

Article

Nonzero-Sum Game-Based Modular Manipulator Optimal Tracking Control: Performance-Index Function Without Control Input

Bing Ma ¹, Zebin Ji ¹, Yi Qin ², Xinye Zhu ¹ and Tianjiao An ^{1,*} ¹ Department of Control Science and Engineering, Changchun University of Technology, Changchun 130012, China² Aviation University of Air Force, Changchun 130022, China

* Correspondence: antianjiao@ccut.edu.cn

Abstract

To address the issue that the performance-index function encompassing both dynamic and control processes in traditional adaptive dynamic programming (ADP) cannot be optimized as time approaches infinity, this paper proposes an optimal tracking control method for a modular manipulator based on a nonzero-sum game. The dynamic model of the modular manipulator is established using the Newton–Euler iterative method. By treating each module of the manipulator as a player in a nonzero-sum game and employing adaptive dynamic programming, the trajectory-tracking problem is transformed into an optimal-control problem. The critic neural network approximates the performance-index function without control input to derive the optimal tracking control policy. A stability theorem proves the stability of the closed-loop system, and an experimental platform validates the accuracy and optimality of the proposed method.

Keywords: adaptive dynamic programming; nonzero-sum game; modular manipulator; optimal control

1. Introduction

Robotic manipulators find broad applications across industrial manufacturing and medical services, among other domains, markedly improving the level of task automation [1,2]. Nevertheless, conventional robotic manipulators are often limited by structural rigidity, inadequate adaptability, and elevated cost. In response, modular manipulators [3,4] have been introduced. Modular manipulator systems consist of standardized, interchangeable modules that support swift reconfiguration and scalability based on operational needs. Such a design makes them highly promising for scenarios requiring substantial adaptability, such as aerospace operations, flexible manufacturing lines, and emergency response missions. Tracking constitutes a representative control task for robotic manipulators [5,6], distinguished by its dynamic nature, stringent accuracy requirement, and intricate environmental interaction. In essence, the manipulator needs to continuously and precisely steer its end-effector or joint states along a time-varying reference trajectory or moving target, while simultaneously accommodating load fluctuations, model uncertainties, and possible external disturbances. Hence, for tracking tasks performed by modular manipulators, it is essential to guarantee not only accuracy but also optimality.

Optimal control aims to devise control policies that fulfill task specifications while optimizing a given performance criterion [7,8]. However, when applied to complex non-



Academic Editor: Iyad Kanj

Received: 14 January 2026

Revised: 5 February 2026

Accepted: 6 February 2026

Published: 9 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

linear and uncertain systems, it encounters difficulties, including heavy reliance on an accurate model and considerable computational expense. Adaptive dynamic programming (ADP) [9–11] has emerged to mitigate these constraints. As a family of approximate dynamic programming techniques, ADP blends neural networks and reinforcement learning concepts to facilitate online learning and approximation of optimal-control policies without depending on an exact model. It is especially well suited for optimizing high-dimensional dynamic systems with intricate models or time-varying environments, such as robotic manipulators [12], smart power grids [13], and unmanned platforms [14]. Yet, traditional ADP formulations embed both dynamic and control terms within the performance index [15–17]. During trajectory-tracking control of a robotic manipulator, control torques remain bounded within a certain interval rather than approaching zero as in regulation control. As a result, the performance index grows unbounded as time approaches infinity, making optimization impossible. Therefore, solving the trajectory-tracking problem for robotic manipulators urgently calls for performance indices that exclude explicit control terms.

Differential game was pioneered in the 1950s by mathematicians, including Rufus Isaacs, who innovatively merged classical game theory with the dynamic system framework of control theory to investigate optimal strategic interactions among adversaries or collaborators in continuous time processes [18–20]. The central model of differential game employs a set of coupled differential equations to capture the state evolution of multiple players, under which Nash equilibrium strategies are derived. Consequently, the theory has been widely adopted in complex multi-agent continuous decision-making contexts, ranging from pursuit evasion games in military engagements and dynamic competition/advertising strategies in economics to ecosystem evolution studies and coordinated control of multi-agent robotic manipulator systems [21–23]. Nonzero-sum differential game lies at the forefront of interdisciplinary research bridging game theory and control theory [24–26]. It studies settings where multiple decision makers (players) operate in continuous dynamic systems, with objectives that are not strictly opposed, thereby allowing a blend of cooperative and competitive interactions. The system's state evolution is governed by several differential equations (state equations), and each participant applies independent control input to influence the state trajectory while aiming to maximize (or minimize) individual, yet interrelated, performance indices. In [24], a learning-based parallel control framework was developed for unknown nonaffine, nonzero-sum games. This was followed by [25], which introduced dynamic self-triggered control for unknown nonlinear constrained systems in a nonzero-sum game setting, utilizing generalized fuzzy hyperbolic models. Subsequently, ref. [26] presented a dynamic event-driven ADP method for N -player nonzero-sum games involving constrained nonlinear systems. It is noteworthy that the performance-index functions in all the aforementioned approaches incorporate a control torque term, which necessitates an explicit solution. In a modular manipulator, every module is equipped with its own control unit, enabling each module to be treated as a player in nonzero sum game. Control strategies are then optimized within the formal framework to attain the Nash equilibrium.

This paper introduces an optimal tracking control scheme for a modular manipulator grounded in nonzero-sum game theory, aiming to resolve the optimization challenge associated with the performance-index function, which includes both dynamic and control processes, in conventional adaptive dynamic programming as time tends toward infinity. The dynamic of the modular manipulator is modeled via the Newton–Euler iterative approach. Within this framework, each module is regarded as a participant in a nonzero-sum game. Leveraging adaptive dynamic programming, the trajectory-tracking task is recast as an optimal-control problem. The optimal tracking control policy is obtained by employing a critic neural network to approximate the performance-index function

in the absence of control input. The stability of the closed-loop system is rigorously demonstrated through the Lyapunov theorem, and experimental results from a dedicated platform confirm the method's accuracy and optimal performance.

This work contributes mainly to the following two aspects:

The proposed performance-index function distinguishes itself from conventional ADP formulations, which typically incorporate both system dynamic and control effort terms [27,28]. In contrast, the present performance index is constructed exclusively based on system state errors. This focused design offers significant advantages: it facilitates not only accurate trajectory tracking in position but also enables direct and precise velocity tracking. Consequently, this formulation inherently circumvents the unboundedness issue that plagues traditional performance indices over infinite horizons, a fundamental limitation that otherwise undermines the optimization process itself.

Within the proposed control architecture, each independent module of the modular manipulator is treated as a rational participant in a nonzero-sum game. The optimal-control strategy for the entire system is subsequently derived within an optimization framework that explicitly enforces Nash equilibrium conditions, ensuring a stable and balanced outcome for all interacting subsystems. The efficacy of this approach is rigorously validated through experimental tests conducted on a physical modular manipulator platform. These results conclusively demonstrate the algorithm's tracking accuracy and confirm its optimal performance in real-world applications.

2. Formulation of Dynamic Representation for Modular Manipulator

The following equation is derived from the Newton–Euler iterative method for modeling robot dynamics:

$$M_a(\theta)\ddot{\theta} + C_o(\theta, \dot{\theta})\dot{\theta} + G_r(\theta) = u, \quad (1)$$

where M_a, C_o, G_r denote the inertia matrix, the Coriolis and centrifugal matrix, and the gravitational torque matrix of the robotic system, respectively; $\theta, \dot{\theta}, \ddot{\theta}$ represent the position, velocity, and acceleration vectors of the robot link; u stands for the control torque.

The dynamic model of the modular manipulator system can be expressed as a set of independent joint subsystems with dynamic coupling terms. By isolating the local variables, each subsystem in the joint space can be represented as:

$$M_{ai}(\theta_i)\ddot{\theta}_i + C_{oi}(\theta_i, \dot{\theta}_i)\dot{\theta}_i + G_{ri}(\theta_i) + I_i(\theta, \dot{\theta}, \ddot{\theta}) = u_i, \quad (2)$$

where $\theta_i, \dot{\theta}_i, \ddot{\theta}_i, u_i$ are i th vectors of $\theta, \dot{\theta}, \ddot{\theta}, u$; M_{ai}, C_{oi}, G_{ri} are separation variables of M_a, C_o, G_r ; $I_i(\theta, \dot{\theta}, \ddot{\theta})$ denotes the dynamic coupling term due to interactions between the joint subsystems.

Assumption 1. M_{ai}, C_{oi}, G_{ri} are continuous as well as Lipschitz for local, and (1) is controllable.

