Journal of Automation and Intelligence xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of Automation and Intelligence

journal homepage: www.keaipublishing.com/en/journals/journal-of-automation-and-intelligence/

Research article

Human–AI interactive optimized shared control

Junkai Tan [a,b], Shuangsi Xue [a,b,*], Hui Cao [a,b], Shuzhi Sam Ge [c]

[a] School of Electrical Engineering, Xi'an Jiaotong University, Xi'an, 710049, China
[b] State Key Laboratory of Electrical Insulation and Power Equipment, Xi'an Jiaotong University, Xi'an, 710049, China
[c] The Department of Electrical and Computer Engineering, National University of Singapore, 117576, Singapore

ARTICLE INFO

Keywords:
Human–AI interaction
Digital-twin system
Adaptive dynamic programming (ADP)
Data-driven
Optimal shared control

ABSTRACT

This paper presents an optimized shared control algorithm for human–AI interaction, implemented through a digital twin framework where the physical system and human operator act as the real agent while an AI-driven digital system functions as the virtual agent. In this digital twin architecture, the real agent acquires an optimal control strategy through observed actions, while the AI virtual agent mirrors the real agent to establish a digital replica system and corresponding control policy. Both the real and virtual optimal controllers are approximated using reinforcement learning (RL) techniques. Specifically, critic neural networks (NNs) are employed to learn the virtual and real optimal value functions, while actor NNs are trained to derive their respective optimal controllers. A novel shared mechanism is introduced to integrate both virtual and real value functions into a unified learning framework, yielding an optimal shared controller. This controller adaptively adjusts the confidence ratio between virtual and real agents, enhancing the system's efficiency and flexibility in handling complex control tasks. The stability of the closed-loop system is rigorously analyzed using the Lyapunov method. The effectiveness of the proposed AI–human interactive system is validated through two numerical examples: a representative nonlinear system and an unmanned aerial vehicle (UAV) control system.

1. Introduction

Human-automation collaboration is an important point of concern in cyber–physical systems (CPS), encompassing the cooperative control between human operators and automated systems to accomplish tasks such as driving, flying, and manufacturing [1–3]. Digital-twin technology provides a novel perspective to the cooperative optimal control problem of the real-world system [4–6], integrating physical systems with digital models for real-time monitoring [7], control [8], and maintenance [9] of complex systems. The digital-twin system could construct a virtual agent that replicates the real-world system by assessing its demonstrated operations, encompassing the model, environment, and behavior of the actual system [10,11]. The digital-twin system facilitates virtual–real interaction between real agents and virtual agents, offering a novel method for modeling real-world systems and operators [12,13].

The interaction between digital and physical systems is a crucial aspect of digital-twin systems, aiming to facilitate efficient and effective control of real-world systems [14,15]. However, the complexity of real-world systems and the unpredictability of human operators render cooperative interaction a complex issue. Learning-based methods have been extensively studied for the design of cooperative control between humans and automation [16–18]. The work [19–21] developed a game-based approximate optimal control algorithm to achieve the optimal control of the human-automation system in a Stackelberg game framework. The research [22] proposed a learning-based hierarchical control algorithm to achieve cooperation control task between human and manipulator arm. However, existing methods have limitations in complex control tasks and the adaptability to the real-world system [23,24]. Recently, reinforcement learning (RL) and adaptive dynamic programming (ADP) techniques provide a new approach to interact with the real-world system and the human operator [25–27]. To model the interaction of human behavior, work [28,29] studied inverse RL techniques, which could learn the optimal value and its corresponding control policy of the human operator from the demonstrated operations. The model-free RL methods are proposed in [30,31] to learn the optimal interaction control of the real-world system without prior knowledge of the system dynamics. For digital-twin systems, RL-based virtual–real interactions are investigated in [32–34] to enable the virtual–real interaction and achieve complex control tasks in the digital-twin system. However, the trustworthiness and reliability of the authority between the real-world and virtual-digital systems remain problematic in the digital-twin framework [35,36].

