Real-Time Adaptive Control of a Flexible Manipulator Using Reinforcement Learning

Santanu Kumar Pradhan and Bidyadhar Subudhi, *Senior Member, IEEE*

***Abstract*—This paper exploits reinforcement learning (RL) for developing real-time adaptive control of tip trajectory and deflection of a two-link flexible manipulator handling variable payloads. This proposed adaptive controller consists of a proportional derivative (PD) tracking loop and an actor-critic-based RL loop that adapts the actor and critic weights in response to payload variations while suppressing the tip deflection and tracking the desired trajectory. The actor-critic-based RL loop uses a recursive least square (RLS)-based temporal difference (TD) learning with eligibility trace and an adaptive memory to estimate the critic weights and a gradient-based estimator for estimating actor weights. Tip trajectory tracking and suppression of tip deflection performances of the proposed RL-based adaptive controller (RLAC) are compared with that of a nonlinear regression-based direct adaptive controller (DAC) and a fuzzy learning-based adaptive controller (FLAC). Simulation and experimental results envisage that the RLAC outperforms both the DAC and FLAC.**

***Note to Practitioners*—This paper shows how to control a system with distributed flexibility. The reinforcement learning approach to develop adaptive control described in the paper can be applied to control also complex flexible space shuttle system and for damping of many vibratory systems.**

***Index Terms*—Adaptive control, flexible-link manipulator, reinforcement learning, tip trajectory tracking.**

NOMENCLATURE

θ_i	Joint position of the i^{th} link.
θ_{di}	Desired joint position of the i^{th} link.
δ_i	Modal displacement for the i^{th} link.
y_{pi}	Redefined tip trajectory of the i^{th} link.
$\dot{y}_{pri}$	Tip reference velocity of the i^{th} link.
y_{mi}	Reference model output of the i^{th} link.
e_i	Tip trajectory error of the i^{th} link.
Δe_i	Change in tip trajectory error of the i^{th} link.
e_{mi}	Error w.r.t. reference model of the i^{th} link.
S_i	Measure of tip trajectory tracking accuracy for link i.
y_{ck}	Critic output.
y_{ak}	Actor output.
r_k	Reward.
$\Re_k$	Value function.
γ	Discount factor.
W_{ak}	Actor weights.
W_{ck}	Critic weights.
δ_{TD_k}	Temporal difference error at k^{th} instant.
Φ_{ck}	Critic regressor vector.
Φ_{ak}	Actor regressor vector.
z_k	Eligibility trace.
G_k	Kalman gain matrix.
P_k	Covariance matrix of the temporal difference error.
μ	Forgetting factor.
K_a	Actor adaptation gain.
a_m	Incremental adaptive memory.
ψ	Gradient vector.
S_k	Gradient updating vector.
λ	Value of the eligibility trace.

Manuscript received July 08, 2011; revised November 30, 2011; accepted February 11, 2012. Date of publication March 07, 2012; date of current version April 03, 2012. This paper was recommended for publication by Associate Editor M. K. Jeong and Editor Y. Narahari upon evaluation of the reviewers' comments.

The authors are with the Department of Electrical Engineering, National Institute of Technology, Rourkela, Orissa 769008, India (e-mail: santanupradhan.nitrkl@gmail.com; bidyadhar@nitrkl.ac.in).

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/TASE.2012.2189004

I. INTRODUCTION

FLEXIBLE-LINK MANIPULATORS (FLMs), i.e., manipulators with thin and lightweight links offer several advantages over rigid-link manipulators such as achieving high-speed operation, lower energy consumption, and increase in payload carrying capacity. These nonconvectional manipulators find applications requiring large workspace like assembly of free-flying space structures and hazardous material management from safer distance [1]. All these advantages and applications motivate towards the accelerated research in FLM control. However, controlling a FLM is difficult owing to distributed link flexibility which makes this type of manipulator system dynamics nonminimum phase and underactuated [1]. Further, control of a FLM becomes more challenging when it has to handle variable payload. In order to achieve good tip trajectory tracking while suppressing tip deflection with varied payloads, adaptive control should be employed, which can provide appropriate control torques to the actuators to achieve the above two-control tasks (good tip trajectory tracking and suppression of tip deflection).

In the past, several papers on design of adaptive controllers for FLMs with variable payloads have been reported. A simple decoupled adaptive controller comprising of the estimation of link's natural frequency for a single link flexible manipulator under variable payload is proposed in [2]. Further advancement

in adaptive controller has been proposed in [3], where a discrete-time nonlinear adaptive controller for a single-link flexible manipulator using RLS-based payload estimation is used. The above adaptive controllers suffer from dependency on identification procedure and excessive tuning of adaptive gains. Intelligent controllers based on supervised learning using neural networks [4] and fuzzy logic have been designed by some investigators [5] for FLMs under parametric uncertainty. However, neural network-based controllers require training of the synaptic weights to an optimal value which consume considerable amount of time and computational complexity. Fuzzy logic-based adaptive controller design depends upon proper formulation of control rule base. A hybrid neuro-fuzzy-based adaptive controller has been proposed in [6]. Although, the above hybrid neuro-fuzzy controller shows better performance compared to neural network and fuzzy logic-based adaptive controllers but it needs *a priori* information about the input output relationship, i.e., supervised and offline learning are essentially required. Also, adaptive control of a multilink flexible manipulator is more complex compared to a single-link flexible manipulator control problem owing to interlink coupling effects. The above discussion reveals that there is a need of a precise real-time adaptive control for FLMs under payload variation. Hence, development of a real-time adaptive control for both tip trajectory tracking and suppression of tip deflection for a two-link flexible manipulator (TLFM) handling variable payload is the objective of this paper. Unlike supervised learning, where the learning is driven by error signal (difference between desired and current response), RL occurs when an agent (manipulator) learns behaviors (tip trajectory tracking) through trial-and-error interaction with the environment (workspace)-based on "reinforcement" signals from the environment [7]–[10].

The contribution of this paper lies in developing a new RL-based real-time adaptive control for a TLFM. Motivated by the successful application of reinforcement learning in many complex systems such as an acrobot, elevator dispatching, dynamic cellular channel allocation, and inverted pendulum, etc., [7], this paper attempts to exploit actor-critic-based RL with modification in critic as well as in actor to develop an adaptive control for a TLFM. Many of the previous works on RL-based control use least square (LS) approach to estimate the weights of the value function [7]. But as the LS is a batch processing technique it is unsuitable for real-time control. Therefore, the proposed actor-critic RLAC uses a RLS-based TD learning to obtain the optimal weights of the value function in the critic. Further, a mechanism of eligibility trace [8] and adaptive memory are embedded to this TD algorithm to enhance learning what we call as Recursive Least Square-Eligibility Trace-Adaptive Memory algorithm (RLS-ET-AM) algorithm. The proposed algorithm calculates the initial critic parameters offline in order to reduce the computational overhead in real-time unlike previous approaches where either zero or random values were taken [8]. To ensure stability of the RLAC, a discrete-time PD controller is supplemented with the above RL learning. The proposed RLAC is compared with a DAC and a FLAC to validate the performances of the proposed RLAC. The rest of this paper is organized as follows. Section II presents the dynamic model of the TLFM. Section III briefly reviews two other controllers such as DAC and FLAC to enable comparison of efficacy of the proposed RLAC. In Section IV, the development of the proposed RLAC is presented. Simulation results are discussed in Section V-A followed by experimental results in Section V-B. Section VI presents conclusions.

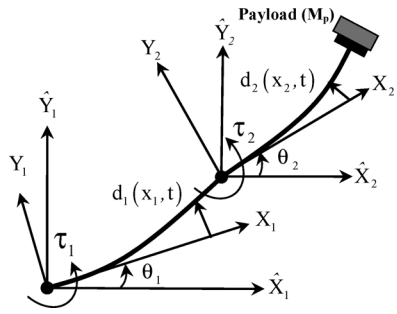

Fig. 1. Schematic diagram of a planar TLFM.

II. Dynamic Model of the TLFM

The schematic diagram of a planar TLFM is shown in Fig. 1, where τ_i is the actuated torque of the i^{th} link, θ_i is the joint angle of the i^{th} joint and $d_i(l_i,t)$ represents the deflection along i^{th} link. The outer free end of the TLFM is attached with payload mass, $\mathbf{M_p}$. The dynamics of the TLFM is given by [11]

$$\mathbf{M}(\theta_i,\delta_i)\begin{bmatrix}\ddot{\theta}_i\\ \ddot{\delta}_i\end{bmatrix}+\begin{bmatrix}\mathbf{c_1}\left(\theta_i,\delta_i,\dot{\theta}_i,\dot{\delta}_i\right)\\ \mathbf{c_2}\left(\theta_i,\delta_i,\dot{\theta}_i,\dot{\delta}_i\right)\end{bmatrix}+\mathbf{K}\begin{bmatrix}0\\ \delta_i\end{bmatrix}+\mathbf{D}\begin{bmatrix}0\\ \dot{\delta}_i\end{bmatrix}=\begin{bmatrix}\tau_i\\ 0\end{bmatrix} \quad (1)$$

where $\mathbf{M}$ is the positive-definite symmetric inertia matrix, $\mathbf{c_1}$ and $\mathbf{c_2}$ are the vectors containing of Coriolis and Centrifugal forces, respectively, $\mathbf{K}$ is the stiffness matrix, and $\mathbf{D}$ is the damping matrix. If the output is taken as tip position, the overall manipulator system becomes nonminimum phase [1]; hence, the redefined output is given by

$$y_{pi}=\theta_i+\left[\frac{d_i(l_i,t)}{l_i}\right] \quad (2)$$

where l_i is length of the i^{th} link. TLFM dynamics (1) can be rewritten in state space form as

$$\dot{x}=f_i(x)+g_i(x)\,u_i \quad (3)$$

with x as the state vector, i.e., $x=\left[\theta_i,\dot{\theta}_i,\delta_i,\dot{\delta}_i\right]^T$ and

$$f_i(x)=\mathbf{M}(\theta_i,\delta_i)^{-1}\cdot\left(-\begin{bmatrix}\mathbf{c_1}\left(\theta_i,\delta_i,\dot{\theta}_i,\dot{\delta}_i\right)\\ \mathbf{c_2}\left(\theta_i,\delta_i,\dot{\theta}_i,\dot{\delta}_i\right)\end{bmatrix}-\mathbf{K}\begin{bmatrix}0\\ \delta_i\end{bmatrix}-\mathbf{D}\begin{bmatrix}0\\ \dot{\delta}_i\end{bmatrix}\right)$$

$$g_i(x)=\mathbf{M}(\theta_i,\delta_i)^{-1}\text{ and }u_i=\begin{bmatrix}\tau_i\\ 0\end{bmatrix}.$$

δ_i and $\dot{\delta}_i$ being the modal displacement and modal velocity for the i^{th} link, respectively, and the actual output vector, $\mathbf{y}$ is given by $y=\left[\theta_i,\dot{\theta}_i\right]$. To express the dynamics of the TLFM in terms

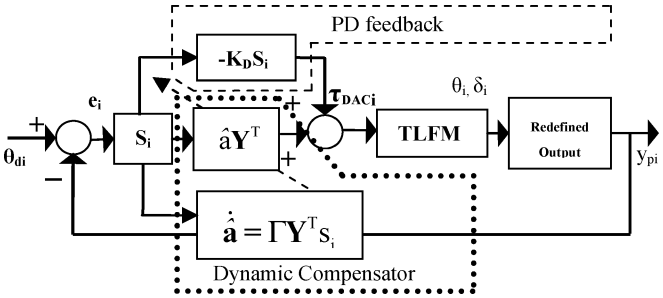

Fig. 2. Nonlinear direct adaptive controller for TLFM.

