalized properly.

The performance of the controller is tested with the closed elliptical trajectory (18). The same trajectory is chosen so that we can compare the results for kinematic limit avoidance.

The end-effector trajectory and the corresponding robot configuration are shown in Fig. 15. The tracking error shown in Fig. 16 indicates that the end effector tracks the trajectory smoothly within an accuracy of 7.4 mm with the proposed adaptive critic-based kinematic limit avoidance scheme. It is observed in experiments that the critic required 1.508 iterative steps in each instant on average along the trajectory. The average iterative steps are higher than the quadratic cost

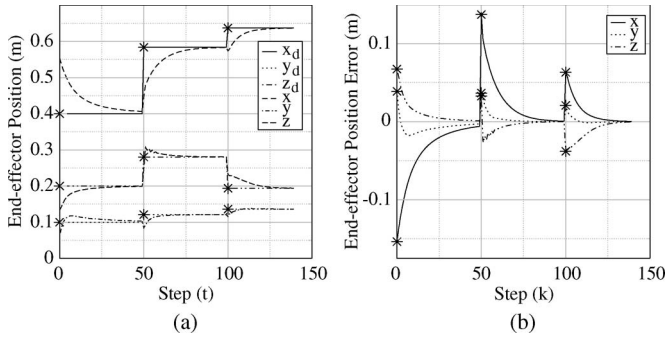


Fig. 13. Closed-loop positioning at the arbitrary desired position from random initial position with kinematic limit avoidance. (a) Position. (b) Error.

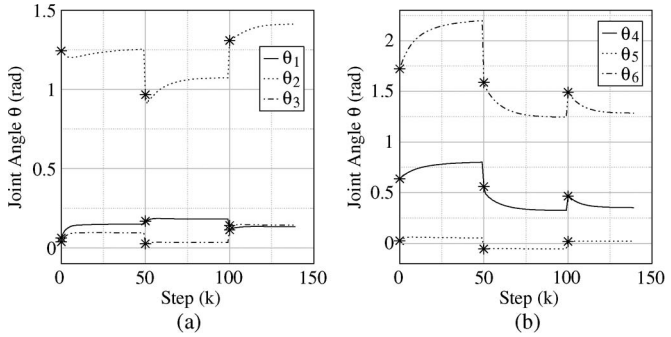


Fig. 14. Joint angle $\boldsymbol{\theta}$ during positioning at the arbitrary desired point from random initial point with kinematic limit avoidance. (a) $(\theta_1, \theta_2, \theta_3)$. (b) $(\theta_4, \theta_5, \theta_6)$.

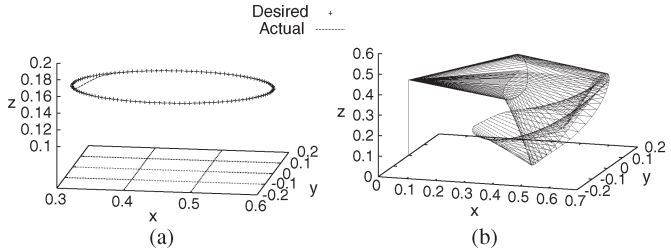


Fig. 15. End-effector motion while tracking the elliptical trajectory with kinematic limit avoidance. (a) Trajectory (in meters). (b) Robot configuration (in meters).

minimization performance since $K = 0.1$. The joint angle input and robot joint configurations are shown in Figs. 17 and 18, respectively. It is clear from the figures that the joint angle of the fourth link is within the kinematic limit.

The joint angle of the fourth link is separately shown in Fig. 19 for better understanding of the effect of kinematic limits on joint configuration. It is clear from the figure that the manipulator avoids the kinematic limit while tracking the elliptical trajectory.

The computation time is then compared with the pseudoinverse-based joint limit avoidance scheme. The robot is simulated to draw the elliptical trajectory defined by (18) for 100 cycles, similar to the analysis of quadratic cost minimization scheme. The average and standard deviation of the computation time taken at each operating point over various runs are tabulated in Table IV. It is clear from the table that the proposed approach takes 33 $\mu$s to compute input for kinematic limit avoidance, while the pseudoinverse-based

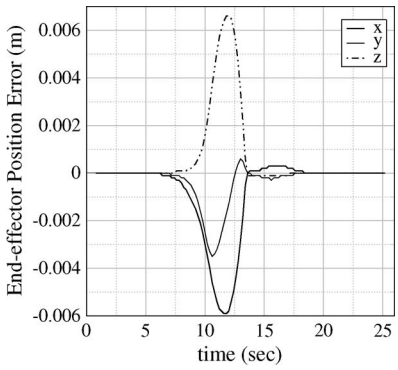


Fig. 16. End-effector position error while tracking the ellipse with kinematic limit avoidance.

