

Adaptive Critic Based Optimal Kinematic Control for a Robot Manipulator

Aiswarya Menon¹, Ravi Prakash², Laxmidhar Behera³, *Senior Member, IEEE*

Abstract—This paper is concerned with the optimal kinematic control of a robot manipulator where the robot end effector position follows a task space trajectory. The joints are actuated with the desired velocity profile to achieve this task. This problem has been solved using a single network adaptive critic (SNAC) by expressing the forward kinematics as input affine system. Usually in SNAC, the critic weights are updated using back propagation algorithm while little attention is given to convergence to the optimal cost. In this paper, we propose a critic weight update law that ensures convergence to the desired optimal cost while guaranteeing the stability of the closed loop kinematic control. In kinematic control, the robot is required to reach a specific target position. This has been solved as an optimal regulation problem in the context of SNAC based kinematic control. When the robot is required to follow a time varying task space trajectory, then the kinematic control has been framed as an optimal tracking problem. For tracking, an augmented system consisting of tracking error and reference trajectory is constructed and the optimal control policy is derived using SNAC framework. The stability and performance of the system under the proposed novel weight tuning law is guaranteed using Lyapunov approach. The proposed kinematic control scheme has been validated in simulations and experimentally executed using a real six degrees of freedom (DOF) Universal Robot (UR) 10 manipulator.

I. INTRODUCTION

Modern robotic systems are becoming complex as the degrees of freedom are increasing to address the complex application scenario such as agriculture, health-care and ware-house automation. A mobile manipulator with Shadow hand mounted on it has 28 degrees of freedom. The inverse kinematic solution to such systems can not be solved analytically. Thus neural and fuzzy neural network based schemes have become popular to design kinematic control [1], [2], [3], [4]. These approaches can not learn the inverse kinematics while optimizing a global cost function. This is very important when one deals with redundant manipulators. One of the approaches to design optimal kinematic control is to use the Hamilton-Jacobi-Bellman (HJB) formulation [5], [6]. The analytical solution of the HJB equation is still a major challenge and as such these solutions are obtained off-line. The real time approximate solutions are known as approximate dynamic programming and are presented in [7], [8], [9]. In these approaches, the framework uses two neural networks - one for action and the other critic. Action network learns to actuate an optimal policy while critic network evaluates the cost function through learning.

The single network adaptive critic (SNAC) is introduced [10] where critic is updated using back-propagation. In

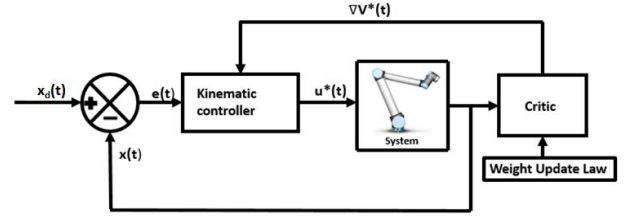


Fig. 1: Block diagram of the kinematic control scheme for a robot manipulator using adaptive critic method

[11], the kinematic control of a robot manipulator using this SNAC framework was presented. The block diagram of the kinematic control scheme for a robot manipulator using SNAC is shown in Fig. 1. As these schemes used back-propagation for critic weight updates, after repeated training, one could show the convergence to near optimal cost through extensive simulations. There exists very few works in the literature that has shown the convergence to the optimal cost in an analytical manner along with the proof of stability. In this work, we are solving the kinematic control problem of any n -DOF robot manipulator where the tasks of reaching a fixed target position and following a time varying task space trajectory have been solved in the framework of optimal regulation and optimal tracking respectively using SNAC. A simple and novel critic weight update rule which ensures that the closed loop system is stable has been proposed. The optimal kinematic control policy using HJB formulation has been developed in the framework of optimal regulation and optimal tracking. Former make the robot to reach fixed target position and later make the robot to follow the task-space time varying trajectory. The analytical proof of stability and convergence has been done using Lyapunov approach. As the degrees of freedom of robotic systems are increasing, learning based strategies will play more important role as compared to model based analytics [12],[13]. Thus the proposed approach has significant relevance in this context. The relevant literature is further scrutinized.

The optimization based kinematic control has been accomplished using local optimization for finding an instantaneous optimal solution [14], [15]. It is centered on the Jacobian pseudo-inverse and null space. The redundancy is resolved by including some constraints into the direct kinematic model or by projecting a particular solution onto the Jacobian null space. The other approach is a global optimization which

All authors are with the Department of Electrical Engineering, Indian Institute of Technology, Kanpur, India-208016. Email id : ¹aiswarya@iitk.ac.in, ²ravipr@iitk.ac.in, ³lbehera@iitk.ac.in

uses an integral type performance index along the whole trajectory [16], [17]. The redundancy resolution is converted to an optimal control problem with the necessary conditions of optimality given by the Pontryagin's principle or by the optimal control theory. Both of the aforementioned methods are suboptimal and require the pseudo-inversion of Jacobian continuously over time which is computationally expensive and suffers from local instability problems [18], [19], [20].

Existing approach towards following a time varying task space trajectory optimally is to find the feedforward term using the dynamics inversion concept and the feedback term by solving an HJB equation [21]. However, such solution is only near optimal because of the feedforward term. Therefore by using an augmented system dynamics consisting of feedback tracking error and reference trajectory and using it to solve the HJB equation results in an optimal control law which is a combination of feedforward and feedback control inputs [22].

