Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Adaptive Critic based Redundancy Resolution scheme for Robot Manipulators

Prem Kumar P.*, Laxmidhar Behera*† and Girijesh Prasad †
*Department of Electrical Engineering
Indian Institute of Technology Kanpur, Kanpur, India
†School of Computing and Intelligent systems
University of Ulster, UK

***Abstract*—A novel adaptive critic based kinematic control scheme for a redundant manipulator has been proposed in this paper. The redundancy resolution has been formulated as a discrete-time optimal control problem. A Takagi-Sugeno (T-S) fuzzy based critic network is proposed to predict the global costate dynamics as fuzzy average of local costate dynamics. The integral cost function is used in the literature earlier, to achieve global optimum in contrast to instantaneous cost functions which gives local optimum. But both the approaches require the computation of pseudo-inverse which is computationally complex and suffers from numerical instability. In contrast, the proposed scheme does not require computation of pseudo-inverse of the Jacobian which makes the method computationally efficient. The proposed scheme is tested on 7 degree of freedom (7DOF) PowerCube manipulator from Amtec Robotics.**

***Index Terms*—Redundant Manipulator, Adaptive Critic, Inverse Kinematic Control, Approximate Dynamic Programming**

I. Introduction

Redundancy is intentionally introduced in robot manipulator to cope with the real-life challenges such as the presence of obstacles, joint limits and better manipulability. But, the control of redundant manipulator is a challenging task since the inverse kinematic relationship from Cartesian space to joint space is a one to many relationship. The redundancy resolution schemes discuss about different methodologies to exploit the redundancy for performing additional task such as obstacle avoidance [1], joint limit avoidance [2], higher manipulability and torque optimization.

Most of the existing redundancy resolution schemes minimize the instantaneous cost functions which involve computation of generalized pseudo-inverse. The reader may refer [3] to know various pseudo-inverse based redundancy resolution schemes. The augmented Jacobian methods [4] are proposed to avoid the pseudo-inverse computation by forming an augmented square Jacobian of joint space dimension using addition tasks. But the control strategy suffers with algorithmic singularity and requires inverse computation. In addition to the computation of inverse, the major drawback of instantaneous cost function is that only local optimum is reached and theoretically it would take infinite time to finish the primary task.

The global optimum can be reached with integral cost functions. Martin et al. [5] used integral cost optimization for path planning with redundant manipulator. Kim et al. [6] derived the $2n$ first order differential equations and boundary conditions for redundancy resolution in optimal control framework. Both of the above methodologies require the computation of pseudo-inverse. Most of the existing redundancy resolution schemes use the complete knowledge of the robot and accurate mathematical model is required. In contrast to these approaches, we propose approximate dynamic programming based adaptive critic formulation for redundancy resolution of manipulators by minimizing the integral cost function, which does not require the computation of pseudo-inverse to achieve global optimality. The closed loop positioning task is modelled as a discrete-time dynamic task and then adaptive critic approach is proposed.

Given a dynamical system with performance cost index, the optimal control problem can be solved using Pontryagin's minimum principle and Bellman's Dynamic programming [7]–[9]. Dynamic programming gives a comprehensive computational technique based on the principle of optimality and results in an optimal controller in state feedback form. Though a feedback form is robust to noise and model uncertainties, the associated Hamilton-Jacobi-Bellman equation demands expensive computation and storage and is an off-line process where the problem is solved in the backward direction from the end point. In [5] and [6], the dynamic programming approach is considered and resulted in an optimal controller which requires the computation of pseudo-inverse.

Werbos [10] proposed Approximate Dynamic Programming (ADP) to overcome these issues, which solved the dynamic programming in forward direction and implemented online. ADP uses dual neural network architecture known as Adaptive Critic (AC) to solve the dynamic programming in forward architecture. In general Adaptive Critic has two networks namely action network and critic network. The action network represents the relationship between the state and the input while the critic network represents the relationship between the state and the costate which evaluates the performance of an actor. Based on the feedback from the critic network, the performance of the actor is improved. The readers may refer [11]–[13] to know about various architectures and the stability proof of adaptive critic.

