Adaptive Compensation and Adaptive Optimal Control of a Single Link Manipulator

Abdullah Almansour*†, Afreen Islam‡, and Guido Herrmann§
*Abdullah Al Salem University, Khaldiya, Kuwait; abdullah.almansour@aasu.edu.kw
†Imperial College London, London, United Kingdom; a.almansour24@imperial.ac.uk
‡University College Dublin, Dublin, Ireland; afreen.islam@ucd.ie
§University of Manchester, Manchester, United Kingdom; guido.herrmann@manchester.ac.uk

***Abstract*—This paper presents an online solution to the finite-horizon optimal tracking control problem for continuous-time nonlinear systems with partially unknown dynamics, based on an Adaptive Dynamic Programming (ADP) approach. The method employs a dual-approximation identifier–critic neural network (NN) architecture, with both networks tuned simultaneously during online implementation. The unknown weights of the identifier and critic activation functions are estimated using a filter-based adaptive algorithm, which provides a simple online validation of the persistence of excitation (PE) condition required for convergence of the control parameters. The controller is evaluated in simulation on an ideal single-link robotic manipulator with partially unknown dynamics and is compared against two classical adaptive nonlinear control strategies: an adaptive Lyapunov-based nonlinear (ALN) controller and an adaptive backstepping (ABS) controller. Performance is assessed in terms of adaptive parameter convergence, tracking accuracy, control input smoothness, and tuning complexity. The ADP-based controller demonstrates the most intuitive tuning process, as its parameters are directly linked to observed system behaviour, and achieves superior tracking performance with the smoothest control input among the three controllers.**

***Index Terms*—adaptive optimal control, trajectory tracking, finite horizon, nonlinear control**

I. Introduction

Optimal control strategies are desirable in many applications [1]. For instance, a vehicle may need to optimise the trade-off between minimising fuel consumption and travel time, or between operational efficiency and financial cost in economic systems. The optimal control solution for a user-defined cost function is obtained by solving the Riccati equation in the case of linear systems, and the Hamilton–Jacobi–Bellman (HJB) equation for nonlinear systems. In general, the optimal solution is computed offline and requires full knowledge of the system dynamics. For nonlinear systems, obtaining an analytical solution to the HJB equation is a challenging problem [2].

Deriving an analytical model that accurately describes certain kinds of systems is challenging due to the presence of various uncertainties [3], [4]. Furthermore, system dynamics may change over time due to the ageing of system components. Adaptive control algorithms address this issue by enabling the system to learn unknown dynamics online using measurements along the system trajectory [5]. Although adaptive controllers are designed to guarantee stability while estimating unknown parameters, they are not optimal with respect to a predefined cost function [6]. In contrast, an adaptive optimal controller simultaneously estimates the system dynamics and computes an optimal or approximately optimal solution with respect to the predefined cost function [7].

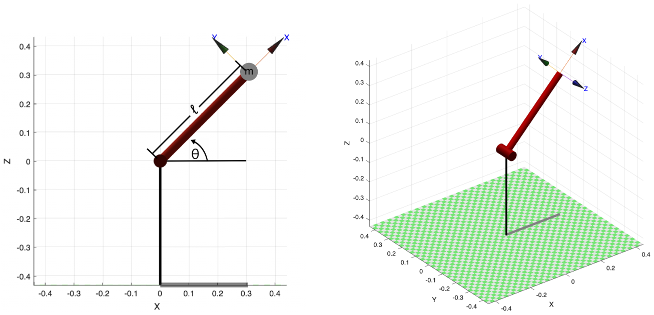
In [8], an adaptive Lyapunov-based nonlinear (ALN) controller was employed to control a rigid robotic system undergoing impact collision with an underactuated, deformable mass–spring system. In [9], the advantages of both the ALN controller and barrier functions were utilised to control an underactuated brachiating robot. In [10], a neural network (NN)-based adaptive backstepping (ABS) controller was used for a manipulator. In [11], an ABS sliding mode control was employed for trajectory tracking of a robotic manipulator. Similarly, in [12], an ABS controller was proposed for trajectory tracking of flexible-joint manipulators.

In this work, a single-link robotic manipulator with partially unknown dynamics is considered. An Adaptive Dynamic Programming (ADP) approach based on an identifier–critic structure is proposed to achieve finite-horizon optimal control of the manipulator while simultaneously estimating the unknown system dynamics using an adaptive identifier. The proposed controller is compared with two adaptive nonlinear control strategies: an ALN controller and an ABS controller. Section II presents the derivation of the three controllers, Section III compares their performance based on simulation results obtained from the single-link manipulator, and Section IV concludes the paper.