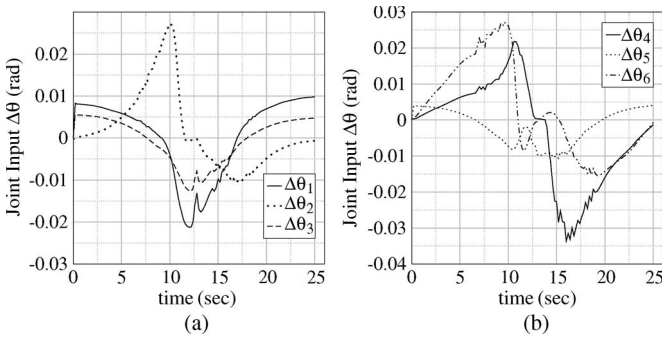


Fig. 17. Joint angle input $\Delta\boldsymbol{\theta}$ while tracking the elliptical trajectory with kinematic limit avoidance. (a) $(\Delta\theta_1, \Delta\theta_2, \Delta\theta_3)$. (b) $(\Delta\theta_4, \Delta\theta_5, \Delta\theta_6)$.

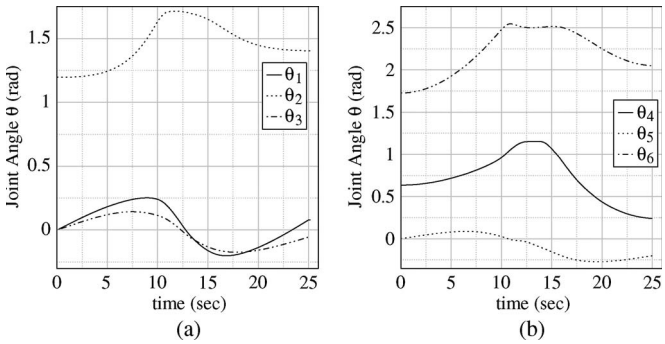


Fig. 18. Joint angle $\boldsymbol{\theta}$ while tracking the elliptical trajectory with kinematic limit avoidance. (a) $(\theta_1, \theta_2, \theta_3)$. (b) $(\theta_4, \theta_5, \theta_6)$.

TABLE IV
COMPUTATION TIME REQUIREMENT FOR JOINT LIMIT AVOIDANCE

| critic ($\mu.sec$) | | Pseudo-inverse ($\mu.sec$) | |
|---|---|---|---|
| Average | Std. Deviation | Average | Std. Deviation |
| 33.33 | 6.57 | 44.44 | 5.73 |
| 34.13 | 5.89 | 44.29 | 5.88 |
| 33.33 | 6.57 | 44.13 | 6.78 |
| 32.94 | 5.93 | 44.29 | 6.66 |
| 32.94 | 7.8 | 44.76 | 6.27 |

technique requires 44 $\mu$s. The computation time is 25% less than the pseudoinverse-based method, which matches with the result of minimum norm simulations discussed earlier. These observations corroborate our claim that critic-based redundancy resolution is computationally efficient and also guarantees the optimal solution.

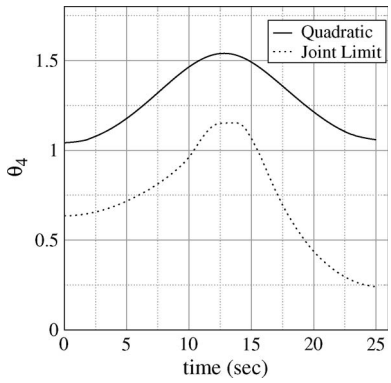


Fig. 19. Joint angle of the fourth link with and without kinematic limit avoidance.