It is shown in this work, that the positioning task can be modelled as a discrete-time input affine nonlinear system and the redundancy is resolved by minimizing the quadratic cost using a single network adaptive critic (SNAC) proposed by

Padhi et al. [14]. SNAC is particularly suitable for optimal control problem for which the optimal control input can be explicitly expressed in terms of state and costate variables. The advantage of SNAC over other architecture is that it has only critic network and it is shown that the network converges to optimal values for linear systems. In general, the existing adaptive critic methodologies employ a multi-layered perceptron as the critic network and the learned weights does not convey any meaningful insight into the optimality. The network architectures which could give information about optimality for control engineers is not analyzed yet. Though the adaptive critic is successfully implemented in real-life problems [15]–[17], the application to kinematic control of redundant manipulator is not yet discusses up-to the author's knowledge.

This paper implements the adaptive critic for kinematic control of redundant manipulator and addresses the issue of meaningful architecture for a single network adaptive critic with T-S fuzzy model. The T-S fuzzy model is chosen in particular because it approximates the complex nonlinear function as a cluster of linear functions which gives meaningful information about optimality in terms of local dynamics. It is shown that the inverse kinematic control in adaptive critic framework does not require the computation of pseudo-inverse. The adaptive critic network replaces the pseudo-inverse computation with training which makes the proposed scheme computationally efficient. The proposed control scheme is simulated on kinematic control of PowerCube manipulator in Cartesian space and then experimental results for vision based control of redundant manipulator with single camera mounted on the end-effector is presented.

The rest of the paper is organized as follows. The robot positioning task is expressed as a discrete-time control problem in the next section followed by a brief introduction of single network adaptive critic with T-S fuzzy system in section III. The simulation results and experimental results are later presented in section IV and V respectively. The paper is finally concluded in section VI.

II. Discrete-time Kinematic Control Problem

A 7DOF PowerCube™ robot manipulator from Amtec Robotics [18] has been considered for both simulation and experimentation. The forward kinematic expression for this manipulator is obtained from D-H parameters [19] presented in Table I. The robot link dimensions of the existing dimension are as follows: $d_1 = 0.318m$, $d_3 = 0.3375m$, $d_5 = 0.3085m$, and $d_7 = 0.2656m$. The derived forward kinematic equation indicates that the joint angle θ_7 is not contributing to the position change of the end-effector since the 7^{th} link exhibits the roll motion of end-effector. Hence, the 7^{th} link is ignored in further discussion.

TABLE I
D-H Parameters of PowerCube

link (i)	α_i	a_i	d_i	θ_i
1	-90	0	d_1	θ_1
2	90	0	0	θ_2
3	-90	0	d_3	θ_3
4	90	0	0	θ_4
5	-90	0	d_5	θ_5
6	90	0	0	θ_6
6	0	0	d_7	θ_7

Given the generic kinematic expression of a manipulator as $\dot{\boldsymbol{x}} = J\dot{\boldsymbol{\theta}}$, the discrete-time kinematic equation looks as:

$$\Delta\boldsymbol{x} = J\Delta\boldsymbol{\theta} \tag{1}$$

where $\Delta\boldsymbol{x} = \begin{bmatrix} \Delta x & \Delta y & \Delta z \end{bmatrix}^T$, $\Delta\boldsymbol{\theta} = \begin{bmatrix} \Delta\theta_1 & \Delta\theta_2 & \Delta\theta_3 & \Delta\theta_4 & \Delta\theta_5 & \Delta\theta_6 \end{bmatrix}^T$, and J is the forward kinematic Jacobian.

The forward differential kinematics can be expressed as a set of discrete motion of end-effector in Cartesian space at different instants as follows,

$$\boldsymbol{x}(k+1) - \boldsymbol{x}(k) = J\Delta\boldsymbol{\theta}(k) \tag{2}$$

where, $\boldsymbol{x}(k+1)$, $\boldsymbol{x}(k)$ are the end-effector position as $(k+1)^{th}$ and k^{th} instants respectively and $\Delta\boldsymbol{\theta}(k)$ is the change in joint angle at the k^{th} instant.