The remainder of this paper is organised as follows. Problem formulation for optimal kinematic control is presented in Section II. In Section III and IV detailed mathematical derivation for the control scheme along with stability proof is presented. In Section V, various simulation and experimental results for kinematic motion control of a 6-DOF robotic manipulator are evaluated with comparison with the state-of-the-art kinematic control solutions. This paper is finally concluded in Section VI.

II. PROBLEM FORMULATION

The forward kinematics of a manipulator involves a non-linear transformation from Joint space to Cartesian space as described by:

$$x(t) = f(\theta(t)) \quad (1)$$

where, $x(t) \in \mathbb{R}^{n \times 1}$ describes the position and orientation of end effector in workspace at time t , $\theta(t) \in \mathbb{R}^{m \times 1}$ each element of which describes the joint angle in the joint space at time t , and $f(\cdot)$ is the non linear mapping. Because of non-linearity and redundancy of mapping, it is usually difficult to directly get $\theta(t)$ for desired $x(t) = x_d(t)$, where $x_d(t)$ is the desired end effector position. By contrast, the mapping from joint space to cartesian space at velocity level is affine mapping. Taking time derivative on both sides of (1) gives:

$$\dot{x}(t) = J\dot{\theta}(t) \quad (2)$$

where, $J = \frac{\partial f}{\partial \theta} \in \mathbb{R}^{n \times m}$ is the Jacobian matrix of $f(\cdot)$. In this paper we focus on design of angular speed of the manipulator which serves as an input to the tracking control loop for robot control. Then the robot manipulator kinematics can be rewritten by replacing $\dot{\theta}(t)$ by $u(t)$:

$$\dot{x}(t) = Ju(t) \quad (3)$$

Thus, we get the dynamics of the system as above. In this paper we propose to solve for an optimal control input $u(t)$ for a robot manipulator described by (3), such that the tracking error given by,

$$e(t) = x(t) - x_d(t) \quad (4)$$

for a given reference trajectory $x_d(t)$, reduces to zero with time.

III. OPTIMAL REGULATION

In this section, we address the optimal kinematic control of a robot manipulator, where the robot end effector has to reach a fixed target position. This is framed as an optimal regulation problem. In the context of kinematic control, optimal regulation means that the robot is made to reach a fixed target position following an optimal trajectory. This trajectory is generated by actuating the desired optimal control policy.

Let $x_d(t) \in \mathbb{R}^{n \times 1}$ be the desired end effector position. Differentiating (4) with respect to time, error dynamics of the system is obtained as,

$$\dot{e} = \dot{x}(t) - \dot{x}_d(t) = Ju(t) - \dot{x}_d(t) \quad (5)$$

A. Formulation of control policy

The infinite horizon HJB cost function for (5) is given by,

$$V(e(t)) = \int_t^\infty e(t)^T Q e(t) + u(t)^T R u(t) dt \quad (6)$$

where, $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ are positive definite design matrices. The control input needs to be admissible so that the cost function equation (6) is finite. The Hamiltonian for the this cost function with the admissible control input $u(t)$ is,

$$H(e(t), u(t), \nabla V(t)) = \nabla V^* \dot{e}(t) + e(t)^T Q e(t) + u(t)^T R u(t) \quad (7)$$

Here, ∇V^* is the gradient of the cost function $V(e(t))$ with respect to $e(t)$. The optimal control input which minimizes the cost function (6) also minimizes the Hamiltonian (7). Therefore, the optimal control is found by solving the following condition $\frac{\partial H(e, u, \nabla V)}{\partial u} = 0$ and obtained to be

$$u^*(e) = -\frac{1}{2} R^{-1} J^T \nabla V^*(e) \quad (8)$$

Then, HJB equation may be re-written as follows,

$$0 = \nabla V^* \dot{e}(t) + e(t)^T Q e(t) + u^*(t)^T R u^*(t) \quad (9)$$

with the cost function $V^*(0) = 0$.

B. Neural Network control design

By using the universal approximation property, $V^*(e(t))$ is constructed by a single hidden layer neural network with a non-linear activation function:

$$V^*(e(t)) = W_c^T \sigma_c(e) + \varepsilon_c \quad (10)$$

where, $W_c \in \mathbb{R}^l$, is the constant target NN weight vector, $\sigma_c \in \mathbb{R}^l$ is the activation function output, l is the number of hidden neurons and ε_c is the function reconstruction error. The target NN vector and reconstruction errors are assumed to be upper bounded according to $\|W_c\| \leq \lambda_{W_c}$ and $\|\varepsilon_c\| \leq \lambda_\varepsilon$.

$$\nabla V^* = \nabla \sigma_c^T W_c + \nabla \varepsilon_c \quad (11)$$

Approximate optimal cost function and its gradient is given by

$$\hat{V}^* = \hat{W}_c^T \sigma_c(e) \quad (12)$$

$$\nabla \hat{V}^* = \nabla \sigma_c^T \hat{W}_c \quad (13)$$

where, \hat{W}_c is the NN estimate of target weight vector W_c . Using (8) and (11), the optimal control law can be written as:

$$u^*(e) = \frac{-1}{2} R^{-1} J^T (\nabla \sigma_c^T W_c + \nabla \varepsilon_c) \quad (14)$$