Property 1 ([4]). The interactive coupling term for the i th subsystem can be defined as:

$$\begin{aligned} I_i(\theta, \dot{\theta}, \ddot{\theta}) = & \left(\sum_{k=1, k \neq i}^n [M_{a ii}(\theta) - M_{ai}(\theta_i)]\ddot{\theta}_i + M_{aik}(\theta)\ddot{\theta}_k \right) \\ & + \left(\sum_{k=1, k \neq i}^n [-C_{oi}(\theta_i, \dot{\theta}_i) + C_{o ii}(\theta, \dot{\theta})]\dot{\theta}_i + C_{o ik}(\theta, \dot{\theta})\dot{\theta}_k \right) \\ & + (-G_{ri}(\theta_i) + \overline{G}_{ri}(\theta)) \end{aligned} \quad (3)$$

where $\bar{G}_{ri}$ is the i th element of G_r ; M_{aik}, C_{oik} are i th row k th line of M_a, C_o , and with bounded satisfied as:

$$\|I_i(\theta, \dot{\theta}, \ddot{\theta})\| \leq \sum_{j=1}^n i_{ij} E_j, \quad (4)$$

where $i_{ij} \geq 0$ and $E_j = 1 + |\dot{e}_j| + |\dot{e}_j|^2$, $\dot{e}_j$ corresponds to the j th joint subsystem's velocity error.

As the modular manipulator system alters its configuration to meet varying task requirements, its dynamic matrices cannot be precisely determined. Therefore, the system uncertainties are explicitly extracted for subsequent treatment.

By reformulating, the dynamic model (1) becomes:

$$\begin{aligned} &(\hat{M}_{ai}(\theta_i) + \tilde{M}_{ai}(\theta_i))\ddot{\theta}_i + (\hat{C}_{oi}(\theta_i, \dot{\theta}_i) + \tilde{C}_{oi}(\theta_i, \dot{\theta}_i))\dot{\theta}_i \\ &+ (\hat{G}_{ri}(\theta_i) + \tilde{G}_{ri}(\theta_i)) + I_i(\theta, \dot{\theta}, \ddot{\theta}) = u_i. \end{aligned} \quad (5)$$

Extracting the uncertainties from the modular manipulator system yields the following reformulated dynamic equation:

$$\hat{M}_{ai}(\theta_i)\ddot{\theta}_i + \hat{C}_{oi}(\theta_i, \dot{\theta}_i)\dot{\theta}_i + \hat{G}_{ri}(\theta_i) + \psi_i = u_i, \quad (6)$$

where $M_{ai} = \hat{M}_{ai} + \tilde{M}_{ai}$, $C_{oi} = \hat{C}_{oi} + \tilde{C}_{oi}$, $G_{ri} = \hat{G}_{ri} + \tilde{G}_{ri}$, $\hat{M}_{ai}, \hat{C}_{oi}, \hat{G}_{ri}$ are the nominal value of M_{ai}, C_{oi}, G_{ri} , $\tilde{M}_{ai}, \tilde{C}_{oi}, \tilde{G}_{ri}$ are the estimated error of M_{ai}, C_{oi}, G_{ri} , $\psi_i = \tilde{M}_{ai}(\theta_i)\ddot{\theta}_i + \tilde{C}_{oi}(\theta_i, \dot{\theta}_i)\dot{\theta}_i + \tilde{G}_{ri}(\theta_i) + I_i(\theta, \dot{\theta}, \ddot{\theta})$ is the term that has not been precisely modeled.

Inspired by [29], the Newton–Euler iterative method can further stabilize the dynamic model of the modular manipulator system.

Define system state $x_i = [x_{i1}, x_{i2}]^T = [\theta_i, \dot{\theta}_i]^T$, then (6) can be converted to the state space equation as follows:

$$\begin{cases} \dot{x}_{i1} = x_{i2} \\ \dot{x}_{i2} = f_i(x_{i1}, x_{i2}) + g_i(x_{i1})u_i + \omega_i(x), \\ y_i = x_{i1} \end{cases} \quad (7)$$

where x is the global system state (to be defined in the subsequent section), while $f_i(x_{i1}, x_{i2})$, $g_i(x_{i1})$, $\omega_i(x)$ correspond to the nominal drift dynamic, nominal control input matrix, and unmodeled dynamic of the modular manipulator system, satisfying:

$$\begin{aligned} f_i(x_{i1}, x_{i2}) &= \hat{M}_{ai}^{-1}(\theta_i)(-\hat{C}_{oi}(\theta_i, \dot{\theta}_i)\dot{\theta}_i - \hat{G}_{ri}(\theta_i)) \\ g_i(x_{i1}) &= \hat{M}_{ai}^{-1}(\theta_i) \\ \omega_i(x) &= -\hat{M}_{ai}^{-1}(\theta_i)\psi_i. \end{aligned} \quad (8)$$

3. Optimal Tracking Control via Nonzero-Sum Game for Modular Manipulator

3.1. Derivation of the Nonzero-Sum Game

Given that each joint module of the modular manipulator system is actuated by a dedicated motor, each of its n -degree-of-freedom subsystems can be modeled as a player in a nonzero-sum game. Consequently, the system is considered with the following state space equation in a nonzero-sum game formulation:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x) + \sum_{m=1}^n G_m u_m + \omega(x), \\ y = x_1 \end{cases} \quad (9)$$

where $x = [x_1^T, x_2^T]^T \in R^{2n}$ are the global state of the modular manipulator. The vectors x_1, x_2 are, respectively, given by the following expressions:

$$\begin{aligned} x_1 &= [x_{11}, \dots, x_{i1}, \dots, x_{n1}]^T \in R^n, \\ x_2 &= [x_{12}, \dots, x_{i2}, \dots, x_{n2}]^T \in R^n. \end{aligned}$$

Moreover, the following equations can be established:

$$\begin{aligned} f(x) &= [f_1(x), \dots, f_i(x), \dots, f_n(x)]^T, \\ G_m &= [0, \dots, 0, g_m, 0, \dots, 0]^T, \\ \omega(x) &= [\omega_1(x), \dots, \omega_i(x), \dots, \omega_n(x)]^T, \\ m &= 1, 2, \dots, n. \end{aligned}$$

Consider the following performance-index function:

$$\begin{aligned} J_i^* &= \min J_i(\dot{e}) = \min \int_t^\infty (\dot{e}^T R_i \dot{e} + \ddot{e}^T Q_i \ddot{e} + W^T W) d\tau \\ &= \int_t^\infty U_{li}(\dot{e}, \ddot{e}) d\tau, \end{aligned} \quad (10)$$

in which x_{1d} corresponds to the desired position, $e = [e_1, \dots, e_i, \dots, e_n]^T = x_1 - x_{1d}$ to the position tracking error, and the velocity tracking error is expressed as $\dot{e} = [\dot{e}_1, \dots, \dot{e}_i, \dots, \dot{e}_n]^T = x_2 - \dot{x}_{1d}$, Q_i, R_i are positive definite matrix, $U_{li}(\dot{e}, \ddot{e})$ is utility function, W defines the upper bound of $\omega(x)$.

This paper presents an optimal-control scheme based on a nonzero-sum game framework, designed to ensure that the modular manipulator accomplishes trajectory tracking while achieving the minimization of every module's performance index under Nash equilibrium conditions. To support controller design, the n modules are accordingly treated as n independent players.

Leveraging the angular acceleration estimation algorithm, the derivative of joint tracking error can be acquired directly from angular velocity measurements. This approach not only mitigates noise typically introduced by differentiators but also enhances real-time control effectiveness for systems with stringent timing requirements, such as modular manipulator.

This study decomposes the optimal-control input for trajectory tracking into two parts: a feedforward component and a feedback component. The feedforward term serves as a steady-state element that guarantees exact tracking of the reference trajectory, whereas the feedback term is obtained by minimizing a prescribed performance-index function. Consequently, conventional cost criteria inevitably retain a persistent steady-state quadratic term. Its integration over an infinite horizon therefore diverges, which undermines the validity of the optimization. In contrast, the performance criterion introduced in this work includes only the error-related term and excludes the control input term. Such a formulation effectively resolves the issue of the traditional quadratic performance-index

function, which incorporates both state and control variables, becoming unbounded as time tends to infinity, thereby preventing the associated failure of the optimization process.