The above discrete motion can be expressed as a dynamic system as,

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

where, I, the identity matrix is the system matrix for dynamic positioning of the manipulator. The above equation, represents the positioning task as a discrete-time dynamic system. The closed loop error dynamics which moves the end-effector from current position $\boldsymbol{x}$ to the desired position $\boldsymbol{x_d}$ can be derived as,

$$\begin{aligned} \boldsymbol{e}(k+1) &= \boldsymbol{e}(k) - J\Delta\boldsymbol{\theta}(k) \\ &= A\boldsymbol{e}(k) + B\boldsymbol{u}(k) \end{aligned} \tag{4}$$

where, $\boldsymbol{e}(k) = \boldsymbol{x_d}(k) - \boldsymbol{x}(k)$, $\boldsymbol{x_d}(k+1) = \boldsymbol{x_d}(k)$, $A = I$, $B = -J$ and $\boldsymbol{u}(k) = \Delta\boldsymbol{\theta}(k)$.

The discrete-time dynamic representation of closed loop system is of input affine form with constant system matrix $A = I$ and a nonlinear input matrix $B = -J$. An adaptive critic architecture which is particularly suitable for such input affine system will be discussed further.

III. A T-S Fuzzy based Adaptive Critic

Padhi et al. [14] has proposed a single network adaptive critic (SNAC) for discrete-time dynamic systems. This architecture is particularly suitable for affine system,

$$\boldsymbol{x}(k+1) = f(\boldsymbol{x}(k)) + g(\boldsymbol{x}(k))\boldsymbol{u}(k) \tag{5}$$

with a quadratic cost function,

$$\begin{aligned} J_c &= \frac{1}{2}\sum_{k=0}^{\infty}(\boldsymbol{x}^T(k)Q\boldsymbol{x}(k) + \boldsymbol{u}^T(k)R\boldsymbol{u}(k)) \\ &= \frac{1}{2}\sum_{k=0}^{\infty} L(\boldsymbol{x}(k), \boldsymbol{u}(k)) \end{aligned} \tag{6}$$

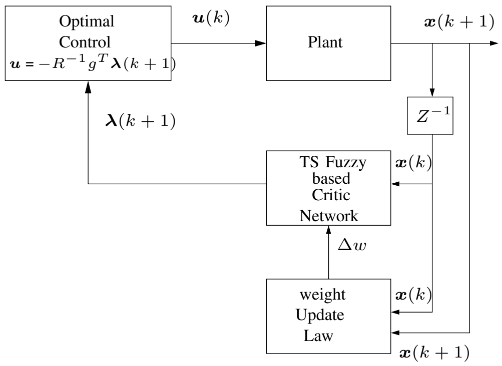


Fig. 1. Control Scheme with Discrete-time Adaptive Critic

where $\boldsymbol{x}(k) \in R^n$, $\boldsymbol{u}(k) \in R^m$, $L(\boldsymbol{x}(k), \boldsymbol{u}(k))$ is the utility function and in general, Q is taken as positive-semi definite matrix and R is a positive definite matrix. In this case, the expression for optimal control input is given by,

$$\boldsymbol{u}(k) = -R^{-1}g^T(\boldsymbol{x}(k))\boldsymbol{\lambda}(k+1) \tag{7}$$

where, $\boldsymbol{\lambda}(k+1) = \frac{\partial J_c^*(k+1)}{\partial \boldsymbol{x}(k+1)}$ and J_c^* is the optimal cost.

A single network adaptive critic is proposed in [14] to learn the costate dynamics along the path given by,

$$\boldsymbol{\lambda}(k) = \frac{\partial L}{\partial \boldsymbol{x}(k)} + \left(\frac{\partial \boldsymbol{x}(k+1)}{\partial \boldsymbol{x}(k)}\right)^T \boldsymbol{\lambda}(k+1) \tag{8}$$

In case of linear systems, a critic network of following architecture is considered.