Peer review under responsibility of Chongqing University.
* Corresponding author at: School of Electrical Engineering, Xi'an Jiaotong University, Xi'an, 710049, China.
E-mail address: xssxjtu@xjtu.edu.cn (S. Xue).

https://doi.org/10.1016/j.jai.2025.01.001
Received 14 November 2024; Received in revised form 14 December 2024; Accepted 4 January 2025
Available online xxxx

The trade-off between expert demonstrations and AI-driven automation operations is an urgent problem in the human-automation cooperation of digital-twin systems, which is essential for achieving optimal control of real-world systems [37,38]. Various shared mechanisms have been studied to quantify the confidence and calibrate the relationship between the human operator and automation [39–41], which could judge the authority ratio of the human operator and the autonomy. Direct shared control methods are investigated in [42,43] to integrate the human operator and the automation in a control ratio allocation manner. Indirect shared control methods are proposed in [44,45] to estimate the intention of the human operator and the autonomy in a shared mechanism, which subsequently adjust the control input of the real-world system. A composite judgement mechanism is proposed in [46] to adjust the confidence of trusted safe behavior and the autonomy in a indirect shared control manner. The literature [47] proposed a safe human-quadrotor interaction shared control scheme using the model predictive control (MPC) method, which optimizes the automation control input using human pilot manuveurs data and feedback control policy. However, the smoothness and adaptability of the shared mechanism stay challenging in the complex control tasks of real-world systems.

Motivated by the above discussions, this paper presents an human–AI interactive optimized shared control algorithm based on digital twin technology, integrating the real-world system and human operator as a real agent with the AI-powered digital-twin system as a virtual agent. The real agent learns a real optimal controller from the demonstrated operations, while the AI virtual agent mirrors the real agent to obtain an auxiliary virtual-digital system and a virtual optimal controller. The real optimal controller and the virtual optimal controller are approximated using AI-based reinforcement learning (RL) technique, and a novel optimal shared mechanism is introduced to train an optimal shared controller jointly, which composite both virtual–real value functions into a composite learning law. Lyapunov stability analysis is conducted to verify the stability of the proposed control scheme. The main contributions of this paper are summarized as follows:

1. A data-driven, AI-enhanced digital twin system is proposed, wherein the real-world system and operator are integrated as a real agent that learns an optimal controller from demonstrated operations. A corresponding digital agent is mirrored using the Koopman operator approach, which establishes an auxiliary virtual-digital system and a virtual optimal controller. This digital twin system enables more efficient and effective interaction between virtual and real agents compared to existing methods [17,22,23].
2. A novel confidence allocation mechanism is introduced to jointly train an optimal shared controller by integrating both virtual and real value functions through an indirect shared approach. This mechanism can adaptively adjust confidence levels for virtual and real agents, facilitating smoother and more flexible management of complex control tasks compared to current methods [40,47,48].
3. A digital twin-based actor–critic algorithm is developed to learn the optimal shared controller for the real-world system. Critic and actor neural networks are designed to learn the virtual–real optimal value functions and controllers, respectively. A composite learning law is proposed for the joint training of the optimal shared controller, eliminating the need for extensive data and prior knowledge of the system dynamics [25,46,49].

The rest of this paper is organized as follows: Section 2 presents the virtual–real system and formulates the optimal control problem. Section 3 introduces the digital-twin-based optimal shared control algorithm. Section 4 provides the stability analysis of the proposed control scheme. Section 5 presents two simulation examples to demonstrate the effectiveness of the proposed control scheme. Section 6 concludes the paper and discusses future works.

Notation: $\mathbb{R}$ denotes the real number set, $\mathbb{R}^n$ denotes the n-dimensional real number space, $\mathbb{R}^{n\times m}$ denotes the $n \times m$ real number matrix, ∇ denotes the gradient operator.

2. System description and problem formulation

In this section, we present the virtual–real system architecture and formulate the optimal control problem for human–AI interactive control. We begin by introducing the concept of a virtual–real system for human–AI interaction through the following formal definition.