Remark 1. *In contrast to traditional performance indices, which account for both control effort and dynamic state, the performance-index function introduced herein is composed solely of an error term related to the system state. This design circumvents the issue of the performance index growing unbounded over an infinite time horizon, which would render the optimization problem intractable.*

Applying the infinite horizon operator to (10), the associated Hamiltonian function is defined as:

$$H_{ai}(\dot{e}, \ddot{e}, J_i) = U_{li}(\dot{e}, \ddot{e}) + \left(\nabla J_i^T \right) \left(f(x) + \sum_{m=1}^n G_m u_m - \ddot{x}_{1d} \right), \quad (11)$$

where $\nabla J_i = \frac{\partial J_i}{\partial \dot{e}}$ denotes the partial derivative of the performance-index function J_i with respect to $\dot{e}$. The optimal performance-index function of the system is defined as:

$$J_i^* = \min_{u_i} \int_t^\infty (U_{li}(\dot{e}, \ddot{e})) d\tau. \quad (12)$$

In accordance with the optimality condition $\frac{\partial H_{ai}}{\partial u_i} = 0$, the optimal-control strategy yields:

$$u_i^*(\dot{e}) = -\frac{1}{2} \left(G_i^T R_i G_i \right)^{-1} \left(2G_i^T R_i (f - \ddot{x}_{1d}) + G_i^T \nabla J_i^* \right). \quad (13)$$

Assumption 2 ([30,31]). *The optimal control is dependent of G_i and $G_i^T R_i G_i$ is invertibility.*

Subsequently, substituting (10) and (13) into the Hamiltonian function (11) yields the coupled Hamilton–Jacobi (HJ) equation:

$$0 = \dot{e}^T R_i \dot{e} + \ddot{e}^T Q_i \ddot{e} + W^T W + \left(\nabla J_i^{*T} \right) \left(f(x) + \sum_{m=1}^n G_m u_m^* - \ddot{x}_{1d} \right) \quad (14)$$

3.2. Approximate Solution to the Nonzero-Sum Game

The aforementioned HJ equation is a nonlinear partial differential equation whose analytic solution is generally intractable. To address this, the critic neural network is utilized herein to approximate the performance-index function, expressed as:

$$J_i(\dot{e}) = W_{fi}^T \delta_{fi}(\dot{e}) + \varepsilon_{fi}, \quad (15)$$

where W_{fi} , $\delta_{fi}(\dot{e})$, and ε_{fi} are, respectively, the ideal neural network weight vector, the activation function, and the bounded approximation error. Differentiating (15) with respect to $\dot{e}$ gives:

$$\nabla J(\dot{e}) = \left(\nabla \delta_{fi}(\dot{e}) \right)^T W_{fi} + \nabla \varepsilon_{fi}, \quad (16)$$

where $\nabla \delta_{fi}(\dot{e}) = \frac{\partial(\delta_{fi}(\dot{e}))}{\partial \dot{e}}$ and $\nabla \varepsilon_{fi}$, respectively, denote the partial derivative of the activation function and that of the approximation error.

From the combination of (13) and (16), it follows that:

$$u_i^*(\dot{e}) = -\frac{1}{2} \left(G_i^T R_i G_i \right)^{-1} \left(2G_i^T R_i (f - \ddot{x}_{1d}) + G_i^T \left(\left(\nabla \delta_{fi}(\dot{e}) \right)^T W_{fi} + \nabla \varepsilon_{fi} \right) \right). \quad (17)$$

Plugging (17) into (11), we obtain:

$$\begin{aligned} H_{ai}(\dot{e}, \ddot{e}, J_i) &= U_{li}(\dot{e}, \ddot{e}) + \left(\nabla J_i^T \right) \\ &\left(\begin{array}{c} f(x) + \sum_{m=1}^n G_m \left(-\frac{1}{2} \left(G_i^T R_i G_i \right)^{-1} \right. \\ \left. \left(2G_m^T R_m (f - \ddot{x}_{md}) + G_m^T \left(\left(\nabla \delta_{fm}(\dot{e}) \right)^T W_{fm} + \nabla \varepsilon_{mi} \right) \right) \right) \\ - \ddot{x}_{1d} \\ - e_{Jci} = 0, \end{array} \right) \end{aligned} \quad (18)$$

where e_{Jci} denotes the approximation residual of the neural network.

Since the ideal neural network weight vector W_{fi} is unknown, the performance-index function can be approximated as:

$$\hat{J}_i(\dot{e}) = \hat{W}_{fi}^T \delta_{fi}(\dot{e}), \quad (19)$$

where $\hat{W}_{fi}$, $\hat{J}_i(\dot{e})$ are the estimated value of W_{fi} , $J_i(\dot{e})$.

The partial derivative of $\hat{J}_i(\dot{e})$ is obtained as follows:

$$\nabla \hat{J}_i(\dot{e}) = \left(\nabla \delta_{fi}(\dot{e}) \right)^T \hat{W}_{fi}. \quad (20)$$

Integrating (17) and (19), the approximate optimal control is derived as:

$$\begin{aligned} \hat{u}_i(\dot{e}) &= -\frac{1}{2} \left(G_i^T R_i G_i \right)^{-1} \\ &\left(2G_i^T R_i (f - \ddot{x}_{1d}) + G_i^T \left(\left(\nabla \delta_{fi}(\dot{e}) \right)^T \hat{W}_{fi} \right) \right). \end{aligned} \quad (21)$$

Consequently, based on (20), the approximated Hamiltonian equation is obtained in the following form:

$$\begin{aligned} \hat{H}_{ai}(\dot{e}, \ddot{e}, \hat{J}_i) &= U_{li}(\dot{e}, \ddot{e}) + \left(\nabla \hat{J}_i^T \right) \\ &\left(\begin{array}{c} f(x) + \sum_{m=1}^n G_m \left(-\frac{1}{2} \left(G_i^T R_i G_i \right)^{-1} \right. \\ \left. \left(2G_m^T R_m (f - \ddot{x}_{md}) + G_m^T \left(\left(\nabla \delta_{fm}(\dot{e}) \right)^T \hat{W}_{fm} \right) \right) \right) \\ - \ddot{x}_{1d} \\ - e_{Ji} = 0, \end{array} \right) \end{aligned} \quad (22)$$

Define the approximation error for the Hamiltonian function as:

$$e_{Ji} = \hat{H}_{ai} - H_{ai}, \quad (23)$$

where $e_{Ji} = \hat{H}_{ai}$ is obtained from (18) and (23).

The weight approximation error is introduced as:

$$\tilde{W}_{fi} = W_{fi} - \hat{W}_{fi}. \quad (24)$$

Based on (18), (22), and (23), we can derive the following:

$$e_{Ji} = e_{Jci} - \tilde{W}_{fi}^T \nabla \delta_{fi}(\dot{e}) \ddot{e}. \quad (25)$$

The update law for the neural network weight is defined as:

$$\dot{\hat{W}}_{fi} = -\alpha_{li} e_{Ji} \nabla \delta_{fi}(\dot{e}) \ddot{e}, \quad (26)$$

where α_{li} denotes the learning law for the neural network weight vector. Define $v_{ai} = \nabla \delta_{fi}(\dot{e}) \ddot{e}$ and assuming there exists a positive constant v_{Li} satisfying $\|v_{ai}\| \leq v_{Li}$, it can be derived that the dynamic equation for the weight approximation error is:

$$\begin{aligned} \dot{\tilde{W}}_{fi} &= -\dot{\hat{W}}_{fi} = \alpha_{li} e_{Ji} v_{ai} \\ &= \alpha_{li} (e_{Jci} - \tilde{W}_{fi}^T v_{ai}) v_{ai}. \end{aligned} \quad (27)$$

Remark 2. Regulation is a special issue of the tracking problem for the desired trajectory from the origin. In addressing the tracking control problem, the optimal-control input consists of two distinct elements: a feedforward component and a feedback component. The feedforward part serves as a steady-state term that achieves exact tracking of the desired trajectory, whereas the feedback component is obtained through minimization of a specified performance-index function. As a consequence, standard performance-index formulations inherently retain a non-removable steady state quadratic term. This leads to an unbounded integral over an infinite time horizon, which ultimately invalidates the optimization objective. In contrast to conventional formulations, the performance-index function introduced in this work includes only the error term and deliberately excludes any control input term. Such a structure successfully resolves the unboundedness issue encountered by classical quadratic performance-index functions, which incorporate both state and control variables, as time tends to infinity, thereby preventing the associated failure of the optimization procedure.

Remark 3. The convergence of the critic neural network relies on the condition of persistent excitation, which is inherently satisfied in trajectory-tracking control tasks. The approximation error is with boundedness.

3.3. Proof of Stability

Theorem 1. Given the subsystem dynamic (1) expressed in the state space form (9) for the modular manipulator, the application of the approximate optimal tracking control law (21), formulated via the nonzero-sum game approach, yields a closed-loop system whose tracking error is uniformly ultimately bounded (UUB).