$$\hat{\boldsymbol{\lambda}}(k+1) = W\boldsymbol{x}(k) \tag{9}$$

It is shown that the weights of the above critic network approaches the optimum value of linear system which is given by,

$$W = (I + PBR^{-1}B^T)^{-1}PA \tag{10}$$

where P is the solution of Algebraic Riccati Equation (ARE). *The reader should note that the analytical expression of optimal weights involve the computation of inverse of* $(I + PBR^{-1}B^T)^{-1}$*, which is being iteratively learnt using adaptive critic framework. With such approximate dynamic programming based scheme, the pseudo-inverse computation is avoided with learning process.* In this work, we exploit the above single network adaptive critic approach for the redundancy resolution of manipulator. It is clear from (4), that the closed loop positioning dynamics of robot manipulator is in input affine form with constant linear system matrix $f(\boldsymbol{x}) = I$ and nonlinear input matrix $g(\boldsymbol{x}) = -J(\boldsymbol{\theta})$. In a small operating zone, the input matrix can be approximated with a constant linear matrix and hence a critic network in (9) would define the costate dynamics effectively. In global operating zone, the costate dynamics can be represented as a cluster of subsystems representing local linear dynamics. Considering these facts, we propose a T-S fuzzy model based critic network such that the costate dynamics of nonlinear system will be modelled with fuzzy clusters of costate dynamics of local linear model. The T-S fuzzy model is particularly chosen because it approximates the nonlinear system as a fuzzy cluster of linear systems. This property will be very much useful for robot implementation since the learned network would represent the optimal weights of local linear model in each fuzzy zone.

The T-S fuzzy based critic network is given by,

IF $x_1(k)$ is F_1^i AND $\cdots$ AND $x_n(k)$ is F_n^i THEN

$$\hat{\boldsymbol{\lambda}}(k+1) = W_i\boldsymbol{x}(k)$$

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

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

where $\mu_i^j(x_j)$ is the membership function of the fuzzy set F_j^i, $i = 1, 2, \cdots, m$.

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

$$\hat{\boldsymbol{\lambda}}(k+1) = \frac{\sum_{i=0}^{m} \mu_i W_i \boldsymbol{x}(k)}{\sum_{i=0}^{N} \mu_i} \tag{12}$$

The reader should note that the costate dynamics in each fuzzy zone is learnt with a network similar to that of costate dynamics of linear system. Hence, the network learns the nonlinear costate dynamics in terms of costate dynamics of local linear model. We know from [14] that the weights would converge to optimum value for linear systems. Hence, with the proposed T-S fuzzy based critic network, the weights would vary smoothly from one zone to another and would converge to the optimal value corresponding to local linear model of each fuzzy zone.

The network should be trained such that the weight would converge to the optimal weights of local linear models. The network is learned from a selected fuzzy zone to the entire workspace such that the weights would converge to optimal values in each zone. To achieve network convergence, we define $S_i = \{\boldsymbol{x}(k) : \| x(k) \| < C_i\}$, 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 control scheme is shown in figure 1. The current state $x(k)$ of the system is given as input to the critic network and the output of the critic network is the costate vector $\lambda(k+1)$. The control input is computed from the output of critic network. The weights of the critic network is learned as follows:

1) Generate N_D random initial operating points for S_i where $i = 1, 2, \ldots, I$, as explained above in each zone of operation. Initialize $i = 1$ and $k = 0$. Repeat the following steps for each member of $\boldsymbol{x}(k)$ in S_i.

2) Give $\boldsymbol{x}(k)$ as input to critic network and compute $\boldsymbol{\lambda}(k+1) = \boldsymbol{\lambda_a}(k+1)$.
3) Compute the input $\boldsymbol{u}(k)$ from (7), using $\boldsymbol{\lambda_a}(k+1)$.
4) Give the input to the plant dynamics (5) and compute $\boldsymbol{x}(k+1)$.
5) Give $\boldsymbol{x}(k+1)$ as input to the critic network and compute $\boldsymbol{\lambda}(k+2)$.
6) Compute $\boldsymbol{\lambda_d}(k+1)$ using $\boldsymbol{\lambda}(k+2)$, $\boldsymbol{x}(k+1)$ in costate dynamics (8). Consider $\boldsymbol{\lambda_d}(k+1)$ as the desired value and update the weights to minimize $\| \boldsymbol{\lambda_d}(k+1) - \boldsymbol{\lambda_a}(k+1) \|$.
7) Increment k and repeat from Step 2 for N instants.
8) Repeat from Step 2 for N_D random points in S_i, with $k = 0$.
9) Check for convergence of the weights of critic network. If convergence is achieved, go to step 1 with $i = i + 1$. Otherwise, repeat steps $2 - 8$ for all the members of S_i.
10) Repeat steps $1 - 9$ till $i = I$.