II. Methodology

A single-link robotic manipulator model is used to evaluate the performance of the proposed ADP method, which combines the optimal trajectory tracking approach proposed in [13] and [14] to achieve finite-horizon optimal trajectory tracking for the system shown in Figure 1, starting from an unstable initial pose aligned with the x-axis.

The performance of the ADP-based adaptive optimal controller is evaluated using simulation results obtained with the MATLAB Robotics Toolbox [15] and compared with ALN and ABS controllers, in terms of their tracking performance, smoothness of control input, and convergence of their adaptive parameters. The dynamics of the single-link manipulator can



(a) Planar view of the single-link manipulator model.

(b) Three-dimensional view of the single-link manipulator.

Fig. 1. Visual representation of the single-link manipulator model from both planar and three-dimensional perspectives.

TABLE I
MODEL SPECIFICATION FOR SINGLE LINK MANIPULATOR.

Parameter	Value	Unit
Link length (l)	0.418	m
Gravitational acceleration g	9.81	m/s^2
Total payload mass (m)	8.05	kg

be described by the simplified state space model:

$$\begin{aligned}\dot{x} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} u + \begin{bmatrix} 0 \\ -\frac{g}{l} \end{bmatrix} \cos(x_1) \\ &= Ax + Bu + W_1^T f(x),\end{aligned} \tag{1}$$

where $x_1 = \theta$ and $x_2 = \dot{\theta}$ are the system states, u is the control input, l is the distance from the joint to the end effector, m is the mass attached to the end effector, and g is the gravitational acceleration constant, with values given in Table I.

A. Adaptive Dynamic Programming Controller

The system dynamics (1) are assumed to be partially known, where the matrices $A \in \mathbb{R}^{2\times 2}$ and $B \in \mathbb{R}^2$ constitute the known part, $W_1 \in \mathbb{R}^2$ is a vector of unknown constant parameters, and $f(x) \in \mathbb{R}$ is an activation function. The cost function is defined as

$$\begin{aligned}V(e, u, t) &= \int_t^{t+T} r(e, u)\, d\tau, \\ r(e, u) &= e^{\mathrm{T}} Q e + u^{\mathrm{T}} R u, \\ e &= x - x_d,\end{aligned} \tag{2}$$

where x_d denotes the desired state trajectory, x is the system state, and e is the tracking error. The matrices $Q \geq 0$ and $R > 0$ are weighting matrices that penalise tracking error and control effort, respectively.

The optimal solution depends on the relative costs of the optimised parameters rather the absolute values of R and Q. Increasing the values in Q with respect to R results in a more aggressive control performance to achieve faster error convergence, and inversely, reducing Q with respect to R may result in poor tracking performance, or instability in extreme scenarios due to the insufficiency of control efforts. Therefore, careful design consideration of the cost function (2) must be taken to achieve desirable results.

The proposed ADP design method utilises an adaptive identifier NN and a critic NN. The adaptive identifier is designed to approximate the unknown system dynamics, while the critic approximates the solution of the HJB equation, which minimises the cost function, defined in equation (2). For the algorithm to work, the PE condition needs to be fulfilled.

Definition 1 (Persistence of Excitation (PE) [16]): A piecewise continuous signal $\zeta(t)$ is said to be persistently excited over the time interval $[t, t+T]$, if there exists a strictly positive constant $\sigma > 0$, such that:

$$\int_t^{t+T} \zeta(\tau)\zeta^T(\tau) d\tau \geq \sigma I, \quad \forall t > 0 \tag{3}$$

The adaptive identifier NN is used to reconstruct the unknown system dynamics based on the filtered variables x_f, u_f, and f_f, which obtained by passing the input-output measurements through a low-pass filter (LPF), given by:

$$\begin{aligned} k\dot{x}_f + x_f &= x, \quad x_f(0) = 0, \\ k\dot{u}_f + u_f &= u, \quad u_f(0) = 0, \\ k\dot{f}_f + f_f &= f, \quad f_f(0) = 0. \end{aligned} \tag{4}$$

The parameter $k > 0$ is a positive constant inversely proportional to the cut-off frequency of the LPF, $\omega_0 = \frac{1}{k}$ rad/s, and should be chosen sufficiently small to avoid filtering out essential information in the variables [3].

The auxiliary filtered regressor matrices [13] are defined as:

$$\begin{aligned}\dot{P}_1 &= -\gamma_1 P_1 + f_f f_f^{\mathrm{T}}, \quad P_1(0) = 0, \\ \dot{Q}_1 &= -\gamma_1 Q_1 + f_f \left[\frac{x - x_f}{k} - Ax_f - Bu_f \right]^{\mathrm{T}}, \quad Q_1(0) = 0,\end{aligned} \tag{5}$$

where the parameter $\gamma_1 > 0$ is a positive definite forgetting factor that ensures the boundedness of $P_1(t)$ and $Q_1(t)$ [3].