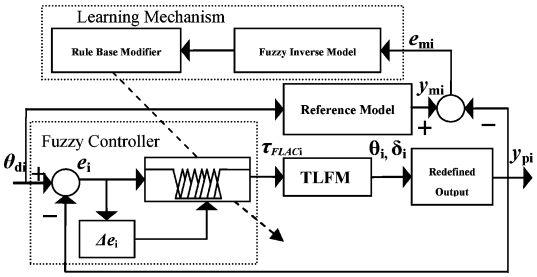

Fig. 3. Fuzzy logic-based adaptive controller for TLFM.

of the redefined tip position and tip velocity, the states are redefined as $\zeta = [y_{pi}, \dot{y}_{pi}]$.

The new state space representation of the TLFM using redefined output can be expressed as

$$\dot{\zeta} = \hbar_i(\zeta) + \lambda_i(\zeta)\, v_i \tag{4}$$

where v_i is the i^{th} torque input with respect to the redefined output ζ

$$\hbar_i(\zeta) = M(\theta_i)^{-1}\left(-\mathbf{c_1}\left(\theta_i, \delta_i, \dot{\theta}_i, \dot{\delta}_i\right) - K\delta_i - D\dot{\delta}_i\right) + \cdots M(\delta_i)^{-1}\left(-\mathbf{c_2}\left(\theta_i, \delta_i, \dot{\theta}_i, \dot{\delta}_i\right)\right)$$

and $\lambda_i(\zeta)\, v_i = M(\theta_i, \delta_i)^{-1}\tau_i$

III. Review of DAC and FLAC

A. Direct Adaptive Controller (DAC)

Fig. 2 shows the structure of the direct adaptive controller (DAC) [12]. It consists of a TLFM dynamic compensator and a PD feedback loop. The dynamic compensator provides joint dynamic torques $\hat{a}Y^T$, necessary to make the desired motions with respect to real-time estimation of the TLFM parameters. The i^{th} PD feedback loop, output $K_D S_i$ regulates the y_{pi} about the θ_{di}. The direct adaptive control law is derived as follows. Define $\mathbf{a}$ as a vector containing the parameters of the TLFM, given by

$$\mathbf{a} = [J_{li}\ J_{hi}\ J_{ieq}\ m_{ieq}\ m_c]^T$$

where J_{li}: i^{th} link inertia; J_{hi}: i^{th} hub inertia; J_{ieq}: total inertia of the i^{th} link; m_{ieq}; total mass of the i^{th} link and m_c: total coupling mass.

The choice of vector $\mathbf{a}$ is made so as to keep the number of manipulator parameter to minimum. Let $\hat{\mathbf{a}}$ be the estimate of $\mathbf{a}$. $\hat{\mathbf{M}}$ and $\hat{\mathbf{c}}_\mathbf{i}$ are the estimates of inertia matrix, Coriolis and Centrifugal force vector, respectively. Then, the TLFM dynamics (4) can be written as

$$g_i(\tilde{x})^{-1}\left[\ddot{\tilde{x}} - \tilde{f}_i(\tilde{x})\right] = \tilde{\mathbf{a}}[Y(y_{pi}, \dot{y}_{pi}, \dot{y}_{pri}, \ddot{y}_{pri},)]^T \tag{5}$$

where $\tilde{x} = y - \zeta$, $\tilde{\mathbf{a}} = \hat{\mathbf{a}} - \mathbf{a}$ is the parameter estimation error, Y is an $i \times m$ matrix independent of the TLFM dynamic parameters with m being equal to dimension of the "$\mathbf{a}$" vector for i^{th} link

$$f_i(\tilde{x}) = f_i(x) - \hbar_i(\zeta),\ g_i(\tilde{x}) = g_i(x) - \lambda_i(\zeta).$$

The direct adaptive control law can be expressed as

$$\tau_{\mathrm{DACi}} = [Y(y_{pi}, \dot{y}_{pi}, \dot{y}_{pri}, \ddot{y}_{pri},)]^T \hat{\mathbf{a}} - K_D S_i \tag{6}$$

$$\dot{\hat{\mathbf{a}}} = -\Gamma Y^T S_i \tag{7}$$

where Γ is the constant positive definite matrix, K_D is the positive definite PD gain matrix, and vector S_i is defined as

$$S_i = \dot{y}_{pi} - \dot{y}_{pri} = \dot{e}_i + \Lambda e_i. \tag{8}$$

B. Fuzzy Learning Based Adaptive Controller (FLAC)

Fig. 3 shows the structure of the FLAC [4]. It utilizes learning mechanism which automatically adjusts the rule base of the fuzzy controller (FC) so that the closed-loop performs according to the user defined reference model containing information of the desired behavior of the controlled system. It consists of three major components namely a FC, a reference model and a learning mechanism described as follows.

1) FC: has e_i and Δe_i as inputs and τ_i as output for i^{th} link with fuzzy implication of the form

$$\text{If } \langle \mathbf{e_i} \text{ and } \mathbf{\Delta e_i} \rangle \text{ Then } \langle \boldsymbol{\tau}_{\mathbf{FLACi}} \rangle$$

where Δe_i is defined as $e_{ik} - e_{ik-1}$, where e_{ik}, and e_{ik-1} are the tip trajectory error terms for i^{th} link at k^{th} and $(k-1)^{th}$ instants, respectively.

The output fuzzy set for both the links are initialized at zero, i.e., $\langle \boldsymbol{\tau}_{\mathbf{FLACi}} \approx \mathbf{0} \rangle$, the rule base is filled online [4] using the learning mechanism as shown in Fig. 3.

2) Reference Model: is chosen according to the desired closed-loop performance such as rise time and settling time. The choice of the reference model is very important as it dictates the FLAC to perform in the desired manner.

3) Learning Mechanism: performs the function of modifying the knowledge base of the FC so that the closed-loop performance behaves as the reference model. The learning block constitutes a fuzzy inverse model and a knowledge base modifier. These are explained next.

The fuzzy inverse model makes an assessment of the derivation of the current closed-loop system behavior from the specified behavior of the reference model. The design of the fuzzy inverse model requires the knowledge of the closed-loop online TLFM tip position error profile and change in error profile.The knowledge base modifier performs the function of modifying the FC so that the better payload adaptability can be achieved. To modify the knowledge base of the fuzzy controller, the rules that are "on" are determined, i.e., the value of e_{mi} is measured,

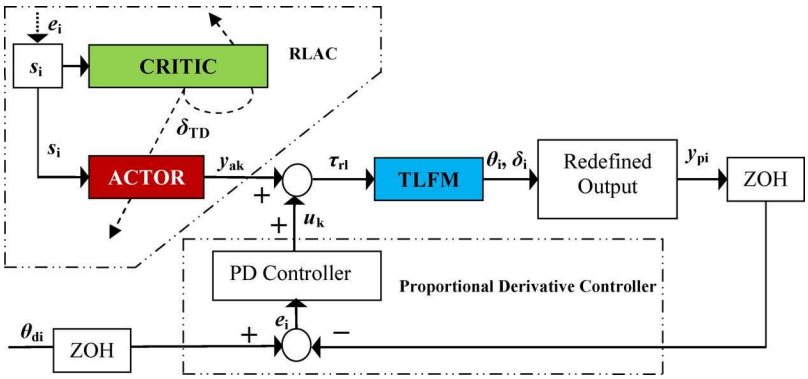

Fig. 4. Structure of reinforcement learning-based adaptive control for a TLFM carrying a variable payload.

where $e_{mi} = y_{mi} - y_{pi}$ and y_{mi} being the reference model error output and reference model output, respectively. The entries of the rule base are modified according to the value of e_{mi} and a scalar output from rule base modifier shifts the center of the fuzzy rule base using the following rule:

$$\text{If } \langle \mathbf{e_{mi}} \neq 0 \rangle \text{ Then } \langle \mathbf{p_{i+1}} = \mathbf{p_0} \rangle$$

where $e_{mi} = y_{mi} - y_{pi}$, $\mathbf{p_i}$ is the fuzzy inverse model output and $\mathbf{p_0}$ is the amount by which the rule base will be modified.

IV. Proposed Real-Time RLAC for TLFM

RL takes places when an agent (TLFM) understands and learns by observing the environment workspace)-based on a scalar internal reinforcement signal called reward r_k and δ_{TD_k} TD error at k^{th} instant TD error and it is the external reinforcement signal that comes from the environment to minimize a long term value function described next. Fig. 4 shows the structure of RLAC for real-time implementation of the TLFM carrying a variable payload. It consists of two important components such as an actor-critic block and a PD control loop. The actor-critic block adapts the actor and critic weights, W_{a_k} and W_{c_k} in order to compensate for the joint torque input under payload uncertainties.

The PD controller provides stable closed-loop performance by regulating the desired tip trajectory. A zero order hold (ZOH) block is used to achieve a discrete value of the desired tip trajectory y_{di} and redefined output y_{pi}. Thus, the net adaptive torque τ_{rl} for i^{th} link is given by

$$\tau_{rl} = y_{ak} + u_k \tag{9}$$

where u_k is the proportional derivative control action and y_{ak} is the estimated actor output. The PD control law uses the past values of tip trajectory tracking error $e_{i_{k-1}}$ for i^{th} link and the past value of the PD control output $u_{i_{k-1}}$. Thus, the i^{th} digital PD control action is generated using the following recursive law:

$$u_{i_k} = K_p \left(e_{i_k} - e_{i_{k-1}}\right) + K_d \left(e_{i_k} - 2e_{i_{k-1}} + e_{i_{k-2}}\right) + u_{i_{k-1}} \tag{10}$$

where K_p and K_d are proportional and derivative gain, respectively, and $e_{i_{k-1}}$ and $e_{i_{k-2}}$ are the tracking errors at sampling instants $(k-1)$ and $(k-2)$, respectively. $u_{i_{k-1}}$ is the control action at $(k-1)^{th}$ instant for i^{th} link.