The linear model of the plant in a given fuzzy zone can be obtained by linearization and the corresponding optimal weights can be easily computed using ARE. The optimal weights in each zone will vary smoothly from the optimal value at the given fuzzy zone, since the system dynamics gradually deviate from linear behavior as the zone of operation increases. Considering this fact, the weights of the entire critic network are always initialized with optimal value at a known fuzzy zone which is computed using ARE.

IV. Simulation Results

As discussed earlier, T-S Fuzzy based critic network is designed for stabilization of states and the states are fuzzified to obtain the network. Hence, for closed loop positioning of robot end-effector, the critic network should be fuzzified with end-effector position error $\mathbf{e}(\mathbf{k})$. But 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 position of the end-effector. Hence, the critic network will be inaccurate if we consider the error as a input to fuzzifier, since error would change based on both the current and desired position and Jacobian depends on the current position of the end-effector.

To cope with this inaccuracies with state $\boldsymbol{e}(k)$ based fuzzification, in this simulation, the current position of the end-effector $\boldsymbol{x} = [x\ y\ z]^T$ is fuzzified, instead of $\boldsymbol{e}(k)$. The current position of the end-effector is particularly chosen because the forward Jacobian of the manipulator continuously changes as the end-effector moves along the closed loop trajectory. With current position as the input to the fuzzifier, the critic network will account for the change in the Jacobian and would get trained for optimal path considering the current robot configuration. As the manipulator changes its joint angle configuration along the trajectory, the different fuzzy zones of critic will be activated and the end-effector would follow the optimal path. The current position of robot end-effector in selecting the zone of operation is suggested in [20] also, where a self-organizing map is used to learn the inverse kinematic relationship for closed loop control from vision space to joint space.

With above modifications, the i^{th} rule of T-S fuzzy based critic network for robot end-effector positioning is given by,

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

The joint angle input $\Delta\boldsymbol{\theta}(k)$ is computed using (7) as,

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

where, $\hat{\boldsymbol{\lambda}}(k+1)$ is obtained from the critic network, which optimizes the given cost. *The advantage of critic based approach is that the redundancy resolution for positioning task is achieved without the computation of pseudo-inverse of Jacobian while minimizing the chosen integral cost also. Hence, we could obtain a computationally efficient global optimum solution with critic based approach.*

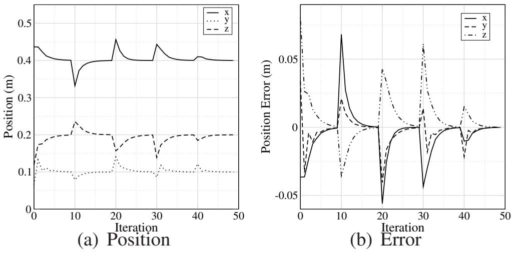
The control task considered is to compute the input $\boldsymbol{u}(k) = \boldsymbol{\Delta\theta}(\mathbf{k})$, which minimizes the cost function,

$$J_c = \frac{1}{2}\sum_{k=0}^{\infty}(\boldsymbol{x}^T(k) Q \boldsymbol{x}(k) + \boldsymbol{u}^T(k) R \boldsymbol{u}(k)) \tag{14}$$

where, Q and R are chosen as Identity matrix of appropriate dimension.