A second auxiliary variable is then defined based on the filtered and integrated regressor matrices P_1 and Q_1 as:

$$M_1 = P_1 \hat{W}_1 - Q_1 = -P_1 \tilde{W}_1, \tag{6}$$

which contains information on the estimation error, and it can improve the convergence of the estimated parameters to their corresponding system values [3].

The adaptive law is then designed as:

$$\dot{\hat{W}}_1 = -\Gamma_1 M_1, \tag{7}$$

where $\Gamma_1 > 0$ is a positive definite constant that constitutes the learning gain. Under the assumption that the activation function f in (1) is persistently excited, the estimation error $\tilde{W}_1$ converges exponentially to zero [17]. This can be validated by testing the positive definiteness of the auxiliary filtered regressor P_1. This is generally done by applying a proper control input u (which sufficiently excites the system) or by adding a probing noise signal to the control input u to ensure the PE condition is met. On convergence, the probing noise signal can be stopped [2].

An adaptive critic NN is designed based on the identified system parameters. The system dynamics (1) can be rewritten in terms of the identified parameters as:

$$\dot{x} = Ax + Bu + \hat{W}_1^{\mathrm{T}} f(x) + \epsilon_N, \tag{8}$$

where ϵ_N is the identifier approximation error. The tracking-error dynamics are then:

$$\dot{e} = \dot{x} - \dot{x}_d = Ax + Bu + \hat{W}_1^{\mathrm{T}} f(x) - \dot{x}_d + \epsilon_N. \tag{9}$$

The optimal tracking control input can be designed in two parts, $u = u_s + u_e$, similar to [13]. The steady-state control input u_s is used to maintain the desired state trajectory x_d, while the stabilising input u_e minimises the cost function (2) during transient period.

The steady-state control input, u_s, is obtained by substituting $x = x_d$ and $u = u_s$ in the system dynamics (8) and solving for u_s, which returns:

$$u_s = B^{\mathrm{g}} \left(\dot{x}_d - Ax_d - \hat{W}_1^{\mathrm{T}} f(x_d) \right), \tag{10}$$

where B^{g} denotes the generalised inverse of the coupling matrix B, which may not be invertible.

By substituting (10) in (9), the error dynamic equation is reformulated as:

$$\dot{e} = Ae + \hat{W}_1^{\mathrm{T}} \big(f(x) - f(x_d) \big) - \dot{x}_d + Bu_e + \epsilon_N + \epsilon_g, \tag{11}$$

where ϵ_g is a bounded residual error that results from the generalised inverse of B. If B is full rank, the residual error ϵ_g will be zero, and the control input (10) can maintain the steady-state error near-zero, given that ϵ_N is small. Furthermore, if $\epsilon_N = 0$, the steady state input u_s can maintain the desired final state trajectory.

The cost function is reformulated in terms of the stabilising input and tracking error as:

$$V(e, u_e, t) = \int_t^{t+T} \left(e^{\mathrm{T}} Q e + u_e^{\mathrm{T}} R u_e \right) d\tau, \tag{12}$$

and the Hamiltonian is then defined as:

$$H(e, u_e, V, t) = L + \lambda^{\mathrm{T}} f = e^{\mathrm{T}} Q e + u_e^{\mathrm{T}} R u_e + \nabla V_e^{*\mathrm{T}} \dot{e}, \tag{13}$$

where $\nabla V_e^* = \frac{\partial V^*}{\partial e}$ is the partial derivative of V^*.

The Hamiltonian (13) is a nonlinear and time-varying function, which is difficult to solve analytically [14]. Therefore, a critic NN with time-varying activation functions $\phi(x, T-t)$ is used to approximate the optimal value function V^* as:

$$V^* = W_2^{*\mathrm{T}} \phi(x, T-t) + \epsilon_v, \tag{14}$$

where $W_2^* \in \mathbb{R}^N$ are the ideal weights of the activation functions, and ϵ_v represents the NN approximation error.

According to the Weierstrass approximation theorem, for a neural network with N activation functions, $\{\phi_i(x, T-t)\}_{i=1}^N$, where $\phi_i(x, T-t)$ denotes the i^{th} element of the activation function vector $\phi(x, T-t)$, the approximation error can converge to zero through appropriate selection of the activation functions, such that $\lim_{N\to\infty} \epsilon_v = 0$ [14].