Proof. Define Lyapunov function as $J_i(\dot{e}) = V_{ci}(t)$, and differentiating $V_{ci}(t)$ with respect to time leads to:

$$\dot{V}_{ci}(t) = \left(\nabla J_i^T \right) \left(f(x) + \sum_{m=1}^n G_m u_m + \varpi(x) - \ddot{x}_{1d} \right). \quad (28)$$

By considering the system's coupled HJ Equation (14), it can be obtained that:

$$\begin{aligned} (\nabla J_i^T)(f(x) - \dot{x}_{1d}) &= -\dot{e}^T R_i \dot{e} \\ &- \ddot{e}^T Q_i \ddot{e} - W^T W - \left(\nabla J_i^T \right) \sum_{m=1}^n G_m u_m \end{aligned} \quad (29)$$

Substituting (29) into (28) yields:

$$\begin{aligned} \dot{V}_{ci}(t) &= -\dot{e}^T R_i \dot{e} - \ddot{e}^T Q_i \ddot{e} - W^T W \\ &- (\nabla J_i^T) \left(\sum_{m=1}^n G_m (u_m^* - \hat{u}_m) \right). \end{aligned} \quad (30)$$

From the combination of (28) and (30), it follows that:

$$\begin{aligned} \dot{V}_{ci}(t) &= -\dot{e}^T R_i \dot{e} - \ddot{e}^T Q_i \ddot{e} - W^T W - \hat{W}_{fi}^T \delta_{fi}(\dot{e}) \\ &\left(\sum_{m=1}^n G_m \left(\begin{aligned} &\left(-\frac{1}{2} (G_i^T R_i G_i)^{-1} \right. \\ &\left. \left(2G_i^T R_i (f - \dot{x}_{1d}) + G_i^T \nabla J_i^* \right) \right) \right. \right. \\ &\left. \left. - \left(\begin{aligned} &-\frac{1}{2} (G_i^T R_i G_i)^{-1} \\ &\left(2G_i^T R_i (f - \dot{x}_{1d}) + \right. \\ &\left. G_i^T \left((\nabla \delta_{fi}(\dot{e}))^T \hat{W}_{fi} \right) \right) \right) \right) \right) \right) \\ &= -\dot{e}^T R_i \dot{e} - \ddot{e}^T Q_i \ddot{e} - W^T W + \hat{W}_{fi}^T \delta_{fi}(\dot{e}) \\ &\left(\sum_{m=1}^n G_m \frac{1}{2} (G_i^T R_i G_i)^{-1} \left(\begin{aligned} &2G_i^T R_i (f - \dot{x}_{1d}) + G_i^T \\ &\left(\nabla J_i^* - (\nabla \delta_{fi}(\dot{e}))^T \hat{W}_{fi} \right) \right) \right) \right) \\ &= -\dot{e}^T R_i \dot{e} - \ddot{e}^T Q_i \ddot{e} - W^T W + \Pi_J, \end{aligned} \quad (31)$$

where upper bound of Π_J is given by:

$$\begin{aligned} \Pi_J &\leq \left\| \begin{aligned} &\hat{W}_{fi}^T \delta_{fi}(\dot{e}) \\ &\left(\sum_{m=1}^n G_m \frac{1}{2} (G_i^T R_i G_i)^{-1} \right. \\ &\left. \left(\begin{aligned} &2G_i^T R_i (f - \dot{x}_{1d}) \\ &+ G_i^T \left(\nabla J_i^* - (\nabla \delta_{fi}(\dot{e}))^T \hat{W}_{fi} \right) \right) \right) \right) \end{aligned} \right\| \\ &\leq \pi_J, \end{aligned} \quad (32)$$

where π_J is a computable positive constant.

When $\pi_J \leq \dot{e}^T R_i \dot{e} + \ddot{e}^T Q_i \ddot{e} + W^T W$, (28) remains negative definite, with the existence of $\dot{V}_{ci}(t) < 0$. Thus, the proof is concluded. $\square$

Remark 4. In practical robotic applications, the control input, equivalent to the actuator output, i.e., the motor's torque, is physically limited and generally does not exceed 5Nm, as indicated in (32). Here, the parameter π_J can be treated as a positive constant that can be determined.

Following the policy iteration algorithm, convergence is declared once the iterative updates to the performance-index function become sufficiently small, i.e., when the difference between consecutive iterates drops below a defined constant.

Remark 5. The performance-index function of conventional ADP is $J_i^* = \min_{u_i} \int_t^\infty (\dot{e}^T Q \dot{e} + u^T R u) d\tau$. Apart from the performance index, there is no essential difference in the tuning parameters or implementation specifics.

4. Experiments

Experiment Setup

Experimental validation of the derived optimal-control policy is carried out on a two DOF modular manipulator testbed, with the platform configuration detailed in Figure 1. Joint control torques are measured by torque sensors, and joint positions are determined using a combination of absolute and incremental encoders. Communication between the Simulink 2016a software environment and the physical robot modules are realized through a QPIDe data acquisition card (Ottawa, Ontario, Canada). As the developed control algorithm is continuous-time in form, it is discretized for real-time experimentation. The sampling time of the control system is set within Simulink. This section employs the manipulator trajectory-tracking experiment, where the desired joint trajectory is specified as:

$$x_{1d} = \frac{1}{6}\pi \sin\left(\frac{1}{45}\pi\right)t + \frac{1}{12}\pi \sin\left(\frac{1}{45}\pi\right)t,$$

$$x_{2d} = -\frac{1}{9}\pi \sin\left(\frac{1}{50}\pi\right)t - \frac{1}{12}\pi \sin\left(\frac{1}{45}\pi\right)t.$$

The conventional ADP optimal control [32,33] and the proposed control strategy are both tested for comparison.

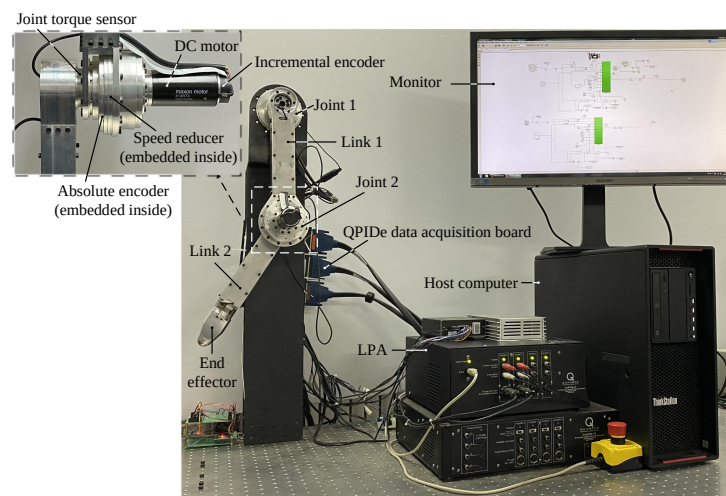


Figure 1. Experimental platform setup.

Position tracking profiles obtained with the baseline method and the proposed method are shown in Figures 2 and 3, respectively. The actual trajectory closely matches the desired reference in both cases, confirming that each controller successfully performs the tracking task. Notably, the initial joint positions align with the starting point of the reference trajectory, a result achieved by an initialization routine. Figures 4 and 5 compare the position error curves of the conventional and proposed methods. The proposed approach clearly yields smaller tracking errors, a benefit derived from the ADP-based optimization of position error. Velocity error curves for the conventional and proposed methods are provided in Figures 6 and 7. Because the performance index explicitly includes the velocity error term, the velocity error is directly optimized, maintaining it within a narrow bound in the steady state.

The control torque produced by the baseline method in Figure 8 exhibits considerable chattering, which could reduce motor lifespan and compromise tracking accuracy. Moreover, sharp torque variations appear at trajectory turning points, risking the activation of safety protection and consequent motor disengagement. By contrast, the proposed method generates a smooth and compliant torque profile, as seen in Figure 9, where the motor output is effectively optimized by the control algorithm. The torque also stays within safe limits at inflection points, owing to the newly designed performance index that optimizes control action.

Finally, Figure 10 displays the weight estimation of the critic neural network under the proposed control scheme. The five curves represent five neurons. The evolving weights capture the solution of the Hamilton–Jacobi equation in real time, illustrating the adaptive learning process inherent to the method.

The root mean square value is shown in Table 1.

Table 1. Root mean square value.

	Joint I	Joint II
Position tracking error (existed method)	1.76×10^{-3} rad	2.03×10^{-3} rad
Position tracking error (proposed method)	0.99×10^{-3} rad	1.37×10^{-3} rad
Velocity tracking error (existed method)	1.96×10^{-3} rad/s	2.01×10^{-3} rad/s
Velocity tracking error (proposed method)	0.35×10^{-3} rad/s	0.89×10^{-3} rad/s
Control torque (existed method)	0.39 Nm	0.21 Nm
Control torque (proposed method)	0.30 Nm	0.17 Nm

A series of comparative experiments was conducted, evaluating the proposed approach against related algorithms, including linear quadratic regulator and model predictive control. The results demonstrate that the proposed method achieves high tracking accuracy, strong robustness, and high computational efficiency.