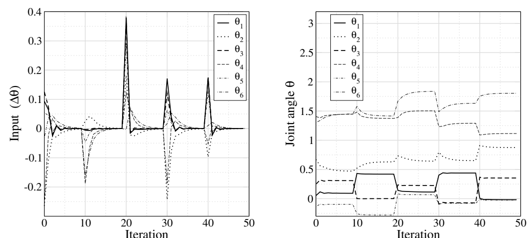
The region of operation is chosen as the cubic volume with diagonal vertices $(0.2, -0.25, 0.0)$ and $(0.7, 0.25, 0.3)$. The operating zone is fuzzified into 5 fuzzy zones in each direction, i.e., along all the three axis. Hence, totally there will be 125 fuzzy zones in the operating region and the Gaussian function is chosen as fuzzy membership function. The width of fuzzy membership function is selected such that at each operating point, the effect of two fuzzy zone will be prominent. The critic network is trained with 500000 random points chosen in the operating zone. The critic network is run for $N = 50$ iterative steps from each operating point and updated with learning rate 0.5. After critic training, the controller performance is analyzed in two stages. In first stage, the positioning accuracy with respect to the given desired position is tested. The desired position of the robot is considered as $\boldsymbol{x_d} = (0.4, 0.1, 0.2)$ and the robot is started from different initial positions. Figure 2(a) shows the position of robot end-effector starting from different initial points towards the desired position $\boldsymbol{x_d}$ and the corresponding error is shown in fig.2(b). Figure 3(a) shows the optimal control input i.e., $\Delta\boldsymbol{\theta}$ and the corresponding joint angle trajectories are shown in fig.3(b). It is clear from the figures that the robot reaches the given desired position from arbitrary initial positions and the joint angles are within the limit. The performance of the controller is checked after 10 iterative steps from each initial position. It is observed in the simulation that the robot could reach the desired position within an accuracy of $0.5mm$ in 10 iterative steps.

The controller performance is further analyzed for random positioning from arbitrary initial position to final position. The



(a) Position (b) Error

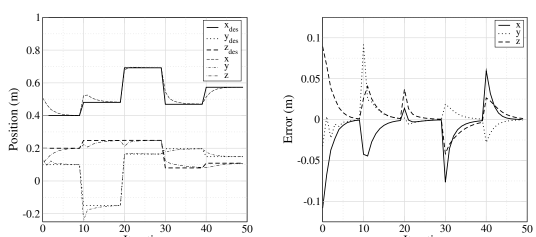
Fig. 2. Robot End-effector Motion to reach a given desired Position :: (a) Position (b) Error



(a) Input (b) Joint Angle

Fig. 3. Joint configuration at each iteration while reaching the given desired position: (a) Input($\Delta\theta$) (b) Joint Angle(θ)

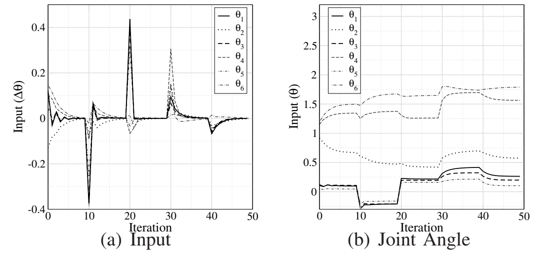
robot end-effector trajectory while moving from current position to the arbitrary desired position in the region of operation is shown in figure 4(a) and the corresponding position error is shown in fig.4(b). The convergence accuracy is checked after 10 iterative steps. It is observed in the simulation that the manipulator could reach any arbitrary position from a random initial position within the accuracy of $0.5mm$ in 10 iterative steps, indicating that the optimal controller is learnt uniformly over the entire workspace. The optimal controller



(a) Position (b) Error

Fig. 4. Robot End-effector Motion :: (a) Position (b) Error

output is shown is fig.5(a) and the corresponding joint angle configurations are shown in fig.5(b). It is clear from the figure that the joint angles are within the limit and the redundancy is resolved with a smooth joint trajectory over the entire workspace. The reader should note that an arbitrary accuracy can be achieved in both the cases by increasing the number of steps.



(a) Input (b) Joint Angle

Fig. 5. Joint configuration at each iteration: (a) Input($\Delta\theta$) (b) Joint Angle(θ)

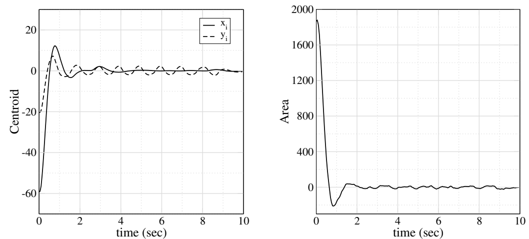
V. Experimental Results

The experimental test is performed for vision based control of PowerCube manipulator with single camera mounted on the end-effector. The end-effector Cartesian velocity is computed from the image features and then the controller is used to compute the joint space movement.