Considering (8) and (13), estimated optimal control law is:

$$\hat{u}^*(e) = \frac{-1}{2} R^{-1} J^T \nabla \sigma_c^T \hat{W}_c \quad (15)$$

The modified error dynamics using (15) is,

$$\dot{e} = -\frac{1}{2} J R^{-1} J^T \nabla \sigma_c^T \hat{W}_c \quad (16)$$

Before proposing the critic weight tuning law and stability proofs, following assumptions need to be made:

Assumption 1: For the system represented by its error dynamics (5), with cost function (6), let $J_s(e)$ be a continuously differentiable Lyapunov function candidate satisfying,

$$J_s(e) = \nabla J_s(e)^T \dot{e} < 0 \quad (17)$$

Then, there exists a positive definite matrix, $M \in \mathbb{R}^{n \times n}$ ensuring,

$$\nabla J_s(e)^T \dot{e} = -\nabla J_s(e)^T M \nabla J_s(e) \leq -\lambda_{\min}(M) \|\nabla J_s(e)\|^2 \quad (18)$$

During the implementation, $J_s(e)$ can be obtained by selecting a polynomial with respect to the vector e , such as $J_s(e) = \frac{1}{2} e^T e$.

Remark 1: Based on the result of [23], the closed loop dynamics with optimal control law can be bounded by a function of the system state. In such situations, it can be assumed that $\|F(e) + G(e)(u^*)\| \leq \eta \|\nabla J_s(e)\|$ with $\eta > 0$, $\|\nabla J_s(e)\|^T \|F(e) + G(e)(u^*)\| \leq \eta \|\nabla J_s(e)\|^2$. Combining (17) with the fact given, $\lambda_{\min} \|\nabla J_s(e)\|^2 \leq \nabla J_s(e)^T M \nabla J_s(e) \leq \lambda_{\max}(M) \|\nabla J_s(e)\|^2$, it implies that *assumption 1* holds.

Moving on, a simple and novel critic weight tuning law is proposed.

$$\dot{\hat{W}}_c = \alpha \nabla \sigma_c J R^{-1} J^T \nabla J_s(e) \quad (19)$$

where, α is the learning rate of critic network and $J_s(e) = \frac{1}{2} e^T e$. By using this weight tuning law, stability and performance is guaranteed theoretically using the Lyapunov approach.

C. Stability Analysis

In this section, the stability of the system for optimal regulation is investigated.

Assumption 2: The Jacobian matrix is bounded as $\|J(\theta)\| \leq \lambda_J$, where λ_J is a positive constant. Here, the control coefficient matrix is the Jacobian matrix. Also, $\nabla \sigma_c, \nabla \varepsilon_c$ are bounded as $\|\nabla \sigma_c\| \leq \lambda_\sigma$ and $\|\nabla \varepsilon_c\| \leq \lambda_{\nabla \varepsilon}$ where λ_σ and

$\lambda_{\nabla \varepsilon}$ are positive constants.

The Lyapunov candidate function is selected as,

$$L(t) = \frac{1}{4\alpha} \widetilde{W}_c^T \widetilde{W}_c + J_s(e) \quad (20)$$

$$\dot{L}(t) = \frac{1}{2\alpha} \widetilde{W}_c^T \dot{\widetilde{W}}_c + \nabla J_s^T \dot{e} \quad (21)$$

where, $\widetilde{W} = W_c - \hat{W}_c$. Using (19),

$$\dot{L}(t) = \frac{-1}{2} \widetilde{W}_c^T \nabla \sigma_c J R^{-1} J^T \nabla J_s + \nabla J_s^T (\dot{e}) \quad (22)$$

The system error dynamics of the system for the optimal control law is $\dot{e}^* = Ju^*$. Using the control laws (14) and (15),

$$\dot{e}^* - \dot{e} = J(u^* - \hat{u}^*) = -\frac{1}{2} J R^{-1} J^T (\nabla \sigma_c^T \widetilde{W}_c + \nabla \varepsilon_c) \quad (23)$$

Using (23) in (22) and on simplification,

$$\dot{L}(t) = \nabla J_s^T \dot{e}^* + \frac{1}{2} \nabla J_s^T J R^{-1} J^T \nabla \varepsilon_c \quad (24)$$

Applying *Assumption 1* and *Assumption 2* here

$$\dot{L} \leq -\lambda_{\min}(M) \|\nabla J_s\|^2 + \frac{1}{2} \|\nabla J_s\| \lambda_J^2 \|R^{-1}\| \lambda_{\nabla \varepsilon} \quad (25)$$

From (25), the following inequality may be derived:

$$\|\nabla J_s\| > \frac{1}{2\lambda_{\min}(M)} \lambda_J^2 \|R^{-1}\| \lambda_{\nabla \varepsilon} \quad (26)$$

With the above condition to be true, $\dot{L} < 0$ which implies that e is bounded.

$\therefore \nabla V^* - \nabla \hat{V}^* = f(e^*) - f(e)$, bounded.

Let $y = \nabla V^* - \nabla \hat{V}^* - \nabla \varepsilon$, bounded.