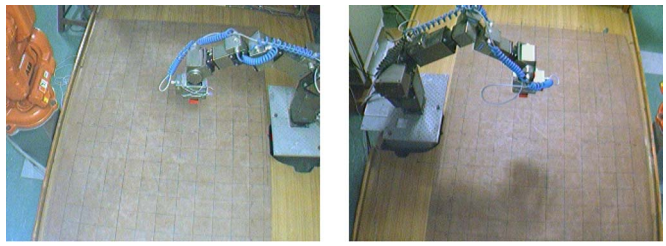


Fig. 20. View of the workspace from stereo vision. The robot tip is identified with red tape. (a) Left camera. (b) Right camera.

## VII. EXPERIMENTAL RESULTS

The experimental test is performed on PowerCube manipulator. The tip of the manipulator is seen through stereo vision fixed on the workspace. Two Fire-i digital cameras [30] have been used for visualization. The workspace visible through the stereo vision is shown in Fig. 20. The workspace is captured through the camera with an image frame of dimension 320 $\times$ 240 pixels. The current position $\mathbf{x}_c$ of the tip of the end effector is projected in the stereo vision. The robot tip is identified with Red Tape wrapped around it. The centroid of the identified region $(u, v)$ is used to identify the current position of the end effector. The real-time experiments have been performed for the elliptical trajectory (18). As mentioned in the simulation, the critic is iterated with the model at each sampling instant, and then, the final control law is given to the manipulator. The sampling interval is chosen as 200 ms based on the computational requirement for image acquisition, processing, and also the manipulator speed. A smaller sampling interval would result in a jerky motion of the manipulator.

### A. Quadratic Cost Minimization

The network trained in Section VI-A is used in the experiment, and the forward kinematic model is used to compute the output during iteration. The tip of the robot manipulator is identified using stereo vision over the entire trajectory.

The end-effector position while tracking the trajectory and the tracking error in real-time experiment are shown in Fig. 21. It is clear from the figure that the manipulator tracks the trajectory with an error of 1.4 cm, which is more than that of the

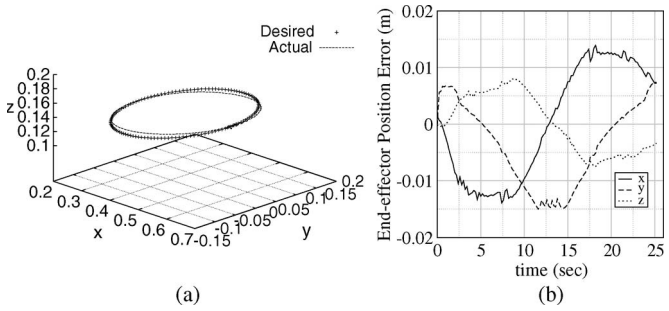


Fig. 21. End-effector motion in real-time experiment with quadratic cost minimization. (a) Trajectory (in meters). (b) Position.

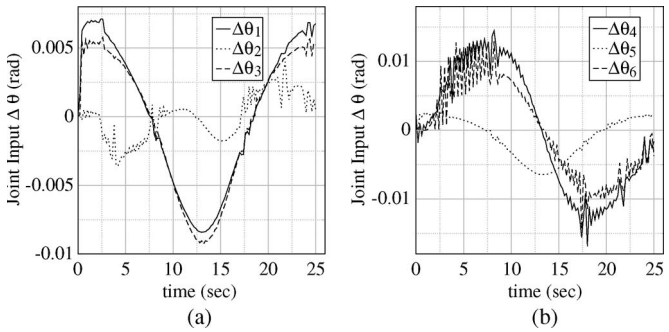


Fig. 22. Joint angle input $\Delta\boldsymbol{\theta}$ in real-time experiment with quadratic cost minimization. (a) $(\Delta\theta_1, \Delta\theta_2, \Delta\theta_3)$. (b) $(\Delta\theta_4, \Delta\theta_5, \Delta\theta_6)$.