The partial derivatives of the optimal cost value function are approximated by:

$$\begin{aligned} \nabla V_e^* &= \nabla \phi_e^{\mathrm{T}}(x, T-t)\, W_2^* + \nabla \epsilon_{v,e}, \\ \nabla V_t^* &= \nabla \phi_t^{\mathrm{T}}(x, T-t)\, W_2^* + \nabla \epsilon_{v,t}, \end{aligned} \tag{15}$$

where $\nabla(.)_e$ and $\nabla(.)_t$ represent the derivatives with respect to the tracking error and time, respectively.

The solution to HJB equation could then be written in terms of the approximated value function as:

$$\begin{aligned} e^{\mathrm{T}} Q e + u_e^{\mathrm{T}} R u_e + W_2^{*\mathrm{T}} \nabla \phi(x, T-t)\, \dot{e} \\ + W_2^{*\mathrm{T}} \nabla \phi_t(x, T-t) + \epsilon_{\mathrm{HJB}} = 0, \end{aligned} \tag{16}$$

where ϵ_{HJB} is the HJB error, which represents the residual error due to the accumulative NN approximation errors and the generalised inverse of B.

By defining the known terms in (16) as:

$$\begin{aligned} \Xi &:= \nabla \phi_t(x, T-t) + \nabla \phi(x, T-t)\, \dot{e}, \\ \Theta &:= e^{\mathrm{T}} Q e + u_e^{\mathrm{T}} R u_e, \end{aligned} \tag{17}$$

the solution to the HJB equation can be written in the form of a recursive neural network (RNN):

$$\Theta + W_2^{*\mathrm{T}}\, \Xi = 0. \tag{18}$$

Using the same adaptive method employed for the identifier NN, the auxiliary variables P_2 and Q_2 for the critic NN [13] are defined as:

$$\begin{aligned} \dot{P}_2 &= -\gamma_2 P_2 + \Xi\, \Xi^{\mathrm{T}}, \quad P_2(0) = 0, \\ \dot{Q}_2 &= -\gamma_2 Q_2 + \Xi\, \Theta, \quad Q_2(0) = 0, \end{aligned} \tag{19}$$

where $\gamma_2 > 0$ is a positive forgetting factor.

The auxiliary variable M_2 is then defined as:

$$M_2 = P_2\, \hat{W}_2 + Q_2 = -P_2\, \tilde{W}_2, \tag{20}$$

which holds the parameter approximation error, as shown in [13], [14], [17]. The adaptive law for the critic NN weight [13] can then be designed as:

$$\dot{\hat{W}}_2 = -\Gamma_2 M_2, \tag{21}$$

where $\Gamma_2 > 0$ is a positive definite learning gain. Provided that the regressor vector Ξ is persistently excited, the adaptive law (21) converges exponentially to the optimal control solution if the approximation error satisfies $\epsilon_{HJB} = 0$. However, in the presence of a bounded approximation error $\epsilon_{HJB} \neq 0$, the adaptive law (21) converges to a near-optimal solution [17].

According to the optimality principle [1], the optimal control input that minimises the cost function (12) must satisfy:

$$\begin{aligned} &\frac{\partial H}{\partial u} = 0, \\ &2Ru_e^* + B^{\mathrm{T}} \frac{\partial V^*}{\partial e} = 0, \\ &u_e^* = -\frac{1}{2} R^{-1} B^{\mathrm{T}} \frac{\partial V^*}{\partial e}. \end{aligned} \tag{22}$$

As the optimal value function $V^*(t)$ and its derivative $\frac{\partial V^*}{\partial e}$ are unknown, the value function approximation from the critic NN (16) is used to estimate the optimal stabilising input as:

$$\hat{u}_e = -\frac{1}{2} R^{-1} B^{\mathrm{T}}\, \hat{W}^{\mathrm{T}} \nabla\phi. \tag{23}$$

By appropriately designing the activation function, the NN approximation error and its derivatives can converge to zero [14], and $\hat{u}_e$ converges to its optimal solution u_e^*.

B. Classic Adaptive Controllers

The tracking performance and parameter estimation of the the adaptive optimal controller is compared with the results obtained by applying two classical adaptive controllers: an adaptive Lyapunov-based nonlinear (ALN) controller and an adaptive backstepping (ABS) controller.

The state space model of the system was reformulated as:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\frac{g}{l}\cos(x_1) + \frac{1}{ml^2} u = \alpha\cos(x_1) + \beta u, \end{aligned} \tag{24}$$

where $x_1 = \theta$ and $x_2 = \dot{\theta}$ denote the system states, u is the control input, l is the distance from the joint to the end effector, m is the mass attached to the end effector, and g is the gravitational acceleration constant. The parameters are defined as $\alpha = -g/l$ and $\beta = 1/(ml^2)$. The corresponding parameter values are listed in Table I. The ALN and ABS controllers are derived in sections II-B1 and II-B2, respectively.