Definition 1 (*Virtual-real System for Human-AI Interaction [34]*)**.** The virtual–real system consists of a real-world system and a virtual-digital system, designated as the real agent and virtual agent respectively:

1. The real agent comprises a physical system that interfaces with and is controlled by an expert human operator.
2. The virtual agent is an AI-powered digital model that mirrors and learns from the real agent's behavior.

The virtual–real system is designed to achieve optimal shared control that integrates both human expertise and AI-driven automation.

Within this framework, the real agent enables direct human–system interaction, capturing expert knowledge and control strategies through demonstrated operations. Concurrently, the virtual agent maintains a high-fidelity digital replica of the real system, enabling predictive analysis and optimization of control policies. The dynamics of this virtual–real composite system can be mathematically formulated as:

$$\begin{cases} \dot{\Theta}_r(t) = F_r(\Theta_r(t)) + G_r(\Theta_r(t))U_r(t) \\ \dot{\Theta}_v(t) = F_v(\Theta_v(t)) + G_v(\Theta_v(t))U_v(t) \end{cases} \tag{1}$$

where $\Theta_r \in \mathbb{R}^{n_r}$, $\Theta_v \in \mathbb{R}^{n_v}$ denote the states of the real and virtual systems, respectively, $U_r \in \mathbb{R}^{m_r}$, $U_v \in \mathbb{R}^{m_v}$ are the control inputs, $F_r : \mathbb{R}^{n_r} \to \mathbb{R}^{n_r}$, $F_v : \mathbb{R}^{n_v} \to \mathbb{R}^{n_v}$ denote the drift dynamics, and $G_r : \mathbb{R}^{n_r} \to \mathbb{R}^{n_r\times m_r}$, $G_v : \mathbb{R}^{n_v} \to \mathbb{R}^{n_v\times m_v}$ denote the control input dynamics. To achieve optimal control for the virtual–real system that effectively integrates both real and virtual agent inputs, we define a shared control input that incorporates both contributions:

$$\mathcal{U}(t) = \mathcal{U}(t, \alpha, U_r, U_v, \Theta_r, \Theta_v) \tag{2}$$

where $\mathcal{U}(t) \in \mathbb{R}^m$ denotes the shared control input of the virtual–real system. This composite control input is designed to integrate both human expertise through U_r and AI-driven automation through U_v. The parameter $\alpha \in [0, 1]$ serves as an adaptive confidence allocation factor that dynamically adjusts the balance between real and virtual agents based on their demonstrated performance and reliability. As α approaches 1, the system assigns higher confidence to the human operator's control decisions, while as α approaches 0, the system relies more heavily on the AI-driven virtual agent's computed optimal actions. With the definition of the shared control input (2), the virtual–real system could be described by the following nonlinear affine input dynamics:

$$\begin{bmatrix} \dot{\Theta}_r(t) \\ \dot{\Theta}_v(t) \end{bmatrix} = \begin{bmatrix} F_r(\Theta_r(t)) \\ F_v(\Theta_v(t)) \end{bmatrix} + \begin{bmatrix} G_r(\Theta_r(t)) \\ G_v(\Theta_v(t)) \end{bmatrix} \times \mathcal{U}(t) \tag{3}$$

which is a composite system of the real-world system and the virtual-digital system.

Remark 1. It is important to note that the primary objective of the virtual–real system is to achieve optimal shared control of the real-world system, which not only stabilizes the physical system but also ensures accurate mirroring by the virtual-digital system. The shared control input (2) is strategically designed to manage the interaction between real and virtual agents, where the adaptive parameter α dynamically adjusts the confidence allocation between these agents. The composite system dynamics (3) integrates both the real-world and virtual-digital systems, with the latter serving as a digital mirror to

facilitate optimal shared control. However, inherent mismatches and uncertainties between physical and virtual systems could potentially compromise system stability. Therefore, proper construction of the digital-twin architecture is crucial for enabling effective optimal shared control, which will be detailed in Section 3.1.