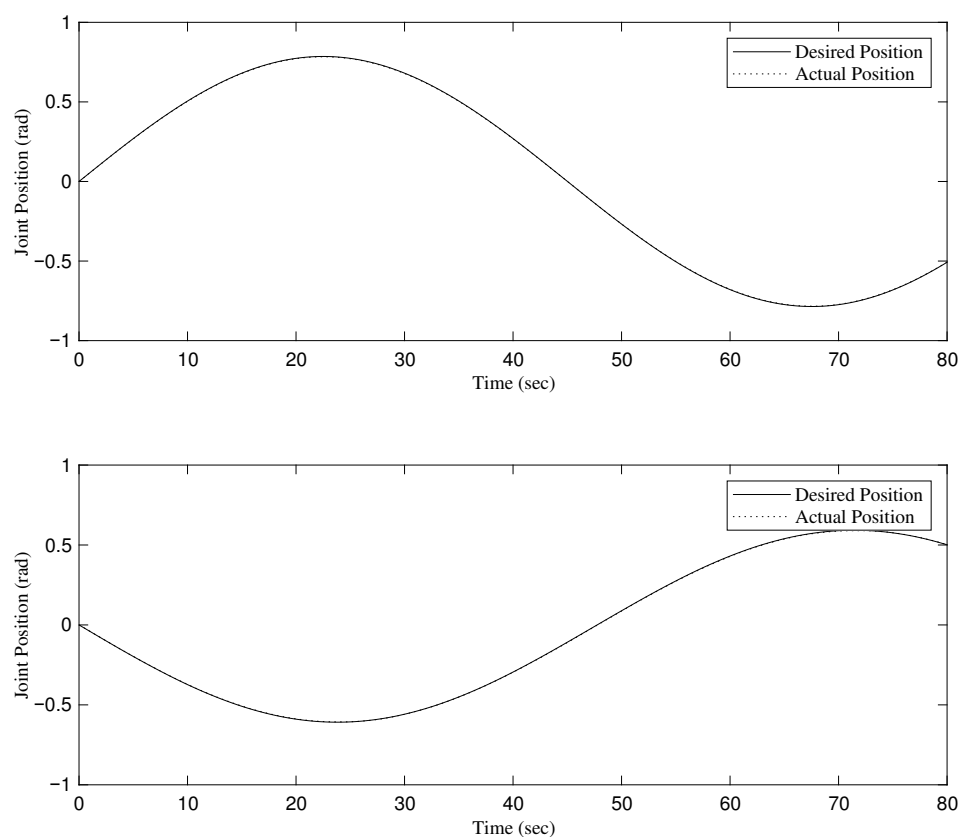


Figure 2. Position tracking under conventional ADP optimal control.

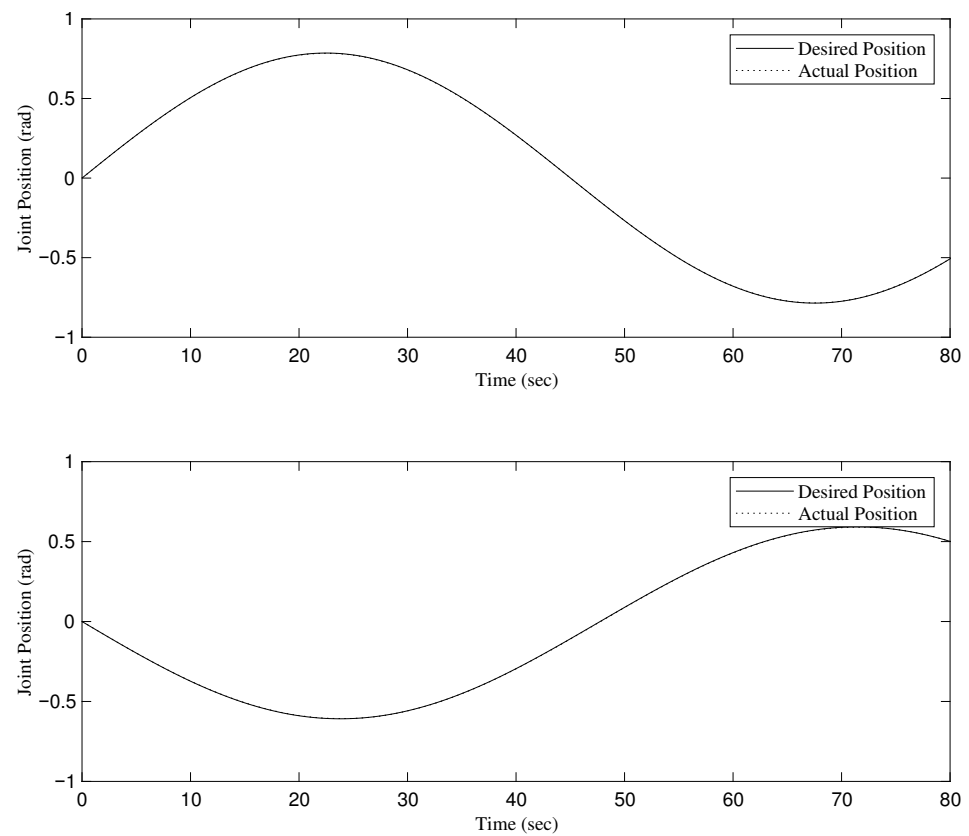


Figure 3. Position tracking under proposed method.

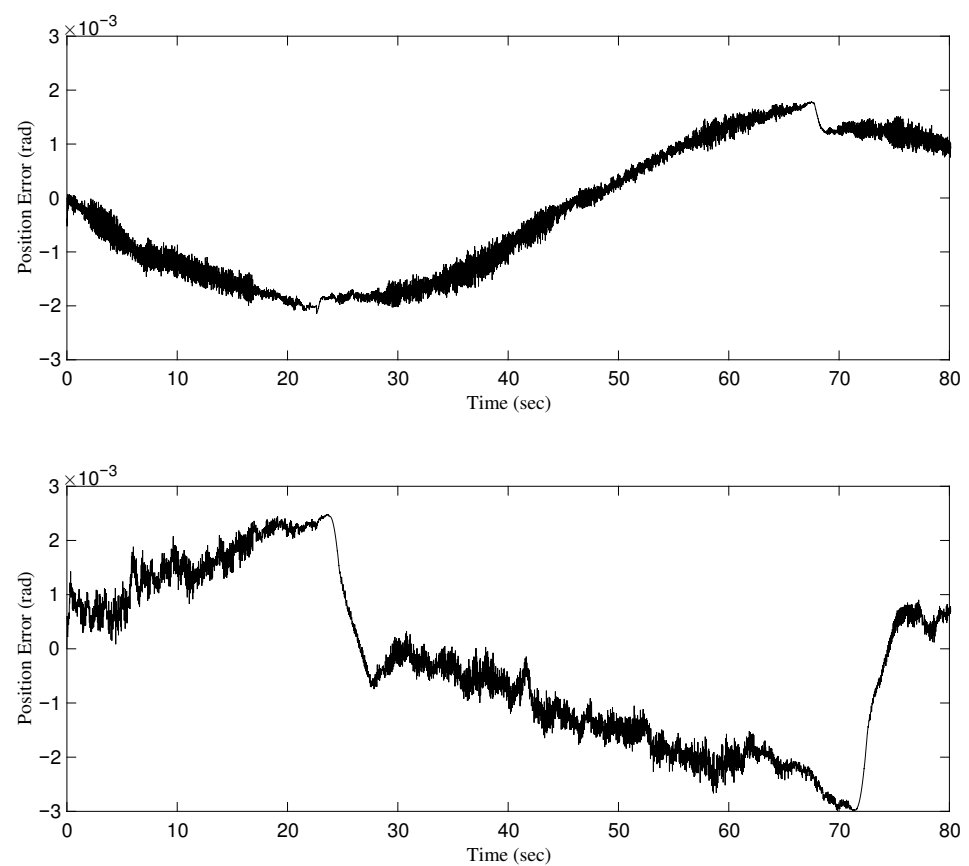


Figure 4. Position error under conventional ADP optimal control.