1) Adaptive Lyapunov-based Nonlinear (ALN) Controller: The trajectory tracking and parameter estimation errors were defined as:

$$\begin{aligned} \dot{e}_d &= \dot{x}_1 - \dot{x}_{1d} = e_v, \\ \dot{e}_v &= \dot{x}_2 - \dot{x}_{2d} = \alpha\cos(x_1) + \beta u - \dot{x}_{2d}, \\ \tilde{\alpha} &= \alpha - \hat{\alpha}, \end{aligned} \tag{25}$$

where x_{1d} and x_{2d} denote the desired angular displacement and angular velocity trajectories, respectively, and $\hat{\alpha}$ is the estimate of the unknown parameter α.

Considering the Lyapunov function candidate

$$V = \frac{1}{2}\,\varepsilon^T P \varepsilon = \frac{1}{2}\begin{bmatrix} e_d & e_v & \tilde{\alpha} \end{bmatrix} \begin{bmatrix} p_1 & K & 0 \\ K & p_2 & 0 \\ 0 & 0 & p_3 \end{bmatrix} \begin{bmatrix} e_d \\ e_v \\ \tilde{\alpha} \end{bmatrix}, \tag{26}$$

and its derivative:

$$\begin{aligned} \dot{V} &= p_1 e_v e_d + K e_v^2 \\ &\quad + (K e_d + p_2 e_v)\,(\alpha\cos(x_1) + \beta u - \dot{x}_{2d}) \\ &\quad - p_3 \dot{\hat{\alpha}}\,\tilde{\alpha}, \end{aligned} \tag{27}$$

and by choosing the control law as:

$$u = -\hat{\alpha}\cos(x_1) + \dot{x}_{2d} - c_1 e_d - c_2 e_v, \tag{28}$$

the derivate of the Lyapunov function candidate becomes

$$\begin{aligned} \dot{V} &= (p_1 - p_2 c_1 - K c_2)\, e_v e_d - c_1 K e_d^2 \\ &\quad - (K - p_2 c_2)\, e_v^2 \\ &\quad + \tilde{\alpha}\Big(\,(K e_d + p_2 e_v)\cos(x_1) - p_3 \dot{\hat{\alpha}}\Big). \end{aligned} \tag{29}$$

Then by setting the adaptive law as:

$$\dot{\hat{\alpha}} = \frac{1}{p_3}\,(K e_d + p_2 e_v)\cos(x_1), \tag{30}$$

and assuming $p_1 = p_2 c_1 + K c_2$, the derivative of the Lyapunov function candidate reduces to

$$\dot{V} = -c_1 K e_d^2 - (p_2 c_2 - K)\, e_v^2. \tag{31}$$

By choosing $c_1 K > 0$, and $p_2 c_2 > K$, and applying the control input (28) together with the adaptive law (30), Barbalat's lemma guarantees that the velocity tracking error e_v, displacement tracking error e_d and parameter estimation error $\tilde{\alpha}$ remain bounded. Furthermore, as time tends to infinite, $\lim_{t\to\infty}(e_v) = 0$ and $\lim_{t\to\infty}(e_d) = 0$.

2) Adaptive Backstepping (ABS) Controller: A set of new state variables are defined for the ABS-based controller as:

$$\begin{aligned} z_1 &= x_1 - x_{1d} = e_d, \\ z_2 &= x_2 - x_{2d} - \varepsilon = e_v - \varepsilon. \end{aligned} \tag{32}$$

By choosing

$$\varepsilon = -c_1 z_1, \tag{33}$$

the state derivatives become

$$\begin{aligned} \dot{z}_1 &= e_v = z_2 + \varepsilon, \\ \dot{z}_2 &= \dot{e}_v - \dot{\varepsilon} = \alpha\cos(x_1) + \beta u - \dot{x}_{2d} + c_1 e_v. \end{aligned} \tag{34}$$

To cancel the nonlinearities in $\dot{z}_2$ and stabilise the system, the control input was designed as:

$$u = \frac{1}{\beta}\Big(-\hat{\alpha}\cos(x_1) - c_1 e_v - z_1 - c_2 z_2 + \dot{x}_{2d}\Big), \tag{35}$$

which makes

$$\dot{z}_2 = \tilde{\alpha}\cos(x_1) - z_1 - c_2 z_2, \tag{36}$$

where $\tilde{\alpha} = \alpha - \hat{\alpha}$ is the parameter estimation error.