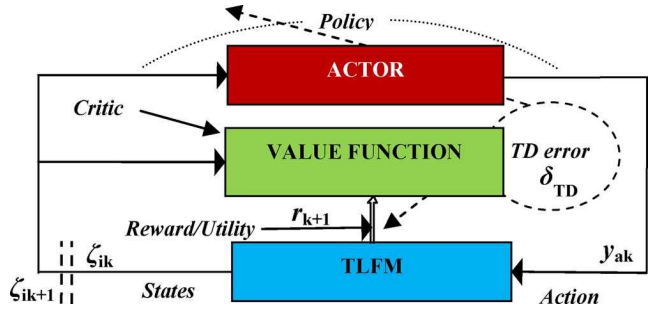

Fig. 5. Actor-Critic-based Reinforcement learning.

A. Actor-Critic Block

Fig. 5 describes the actor-critic-based RL, where y_{ak} denotes the control policy applied to the actuators of the TLFM, $\zeta_{i_k} = y_{pi_k}$ is the measured redefined tip trajectory given in (2) for the i^{th} link at k^{th} instant. Reward at $(k+1)^{th}$ instant, r_{k+1} is the result of the transition $(\zeta_{i_k}, y_{ak}, \zeta_{i_{k+1}})$, where $\zeta_{i_{k+1}}$ is the successive value of ζ_{i_k} at $(k+1)^{th}$ instant. Let a value or cost be assigned to the total cumulative reward function say $\Re_k(\zeta_k)$ expressed as

$$\Re_k(\zeta_k) = \sum_{k=0}^{\infty} \gamma^k r_{k+1} \tag{11}$$

where γ^k is the discount factor at the k^{th} instant. The value of the discount factor decides as how much weightage is to be given to future rewards. The RL searches for a control policy, y_{ak} in the actor so that it minimizes the value function defined in (11)

$$\Re_k(\zeta_k) = \min_{y_{ak}} \left(\sum_{k=0}^{\infty} \gamma^k r_{k+1} \right). \tag{12}$$

It is difficult to achieve minimization of the $\Re_k(\zeta_k)$ in real-time as (12) needs evaluation of an infinite sum backward in time. To provide forward in time solution of (12) approximation of the $\Re_k(\zeta_k)$ is necessary. In order to approximate the $\Re_k(\zeta_k)$ (12) can be rewritten as sum of k^{th} step reward r_k and discount times infinite sum of the future value function in compact form as

$$\Re_k(\zeta_k) = r_k + \gamma \sum_{k=1}^{\infty} \gamma^{k-1} r_{k+1}. \tag{13}$$

The difference equation equivalent of (13) is given [8] as

$$\Re_k(\zeta_k) = r_k + \gamma \Re_k(\zeta_{k+1}) \tag{14}$$

where

$$\Re_k(\zeta_{k+1}) = \sum_{k=1}^{\infty} \gamma^{k-1} r_{k+1}.$$

(14) is also known as Bellman equation. Based on this equation, δ_{TD_k} can be defined as

$$\delta_{\mathrm{TD}_k} = r_k + \gamma \Re_k(\zeta_{k+1}) - \Re_k(\zeta_k). \tag{15}$$

δ_{TD_k} is a prediction error between predicted and observed performance. If (15) holds good for some value of y_{ak}, then δ_{TD_k} must approach zero. Thus, (15) becomes

$$0 \cong r_k + \gamma \Re_k(\zeta_{k+1}) - \Re_k(\zeta_k). \tag{16}$$

The RLAC-based actor-critic RL consists of two separate blocks, actor and critic. In actor, the policy y_{ak} is updated and on critic, the $\Re_k(\zeta_k)$ is updated using a linear function approximator-based on RLS algorithm. Hence, the value function $\Re_k(\zeta_k)$ relating the critic weights W_{ck} can be expressed as

$$\Re_k(x_k) = \Phi_{c_k}^T W_{c_k} \tag{17}$$

where $\Phi_{c_k}^T$ is the regressor vector, $\Phi_{c_k} = (x_k \otimes x_k)$, where $\otimes$ is the Kronecker product. Similarly, y_{ak} can be expressed in regressor form as

$$y_{ak} = \Phi_{a_k}^T W_{a_k} \tag{18}$$

where W_{a_k} is the matrix of actor weight estimates and $\Phi_{a_k}^T$ is the actor regressor vector. Signals δ_{TD_k} and r_k play vital role in determining the performance of the control policy by minimizing δ_{TD_k} defined in (16). The performance measure of the TLFM control is attributed to achieve the desired tip trajectory tracking while simultaneously damping out the tip deflection. Therefore, S_i defined in (8) which measures the accuracy of the tip trajectory tracking for i^{th} link is used to formulate the r_k and is given as

$$\begin{aligned} r_k &= 0, \quad \text{if } \left(S_i{}^2 \le \varepsilon\right); \text{else} \\ r_k &= -0.5 \end{aligned} \tag{19}$$

where ε is a predefined tolerance value and a reward (negative) is taken in (19) to improve the closed-loop performance. Substituting for $\Re_k(\zeta_k)$ from (17) in (15), one obtains

$$\delta_{\mathrm{TD}_k} = r_k + \gamma \left(\Phi_{c_{k+1}}^T W_{c_k}\right) - \Phi_{c_k}^T W_{c_k}. \tag{20}$$

B. Critic Weight (W_{ck}) Update Using the Proposed (RLS-ET-AM) Algorithm

The objective of the critic is to estimate $\Re_k(\zeta_k)$ using proposed (RLS-ET-AM) algorithm. Let $\hat{\Re}_k(\zeta_k)$ be the estimate of the value $\Re_k(\zeta_k)$ and a cost function J for N measurements is chosen so as to minimize the temporal difference error δ_{TD_k} defined in (15). J is given as

$$J = \sum_{k=1}^{N-1} \left[z_k \left(r_k + \gamma \hat{\Re}_k(\zeta_{k+1}) - \hat{\Re}_k(\zeta_k) \right) \right]^2 \tag{21}$$

where r_k is the reward function, $\hat{\Re}_k(\zeta_k)$ is the estimate value of the value function $\Re_k(\zeta_k)$ and $z_k(\zeta_k)$ is the eligibility trace used to improve the temporal difference learning by selecting the eligible state embedded in (ζ_k). We call this algorithm as RLS-ET algorithm. The eligibility trace is being defined as

$$z_k(\zeta_k) = \begin{cases} \gamma\lambda z_k(\zeta_k) + 1, & \text{if } \zeta_k = \theta_{di} \\ \gamma\lambda z_k(\zeta_k), & \text{if } \zeta_k \ne \theta_{di} \end{cases}$$

where γ is discount factor, λ is the value of the eligibility trace and θ_{di} is the desired tip trajectory for i^{th} link, it is to be noted that both the values of γ and λ are less than unity. Substituting the value of $\Re_k(\zeta_k)$ from (17) in (21) gives

$$J = \sum_{k=1}^{N-1} \left[z_k \left(r_k + \gamma \Phi_{c_{k+1}}^T \hat{W}_{c_{k+1}} - \Phi_{c_k}^T \hat{W}_{c_k} \right) \right]^2. \tag{22}$$

(22) can be modified in terms of predicted critic weights

$$\hat{W}_{c_{k+1}} \; as \; J = \sum_{k=1}^{N-1} \left[z_k \left(r_k + \hat{W}_{c_{k+1}} \left(\gamma \Phi_{c_{k+1}}^T - \Phi_{c_k}^T \right) \right) \right]^2. \tag{23}$$

The least square solution of (23) is given as

$$\hat{W}_{c_{k+1}} = \left(\sum_{k=1}^{N} \left(\Phi_{c_k} (\Phi_{c_k} - \gamma \Phi_{c_k})^T \right) \right)^{-1} \left(\sum_{k=1}^{N} (\Phi_{c_k} r_k z_k) \right). \tag{24}$$

A recursive form of the above equation with forgetting factor, μ can be obtained easily as follows:

$$W_{c_{k+1}} = W_{c_k} + G_{k+1} \left[r_k + \left(\Phi_{c_k}^T - \gamma \Phi_{c_{k+1}}^T \right) W_{c_k} \right] \tag{25}$$

with the Kalman gain G_{k+1} and covariance matrix P_{k+1} updatation are given as follows:

$$\begin{aligned} P_{k+1} = \frac{1}{\mu} \Bigg[P_k - P_k z_k \Big[\mu + \left(\Phi_{c_k}^T - \gamma \Phi_{c_{k+1}}^T \right) P_k z_k \Big]^{-1} \\ \cdot \left(\Phi_{c_k}^T - \gamma \Phi_{c_{k+1}}^T \right) P_k \Bigg] \end{aligned} \tag{26}$$

$$G_{k+1} = P_k z_k \left(\mu + \left(\Phi_{c_k}^T - \gamma \Phi_{c_{k+1}}^T \right) P_k z_k \right)^{-1}. \tag{27}$$

Equations (25)–(27) constitute the RLS-based TD learning with eligibility trace. Further, an incremental adaptive memory $\mathbf{a_m}$ can be added to the above RLS-ET algorithm to enhance the learning speed of the critic. The resulting weight updatataion

expressions with RLS-based TD learning with eligibility trace and an adaptive memory (RLS-ET-AM) are given in (28)–(32)

$$W_{c_{k+1}} = W_{c_k} + G_{k+1}\left[r_k + \left(\Phi_{c_k}^T - \gamma\,\Phi_{c_{k+1}}^T\right) W_{c_k}\right] \quad (28)$$

$$G_{k+1} = P_k z_k \mathbf{a}_{m_k}^{-1}\left(\mu + \left(\Phi_{c_k}^T - \gamma\,\Phi_{c_{k+1}}^T\right) P_k z_k \mathbf{a}_{m_k}^{-1}\right)^{-1} \quad (29)$$

$$P_{k+1} = \frac{1}{\mu}\left(\mathbf{a}_{m_k}^{-1}\left[P_k - P_k z_k\left[\mu + \left(\Phi_{c_k}^T - \gamma\,\Phi_{c_{k+1}}^T\right) P_k z_k\right]^{-1} \cdot \left(\Phi_{c_k}^T - \gamma\,\Phi_{c_{k+1}}^T\right) P_k\right]\right) \quad (30)$$

$$\mathbf{a}_{m_k}^{-1} = \mathbf{a}_{m_k} + k_r\left[\hat{\psi}_k^T \Phi_{c_k} \delta_{\mathrm{TD}_k}\right]. \quad (31)$$