Next subsection will present the problem formulation of the optimal shared control of the virtual–real system.

2.1. Formulation of optimal control

To achieve optimal control of the virtual–real system, we first formulate the optimal control problem by defining value functions for both the real and virtual agents. For each agent $i \in \{r, v\}$, the value function is defined as the quadratic cost function:

$$V_i(\Theta_i, U_i) = \int_t^{\infty} r_i(\Theta_i, U_i)\, d\tau, \quad i \in \{v, r\} \tag{4}$$

where $U_i(t)$ is bounded by symmetric constraints $|U_i| \le \bar{U}_i$ for each agent i. The instantaneous reward function $r_i(\Theta, U)$ for deriving the optimal control policies is defined as:

$$r_i(\Theta_i, U_i) = \Theta_i^{\top} Q_i \Theta_i + \psi_i(U_i), \ \forall i \in \{v, r\} \tag{5}$$

where $Q_i \in \mathbb{R}^{n\times n}, (i = v, r)$ represents the state weighting matrices that penalize deviations in Θ_i. To ensure bounded control signals in both virtual and real systems, we adopt a hyperbolic tangent penalty function $\psi_i(U_i)$ to constrain the magnitude of control input U_i. Similar approaches have been studied in [50,51]. The penalty function takes the form:

$$\psi_i(U_i) = \int_0^{U_i} 2\mu_i R_i \tanh^{-1}\left(\gamma_{U_i}/\mu_i\right) \mathrm{d}\gamma_{U_i}, \ \forall i \in \{v, r\} \tag{6}$$

where $\mu_i \in \mathbb{R}^{m\times 1}$ represents the symmetric control input constraints satisfying $|U_i| \le \bar{U}_i = \mu_i$, $R_i \in \mathbb{R}^{n\times n}$ is the positive definite control penalty matrix, and γ_U denotes an integral variable. For developing an optimal controller for the virtual–real system, we need to derive both the optimal value function $V_i^*(\Theta_i)$ and optimal control input $U_i^*(\Theta_i)$. To facilitate the subsequent stability analysis and controller synthesis based on dynamics (3), we first introduce the following assumption and definition.

Assumption 1 ([16,28]). For the augmented dynamics (3), the following standard conditions hold:

1. The functions $F(\Theta)$ and $G_i(\Theta)$ are Lipschitz continuous on a compact set $\chi \subset \mathbb{R}^n$ containing the origin, with $F(0) = 0$. Furthermore, the control input matrix $G_i(\Theta)$ satisfies the boundedness condition: $\|G_i(\Theta)\| \le G_{Hi}, \forall \Theta \in \chi$.
2. The weighting matrices Q_i and R_i in the cost function are bounded as follows:

$$\underline{\lambda}_{Q_i} I \le Q_i \le \bar{\lambda}_{Q_i} I, \quad \underline{\lambda}_{R_i} I \le R_i \le \bar{\lambda}_{R_i} I$$

where $\underline{\lambda}_{Q_i}, \underline{\lambda}_{R_i} \ge 0$ and $\bar{\lambda}_{Q_i}, \bar{\lambda}_{R_i} > 0$ are finite constants, and I denotes the identity matrix of appropriate dimension.

Based on Assumption 1, the augmented system dynamics (3) satisfy fundamental stability and control synthesis requirements. The optimal value function $V_i^*(\Theta)$ for each agent $i \in \{v, r\}$ minimizes the quadratic cost (24) over admissible control inputs:

$$V_i^*(\Theta) = \min_{U_i \in \Omega_U} \int_t^{\infty} r_i(\Theta, U_i) d\tau, \ \forall i \in \{v, r\} \tag{7}$$

where Ω_U denotes the admissible control input set. To minimize the value function (4), we define the Hamilton function:

$$H_i(\Theta_i, U_i, \nabla V_i^*) = \Theta_i^{\top} Q_i \Theta + \psi_i(U_i) + (\nabla V_i^*)^{\top}(F_i + G_i U_i) \tag{8}$$

where ∇V_i^* denotes the gradient of the optimal value function with respect to the state Θ_i. By taking the derivative of the Hamilton function (8) with respect to the control input U_i and setting it to zero, the optimal control law is obtained as:

$$U_i^*(\Theta_i) = \frac{\partial H_i}{\partial U_i} = -\mu_i \tanh\left(\frac{R^{-1} G_i^{\top}}{2\mu_i}\left(\nabla V_i^{*\top}\right)\right), \ \forall i \in \{v, r\} \tag{9}$$