Considering the Lyapunov function candidate:

$$V = \frac{1}{2} z_1^2 + \frac{1}{2} z_2^2 + \frac{1}{2}\Gamma\tilde{\alpha}^2, \tag{37}$$

where $\Gamma \in \mathbb{R}^+$ is the adaptation (learning) gain, and it derivative:

$$\begin{aligned} \dot{V} &= z_1\dot{z}_1 + z_2\dot{z}_2 - \tilde{\alpha}\Gamma\dot{\hat{\alpha}} \\ &= z_1(z_2 + c_1 z_1) + z_2(\tilde{\alpha}\cos(x_1) - z_1 - c_2 z_2) - \tilde{\alpha}\Gamma\dot{\hat{\alpha}} \\ &= -c_1 z_1^2 - c_2 z_2^2 + \tilde{\alpha}\Big(z_2\cos(x_1) - \Gamma\dot{\hat{\alpha}}\Big). \end{aligned} \tag{38}$$

By choosing the adaptive law:

$$\dot{\hat{\alpha}} = \Gamma^{-1}\, z_2\cos(x_1), \tag{39}$$

the derivative of the Lyapunov function candidate becomes:

$$\dot{V} = -c_1 z_1^2 - c_2 z_2^2 \leq 0. \tag{40}$$

According to Barbalat's lemma, for the system defined in (24), by applying the control law (35) and the adaptive law (39), where c_1, c_2 and Γ are positive definite constants, it can be established that z_1, z_2, and $\tilde{\alpha}$ are bounded, and that $\lim_{t\to\infty} z_1 = 0$ and $\lim_{t\to\infty} z_2 = 0$. Therefore, from equation (32) and (33), it can be concluded that $\lim_{t\to\infty} e_d = 0$ and $\lim_{t\to\infty} e_v = 0$.

TABLE II
CONTROLLERS' DESIGN PARAMETERS.

Controller	Parameters
ADP	$k = 0.01, \gamma_1 = 1,\ \gamma_2 = 1,$ $\Gamma_1 = 350,\ \Gamma_2 = \mathrm{diag}([10,\ 10,\ 10])$
ABS	$c_1 = c_2 = 40,$ $\Gamma = 350$
ALN	$c_1 = c_2 = 40,$ $K = p_2 = 1,\ p_3 = \frac{1}{350}$

III. RESULTS

This section compares the performance of the three controllers in terms of adaptive parameter convergence, quality of their control input, and trajectory tracking performance.

A. Chosen Design Parameters

The activation functions for the ADP controller $\phi(x, T-t)$ are usually designed based on engineering experience [14]. As the solution of the HJB equation is time-dependent, the activation function was designed based on the normalised time-to-go function $t_n = (T-t)/T$. The time-dependent term used in [14] was an exponentially decreasing function within the horizon length, $\phi(x, T-t) = \frac{1}{2}\left[e_1 t_n^5 \quad e_2 t_n^5 \quad e_1 e_2 t_n^5\right]$. When a similar activation function was implemented, it was found to be insufficient for stabilising the system, as the time-dependent term reached near-zero values within the time-horizon. This can be tackled by reducing the order of the time-dependent term. Alternatively, a new activation function was designed based on a squared sinusoidal function of time, with a period proportional to the horizon length:

$$\phi(x, T-t) = \frac{1}{2}\begin{bmatrix} e_1^2 \sin^2\left(\frac{\pi}{2} t_n\right) \\ e_2^2 \sin^2\left(\frac{\pi}{2} t_n\right) \\ e_1 e_2 \sin^2\left(\frac{\pi}{2} t_n\right) \end{bmatrix}. \quad (41)$$

For the ADP controller, the learning gains Γ_1 and Γ_2 are positive values that control the convergence of the adaptive laws (7) and (21), respectively. The forgetting factor parameters γ_1 and γ_2 must also be chosen as positive values, and they ensure the boundedness of the auxiliary variables (5) and (19), respectively. The LPF parameters k is inversely proportional to its cut-off frequency, and it must be chosen as a small value to avoid filtering essential information form the state variables. The tuning parameters for the three controllers were chosen heuristically in the simulation process. Table II provides a summary of the chosen parameters.

B. Adaptive Parameter Convergence

Figure 2 shows the identified parameters by the adaptive component of the three methods and the ground truth system parameters. The ADP-based method exhibits the smallest overshoot; however, its parameter convergence is slightly slower than that of the other two methods. This difference can be reduced by increasing the learning gain of the identifier Γ_1, but this may result in increased overshoot. The identified parameters slightly deviated from the constant ground-truth value between $t = 10\,\mathrm{s}$ and $t = 15\,\mathrm{s}$ as the manipulator transitioned to the second goal position, which is likely due to some unmodelled dynamics in the simulated manipulator.