The gradient vector $\hat{\psi}_k$ is updated using following expression:

$$\hat{\psi}_{k+1} = \left[I - G_{k+1}\gamma\Phi^T{}_{c_{k+1}}\right]\hat{\psi}_k + S_{k+1}\Phi_{c_{k+1}}\delta_{\mathrm{TD}_k} \quad (32)$$

where I is the identity matrix and

$$S_{k+1} = \mathbf{a}_{m_{k+1}}^{-1}\left[I - G_{k+1}\gamma\Phi^T{}_{c_{k+1}}\right] S_k\left[I - \Phi_{c_k} G_{k+1}\right] + \cdots \mathbf{a}_{m_{k+1}}^{-1}\left[G_{k+1} G^T{}_k P_{k+1}\right]. \quad (33)$$

C. Actor Weight (W_{ak}) Update Using Gradient Based Estimator and the Proposed RLS-ET-AM Algorithm

The actor weight vector W_{c_k} can be updated using gradient-based estimator as described below. The control policy y_{ak} can be written in parametric form as

$$y_{a_k} = \Phi_{a_k}^T W_{a_k} \quad (34)$$

and $\hat{y}_{a_k} = \Phi_{a_k}^T \hat{W}_{a_k}$ be its estimate. Then, the control policy estimation error can be written as

$$\tilde{y}_{a_k} = y_{a_k} - \hat{y}_{a_k} = \Phi_{a_k}^T\left(W_{a_k} - \hat{W}_{a_k}\right) \quad (35)$$

$$\hat{W}_{a_{k+1}} = \hat{W}_{a_k} - K_a \Phi_{a_k}^T\left(\hat{\Re}_k(\zeta_k) - \tilde{y}_{a_k}\right). \quad (36)$$

The control policy y_{ak} can also be rewritten in terms of the critic parameter W_{c_k} as follows:

$$\hat{W}_{a_{k+1}} = \hat{W}_{a_k} - K_a \Phi_{a_k}^T\left(\Phi_{c_k}^T \hat{W}_{c_k} - \hat{y}_{a_k}\right) \quad (37)$$

$0 < K_a \leq 1$ is the adaptation gain.

By measuring the external reinforcement signal δ_{TD_k} and internal reinforcement signal r_k, the critic as well as actor weights are updated. The learning terminates as soon as approximation error tends towards zero. The proposed RLS-ET-AM algorithm is shown in Table I.

D. Convergence Analysis of the Critic Weights W_{c_k} Using the Proposed RLS-ET-AM Algorithm

The existing RLS_TD learning algorithm [8] is modified by adding an incremental adaptive memory to RLS-based linear function approximator with offline calculated critic weights. In order to prove the convergence of above RLS-ET-AM algorithm, certain assumptions are used. These are as follows.

TABLE I
PROPOSED RLS-ET-AM ALGORITHM

Step 1:	for k=0 **begin {** Define the performance index as given in (26)
Step 2:	Initialize initial values of Wc_k, P_k, S_k, a_m, μ , K_a, γ, z_k
Step 3:	Observe the transition states of y_{pi}, y_{di}, r_{k+1} and δ_{TD}
Step 4:	Apply equations (28)-(32) to update critic weights
Step 5:	Apply equations (35)-(37) to update actor weights
Step 6:	Check the termination criteria from step: 1 update k=k+1 till criteria is satisfied **} end**

Assumption 1: The discrete event of states $\{\zeta_k\}$, with transition probability matrix $\mathbf{P}$, and distribution χ satisfy

$$\chi^T P = \chi^T. \quad (38)$$

Assumption 2: The transition reward $r(\zeta_k, \zeta_{k+1})$ satisfies

$$E_0\left[r^2(\zeta_k, \zeta_{k+1})\right] < \infty \quad (39)$$

where $E_0[\,]$ is the expectation with respect to distribution χ.

Assumption 3: The matrix Φ_{ck}^T is linearly independent.

Assumption 4: For every "k," the function Φ_{ck}^T satisfies

$$E_0\left[\Phi^2{}_{ck}(\zeta_k)\right] < \infty. \quad (40)$$

Assumption 5: $\left[P_k{}^{-1} + (1/k)\sum_{k=1}^{K}\Omega(\zeta_k)\right]$ is nonsingular $\forall\ k > 0$

Theorem 1: Considering the above assumptions (1–5) and using the proposed RLS-ET-AM algorithm given in (28)–(32), the critic weights Wc_k converge to Wc_k^* (optimal critic weights).

Proof: Applying matrix inversion Lemma $(A+BC)^{-1} = A^{-1} - A^{-1}B(I+C\,A^{-1}B)^{-1}C\,A^{-1}$ to (30), it can be rewritten as

$$P_{k+1} = \mathbf{a}_{m_k}^{-1}\left[P_k{}^{-1} + z_k\left(\Phi_{c_k}^T - \gamma\Phi_{c_{k+1}}^T\right)\right]^{-1} \quad (41)$$

where $A^{-1} = P_k$; $B = z_k$, $C = \left(\Phi_{c_k}^T - \gamma\Phi_{c_{k+1}}^T\right)$ and assuming $\mu = 1$.

G_k given in (29) is multiplied by P_k^{-1} giving

$$P_k{}^{-1} G_{k+1} = P_k{}^{-1} P_k z_k \mathbf{a}_{m_k}^{-1}\left(1 + \left(\Phi_{c_k}^T - \gamma\,\Phi_{c_{k+1}}^T\right) P_k z_k\right)^{-1} \quad (42)$$

$$G_{k+1} = z_k \mathbf{a}_{m_k}^{-1}\left(P_k{}^{-1} + \left(\Phi_{c_k}^T - \gamma\,\Phi_{c_{k+1}}^T\right) z_k\right)^{-1}. \quad (43)$$

(43) can be rewritten using the expression for P_{k+1} from (41) as

$$G_{k+1} = P_{k+1} z_k \mathbf{a}_{m_k}^{-1}. \quad (44)$$

Using the results obtained for updatation of covariance matrix of the TD error P_{k+1} in (41) and $G_{k+1}\,G_k$ in (44), the updatation of the critic weights Wc_{k+1} defined in (28) can be rewritten as

$$\begin{aligned} &W_{c_{k+1}} \\ &= W_{c_k} + P_{k+1} z_k \mathbf{a}_{m_k}^{-1} \left[r_k + \left(\Phi_{c_k}^T - \gamma\, \Phi_{c_{k+1}}^T \right) W_{c_k} \right] \\ &= W_{c_k} + P_{k+1} \left[z_k r_k \mathbf{a}_{m_k}^{-1} + \left(\Phi_{c_k}^T - \gamma\, \Phi_{c_{k+1}}^T \right) W_{c_k} z_k \mathbf{a}_{m_k}^{-1} \right] \\ &= P_{k+1} \left[\left({P_{k+1}}^{-1} - z_k \left(\Phi_{c_k}^T - \gamma \Phi_{c_{k+1}}^T \right) \right) W_{c_k} + z_k r_k \mathbf{a}_{m_k}^{-1} \right]. \end{aligned} \tag{45}$$

Substituting for P_{k+1} from (41) in (45) gives

$$\begin{aligned} &= P_{k+1} \left(\left[{P_k}^{-1} + z_k \left(\Phi_{c_k}^T - \gamma \Phi_{c_{k+1}}^T \right) \right] \right. \\ &\quad \left. - z_k \left(\Phi_{c_k}^T - \gamma \Phi_{c_{k+1}}^T \right) \right) W_{c_k} + z_k r_k \mathbf{a}_{m_k}^{-1} \\ W_{c_{k+1}} &= P_{k+1} \left({P_k}^{-1} W_{c_k} + z_k r_k \mathbf{a}_{m_k}^{-1} \right). \end{aligned} \tag{46}$$

Using (41) in (46) gives

$$\begin{aligned} &W_{c_{k+1}} \\ &= \left[{P_k}^{-1} + z_k \left(\Phi_{c_k}^T - \gamma\, \Phi_{c_{k+1}}^T \right) \right]^{-1} \left({P_k}^{-1} W_{c_k} + z_k r_k \mathbf{a}_{m_k}^{-1} \right) \end{aligned} \tag{47}$$

denoting $\Omega(\zeta_k) = z_k \left(\Phi_{c_k}^T - \gamma\, \Phi_{c_{k+1}}^T \right)$, $p_k = z_k r_k \mathbf{a}_{m_k}^{-1}$ in (47) and Wc_{k+1} as $W_{\mathrm{RLS-ET-AM}}$ one obtains

$$\begin{aligned} &W_{\mathrm{RLS-ET-AM}} \\ &= \left[{P_k}^{-1} + \sum_{k=1}^{K} \Omega(\zeta_k) \right]^{-1} \left[{P_k}^{-1} W_{ck} + \sum_{k=1}^{K} p_k \right] \\ &= \left[\frac{1}{N} {P_k}^{-1} + \frac{1}{N} \sum_{k=1}^{N} \Omega(\zeta_k) \right]^{-1} \left[\frac{1}{N} {P_k}^{-1} W_{c_k} + \frac{1}{N} \sum_{k=1}^{N} p_k \right] \end{aligned} \tag{48}$$

since

$$E_0\left[\Omega(\zeta_k)\right] = \lim_{k\to\infty} \frac{1}{N} \sum_{k=1}^{N} \Omega(\zeta_k) \tag{49}$$

$$E_0\left[p_k\right] = \lim_{k\to\infty} \frac{1}{K} \sum_{k=1}^{N} p_k \tag{50}$$

where N denotes the number of measurements and from assumptions (1–5), it is known that $E_0\left[\Omega(\zeta_k)\right]$ is invertible, i.e.,

$$\left(E_0\left[\Omega(\zeta_k)\right]\right) \lim_{k\to\infty} W_{\mathrm{RLS-ET-AM}} = E_0 p_k = W^*. \tag{51}$$

Thus, from (51) it is clear that $W_{\mathrm{RLS-ET-AM}}$ converges to W^* (optimal critic weights).

V. Results and Discussions

A. Simulation Results

The numerical simulation of the DAC, FLAC, and RLAC has been performed using MATLAB/SIMULINK®. The DAC, FLAC, and the proposed RLAC have been applied to the TLFM available in the Advanced Robotics Research Laboratory, National Institute of Technology (NIT), Rourkela. The physical parameters of the studied TLFM are given in Table II and the controller parameters for RLAC, FLAC, and DAC are given in Table III.

TABLE II
PHYSICAL PARAMETERS OF THE TLFM.