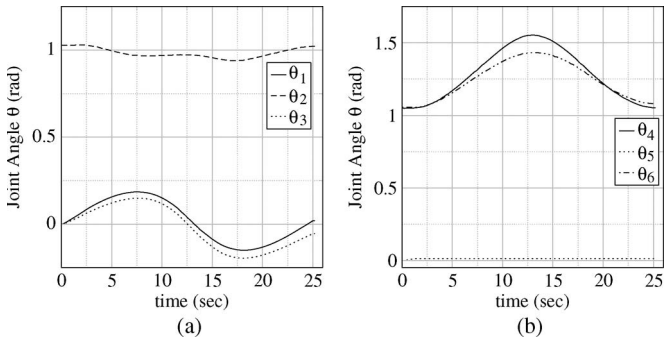


Fig. 23. Joint angle $\boldsymbol{\theta}$ in real-time experiment while minimizing quadratic cost. (a) $(\theta_1, \theta_2, \theta_3)$. (b) $(\theta_4, \theta_5, \theta_6)$.

simulation results due to model inaccuracies. This performance can be further improved by updating the critic in real time.

The corresponding input and the joint angle configurations are shown in Figs. 22 and 23, respectively, which clearly show that the real-time performance is closer to the simulation results in spite of model inaccuracies.

The end-effector trajectory in vision space is shown in Fig. 24. The trajectory is noisy due to the image processing inaccuracy associated with the identification of the end effector using the centroid of the red tape.

### *B. Joint Limit Avoidance*

The elliptical trajectory is tracked considering the kinematic limits with the network obtained in Section VI-B. The kinematic limit of the fourth joint due to engineering design is $[-90^\circ, 90^\circ]$, as tabulated in Table II. An additional constraint

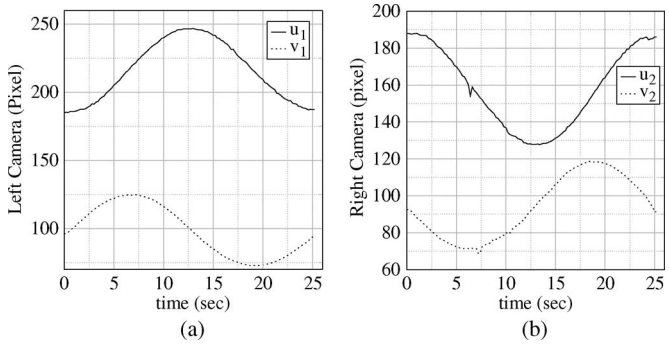


Fig. 24. Trajectory of the end-effector in the vision space during quadratic cost minimization. (a) Left camera $x-y$ coordinates $(u_1, v_1)$. (b) Right camera $x-y$ coordinates $(u_2, v_2)$.

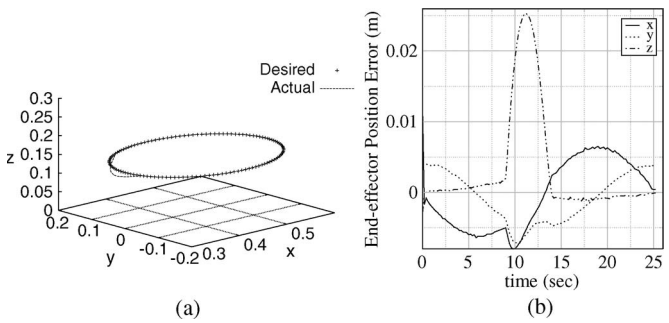


Fig. 25. End-effector motion in real-time experiment with kinematic limit avoidance. (a) Trajectory (in meters). (b) Error.

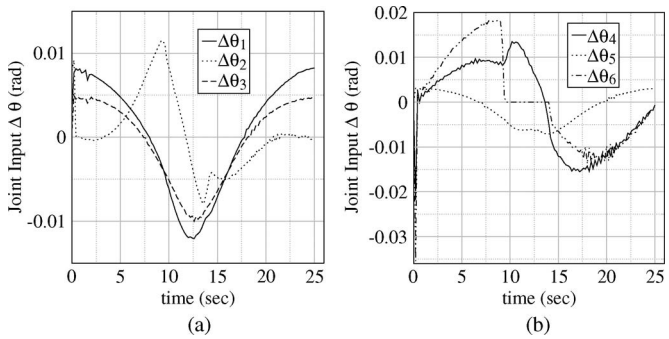


Fig. 26. Real-time experiment: Joint angle input $\Delta\boldsymbol{\theta}$ while tracking an elliptical trajectory with kinematic limit avoidance. (a) $(\Delta\theta_1, \Delta\theta_2, \Delta\theta_3)$. (b) $(\Delta\theta_4, \Delta\theta_5, \Delta\theta_6)$.