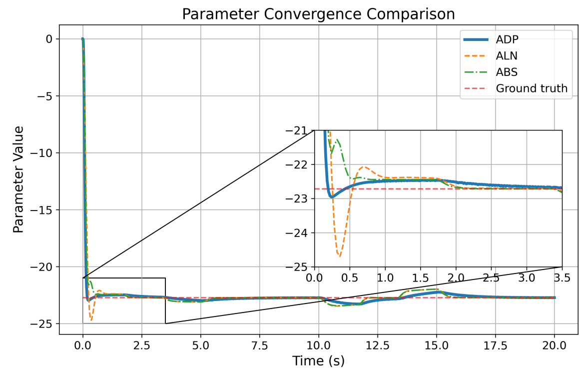


Fig. 2. Adaptive parameters convergence of the three controllers in comparison with the ground truth system parameter.

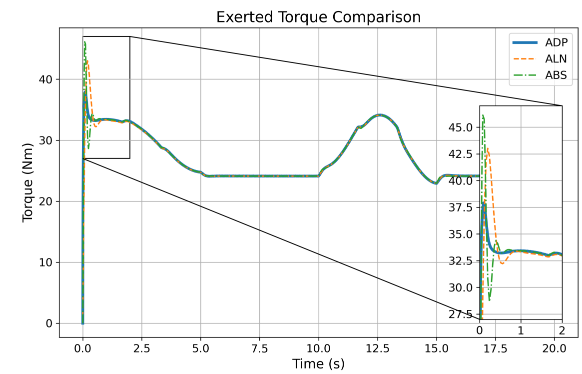


Fig. 3. Exerted torque by the three controllers.

C. Control Effort

Figure 3 shows the torque commands generated by the three controllers. The ADP-based method produced the smoothest control commands and the lowest torque magnitudes at any point, whereas the ABS controller was the most aggressive among the three, exhibiting the largest torque commands with the steepest gradients. The non-aggressive behaviour of the ADP controller is attributed to its approximately optimal performance with respect to the cost function defined in (2).

D. Tracking Performance

Figure 4 compares the trajectory tracking performance of the three controllers. While the three controllers were able to track the desired trajectory, the ADP and ABS controllers converged to the desired trajectory faster than the ALN controller. Furthermore, the ADP controller had the smallest overshoot between the three controllers, while the ABS controller experienced the largest overshoot from the velocity trajectory.

Table III provides a summary of the tracking root-mean squared error for the angular displacement $\theta_{RMSE}(rad)$ and

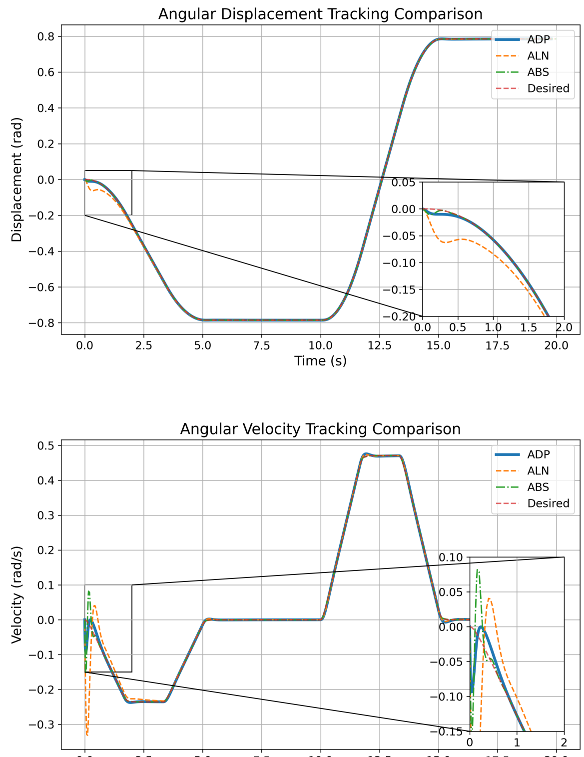


Fig. 4. Comparison of trajectory tracking performance of the three controllers. (Top) Angular displacement tracking performance. (Bottom) Angular velocity tracking performance.

TABLE III
COMPARISON BETWEEN THE CONTROLLERS' PERFORMANCE.

Controller	$\theta_{RMSE}(rad)$	$\dot{\theta}_{RMSE}(rad/s)$	$\tau_{max}(Nm)$
ADP	0.0013	0.0075	37.9026
ABS	0.0010	0.0147	46.1266
ALN	0.0098	0.0294	43.0126

angular velocity $\dot{\theta}_{RMSE}(rad/s)$ trajectories, and the maximum torque input $\tau_{max}(Nm)$. Overall, the results indicate satisfactory performance of the ADP-based controller for the manipulator, as it achieved the best tracking performance with the smallest torque requirement. The ABS controller, on the other hand, achieved better trajectory tracking performance than the ALN controller, at the expense of a more aggressive control commands.