Combining the optimal control input (23) with the Hamilton function (8), the HJB equation is obtained as:

$$0 = \Theta_i^{\top} Q_i \Theta_i + \psi_i(U_i^*) + (\nabla V_i^*)^{\top}(F_i + G_i U_i^*), \ \forall i \in \{v, r\} \tag{10}$$

The optimal value function (24) and the corresponding saturated optimal control input (23) could be derived by solving the HJB equation (10). Now the problem of optimal control of the virtual–real system is formulated. However, solving the HJB equation (10) is still a complex and challenging problem due to its nonlinearity and high dimensionality. The next section will introduce a novel shared mechanism that collects and allocates control inputs from the human operator and the optimal controller of autonomy.

3. Main results: human-AI interactive optimized shared control algorithm

In this section, a digital-twin-based optimal shared control algorithm is proposed to achieve the optimal shared control of the virtual–real system. The proposed algorithm is composed of three parts: the digital-twin system construction, the shared confidence allocation and the optimal shared control design. The architecture of the proposed digital-twin system is illustrated in Fig. 1, where the real-world system and the virtual-digital system are connected by the shared control input. The real-world system learns from the demonstrated operations to obtain the real optimal controller, and the virtual-digital system mirrors the real-world system to obtain the virtual optimal controller. While the virtual–real optimal controllers are approximated using the reinforcement learning (RL) technique, and the shared confidence allocation adjusts the confidence α to the real agent and the virtual agent. Then the optimal shared controller is constructed by compositing the virtual–real optimal controllers with the shared confidence allocation. First, the digital-twin system construction is introduced in the next subsection.

3.1. Digital-twin system construction

Consider an uncontrolled real-world nonlinear system with discrete-time dynamics:

$$\Theta_r(t_{i+1}) = \int_t^{t+1} F_r(\Theta_r\, d\tau + \Theta_r(t_i)) \tag{11}$$

which represents the discrete-time evolution of system (1), where $\Theta_r(t_i) \in \mathbb{R}^n$ is the system state at time step t_i. This real-world system operates in the state–space $\mathbb{R}^n$. To establish a virtual-digital model that accurately mirrors this physical system, we employ the Koopman operator framework to characterize the nonlinear system dynamics. This approach lifts the system states into an expanded Hilbert space, enabling more effective prediction and analysis of the virtual-digital system behavior. For analyzing the system dynamics in a lifted functional space, we define the Koopman operator $\mathcal{K} : \mathcal{M} \to \mathcal{M}$ as:

$$\left(\mathcal{K}\Psi_{\mathcal{K}_1}\right)\left[\Theta_r(t_{i+1})\right] = \Psi_{\mathcal{K}_1}\left\{F_r\left[\Theta_r(t_i)\right]\right\} \tag{12}$$

where $\Psi_{\mathcal{K}_1} : \mathbb{R}^{n_r} \to \mathcal{M}$ maps the original state–space to the Hilbert space associated with the Koopman operator. While this formulation applies to autonomous systems, we need to extend it to incorporate control inputs. Let $\chi_r(t_i) = \left[\Theta_r(t_i)^{\top}, U_r(t_i)^{\top}\right]^{\top}$ denote the augmented state including both system states and control inputs. The augmented dynamics can then be expressed as:

$$\chi_r(t_{i+1}) = \overline{F}(\chi_r(t_{i+1})) := \left[F_r(\Theta_r, U_r(0))^{\top} \quad (SU_r)^{\top}\right]^{\top} \tag{13}$$