(20) is considered, and the critic network is learned. The learned network is used in real-time experiment.

The end-effector trajectory and its corresponding robot configuration are shown in Fig. 25. It is observed that the end effector tracks the trajectory with an accuracy of 2.55 cm in a narrow operating zone, which is more than the theoretical accuracy of 7 mm, indicating that the network has to be updated more in those operating zones. The corresponding input and joint angle configurations are shown in Figs. 26 and 27. It is clear from the figures that the joint angle of the fourth link is within the kinematic limit, similar to the simulation results. The fourth joint angle is separately shown in Fig. 28 for better comparison of simulation and experimental results. The experimental result is similar to the simulation, and the manipulator avoids the kinematic limit with SNAC-based approach effectively. The

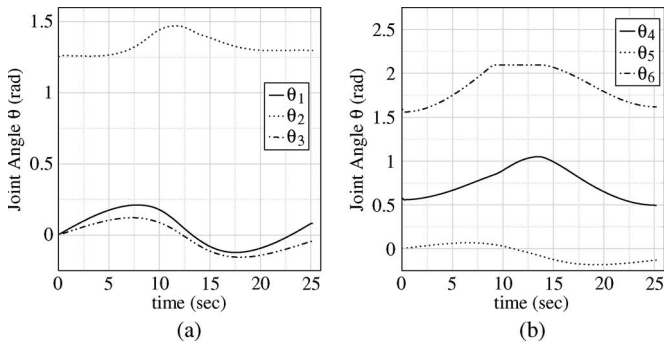


Fig. 27. Real-time experiment: Joint angle $\boldsymbol{\theta}$ while tracking an elliptical trajectory with kinematic limit avoidance. (a) $(\theta_1, \theta_2, \theta_3)$. (b) $(\theta_4, \theta_5, \theta_6)$.

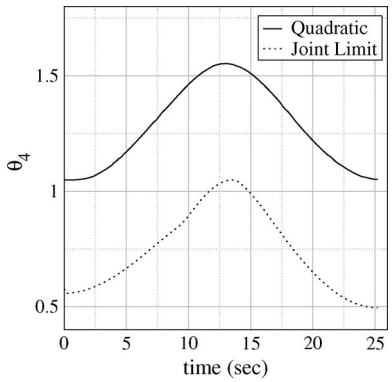


Fig. 28. Real-time experiment: Joint angle of the fourth link with and without kinematic limit avoidance.

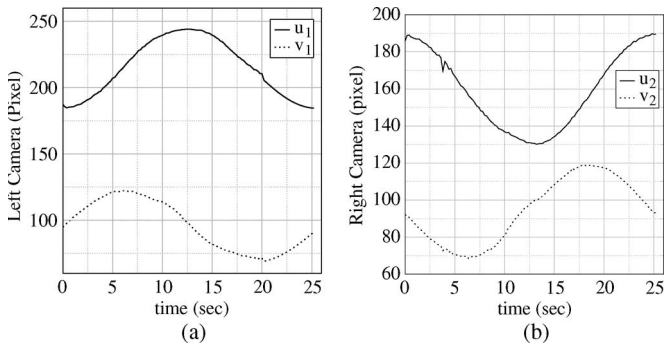


Fig. 29. Trajectory of the end effector in the vision space during kinematic limit avoidance. (a) Left camera $x-y$ coordinates $(u_1, v_1)$. (b) Right camera $x-y$ coordinates $(u_2, v_2)$.

trajectory of the end effector as seen through the vision space is shown in Fig. 29. It is easy to note that the trajectory is similar to that of quadratic cost minimization shown in Fig. 24 due to accurate tracking.