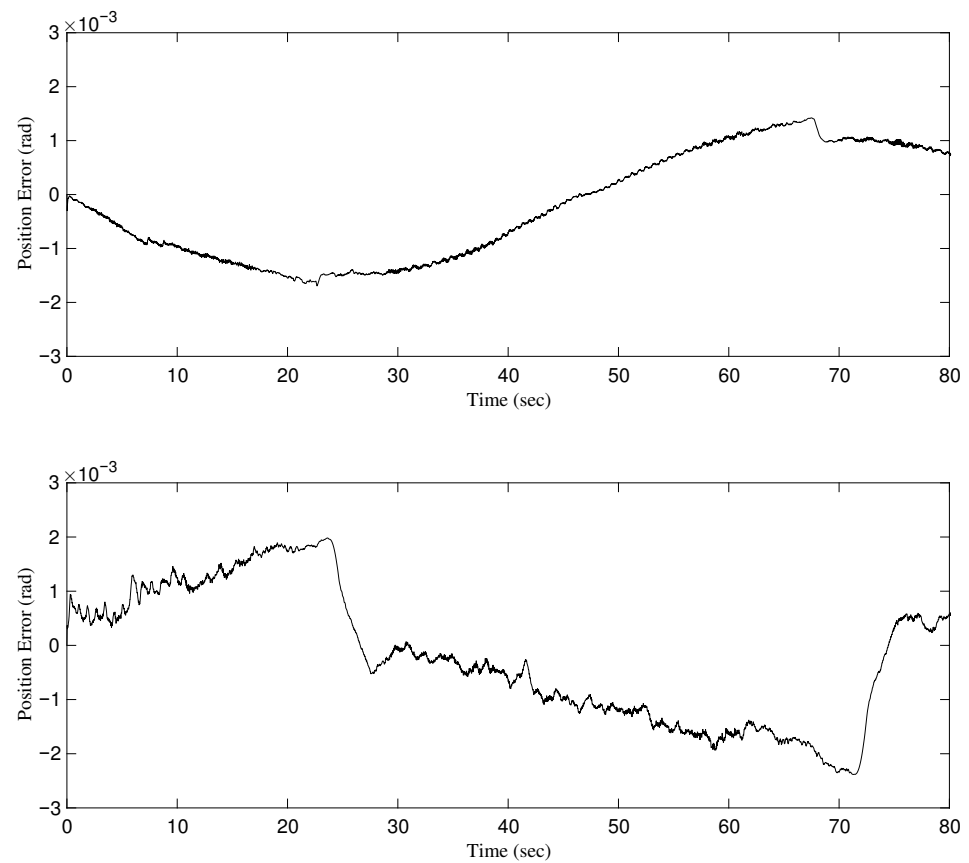


Figure 5. Position error under proposed method.

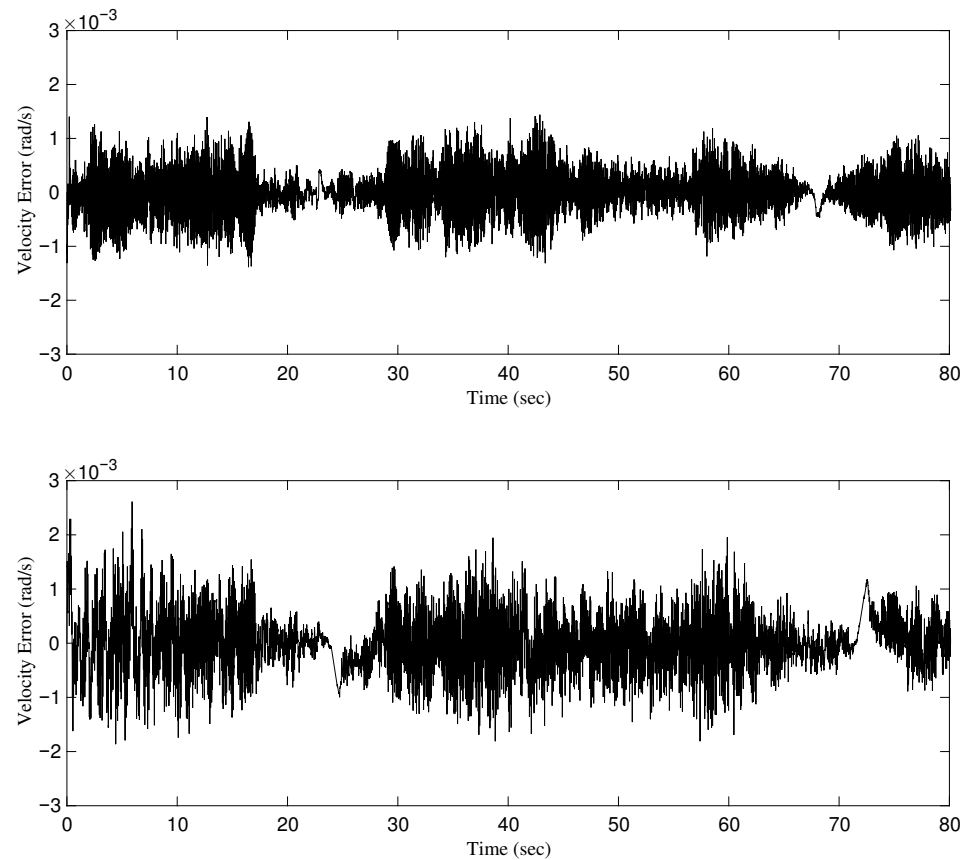


Figure 6. Velocity error under conventional ADP optimal control.

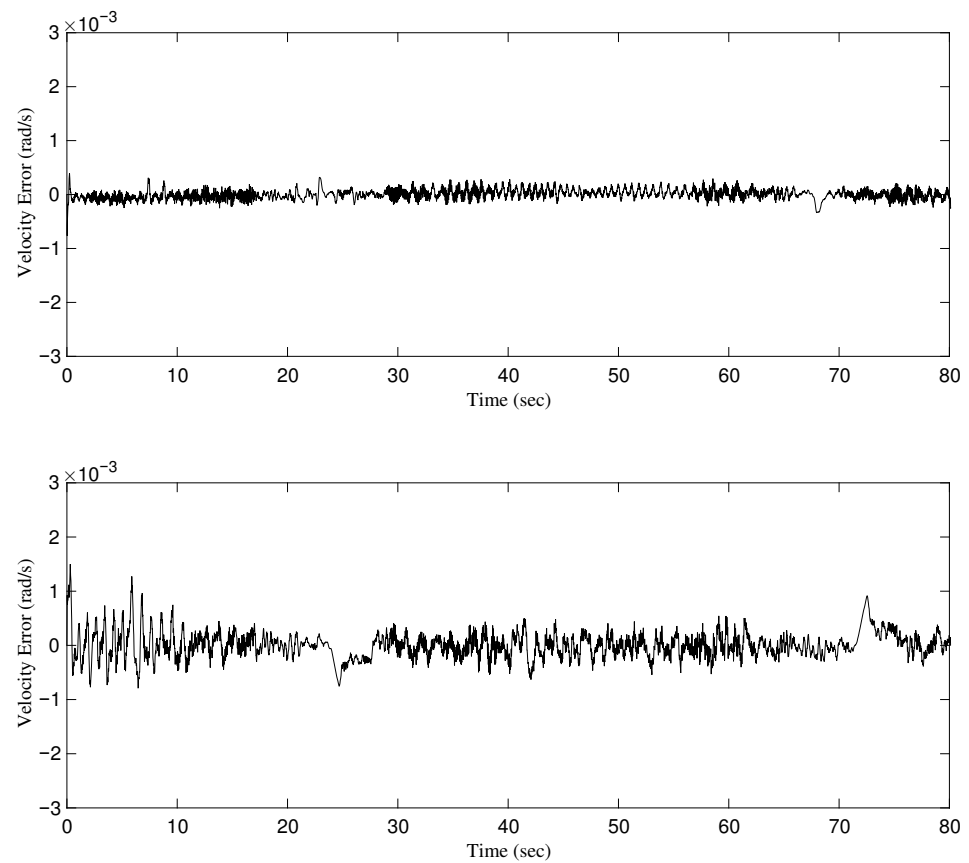


Figure 7. Velocity error under proposed method.

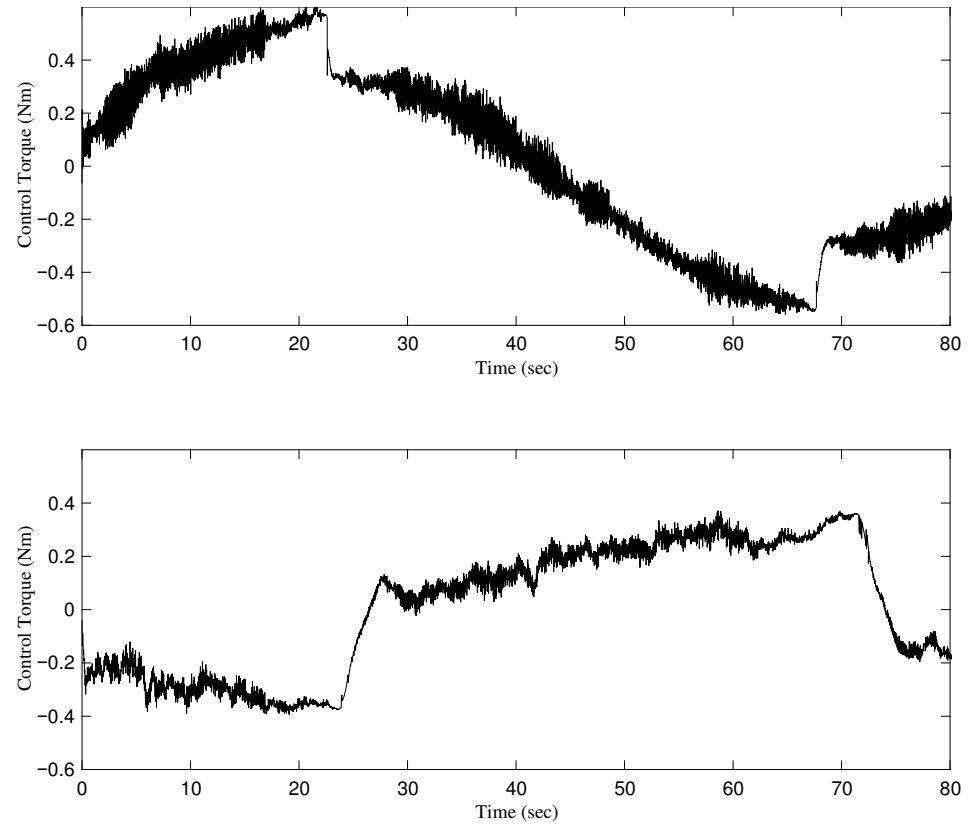


Figure 8. Control torque under conventional ADP optimal control.