Then,

$$y = \nabla \sigma_c^T \widetilde{W}_c$$

$$\dot{\widetilde{W}}_c = -\alpha \nabla \sigma_c J R^{-1} J^T \nabla J_s(e)$$

Using the condition of Persistence of Excitation on $\nabla \sigma_c$, boundedness on y and e ensures the boundedness of \widetilde{W}_c and $\therefore \hat{W}_c$. Assuming $\|\widetilde{W}_c\| \leq \lambda_{\widetilde{W}_c}$, it may be also observed that the estimated optimal controller (15) converges to a neighborhood of the optimal feedback controller (14) with a finite bound λ_u as,

$$\|u^*(e) - \hat{u}^*(e)\| \leq \frac{1}{2} \|R^{-1}\| \lambda_J (\lambda_\sigma \lambda_{\widetilde{W}_c} + \lambda_{\nabla \varepsilon}) \triangleq \lambda_u \quad (27)$$

Hence, it may be concluded that the instantaneous cost function $V(e, u, t) = e(t)^T Q e(t) + u(t)^T R u(t)$ is also bounded.

IV. OPTIMAL TRACKING CONTROL

In this section, the kinematic control of a robot manipulator following a time varying task space trajectory is solved in the framework of optimal tracking.

A. Formation of Augmented System

Let the time varying reference trajectory, denoted by $x_d(t) \in \mathbb{R}^n$ be possessing the dynamics

$$\dot{x}_d(t) = \varphi(x_d(t)) \quad (28)$$

with the initial condition $x_d(0) = x_{d0}$, where, $\varphi(x_d(t))$ is a Lipschitz continuous function satisfying $\varphi(0) = 0$. Let the trajectory tracking error be $e(t) = x(t) - x_d(t)$ with the initial condition $e(0) = e_0 = x_0 - x_{d0}$. Considering (3) and (29), the tracking error dynamics is:

$$\dot{e}(t) = -\varphi(x_d(t)) + Ju(t) \quad (29)$$

Next, an augmented system is constructed as in [22], in the form $\xi(t) = [e(t)^T, x_d(t)^T]^T \in \mathbb{R}^{2n}$ with initial condition $\xi(0) = \xi_0 = [e_0^T, x_{d0}^T]^T$. The augmented dynamics based on (29) and (30) can be formulated as:

$$\dot{\xi}(t) = F(\xi(t)) + G(\xi(t))u(t) \quad (30)$$

where, $F(\cdot)$ and $G(\cdot)$ are the new system matrices.

B. Formulation of Control Policy

The infinite horizon HJB cost function for the system in (31) is given by,

$$V(\xi(t)) = \int_t^\infty \xi(t)^T \bar{Q} \xi(t) + u(t)^T R u(t) dt \quad (31)$$

where, $\bar{Q} = \text{diag}\{Q, 0_{n \times n}\}$. $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ are positive definite matrices. The control input must be admissible so that cost function given by (32) is finite.

Applying the same methods of solving as in Section III-A, the control policy is obtained as,

$$u^*(\xi) = -\frac{1}{2} R^{-1} G(\xi)^T \nabla V^*(\xi) \quad (32)$$

C. Neural Network control design

Here, $V^*(\xi(t))$ is constructed by a single hidden layer neural network with a non-linear activation function:

$$V^*(\xi(t)) = W_c^T \phi_c(\xi) + \varepsilon_c \quad (33)$$

where, $W_c \in \mathbb{R}^p$, is the ideal weight, $\|W_c\| \leq \lambda_{W_c}$, $\phi_c \in \mathbb{R}^p$ is the activation function output, p is the number of hidden neurons and ε_c is the function approximation error. Using the same approach as in Section III-B, optimal control law and approximate control law may be obtained as,

$$u^*(\xi) = -\frac{1}{2} R^{-1} G(\xi)^T (\nabla \phi_c^T W_c + \nabla \varepsilon_c) \quad (34)$$

$$\hat{u}^*(\xi) = -\frac{1}{2} R^{-1} G(\xi)^T \nabla \phi_c^T \hat{W}_c \quad (35)$$

Applying (35) into the augmented system dynamics (31), $\dot{\xi}(t) = F(\xi) + G(\xi)\hat{u}^*(t)$ can be formulated as:

$$\dot{\xi} = F(\xi) - \frac{1}{2} G(\xi) R^{-1} G(\xi)^T \nabla \phi_c^T \hat{W}_c \quad (36)$$

The weight update law proposed for the critic of optimal tracking control is,

$$\dot{\hat{W}}_c = \alpha \nabla \phi_c G(\xi) R^{-1} G(\xi)^T \nabla J_s(\xi) \quad (37)$$

where, α is the learning rate of critic network and $J_s(\xi) = \frac{1}{2} \xi^T \xi$.

Remark 3: Assumption 1 and Remark 1 holds here. The augmented system states shall replace the system error states as follows ; For the augmented system (31), with cost function (32), let $J_s(\xi)$ be a continuously differentiable Lyapunov function candidate satisfying,

$$\dot{J}_s(\xi) = \nabla J_s(\xi)^T \dot{\xi} < 0 \quad (38)$$

Then, there exists a positive definite matrix, $M \in \mathbb{R}^{2n \times 2n}$ ensuring,

$$\nabla J_s(\xi)^T \dot{\xi} = -\nabla J_s(\xi)^T M \nabla J_s(\xi) \leq -\lambda_{\min}(M) \|\nabla J_s(\xi)\|^2 \quad (39)$$

During the implementation, $J_s(\xi) = \frac{1}{2} \xi^T \xi$.