## VIII. Conclusion

A novel SNAC-based scheme has been proposed for optimal redundancy resolution scheme of the kinematically controlled robot manipulators. The proposed scheme formulates the end-effector closed-loop positioning task as a discrete-time input affine system, and the desired additional task is modeled as an integral quadratic cost function. With such formulation, the optimal control strategies can be used to resolve the redundancy. The real-time optimal redundancy resolution is achieved by implementing the optimal controller using SNAC. The TS fuzzy system is used as the critic network to learn the costate dynamics in terms of the local linear dynamics. The proposed scheme resolves the redundancy as the end effector moves in discrete steps toward the desired position. With such simplicity, a global optimal solution is achieved with integral cost function, even without computing the pseudoinverse of the forward Jacobian. The computational cost of the proposed adaptive critic-based scheme is analyzed and compared with the pseudoinverse-based redundancy resolution. It is shown that the computational requirement of the critic-based method is linear with the degree of redundancy, while the pseudoinverse-based technique is in quadratic order. The proposed adaptive critic-based kinematic control scheme has been tested in both simulation and real time on a 7DOF PowerCube manipulator with quadratic cost function and kinematic limits. The adaptive critic-based redundancy resolution will be further tested for obstacle avoidance.

## References

[1] Y. Zhang and J. Wang, "Obstacle avoidance for kinematic control redundant manipulators using a dual neural network," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 752–759, Feb. 2004.

[2] T. F. Chan and R. V. Dubey, "A weighted least-norm based solution scheme for avoiding joint limits for redundant joint manipulators," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 286–292, Apr. 1995.

[3] Y. Zhang, S. S. Ge, and T. H. Lee, "A unified quadratic programming-based dynamical system approach to torque minimization of physically constrained redundant manipulators," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 5, pp. 2126–2132, Oct. 2004.

[4] A. Ryberg, M. Ericsson, A. K. Christiansson, K. Eriksson, J. Nilsson, and M. Larsson, "Stereo vision for path correction in off-line programmed robot welding," in *Proc. IEEE ICIT*, Mar. 2010, pp. 1700–1705.

[5] W. Choi, M. Akbarian, V. Rubstov, and C.-J. Kim, "Microhand with internal visual system," *IEEE Trans. Ind. Electron.*, vol. 56, no. 4, pp. 1005–1011, Apr. 2009.

[6] Y. Motai and A. Kosaka, "Hand-eye calibration applied to viewpoint selection for robotic vision," *IEEE Trans. Ind. Electron.*, vol. 55, no. 10, pp. 3731–3831, Oct. 2008.

[7] F. Chaumette, "Image moments: A general and useful set of features for visual servoing," *IEEE Trans. Robot. Autom.*, vol. 20, no. 4, pp. 713–723, Aug. 2004.

[8] W. Wilson, C. Hulls, and G. Bell, "Relative end-effector control using Cartesian position based servoing," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 684–696, Oct. 1996.

[9] L. Deng, F. Janabi-Sharifi, and W. J. Wilson, "Hybrid motion control and planning strategies for visual servoing," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1024–1040, Aug. 2005.

[10] F. Chaumette and S. Hutchinson, "Visual servo control Part I: Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[11] F. Chaumette and S. Hutchinson, "Visual servo control Part II: Advanced approaches (tutorial)," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 109–118, Mar. 2007.

[12] F. Chaumette and E. Marchand, "A redundancy-based iterative approach for avoiding joint limits: Application to visual servoing," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 719–730, Oct. 2001.

[13] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *J. Intell. Robot. Syst.*, vol. 4, no. 4, pp. 201–212, Aug. 1990.

[14] D. P. Martin, J. Baillieul, and J. M. Hollerbach, "Resolution of kinematic redundancy using optimization," *IEEE Trans. Robot. Autom.*, vol. 5, no. 4, pp. 529–533, Aug. 1989.

[15] S.-W. Kim, K.-B. Park, and J.-J. Lee, "Redundancy resolution of robot manipulators using optimal kinematic control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1994, vol. 1, pp. 683–688.

[16] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control*, D. A. White and D. A Sofge, Eds. New York: Multiscience, 1992.