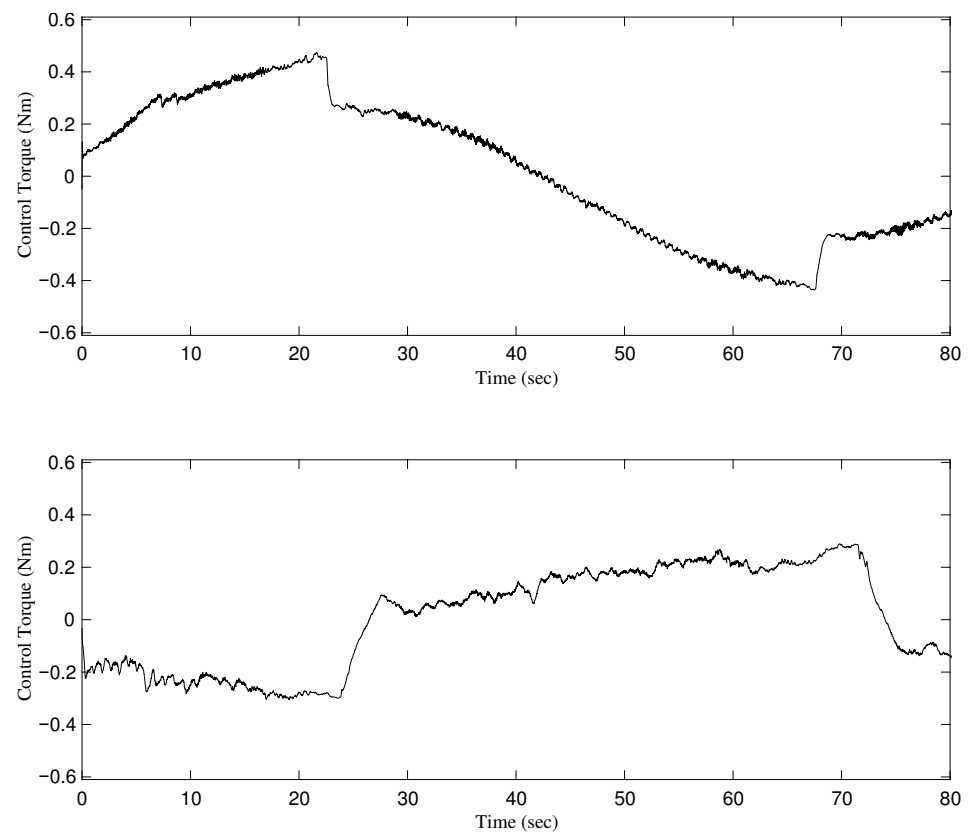


Figure 9. Control torque under proposed method.

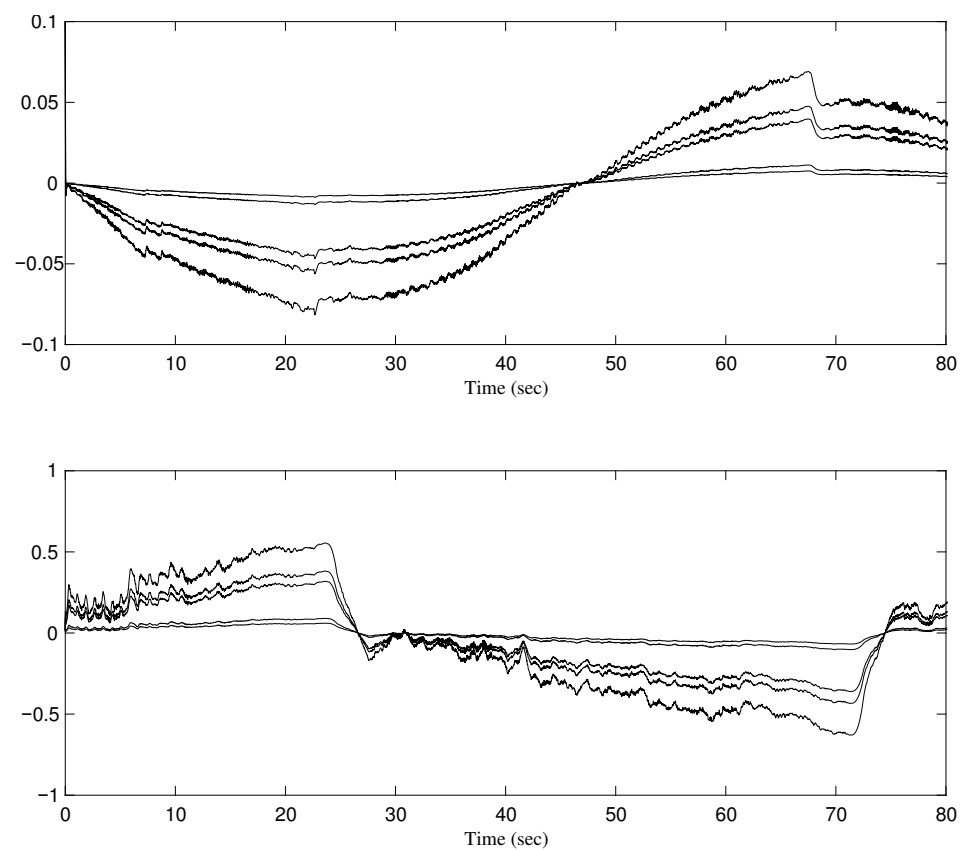


Figure 10. Critic weight under proposed method.

5. Conclusions

To overcome the limitation in optimizing the infinite performance-index function that integrates dynamic and control processes within traditional adaptive dynamic programming over an infinite time horizon, a novel optimal tracking control approach for a modular manipulator is presented, which employs nonzero-sum game theory. The manipulator's dynamic model is derived using the Newton–Euler iteration. In this method, each modular unit is considered an independent player in a nonzero-sum game. Through the application of adaptive dynamic programming, the tracking control problem is translated into an optimal-control formulation. Approximation of the performance-index function without control input is achieved via a critic neural network, yielding the optimal tracking control strategy. The closed-loop system's stability is established by a dedicated theorem, while the efficacy and optimality of the proposed strategy are verified through experimental tests on a physical platform. Building on the insight from [34], we will explore the incorporation of a feedforward compensator within a predictive control framework as a future research direction. Moreover, designed for dual position-velocity tracking, the proposed method is not directly applicable to systems where stabilization is the key concern, as is the case with a flexible robot. Consequently, future work will need to adapt the methodology accordingly.

Author Contributions: Methodology, Z.J.; Software, Y.Q.; Validation, X.Z.; Writing—original draft, B.M.; Writing—review and editing, T.A. All authors have read and agreed to the published version of the manuscript.

Funding: The work is supported by the Scientific Technological Development Plan Project in Jilin Province of China (20260602029RC).