D. Stability Analysis

In this section, the stability of the system is investigated.

Remark 4: Assumption 2 holds here. However, the control coefficient matrix is modified. The assumption $\|G(\xi)\| \leq \lambda_g$, where λ_g is a positive constant shall be used. Lyapunov function,

$$L(t) = \frac{1}{4\alpha} \tilde{W}_c^T \tilde{W}_c + J_s(\xi) \quad (40)$$

$$\dot{L}(t) = \frac{1}{2\alpha} \tilde{W}_c^T \dot{\tilde{W}}_c + \nabla J_s^T \dot{\xi} \quad (41)$$

Using the similar approach as in Section III-C,

$$\dot{L}(t) = \nabla J_s^T \dot{\xi}^* + \frac{1}{2} \nabla J_s^T G(\xi) R^{-1} G(\xi)^T \tilde{W}_c \quad (42)$$

Applying the Remark 3 and Remark 4 here

$$\dot{L} \leq -\lambda_{\min}(M) \|\nabla J_s\|^2 + \frac{1}{2} \|\nabla J_s\| \lambda_g^2 \|R^{-1}\| \lambda_{\nabla \varepsilon} \quad (43)$$

From this, we observe that if the condition below holds, then, $\dot{L}(t) < 0$ which implies that ξ is bounded.

$$\|\nabla J_s\| \geq \frac{1}{2\lambda_{\min}(M)} \lambda_g^2 \|R^{-1}\| \lambda_{\nabla \varepsilon} \quad (44)$$

$\therefore \nabla V^* - \nabla \hat{V}^* = f(\xi^*) - f(\xi)$, bounded.

Let $y = \nabla V^* - \nabla \hat{V}^* - \nabla \varepsilon$, bounded.

Then,

$$y = \nabla \phi_c^T \tilde{W}_c$$

$$\dot{\tilde{W}}_c = -\alpha \nabla \phi_c G R^{-1} G^T \nabla J_s(\xi)$$

Using the condition of Persistence of Excitation on $\nabla \phi_c$, boundedness on y and ξ ensures the boundedness of \tilde{W}_c and $\therefore \hat{W}_c$. Assuming $\|\tilde{W}_c\| \leq \lambda_{\tilde{W}_c}$, it may be noted that estimated controller input converges to the neighborhood of the optimal control input as a finite bound λ_u exists just as in the case of regulation as shown below:

$$\|u^*(\xi) - \hat{u}^*(\xi)\| \leq \frac{1}{2} \|R^{-1}\| \lambda_g (\lambda_\sigma \lambda_{\tilde{W}_c} + \lambda_{\nabla \varepsilon_c}) \triangleq \lambda_u \quad (45)$$

It is also observed that the cost function $V(\xi, u, t) = \xi(t)^T \bar{Q} \xi(t) + u(t)^T u(t)$ is also bounded.

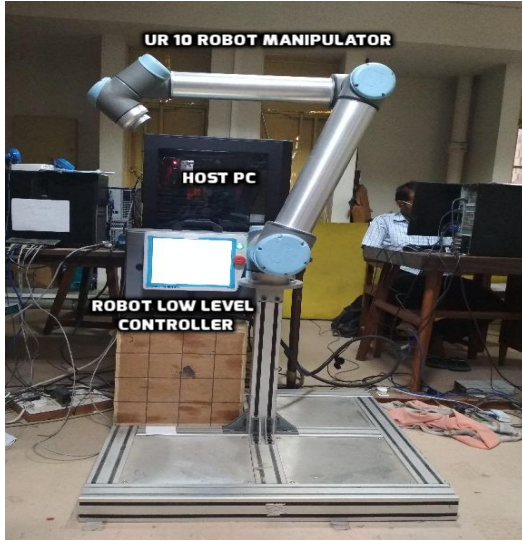


Fig. 2: Hardware Setup

V. RESULTS AND DISCUSSION

In this section, we consider the numerical simulations followed by real time experimental validations on a real 6 DOF UR 10 robot manipulator to demonstrate the effectiveness of the proposed kinematic control.

A. Experimental Setup

Our experimental setup shown in Fig. 2 consists of a UR10 robot manipulator with its controller box/internal computer and a host PC/external computer. The UR10 robot manipulator is a 6 DOF robot arm designed to safely work alongside and in collaboration with a human. This arm can follow position commands like a traditional industrial robot, as well as take velocity commands to apply a given velocity in/around a specified axis. The low level robot controller is a program running on UR10's internal computer broadcasting robot arm data, receiving and interpreting the commands and controlling the arm accordingly. There are several options for communicating with the robot low level controller to control the robot including the teach pendant or opening a TCP socket (C++/Python) on a host computer. We used open source C++ based UrDriver wrapper class integrated with ROS on a host PC to implement our proposed velocity based kinematic control scheme. The host PC streams joint velocity commands via URScript to the robot real time interface over Ethernet at 125Hz. The driver was configured with necessary parameters like IP address of the robot at startup using ROS parameter server.