[17] D. V. Prokhorov and D. C. Wunsch, II, "Adaptive critic designs," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 997–1007, Sep. 1997.
[18] S. Ferrari and R. F. Stengel, "Model based adaptive critic designs," in *Handbook of Learning and Approximate Dynamic Programming*, J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, II, Eds. Piscataway, NJ: IEEE Press, 2004.
[19] X. Liu and S. N. Balakrishnan, "Convergence analysis of adaptive critic based optimal control," in *Proc. Amer. Control Conf.*, Chicago, IL, 2000, pp. 1929–1933.
[20] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.
[21] T. Cheng, F. L. Lewis, and M. Abu-Khalaf, "Fixed-final-time-constrained optimal control of nonlinear systems using neural network HJB approach," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1725–1737, Nov. 2007.
[22] S. Mohagheghi, Y. del Valle, G. K. Venayagamoorthy, and R. G. Harley, "A proportional-integrator type adaptive critic design-based neurocontroller for a static compensator in a multimachine power system," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 86–96, Feb. 2007.
[23] R. Padhi, N. Unnikrishnan, X. Wang, and S. N. Balakrishnan, "A single network adaptive critic (SNAC) architecture for optimal control," *Neural Netw.*, vol. 19, no. 10, pp. 1648–1660, Dec. 2006.
[24] P. Prem Kumar, L. Behera, and G. Prasad, "Adaptive critic based redundancy resolution scheme for robot manipulators," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2009, vol. 1, pp. 3181–3186.
[25] Amtec Robotics. [Online]. Available: http://www.amtec-robotics.com
[26] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Hoboken, NJ: Wiley, 2005.
[27] P. Prem Kumar and L. Behera, "Visual servoing of redundant manipulator with Jacobian estimation using self-organizing map," *Robot. Auton. Syst.*, vol. 58, no. 8, pp. 978–990, Aug. 2010.
[28] H. Zghal, R. V. Dubey, and J. A. Euler, "Efficient gradient projection optimization for manipulators with multiple degrees of redundancy," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1990, vol. 2, pp. 1006–1011.
[29] G. H. Golub and C. Reinsch, "Singular value decomposition and least square solutions," *Numer. Math.*, vol. 14, no. 5, pp. 403–420, Apr. 1970.
[30] Unibrain. [Online]. Available: http://www.unibrain.com/

**Prem Kumar Patchaikani** (S'09) received the B.E degree in electrical and electronics engineering from Thiagarajar College of Engineering, Madurai, India, in 2003 and the M.Tech degree in control system engineering from Indian Institute of Technology, Kanpur, India, in 2005, where he is currently working toward the Ph.D. degree in the Department of Electrical Engineering.

From 2005 to 2006, he was a Design Engineer with Larsen and Toubro Ltd., Mumbai, India. His primary research interests include vision-based redundant manipulator control in dynamic environment and neuro–fuzzy control schemes.

**Laxmidhar Behera** (S'92–M'03–SM'03) received the B.Sc. and M.Sc. degrees in engineering from National Institute of Technology Rourkela, Rourkela, India, in 1988 and 1990, respectively, and the Ph.D. degree from the Indian Institute of Technology (IIT) Delhi, New Delhi, India. He pursued his postdoctoral studies in the German National Research Center for Information Technology (GMD), Sank Augustin, Germany, in 2000–2001.

He was an Assistant Professor with the Birla Institute of Technology and Science, Pilani, India, in 1995–1999. He has also worked as a Reader with the University of Ulster, Londonderry, U.K., and as a Visiting Researcher at Fraunhofer-Gesellschaft, Sankt Augustin, Bonn, and at Eidgenoessische Technische Hochschule Zurich, Zurich, Switzerland. He is currently a Professor with the Department of Electrical Engineering, IIT, Kanpur, India. He has more than 120 papers to his credit published in refereed journals and presented in conference proceedings. His research interests include intelligent control, robotics, neural networks, and cognitive modeling.

**Girijesh Prasad** (M'98–SM'07) received the B.Tech. degree in electrical engineering from the Regional Engineering College, Calicut, India, in 1987, the M.Tech. degree in computer science and technology from the University of Roorkee, Roorkee, India, in 1992, and the Ph.D. degree from Queen's University Belfast, Belfast, U.K., in 1997.

Since 1999, he has been with the School of Computing and Intelligent Systems, Faculty of Computing and Engineering, University of Ulster, Londonderry, U.K., where he is currently a Reader and an Executive Member of the Intelligent Systems Research Center, leading the Brain–Computer Interface (BCI) and Assistive Technology Team. He is the author of more than 120 peer-reviewed academic papers in international journals, books, and conference proceedings. His research interests are in self-organizing hybrid intelligent systems, neural computation, fuzzy neural networks, evolutionary algorithms, adaptive predictive modeling and control, and BCI and assistive robotics.

Dr. Prasad is a Chartered Engineer and a member of the Institution of Engineering and Technology.