Parameter	Link-1	Link-2
Link length	0.201m	0.2m
Elasticity	2.0684×10^{11}(N/m^2)	2.0684×10^{11}(N/m^2)
Rotor moment of Inertia	6.28×10^{-6}(kg m^2)	1.03×10^{-6}(kg m^2)
Drive moment of Inertia	7.361×10^{-4}(kg m^2)	44.55×10^{-6} (kg m^2)
Link moment of Inertia	0.17043 (kg m^2)	0.0064387(kg m^2)
Gear ratio	100	50
Maximum Rotation	(+/-90, +/-90)deg.	(+/-90, +/-90)deg.
Drive Torque constant	0.119(Nm/A)	0.0234(Nm/A)

TABLE III
CONTROLLER PARAMETERS FOR TLFM

Type of Controller	Controller Parameters
DAC	Γ=diag([20, 0.2, 0.5, 0.2, 20, 50, 10]), Λ= 0.5, K_{D1}=20 and K_{D2}=15
FLAC	g11=0.5, g12= 1.25, g21=0.75, g22= 2.25 (Scaling gains for TLFM) g11=0.5, g12= 1.25, g21=0.75, g22= 2.25 (Scaling gains for Learning Mechanism).
RLAC	ω_{n1}= 3.15rad/sec, ω_{n2}= 10.054rad/sec I_{eq1}= 0.17043Kg/m^2, I_{eq2}= 0.0064Kg/m^2 K_{p1}=1.75, K_{p2}=0.65, K_{d1}=1.25, K_{d2}=0.15 P_0=1000×I$^{4\times4}$, α=0.5, z=0.25, γ=0.98

To validate the tip trajectory tracking performances, the desired trajectory vector for two joints, $\theta_{di}(t) i = 1{,}2$ are chosen as where $\theta_{di}(t) = [\theta_{d1}, \theta_{d2}]^T$, $\theta_d(0) = \{0, 0\}$ are the initial positions of the links and $\theta_f(0) = \{\pi/4, \pi/6\}$ are the final positions for link-1 and link-2, t_d is the time taken to reach the final positions which is taken 4 s and total simulation time is set as 10 s

$$\theta_{d_i}(t) = \theta_0(t) + \left[6\frac{t^5}{{t_d}^5} - 15\frac{t^4}{{t_d}^4} + 10\frac{t^3}{{t_d}^3} \right] \left(\theta_f(t) - \theta_0(t) \right). \tag{52}$$

The universe of discourses for FLAC for link-1and link-2 tip position errors were chosen as $[-\pi/2, \pi/2]$ rad, tip position change in error was chosen to be $[-2, 2]$, respectively. The control torque universe of discourse is $[-5, 5]$ Nm was chosen to keep the control input within reasonable limits The fuzzy rule base was taken from for the TLFM is a 9×9 array. The reference model is taken as $5/(s+5)$ for both the links . Gains of the discrete PD controller for the RLAC were determined by assuming the manipulator's links to be rigid, i.e., for $d_i(l_i, t) = 0$. The gains were obtained from closed-loop error equation (53) knowing the values of ω_{ni}:

$$I_{\mathrm{eq}_i} \ddot{e}_{i_k} + K_{d_i} \dot{e}_{i_k} + K_{p_i} e_{i_k} = 0, i = 1, 2 \tag{53}$$

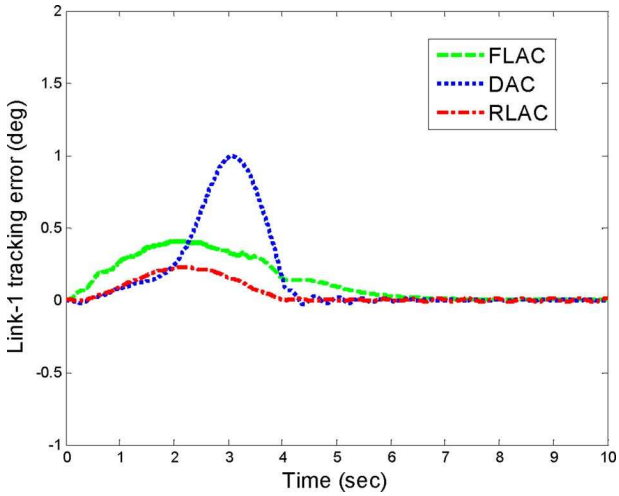

Fig. 6. Tip trajectory tracking errors (link-1).

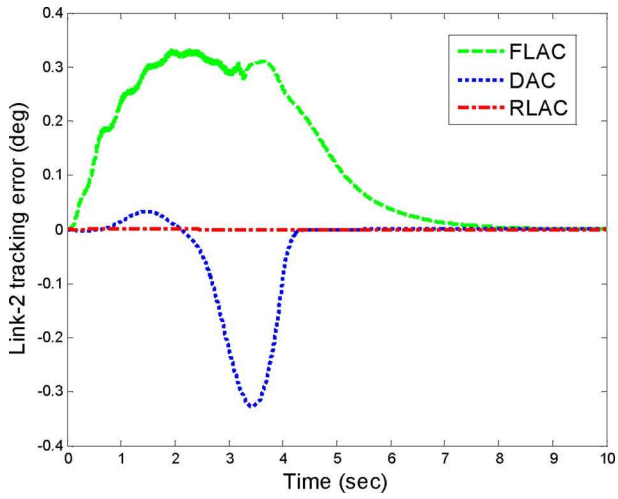

Fig. 7. Tip trajectory tracking errors (link-2).

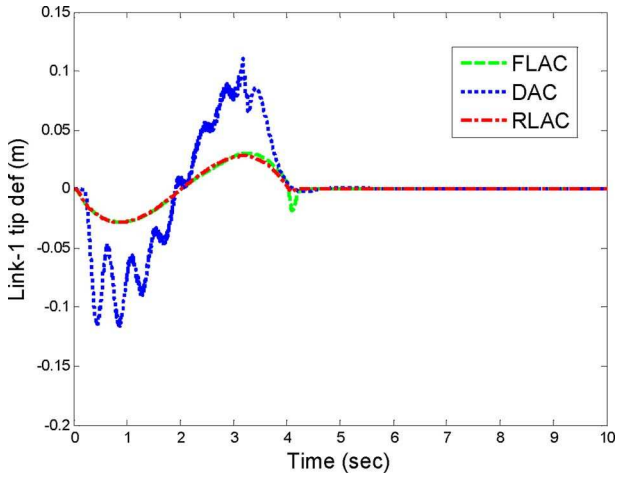

Fig. 8. Comparison of link-1 tip deflection.

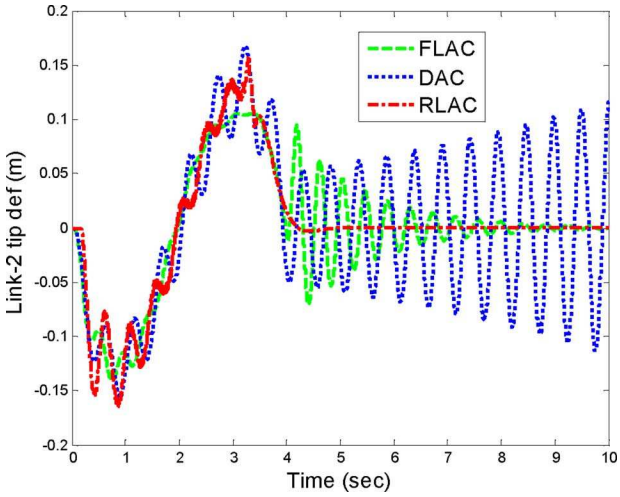

Fig. 9. Comparison of link-2 tip deflection.

where I_{eqi} denotes the equivalent inertia of the i^{th} joint. From (53), assuming critical damping, K_{pi} and K_{di} can be determined as

$$K_{pi} = I_{\text{eq}_i}\omega_{ni}^2;\ K_{di} = 2I_{\text{eq}_i}\omega_{ni} \tag{54}$$

where ω_{ni} is the i^{th} link's natural frequency

1) Simulation Results for an Initial Payload of 0.157 Kg: Figs. 6 and 7 show the tip trajectory tracking error curves for link-1 and link-2, respectively. From Fig. 6, for link-1, it is seen that there exists a tracking error of 0.4° in case of the FLAC and 1° in case of DAC. However, the tracking error by the RLAC is almost zero. Link-2 tracking error profiles in Fig. 7 reveal that the tracking errors are 0.45° for both DAC and FLAC, whereas it is almost zero in case of the RLAC. Thus, RLAC provides excellent tracking performance.Figs. 8 and 9 show the tip deflection trajectories for link-1 and link-2 carrying 0.157 kg of payload. From these figures, it is seen that the RLAC suppresses the tip deflection faster compared to the DAC and FLAC by damping it within 4 s. Figs. 10 and 11 show the control torque profiles generated by DAC, FLAC and RLAC for joint-1 and joint-2, respectively. From Figs. 10 and 11, it seen that the control input generated by the RLAC becomes zero compared to DAC and FLAC for link-1 and link-2 when the desired tip position is tracked. Thus, RLAC needs less control excitation for handling a payload of 0.157 kg compared to DAC and FLAC.

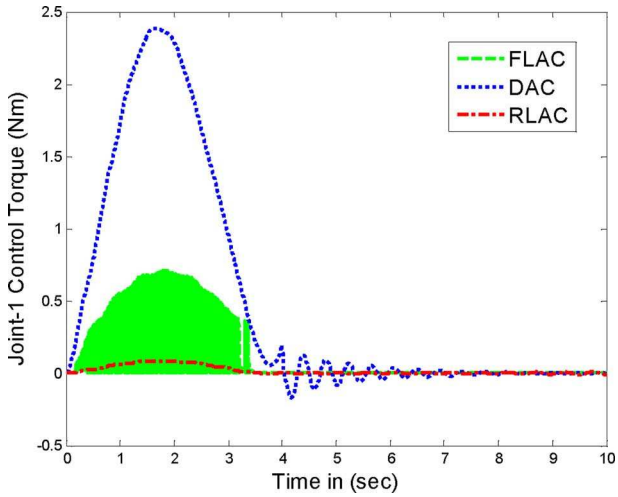

Fig. 10. Torque profiles (joint-1).

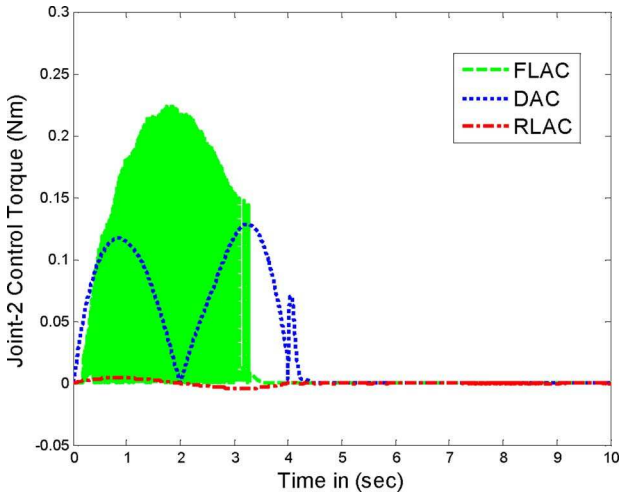

Fig. 11. Torque profiles (joint-2).