B. Reaching a Fixed Position

Simulations are first performed to verify the kinematic control of the robot manipulator to reach a fixed position using the control law expressed in Equation (15). A typical simulation run generated with a random seed pose ($\theta_0 = [-0.51 - 1.041.48 - 1.82 - 1.45 - 1.62]rad$) to a fixed target pose $[-0.658, 0.626, 0.407]m$ in cartesian space is shown in Figure 3(a). The weights were initialized randomly and $\alpha = \alpha_{initial}(tanh(n-k)) + \alpha_{final}$, where, $n=50$, $k=time$ instance,

$\alpha_{initial} = 100$ and $\alpha_{final} = 150$. The predicted cost function was: $w_1e_1^2 + w_2e_1e_2 + w_3e_1e_3 + w_4e_2^2 + w_5e_2e_3 + w_6e_3^2$

C. Following a Time Varying Reference Trajectory

In this section, the kinematic control of the robot manipulator to follow a time varying reference trajectory is validated. Simulations are first performed using the control law derived in Equation (35). A typical simulation generated with a random seed pose ($\theta_0 = [-0.51 - 1.041.48 - 1.82 - 1.45 - 1.62]rad$) and following a time varying reference trajectory moving at an angular speed of $0.075 rad/s$ along a circle centered at $[-0.7, 0, 0.5]m$ with a radius of $0.2m$ is shown in Figure 3(e). Weights were initialized randomly and $\alpha = \alpha_{initial}(tanh(n-k)) + \alpha_{final}$, where, $n=10$, $k=time$ instance, $\alpha_{initial} = 20$ and $\alpha_{final} = 70$. In this work, we take the predicted cost to be:

$$w_1\xi_1^2 + w_2\xi_1\xi_2 + w_3\xi_1\xi_3 + w_4\xi_1\xi_4 + w_5\xi_1\xi_5 + w_6\xi_1\xi_6 + w_7\xi_2^2 + w_8\xi_2\xi_3 + w_9\xi_2\xi_4 + w_{10}\xi_2\xi_5 + w_{11}\xi_2\xi_6 + w_{12}\xi_3^2 + w_{13}\xi_3\xi_4 + w_{14}\xi_3\xi_5 + w_{15}\xi_3\xi_6 + w_{16}\xi_4^2 + w_{17}\xi_4\xi_5 + w_{18}\xi_4\xi_6 + w_{19}\xi_5^2 + w_{20}\xi_5\xi_6 + w_{21}\xi_6^2.$$

D. Observations

After a short transient time, the error trajectory $e(t) = x(t) - x_d(t)$ converges to zero in both cases as shown in Fig. 3(b) and 3(f) respectively. Note that the control input $u = \theta$ remain within the maximum joint velocity limits shown in Fig. 3(c) and Fig. 3(g). However unlike in optimal regulation, it does not converge to zero as the velocity compensation is required for perfect tracking control. The time history plot of the associated cost function is shown in Fig. 3(d,h). The experimental validation for the corresponding target pose was performed using a real UR 10 robot manipulator. The results from Fig. 4 shows that the end effector of UR 10 robot manipulator successfully reaches the target pose under the proposed control scheme and successfully follows the reference time varying circular trajectory starting from a random seed pose. The time history plot of the associated cost function is shown in Fig. 4(d,h). Unlike in simulations, the control effort shows some chatter due to the inertia of the real hardware.

E. Quantitative Test Comparison

In order to quantify the performance of the proposed optimal kinematic control, an automated test process was used where a large, statistically-valid number of random samples were used as inputs. Kinematic models of Universal Robot (UR) 10 was used to demonstrate the tests. The quantitative test methodology for comparing the proposed method for kinematic control against the state-of-the-art kinematic control using an RNN [3] and Singular Value Filtering (SVF) approach [24] is entailed.

First, a total of 1,000 random samples and 1000 different feasible elliptical trajectories were generated in the sample space for regulation and tracking respectively. The sample space is a cuboid volume of task space within robot's reach and every sample is a pair of seed pose and target pose in the case of regulation. The three kinematic control schemes mentioned above are tested