Data Availability Statement: Some or all data, models, or code generated or used during the study are available from the corresponding author by request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kermenov, R.; Biase, A.D.; Pellicani, I.; Longhi, S.; Bonci, A. Near Time-Optimal Trajectories with ISO Standard Constraints for Human–Robot Collaboration in Fabric Co-Transportation. *Robotics* **2025**, *14*, 10. [\[CrossRef\]](#)
2. Enderica, C.S.; Llata, J.R.; Torre-Ferrero, C. Guided Reinforcement Learning with Twin Delayed Deep Deterministic Policy Gradient for a Rotary Flexible-Link System. *Robotics* **2025**, *14*, 76. [\[CrossRef\]](#)
3. An, T.; Dong, X.; Dong, B.; Jiang, H.; Liu, L.; Ma, B. Event-Triggered Mixed Nonzero-Sum Game Optimal Control for Modular Robotic Manipulator Performing Coordinated Operation Tasks. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 20371–20385. [\[CrossRef\]](#)
4. Zhao, B.; Zhang, Y.; Liu, D. Adaptive Dynamic Programming-Based Cooperative Motion/Force Control for Modular Reconfigurable Manipulators: A Joint Task Assignment Approach. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 10944–10954. [\[CrossRef\]](#)
5. Zhang, N.; Zhang, F.; Wang, J.; Li, Y.; Yang, C.; Wang, C.; Li, K. Precision Trajectory Tracking of Robot Manipulator Using a Discrete-Time Learning-Based Neural Network Control with Prescribed Performance. *IEEE Trans. Ind. Electron.* **2025**, *72*, 10640–10650. [\[CrossRef\]](#)
6. Jiang, Y.; Wang, Y.; Miao, Z.; Na, J.; Zhao, Z.; Yang, C. Composite-Learning-Based Adaptive Neural Control for Dual-Arm Robots with Relative Motion. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *33*, 1010–1021. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Xia, H.; Mu, C.; Sun, C. Output-Feedback Safe Tracking Control for Nonlinear Systems with Sensor Faults via Adaptive Critic Learning. *IEEE Trans. Syst. Man Cybern. Syst.* **2025**, *55*, 6975–6985. [\[CrossRef\]](#)
8. Yan, J.; Markovsky, I.; Lygeros, J. Secure Data Reconstruction: A Direct Data-Driven Approach. *IEEE Trans. Autom. Control* **2025**, *70*, 8361–8367. [\[CrossRef\]](#)
9. Zhang, Y.; Zhao, B.; Liu, D. Distributed Optimal Containment Control of Wheeled Mobile Robots via Adaptive Dynamic Programming. *IEEE Trans. Syst. Man Cybern. Syst.* **2025**, *55*, 5876–5886. [\[CrossRef\]](#)

10. Xu, F.; Xie, L.; Dimarogonas, V. Simultaneous Distributed Localization and Formation Tracking Control via Matrix-weighted Position constraints. *Automatica* **2025**, *175*, 112188. [[CrossRef](#)]
11. Yan, J.; Mo, Y.; Ishii, H. A Framework for Distributed Estimation with Limited Information and Event-Based Communications. *IEEE Trans. Autom. Control* **2023**, *69*, 3544–3559. [[CrossRef](#)]
12. An, T.; Dong, X.; Dong, B.; Cong, R.; Liu, L.; Ma, B. User-Led Modular Robot Manipulator Systems Interaction Tasks-Oriented Hierarchical Approximate Optimal Control: A Stackelberg-Pareto Differential Game Perspective. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 17801–17813. [[CrossRef](#)]
13. Ruan, S.; Ma, Y.; Wei, Z.; Zhang, C.; Xiang, C. Data-Driven Cooperative Differential Game Based Energy Management Strategy for Hybrid Electric Propulsion System of a Flying Car. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 14770–14784. [[CrossRef](#)]
14. Liu, D.; Mao, Z.; Jiang, B.; Xu, L. Simplified ADP-Based Distributed Event-Triggered Fault-Tolerant Control of Heterogeneous Nonlinear Multiagent Systems with Full-State Constraints. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2024**, *71*, 3820–3832. [[CrossRef](#)]
15. Zhao, B.; Zhang, S.; Liu, D. Self-Triggered Approximate Optimal Neuro-Control for Nonlinear Systems Through Adaptive Dynamic Programming. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 4713–4723. [[CrossRef](#)]
16. Xu, W.; Wang, T.; Qiu, J.; Liu, X. A Novel Framework for Game-Based Optimal Event-Triggered Control of Multi-Input Nonlinear Systems. *IEEE Trans. Autom. Control* **2024**, *69*, 4867–4874. [[CrossRef](#)]
17. Li, J.; Nagamune, R.; Zhang, Y.; Li, S.E. Robust Approximate Dynamic Programming for Nonlinear Systems with Both Model Error and External Disturbance. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 896–910. [[CrossRef](#)]
18. Lin, M.; Zhao, B.; Liu, D. Event-Triggered Robust Adaptive Dynamic Programming for Multiplayer Stackelberg–Nash Games of Uncertain Nonlinear Systems. *IEEE Trans. Cybern.* **2024**, *54*, 273–286. [[CrossRef](#)]
19. Qin, C.; Qiao, X.; Wang, J.; Zhang, D.; Hou, Y.; Hu, S. Barrier-Critic Adaptive Robust Control of Nonzero-Sum Differential Games for Uncertain Nonlinear Systems with State Constraints. *IEEE Trans. Syst. Man Cybern. Syst.* **2024**, *54*, 50–63. [[CrossRef](#)]
20. Ji, Z.; Ma, B.; Jiang, H.; Liang, J.; Dong, B.; An, T. Event-triggered-based Optimal Control for Reconfigurable Robot via Mixed Nonzero-sum Game. *IFAC-PapersOnLine* **2025**, *59*, 332–337. [[CrossRef](#)]
21. Ren, H.; Jiang, B.; Ma, Y. Zero-Sum Differential Game-Based Fault-Tolerant Control for a Class of Affine Nonlinear Systems. *IEEE Trans. Cybern.* **2024**, *54*, 1272–1282. [[CrossRef](#)]
22. Zhang, H.; Zhang, Y.; Zhao, X.; Su, C.-Y. HSMS-Based Event-Triggered Adaptive Dynamic Programming for Pursuit–Evasion Differential Games of Multiagent Systems. *IEEE Trans. Cybern.* **2025**, early access. [[CrossRef](#)] [[PubMed](#)]
23. Zhang, Y.; Wang, Y.; Cai, Y. Value Iteration-Based Distributed Adaptive Dynamic Programming for Multi-Player Differential Game with Incomplete Information. *IEEE/CAA J. Autom. Sin.* **2025**, *12*, 436–447. [[CrossRef](#)]
24. Lu, J.; Wei, Q.; Li, L. Learning-Based Parallel Control for Unknown Nonaffine Nonzero-Sum Games. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 19317–19327. [[CrossRef](#)]
25. Liu, F.; Su, H.; Zhang, H.; Luo, B.; Wang, J. Dynamic Self-Triggered Control for Nonzero-Sum Games of Unknown Nonlinear Constrained Systems via Generalized Fuzzy Hyperbolic Models. *IEEE Trans. Syst. Man Cybern. Syst.* **2025**, *55*, 3506–3518. [[CrossRef](#)]
26. Guo, S.; Pan, Y.; Li, H.; Cao, L. Dynamic Event-Driven ADP for N-Player Nonzero-Sum Games of Constrained Nonlinear Systems. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 7657–7669. [[CrossRef](#)]
27. Xia, H.; Lindquist, A.; Mu, C.; Sun, C. Decentralized Secure Tracking Control for Nonlinear Interconnected Systems: A Synergetic Learning-Based Strategy. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2025**, *72*, 8370–8382. [[CrossRef](#)]
28. Qiao, J.; Li, M.; Wang, D. Asymmetric Constrained Optimal Tracking Control with Critic Learning of Nonlinear Multiplayer Zero-Sum Games. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 5671–5683. [[CrossRef](#)] [[PubMed](#)]
29. Marano, D.; Cammarata, A.; Fichera, G.; Sinatra, R.; Prati, D. Modeling of a Three-axis MEMS Gyroscope with Feedforward PI Quadrature Compensation. In *Advances on Mechanics, Design Engineering and Manufacturing*; Lecture Notes in Mechanical Engineering; Springer: Cham, Switzerland, 2017; pp. 71–80.
30. García-Beltrán, C.; Miranda-Araujo, E.; Guerrero-Sanchez, M.; Valencia-Palomo, G.; Hernández-González, O.; Gómez-Peñate, S. Passivity-based Control Laws for an Unmanned Powered Parachute Aircraft. *Asian J. Control* **2021**, *23*, 2087–2096. [[CrossRef](#)]
31. Donaire, A.; Romero, J.; Ortega, R.; Siciliano, B.; Crespo, M. Robust IDA-PBC for Underactuated Mechanical Systems Subject to matched Disturbances. *Int. J. Robust Nonlinear Control* **2017**, *27*, 1000–1016. [[CrossRef](#)]
32. Xue, S.; Zhao, N.; Zhang, W.; Luo, B.; Liu, D. A Hybrid Adaptive Dynamic Programming for Optimal Tracking Control of USVs. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 9961–9969. [[CrossRef](#)] [[PubMed](#)]

33. Xue, S.; Zhang, W.; Luo, B.; Liu, D. Integral Reinforcement Learning-Based Dynamic Event-Triggered Nonzero-Sum Games of USVs. *IEEE Trans. Cybern.* **2025**, *55*, 1706–1716. [[CrossRef](#)] [[PubMed](#)]
34. Valencia-Palomo, G.; Rossiter, J.; López-Estrada, F. Improving the Feed-Forward Compensator in Predictive Control for Setpoint Tracking. *ISA Trans.* **2014**, *53*, 755–766. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.