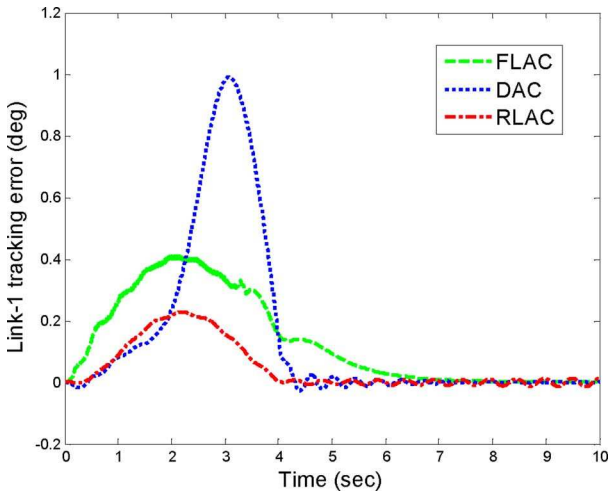

Fig. 12. Tip trajectory tracking errors (Link-1).

2) Simulation Results for an Additional Payload of 0.3 Kg: An additional payload of 0.3 kg is now attached to the existing initial payload of 0.157 kg making the overall payload 0.457 kg. Performances of three controllers RLAC, FLAC and DAC for 0.457 kg payload were compared in Figs. 12–18. Figs. 12 and 13 depict the tip trajectory tracking performance for link-1 and link-2 From Fig. 12, it can be seen that the time evolution of the error trajectory achieved by employing DAC has yielded maximum overshoot compared to the FLAC and RLAC. Fig. 13 shows that FLAC has yielded maximum overshoot compared to the DAC and RLAC controllers. Suppressing the tip deflection performances of RLAC, FLAC and DAC were compared in Figs. 14 and 15 for link-1 and link-2, respectively. From Fig. 14, it is seen that tip deflection is maximum in case of DAC compared to FLAC and also RLAC when a payload of 0.457 kg is attached for link-1. From Fig. 15, it is seen that the tip deflection trajectories for link-2 is more oscillatory when carrying 0.457 kg of payload in case of DAC compared to FLAC and RLAC.

Joint torque signals generated from DAC, FLAC and RLAC are compared in Figs. 16 and 17. The adaptation of the actor and critic weights for RLAC carrying payload of 0.457 kg using simulation model is shown in Fig. 18.The results show that as the learning progresses, the updated critic weights converge to their optimal values.

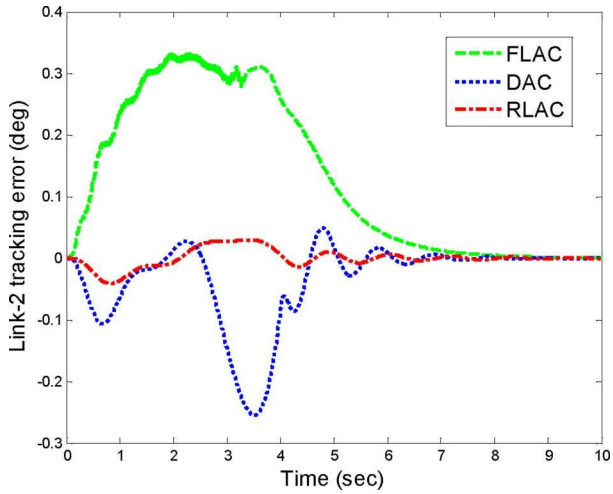

Fig. 13. Tip trajectory tracking errors (Link-2).

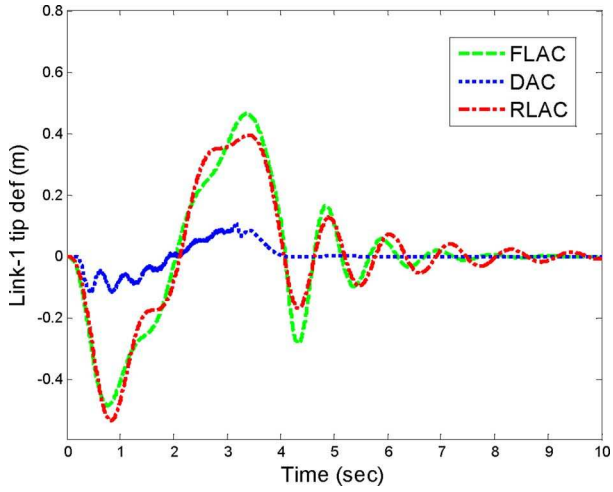

Fig. 14. Comparison of link-1 tip deflection.

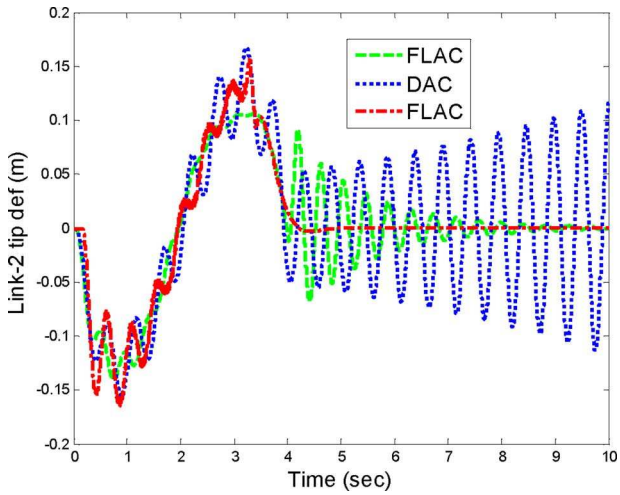

Fig. 15. Comparison of link-2 tip deflection.

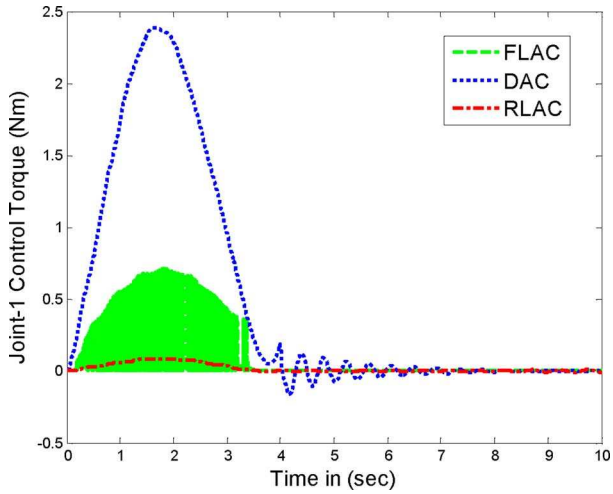

Fig. 16. Torque profiles (joint-1).

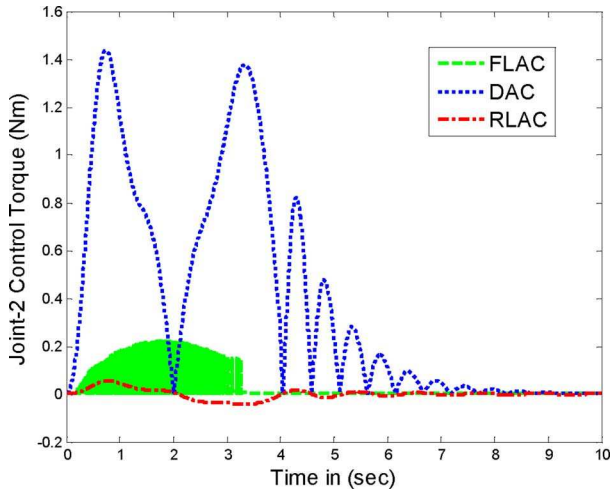

Fig. 17. Torque profiles (joint-2).

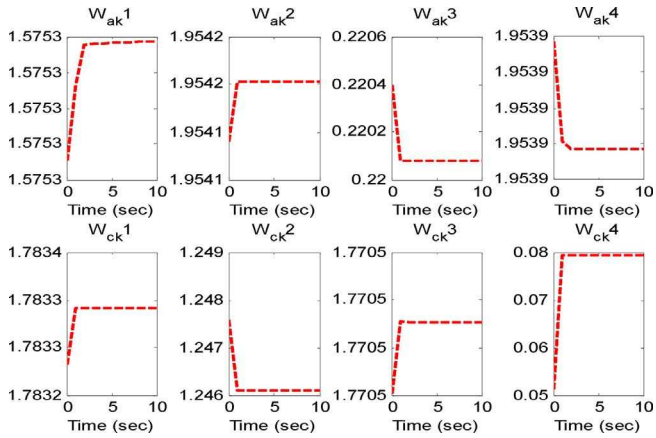

Fig. 18. Simulation results for adaptation of the actor and critic weights to optimal values.

B. Experimental Setup

The experimental setup used to implement the proposed RLAC and the other two for comparison is shown in Fig. 19. The setup has two links and two joints and an end effecter to carry the variable payload. These two joints are excited by two

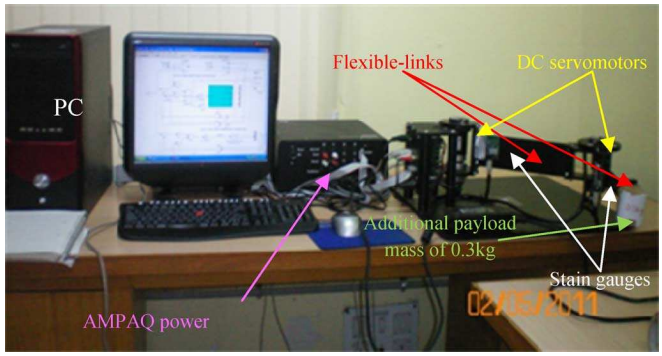

Fig. 19. Photograph of the experimental setup.

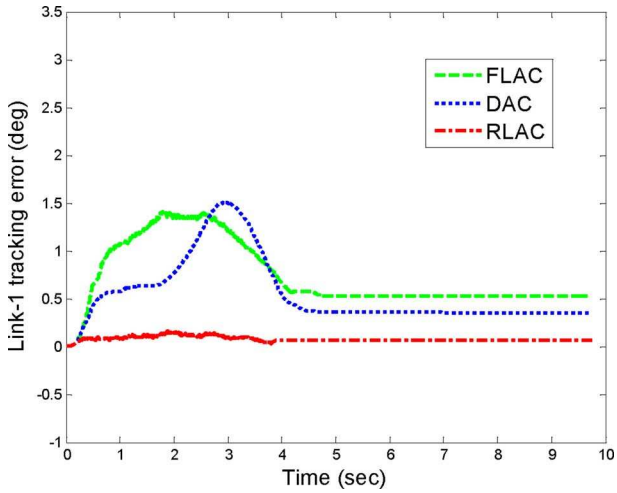

Fig. 20. Tip trajectory tracking errors (link-1).