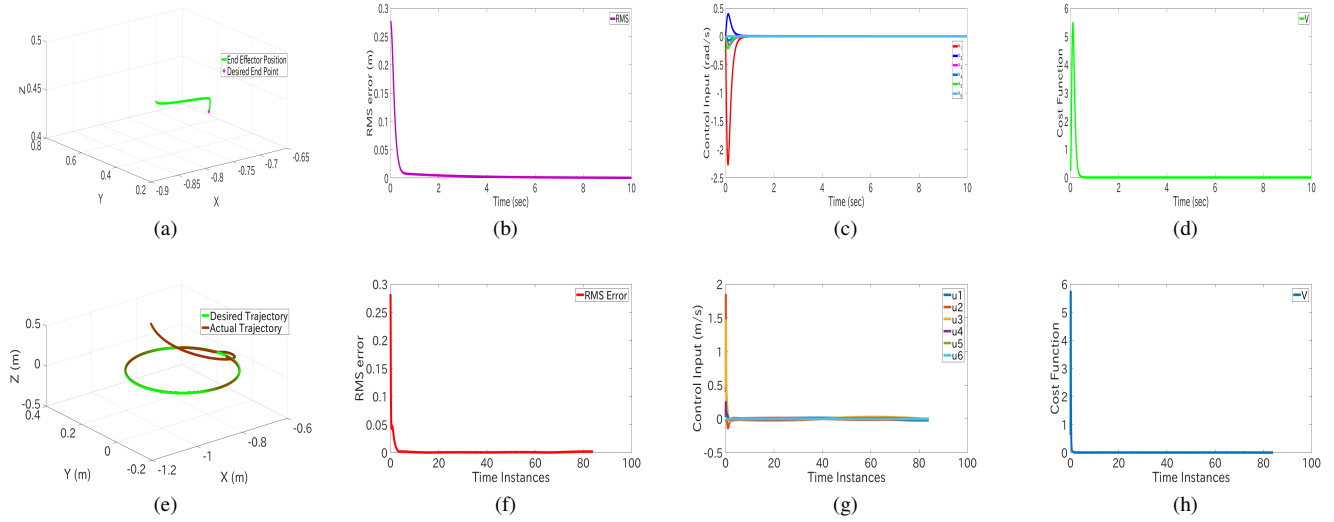


Fig. 3: Simulation results for Optimal regulation of a fixed target (a-d) and optimal tracking for a time varying elliptical trajectory (e-h).

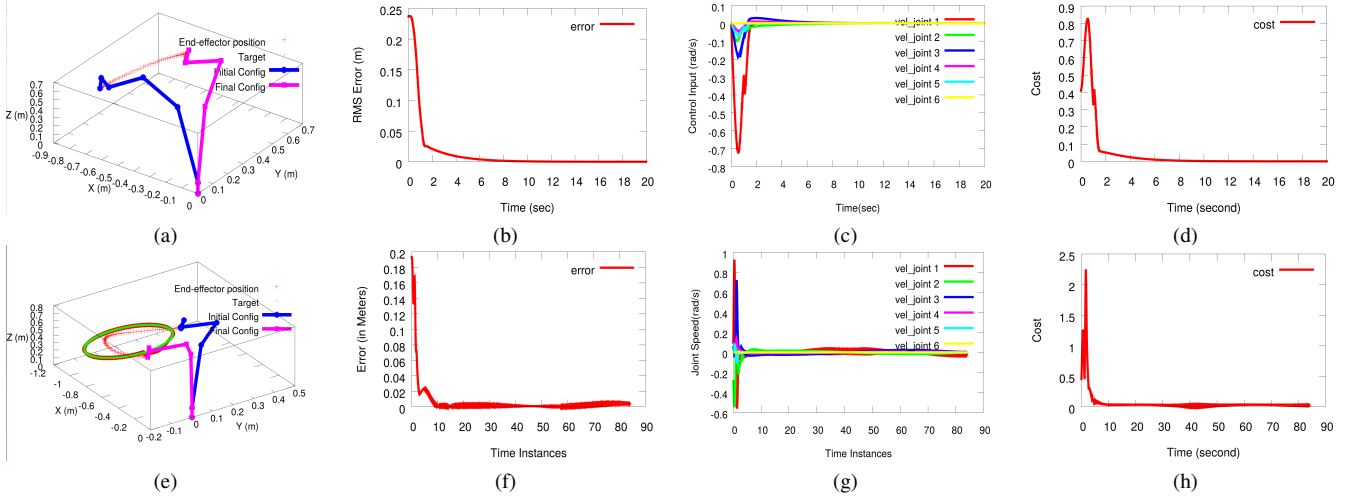


Fig. 4: Experimental results for Optimal regulation of a fixed target (a-d) and optimal tracking for a time varying elliptical trajectory (e-h).

and compared on trajectory cost. The trajectory cost is defined as normalized total cost: V/N , where $V = \int_t^\infty e(t)^T Q e(t) + u(t)^T R u(t) + \dot{u}(t)^T R \dot{u}(t) dt$ for the same $Q = I_{3 \times 3}$ and $R = I_{6 \times 6}$ and N is the number of total sample instances. The design parameters were selected such that the maximum control effort is the same for all three approaches.

Controller	Trajectory cost Regulation	Trajectory Cost Tracking
Controller [3]	49.74	14.39
Controller [24]	48.32	32.21
Proposed	5.78	3.30

TABLE I: Comparisons of different algorithms for Kinematic Control of UR10 Robot Manipulator

The cost matrix contain two terms. One for the measure

of optimal control action and another for smoothness of the motion. The comparison from Table I shows that the proposed kinematic control has the optimal control action than the state-of-the-art kinematic control.

VI. CONCLUSION

In this work, we have designed an optimal kinematic controller for a robot manipulator using SNAC framework. A simple critic weight update law was proposed which ensured that the closed loop system becomes stable in the sense of Lyapunov while following an optimal trajectory. The robot was expected to reach a target position or follow a time varying trajectory in the task space while optimizing a global cost function. Using the proposed optimal regulation and optimal tracking framework in the context of SNAC, it has been experimentally demonstrated that robot performs desired tasks with optimal cost as evident from Table 1.