IV. Conclusion

In conclusion, this paper presented an online approximate optimal tracking controller for a continuous-time nonlinear system with partially unknown dynamics. The controller utilised an Adaptive Dynamic Programming (ADP) algorithm, consisting of a dual-approximation identifier–critic neural network (NN) architecture that simultaneously identifies the unknown dynamics and estimates the solution to Hamilton-Jacobi-Bellman (HJB) equation. Furthermore, the algorithm provides a simple online validation method for the persistence of excitation (PE) condition that is necessary for the convergence of the control parameters. The proposed controller was validated in simulation on a single-link robotic manipulator. It was also compared with two classical adaptive control algorithms, namely an adaptive Lyapunov-based nonlinear (ALN) controller and an adaptive backstepping (ABS) controller. Simulation results validate the theoretical soundness and efficiency of the approximate optimal ADP controller under ideal conditions. Future work will focus on scaling this architecture to robotic manipulators with multiple-joints, and consider practical models with non-ideal system dynamics.

References

[1] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*. John Wiley & Sons, 2012.

[2] K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[3] J. Na, M. N. Mahyuddin, G. Herrmann, X. Ren, and P. Barber, "Robust adaptive finite-time parameter estimation and control for robotic systems," *Int. J. Robust. Nonlinear Control*, vol. 25, no. 16, pp. 3045–3071, 2015.

[4] A. Islam, A. Siming Chen, and G. Herrmann, "A q-learning algorithm to solve the two-player zero-sum game problem for nonlinear systems," *Int. J. Adapt. Control Signal Process.*, vol. 39, no. 3, pp. 566–581, 2025.

[5] D. Wang, H. He, and D. Liu, "Adaptive critic nonlinear robust control: A survey," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3429–3451, 2017.

[6] C. Chen, H. Modares, K. Xie, F. L. Lewis, Y. Wan, and S. Xie, "Reinforcement learning-based adaptive optimal exponential tracking control of linear systems with unknown dynamics," *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4423–4438, 2019.

[7] D. Vrabie and F. L. Lewis, "Adaptive optimal control algorithm for continuous-time nonlinear systems based on policy iteration," in *2008 47th IEEE Conf. on Decision and Control*. IEEE, 2008, pp. 73–79.

[8] K. Dupree, C.-H. Liang, G. Hu, and W. E. Dixon, "Adaptive lyapunov-based control of a robot and mass–spring system undergoing an impact collision," *IEEE Trans. Systems, Man, & Cybernetics, Part B (Cybernetics)*, vol. 38, no. 4, pp. 1050–1061, 2008.

[9] S. Farzan, V. Azimi, A.-P. Hu, and J. Rogers, "Adaptive control of wire-borne underactuated brachiating robots using control lyapunov and barrier functions," *IEEE Trans. on Control Sys. Technology*, vol. 30, no. 6, pp. 2598–2614, 2022.

[10] Q. Guo, Y. Zhang, B. G. Celler, and S. W. Su, "Neural adaptive backstepping control of a robotic manipulator with prescribed performance constraint," *IEEE trans. neural networks & learning systems*, vol. 30, no. 12, pp. 3572–3583, 2018.

[11] Z. Dachang, D. Baolin, Z. Puchen, and W. Wu, "Adaptive backstepping sliding mode control of trajectory tracking for robotic manipulators," *Complexity*, vol. 2020, no. 1, p. 3156787, 2020.

[12] C. Pan, H. Huang, C. Chen, L. Wu, and J. Xiao, "Neuro-adaptive command-filtered backstepping control of uncertain flexible-joint manipulators with input saturation via output," *Scientific Reports*, vol. 15, no. 1, p. 17254, 2025.

[13] J. Na and G. Herrmann, "Online adaptive approximate optimal tracking control with simplified dual approximation structure for continuous-time unknown nonlinear systems," *IEEE/CAA J. Autom. Sin.*, vol. 1, no. 4, pp. 412–422, 2014.

[14] J. Na, G. Li, B. Wang, G. Herrmann, and S. Zhan, "Robust optimal control of wave energy converters based on adaptive dynamic programming," *IEEE Trans. Sustain. Energy*, vol. 10, no. 2, pp. 961–970, 2018.

[15] P. Corke, "MATLAB toolboxes: Robotics and vision for students and teachers," *IEEE Robot. Autom. Mag.*, vol. 14, no. 4, pp. 16–17, 2008.

[16] P. Ioannou and B. Fidan, *Adaptive control tutorial*. SIAM, 2006.

[17] G. Herrmann, J. Na, and M. N. Mahyuddin, "Novel robust adaptive algorithms for estimation and control: Theory and practical examples," in *Control of Complex Systems*. Elsevier, 2016, pp. 661–709.