DC servo motors powered at 42 watts with $\pm 15\ V$ DC power supply. The amplifier signal is provided by a two channel amplifier package (AMPAQ) power module shown in Fig. 19. The drives for link-1 and link-2 offer zero backlashes with gear ratios of 100:1 and 50:1, respectively. Digital encoders are used to measure the angular speed and position and two stain gauges for each link are used to measure the deflection in links. MATLAB real-time tool box issued to generate the target-logic C-code. CORE(TM) 2 Duo processor is used using real-time win target to run the compiled code in real-time.

Analog to digital and digital to analog signal are processed using an in built hardware-in-the-loop (HIL) board. All the three controllers (RLAC, FLAC, and DAC) have been implemented using MATLAB/SIMULINK®

C. Experimental Results

1) Experimental Results for an Initial Payload of 0.157 Kg: Figs. 20–25 show comparison of the experimental results for TLFM obtained by employing RLAC, FLAC, and DAC with an initial payload of 0.157 kg. Figs. 20 and 21 show the comparison of the tip trajectory tracking, after 4 s when the tip attains the final position, the steady-sate error is almost zero in case of RLAC for link-1 and link-2, whereas the DAC and FLAC yield steady-state errors of 0.1 and 0.2 for link-1 and link-2, respectively, after 4 s. Figs. 22 and 23 show the tip deflection trajectories for the link-1 and link-2 when loaded for a 0.457 kg payload. From Fig. 22, it can be seen that RLAC yields 0.1 m

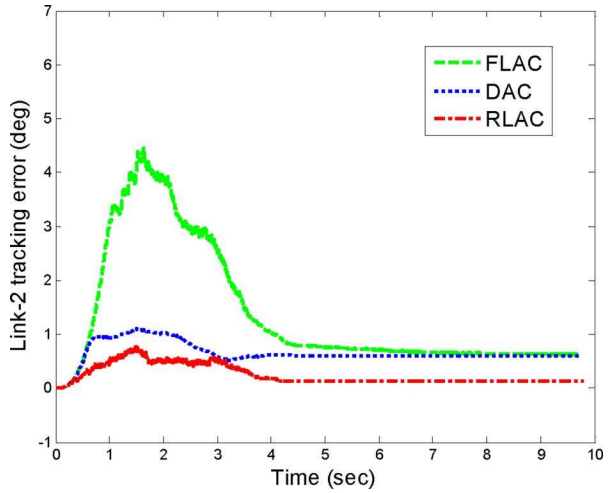

Fig. 21. Tip trajectory tracking errors (link-2).

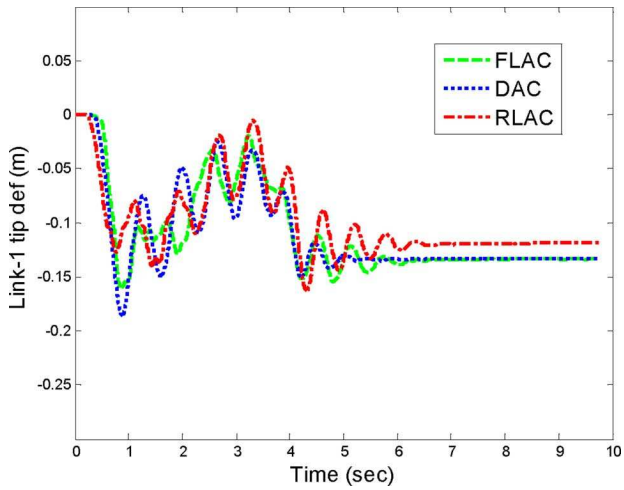

Fig. 22. Comparison of link-1 tip deflection.

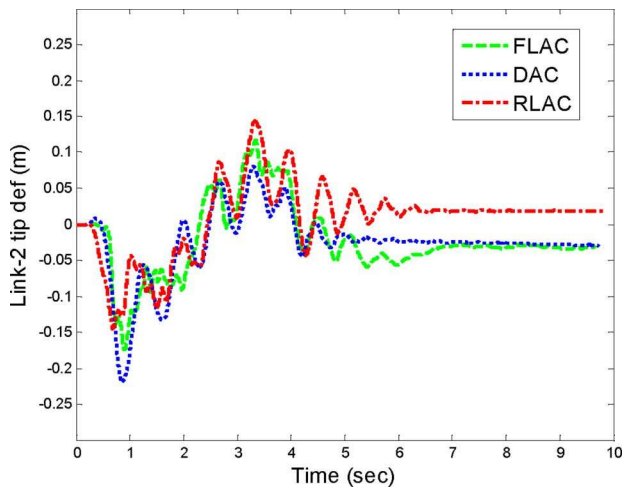

Fig. 23. Comparison of link-2 tip deflection.

of initial deviation as compared to FLAC and DAC where the deflection are 0.16 and 0.18 m for link-1. Link-2 tip deflection characteristics are shown in Fig. 23, from which it is seen that RLAC has 0.15 m of initial deviation as compared to who have

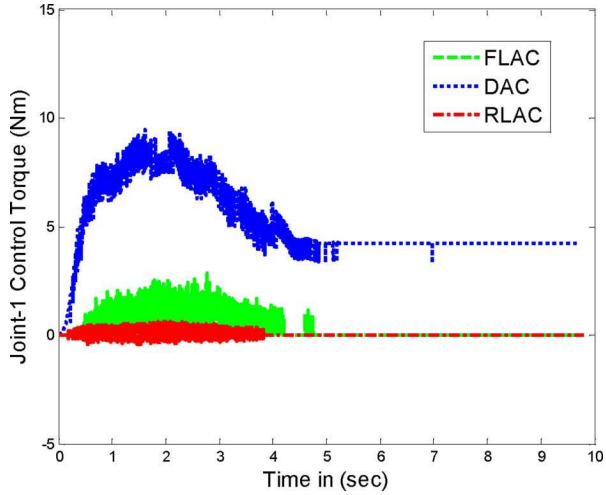

Fig. 24. Torque profiles (joint-1).

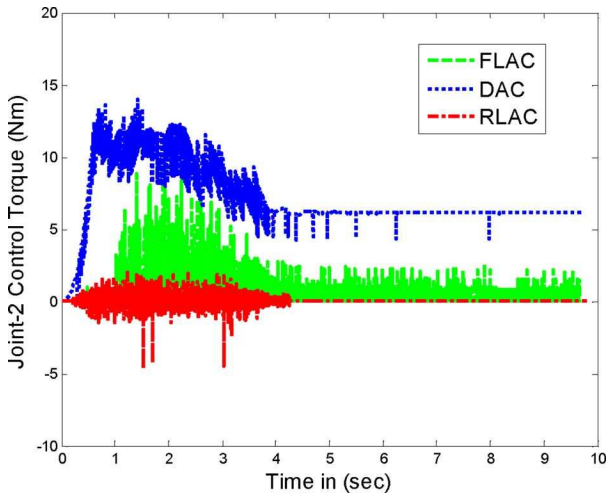

Fig. 25. Torque profiles (joint-2).

0.18 and 0.22 m of initial deviation, respectively, for FLAC and DAC.

Torque profiles for joint-1 generated by employing the three controllers are shown in Fig. 24, and that for joint-2 is shown in Fig. 25. The joint torque control input for link-1 obtained by DAC reaches to a maximum value (9 Nm) at 2 s when the tip reaches to the final position at 4 s the control input reduces to 5 Nm. In case of FLAC where control input reaches to a maximum value (2 Nm) at 2 s and 0.5 Nm for RLAC and torque becomes zero when the tip reaches the final position at 4 s. From Figs. 24 and 25, the joint control torque signals generated by DAC, FLAC and RLAC for link-2 have maximum of 12, 10, and 2.5 Nm, respectively.

2) Experimental Results for an Additional Payload of 0.3 Kg: An additional payload of 0.3 kg is added to the initial payload of 0.157 kg. Figs. 26–32 show comparison of the experimental results for TLFM obtained by employing RLAC, FLAC, and DAC with a payload of 0.457 kg. Figs. 26 and 27 compare the tip trajectory tracking performances for link-1 and link-2, respectively.

From Figs. 26 and 27, it is clear that when the final position is attained, the steady-state error in case of RLAC is almost zero,

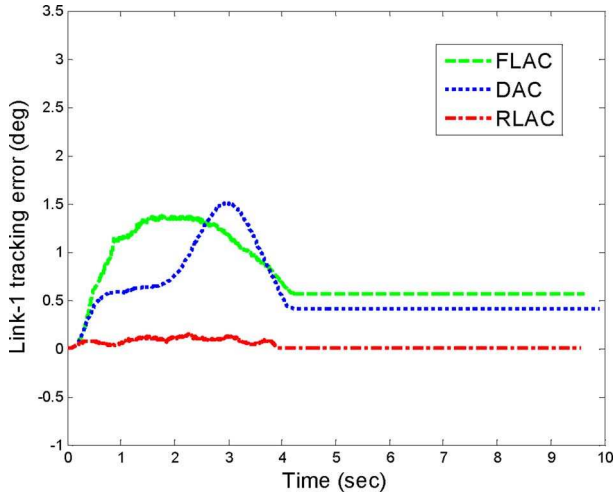

Fig. 26. Tip trajectory tracking errors (link-1).

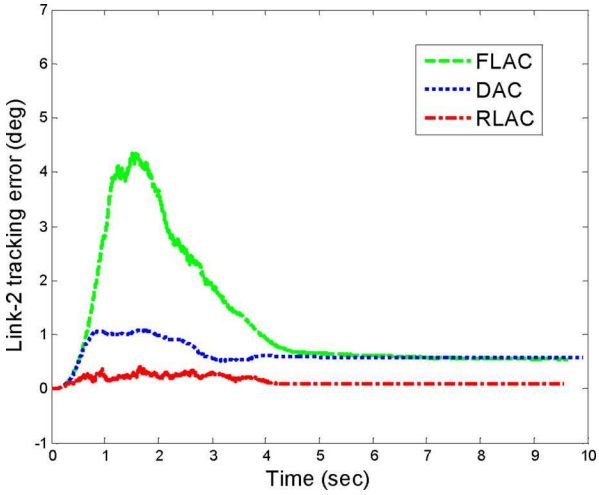

Fig. 27. Tip trajectory tracking errors (link-2).