The image moments are considered as visual features and a planar rectangular object is considered for visual servoing task. The image feature vector is taken as $\boldsymbol{U} = [x_i, y_i, a, \alpha]^T$ where, (x_i, y_i) represents the centroid of the projection of object in image plane, a is the area of the object and α represents the major axis angle of the object with respect to the camera x-axis. The image features are considered to represent the Cartesian space translational velocities $\dot{x}, \dot{y}, \dot{z}$ and z-axis rotation ω_z of the end-effector. The angular velocity around the x and y axis of the end-effector is considered as zero throughout the experimentation and hence not used in computing image Jacobian L.

The image Jacobian obtained in [21] is used to compute the camera velocity $\dot{\boldsymbol{x}}_{\boldsymbol{c}} = [\dot{x}_c, \dot{y}_c, \dot{z}_c, \omega_c]^T$ required to move the camera from the current image feature $\boldsymbol{x}$ to the desired image feature $\boldsymbol{x_d}$. The camera velocity is computed using $\dot{\boldsymbol{x}}_{\boldsymbol{c}} = -K_i L^{-1}\boldsymbol{e}$, where K_i is a small positive gain and $\boldsymbol{e} = \boldsymbol{U_d} - \boldsymbol{U}$. The end-effector velocity $\dot{\boldsymbol{x}} = [\dot{x}, \dot{y}, \dot{z}, \omega_z]$ is computed from the camera motion using the coordinate transformation $\dot{\boldsymbol{x}} = P\dot{\boldsymbol{x}}_{\boldsymbol{c}}$, where P represents the coordinate transformation from camera frame to end-effector frame. In our experimental setup the Camera is mounted inside the gripper such that the Camera frame x-axis coincides with y-axis of end-effector and y-axis coincides with the x-axis of end-effector. Hence, the coordinate transformation represents the interchange of velocities in x and y axis direction. The joint velocity which minimizes the cost $\dot{\boldsymbol{\theta}}^T\dot{\boldsymbol{\theta}}$ is computed from the obtained end-effector velocity.

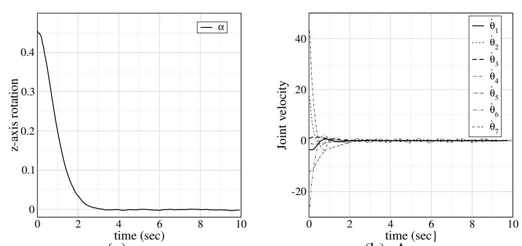
In our experiment, the desired position is considered as $10cm$ distance in z-direction, with the object plane is parallel to the camera x-y plane. The image Jacobian is computed at the desired location and is used over the visual servoing trajectory. The experimental results are shown in the figures 6 and 7. The image feature errors are shown in figures 6(a), 6(b) and 7(a), which clearly shows that the end-effector finally reaches the desired position minimizing the given cost function. The corresponding joint velocities are shown in figure



(a) Centroid (b) Area

Fig. 6. Image Feature Error: (a) Centroid (x_i, y_i) of the object (b) area of the object

7(b) indicating that the joint velocities are within the limit. As discussed above, the existing cost minimization based



(a) α (b) Area

Fig. 7. (a)Image Feature Error (α) (b) Joint Velocity with visual servoing

redundancy resolution schemes require the computation of pseudo-inverse which is computationally complex. With adaptive critic based integral cost minimization, global optimum can be reached without the computation of pseudo-inverse of forward Jacobian.

VI. Conclusion

A novel adaptive critic based kinematic control scheme has been proposed for a redundant manipulator. The redundancy resolution has been formulated as a discrete time optimal control problem. A T-S fuzzy based critic network has been used to model the costate dynamics. The proposed adaptive critic based scheme is computationally efficient compared to the existing optimization techniques since the computation of the pseudo-inverse of the forward Jacobian, J, is not required. Since redundancy resolution is achieved through real-time optimization using the proposed adaptive critic, this work is remarkable in terms of following two foundational issues in visual servoing:

- Kinematic control can be achieved without the computation of the pseudo-inverse of the forward Jacobian.
- Redundancy resolution can be achieved through real-time optimization of a task-specific cost.