REFERENCES

- [1] S. Kumar, L. Behera, and T. McGinnity, "Kinematic control of a redundant manipulator using inverse-forward adaptive scheme with a ksom based hint generator," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 622–633, 2010.
- [2] P. Prem Kumar and L. Behera, "Visual servoing of a redundant manipulator with jacobian matrix estimation using self-organizing map," *Robotics and Autonomous Systems*, vol. 58, no. 8, pp. 978–990, 2010.
- [3] S. Li, Y. Zhang, and L. Jin, "Kinematic control of redundant manipulators using neural networks," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2243–2254, 2017.
- [4] I. Sirazuddin, L. Behera, T. McGinnity, and S. Coleman, "Image based visual servoing of a 7 dof robot manipulator using an adaptive distributed fuzzy pd controller," *IEEE/ASME Trans on Mechatronics*, vol. 19, no. 2, pp. 512–523, 2014.
- [5] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012.
- [6] S. Keerthi and E. Gilbert, "An existence theorem for discrete-time infinite-horizon optimal control problems," *IEEE Transactions on Automatic Control*, vol. 30, no. 9, pp. 907–909, 1985.
- [7] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 2005, vol. 1, no. 3.
- [8] Z. Chen and S. Jagannathan, "Generalized hamilton–jacobi–bellman formulation-based neural network control of affine nonlinear discrete-time systems," *IEEE Transactions on Neural Networks*, vol. 19, no. 1, pp. 90–106, 2008.
- [9] J. Si, A. Barto, W. Powell, and D. Wunsch, "Handbook of learning and approx. dynamics prog," 2004.
- [10] R. Padhi, N. Unnikrishnan, X. Wang, and S. Balakrishnan, "A single network adaptive critic (snac) architecture for optimal control synthesis for a class of nonlinear systems," *Neural Networks*, vol. 19, no. 10, pp. 1648–1660, 2006.
- [11] P. K. Patchaikani, L. Behera, and G. Prasad, "A single network adaptive critic-based redundancy resolution scheme for robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3241–3253, 2012.
- [12] C. S. . T. B. . J. B. . J. P. . H. Ulbrich, "Predictive online inverse kinematics for redundant manipulators," in *ICRA Proceedings of the 2014 IEEE International Conference on*. IEEE, 2014, pp. 5056–5061.
- [13] R. C. L. . T.-W. L. . Y.-H. Tsai, "Analytical inverse kinematic solution for modularized 7-dof redundant manipulators with offsets at shoulder and wrist," in *IROS Proceedings of the 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 516–521.
- [14] M. Giftthaler, F. Farshidian, T. Sandy, L. Stadelmann, and J. Buchli, "Efficient kinematic planning for mobile manipulators with non-holonomic constraints using optimal control," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 3411–3417.
- [15] J. Hollerbach and K. Suh, "Redundancy resolution of manipulators through torque optimization," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2. IEEE, 1985, pp. 1016–1021.
- [16] S.-W. Kim, K.-B. Park, and J.-J. Lee, "Redundancy resolution of robot manipulators using optimal kinematic control," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*. IEEE, 1994, pp. 683–688.
- [17] D. P. Martin, J. Baillieul, and J. M. Hollerbach, "Resolution of kinematic redundancy using optimization techniques," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 4, pp. 529–533, 1989.
- [18] A. Maciejewski, "Kinetic limitations on the use of redundancy in robotic manipulators," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*. IEEE, 1989, pp. 113–118.
- [19] K. A. O'Neil, "Divergence of linear acceleration-based redundancy resolution schemes," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 625–631, 2002.
- [20] S. Cocuzza, I. Pretto, and S. Debei, "Novel reaction control techniques for redundant space manipulators: Theory and simulated microgravity tests," *Acta Astronautica*, vol. 68, no. 11-12, pp. 1712–1721, 2011.
- [21] B. Kiumarsi and F. L. Lewis, "Actor–critic-based optimal tracking for partially unknown nonlinear discrete-time systems," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 1, pp. 140–151, 2015.
- [22] D. Wang, D. Liu, Y. Zhang, and H. Li, "Neural network robust tracking control with adaptive critic framework for uncertain nonlinear systems," *Neural Networks*, vol. 97, pp. 11–18, 2018.
- [23] T. Dierks and S. Jagannathan, "Optimal control of affine nonlinear continuous-time systems," in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 1568–1573.
- [24] A. Colomé and C. Torras, "Redundant inverse kinematics: Experimental comparative review and two enhancements," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 5333–5340.
- [25] K. Vamvoudakis and S. Jagannathan, *Control of Complex Systems: Theory and Applications*. Butterworth-Heinemann, 2016.
- [26] K. G. Vamvoudakis and F. L. Lewis, "Multi-player non-zero-sum games: Online adaptive learning solution of coupled hamilton–jacobi equations," *Automatica*, vol. 47, no. 8, pp. 1556–1569, 2011.
- [27] T. Dierks and S. Jagannathan, "Optimal tracking control of affine nonlinear discrete-time systems with unknown internal dynamics," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. IEEE, 2009, pp. 6750–6755.
- [28] D. Vrabie and F. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Networks*, vol. 22, no. 3, pp. 237–246, 2009.
- [29] R. R. Selmic and F. L. Lewis, "Backlash compensation in nonlinear systems using dynamic inversion by neural networks," in *Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 1163–1168.