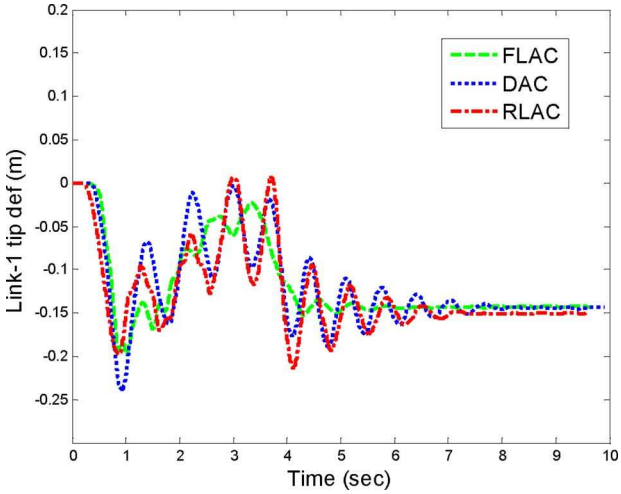

Fig. 28. Comparison of link-1 tip deflection.

while a finite steady-state error exists in case of both DAC and FLAC. The TLFM is an infinite dimensional system due to distributed link flexure. Higher modes have been neglected in modeling therefore there is a difference in steady-state error for simulation and experimental results. Figs. 28 and 29 show the tip

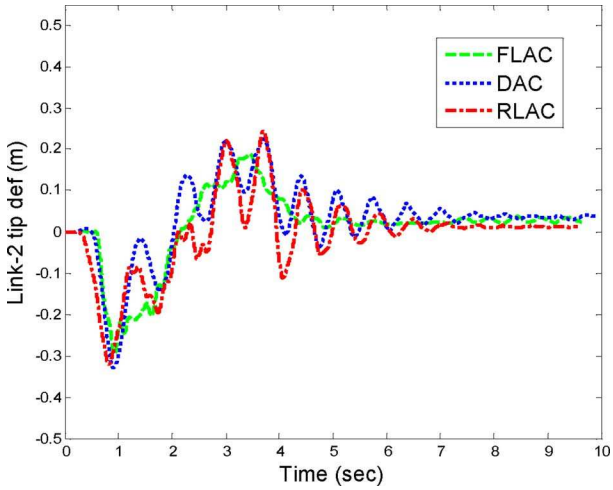

Fig. 29. Comparison of link-2 tip deflection.

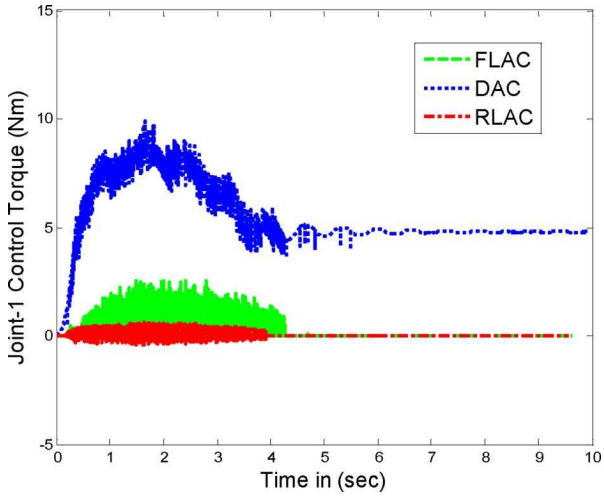

Fig. 30. Torque profiles (joint-1).

deflection trajectories for the link-1 and link-2 when asked for a payload of 0.457 kg. From Fig. 28, it is seen in case of RLAC there exists an initial deviation of 0.2 m as compared to FLAC and DAC in which deflections are 0.2 and 0.25 m, respectively, for link-1. Link-2 tip deflection responses are shown in Fig. 29. RLAC has 0.3 m of initial deviation as compared to FLAC and DAC where initial deviations are 0.28 and 0.32 m, respectively.

Torque profile generated for joint-1 by the three controllers is shown in Fig. 30. From this figure, it is seen that the DAC torque signal reaches to a maximum value of 9.5 Nm and reduces to 5 Nm at 4 s when the final position is tracked. FLAC torque signal becomes the maximum (2.5 Nm) at 2 s and almost reduces to zero when the final position is tracked.

FLAC torque signal reaches to maximum value of 9 Nm at 1.5 s and reduces to 2 Nm at 4 s, whereas RLAC generates appropriate control torques with zero at the final position.

But RLAC generates control torque signal with less amplitude initially and zero magnitude, while the desired position has been tracked. From Fig. 31, torque profile generated for joint-2, it is seen that the DAC torque signal reaches to maximum value of 15 Nm at 1 s and reduces to 6 Nm at 4 s maximum value of 2 Nm at 1.5 s with almost zero value at the final position. The

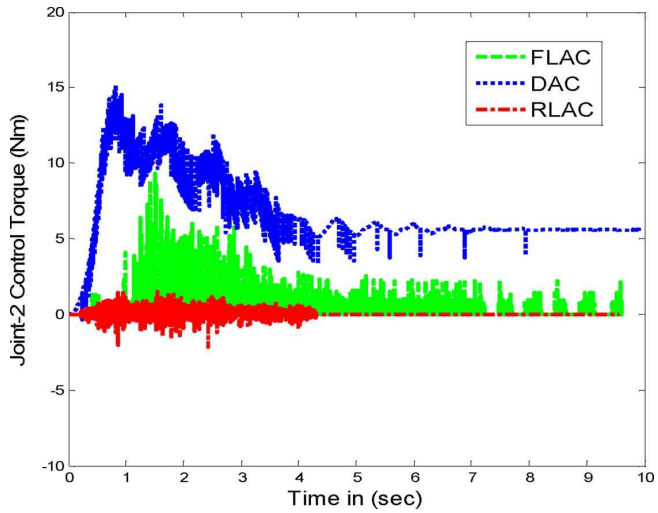

Fig. 31. Torque profiles (joint-2).

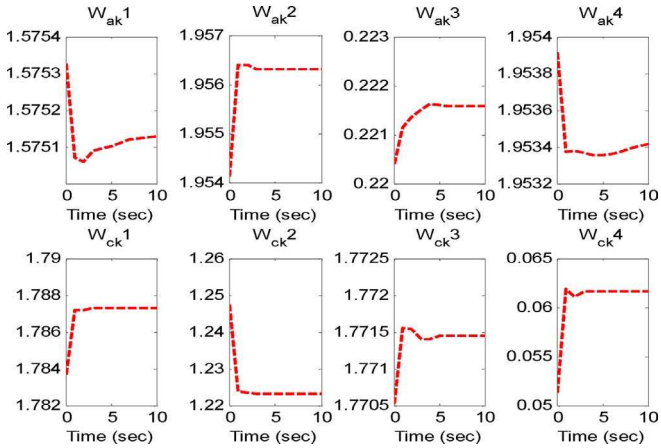

Fig. 32. Experimental results for adaptation of the actor and critic weights to optimal values.

experimental values while updating the actor and critic weights under payload of 0.457 kg are shown in Fig. 32. The results show that despite changes in payload, the critic weights converge to their optimal values. However, there is difference in critic weights in experiment and simulations. This is because of approximations in modeling of the TLFM.

VI. Conclusion

This paper has proposed a new real-time adaptive controller for tracking control of tip trajectory and suppressing tip deflection for a TLFM, while subjected to handle variable payload-based on RL technique. The proposed RLAC provides better tracking and tip deflection damping performance compared to both a nonlinear DAC and a FLAC. The reason for superiority in performance exhibited by RLAC is due to the integration of optimality with the adaptivity, i.e., its ability to adapt the actor and critic weights to their optimal values using Recursive Least Square-Eligibility Trace-Adaptive Memory feature embedded. However, DAC and FLAC are only adaptive controllers. The proposed RLAC has been applied successfully to a laboratory flexible robot setup. The RLAC has exhibited excellent performance in real-time control of this manipulator which has distributed flexure along it links. Thus, it is expected that the RLAC will be useful in similar control applications such as in space manipulators.

Acknowledgment

The authors thank the editor and reviewers for their valued comments in improving the quality of this paper.

References

[1] M. O. Tokhi and A. K. M. Azad, *Flexible Robot Manipulators: Modeling, Simulation and Control*. London, U.K.: IET, 2008.
[2] V. Feliu, K. S. Rattan, and B. H. Brown, "Adaptive control of a single-link flexible manipulator," *IEEE Contr. Syst. Mag.*, vol. 10, no. 2, pp. 29–33, 1990.
[3] M. R. Rokui and K. Khorsani, "Experimental results on discrete time nonlinear adaptive tracking control of a flexible-link manipulator," *IEEE Trans. Syst. Man, Cybern.*, vol. 30, no. 1, pp. 151–164, 2000.
[4] V. G. Moudgal, W. A. Kwong, K. M. Passino, and S. Yurkovich, "Fuzzy learning control for a flexible-link robot," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 2, pp. 199–210, May 1995.
[5] L. B. Gutierrez, F. L. Lewis, and J. A. Lowe, "Implementation of a neural network tracking control for a single flexible link: Comparison with PD and PID controllers," *IEEE Trans. Ind. Electron.*, vol. 45, no. 2, pp. 307–318, Apr. 1998.
[6] B. Subudhi and A. S. Morris, "Soft computing methods applied to the control of a flexible robot manipulator," *Appl. Soft Comput.*, vol. 9, no. 1, pp. 149–158, 2009.
[7] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.
[8] S. J. Bradtke and A. G. Barto, "Linear least-squares algorithms for temporal difference learning," *Mach. Learn.*, vol. 22, pp. 33–57, 1996.
[9] L. A. Prashanth and S. Bhatnagar, "Reinforcement learning with function approximation for traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 412–421, Jun. 2011.
[10] Q. Yang and S. Jagannathan, "Reinforcement learning control for affine nonlinear discrete-time systems using online approximators," *IEEE Trans. Syst. Man, Cybern.-Part B: Cybern.*, pp. 1–14, 2011.
[11] A. de Luca and B. Siciliano, "Closed-form dynamic model of planar multilink lightweight robots," *IEEE Trans. Syst. Man, Cybern.*, vol. 21, no. 4, pp. 826–839, 1991.
[12] J. J. E. Slotine and L. Weiping, "Adaptive manipulator control: A case study," *IEEE Trans. Automat. Control*, vol. 33, no. 11, pp. 995–1003, Nov. 1988.

Santanu Kumar Pradhan received the B.E. degree in electrical and electronics engineering from Biju Patnaik University of Technology, Rourkela, India, in 2006 and the Master of Technology degree in energy systems from the Indian Institute of Technology (IIT), Roorkee, in 2009. He is currently working towards the Ph.D. degree in electrical engineering at the National Institute of Technology (NIT), Rourkela.

His research interests include adaptive control of flexible robots.

Bidyadhar Subudhi (M'94–SM'08) received B.S. degree in electrical engineering from the National Institute of Technology (NIT), Rourkela, India, the Master of Technology in control and instrumentation from the Indian Institute of Technology (IIT), Delhi, in 1994, and the Ph.D. degree in control system engineering from the University of Sheffield, Sheffield, U.K., in 2003.

Currently, he is working as a Professor and Head of the Electrical Engineering Department, NIT. His research interests include adaptive control, robotics, and industrial electronics.