The control scheme is simulated on 7DOF PowerCube manipulator with a quadratic cost function and performance has been found satisfactory. The controller is finally implemented with the Cartesian end-effector velocity obtained from the image features in single camera through eye-in-hand visual servoing. The proposed adaptive critic method will be extended further with various integral cost functions such that the manipulator can perform additional tasks such as obstacle avoidance and coordinate with other manipulators.

Acknowledgment

This work has been supported by UK-India Education and Research Initiative (UKIERI) grant 'Innovations in Intelligent Assistive Robotics'.

References

[1] Yunong Zhang and Jun Wang, "Obstacle avoidance for kinematic control redundant manipulators using a dual neural network," *IEEE Tr. on Robotics and Automation*, vol. 18, no. 6, February 2002.

[2] T. F. Chan and R. V. Dubey, "A weighted least-norm based solution scheme for avoiding joint limits for redundant joint manipulators," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 2, April 1995.

[3] Bruno Siciliano, "Kinematic control of redundant robot manipulators : A tutorial," *Journal of Intelligent and Robotic systems*, vol. 4, no. 4, August 1990.

[4] H. Seraji, "Configuration control of redundant manipulators: theory and implementation," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 4, pp. 472–490, 1989.

[5] D.P. Martin, John Baillieul, and J. M. Hollerbach, "Resolution of kinematic redundancy using optimization," *IEEE Tr. on Robotics and Automation*, vol. 5, no. 4, 1989.

[6] Sung-Woo Kim, Kang-Bark Park, and Ju-Jang Lee, "Redundancy resolution of robot manipulators using optimal kinematic control," in *IEEE International Conference on Robotics and Automation*, 1994, vol. 1, pp. 683–688.

[7] A.E. Bryson and Y.C. Ho, *Applied Optimal Control*, Taylor and Francis, 1975.

[8] Desineni S. Naidu, *Optimal Control Systems*, CRC Press, 2003.

[9] Dimitry P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, Belmount, 1995.

[10] P. J. Werbos, *Approximate dynamic programming for real-time control and neural modeling*, In D. A. White, and D. A Sofge (Eds.), *Handbook of Intelligent control, Multiscience Press*, 1992.

[11] D. V. Prokhorov an D. C. Wunsch II, "Adaptive critic designs," *IEEE Tr. on Neural Networks*, vol. 8, no. 5, pp. 997–1007, September 1997.

[12] S. Ferrari and R. F. Stengel, *Model based adaptive critic designs.*, In Jennie Si, A. G. Barto, W. B. Powell, and D. Wunsch II (Eds.), *Handbook of learning and Approximate Dynamic Programming*, IEEE Press, 2004.

[13] X. Liu and S. N. Balakrishnan, "Convergence analysis of adaptive critic based optimal control," in *Proceedings of the American Control Conference*, Chicago, USA, 2000, pp. 1929–1933.

[14] Radhakant Padhi, Nishant Unnikrishnan, Xiaohua Wang, and S.N. Balakrishnan, "A single network adaptic critic (snac) architecture for optimal control," *Neural Networks*, vol. 19, pp. 1648–1660, 2006.

[15] P. Jung-Wook, R. G. Harle, and G. K. Venayagamoorthy, "Adaptive-critc based optimal neurocontrol for synchronous generators in a power system using mlp/rbf neural networks," *IEEE Transactions on Industry Applications*, vol. 39, no. 5, pp. 1529–1540, October 2003.

[16] R. Padhi, S. N. Balakrishnan, and T. Randolph, "Adaptive critic based optimal neuro control synthesis for distributed parameter systems," *Automatica*, vol. 37, pp. 1223–1234, 2001.

[17] S. N. Balakrishnan and V. Biega, "Adaptive critic based neural network for aircraft optimal control," *Journal of Guidance, Control and Dynamics*, vol. 19, no. 4, pp. 893–898, July-August 1996.

[18] "Amtec robotics," www.amtec-robotics.com.

[19] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, John Wiley & sons Inc., 2005.

[20] Prem Kumar P., Naman R. Patel, and Laxmidhar Behera, "Visual servoing of redundant manipulator with jacobian estimation using self-organizing map," in *Proceedings of Fifth International Conference on Computation Intelligence, Robotics and Autonomous Systems (CIRAS 2008)*, 2008.

[21] F. Chaumette, "Image moments: A general and useful set of features for visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 4, August 2004.