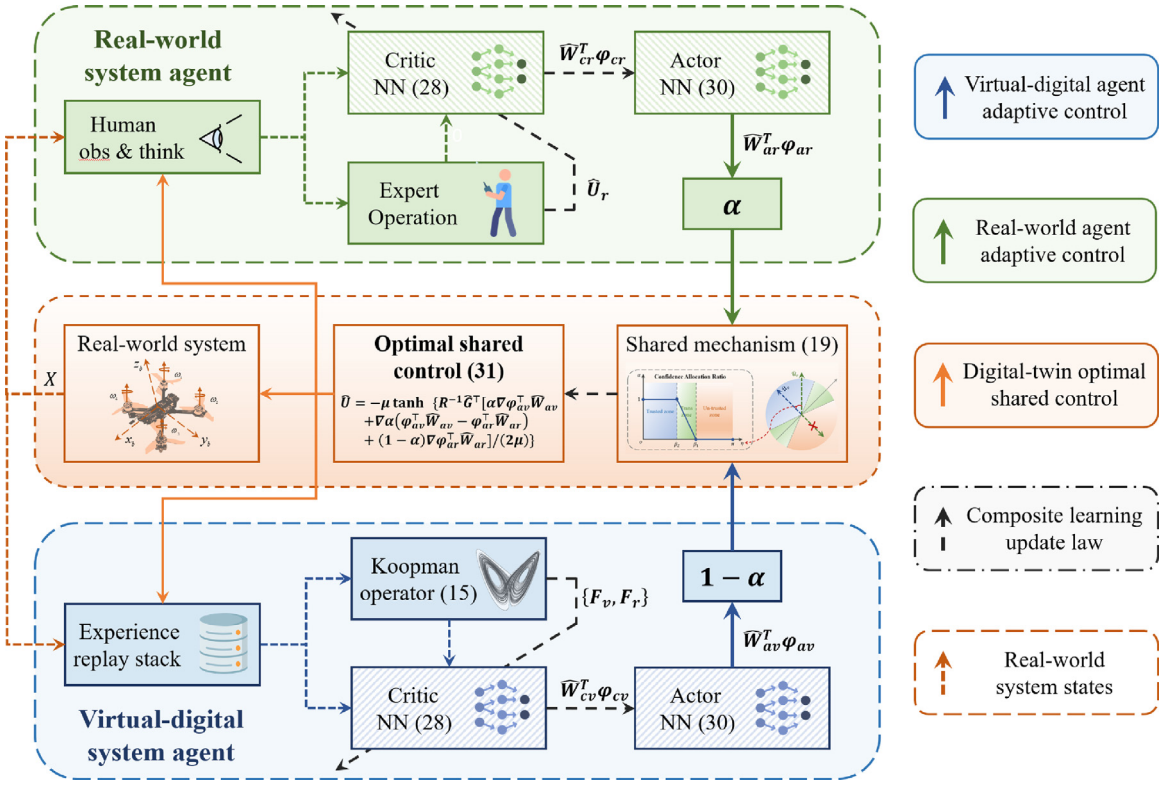

Fig. 1. The architecture of the proposed digital-twin system.

where S represents the shift operator that advances the control sequence: $(SU_r)(t_i) = U_r(t_{i+1})$. This allows us to reformulate the Koopman operator $\mathcal{K} : \mathcal{M} \to \mathcal{M}$ to analyze the controlled system dynamics:

$$(\mathcal{K}\Psi_{\mathcal{K}_2})(\chi_r(t_i)) = \Psi_{\mathcal{K}_2}(\overline{F}_r(\chi_r(t_i))) \tag{14}$$

where $\Psi_{\mathcal{K}_2} : \mathbb{R}^{n_r+l} \to \mathbb{R}$ maps the augmented state–space to the Koopman operator's Hilbert space. This operator framework enables analysis of controlled nonlinear systems in an expanded functional space. The Koopman approach lifts the state dynamics into a higher-dimensional space where linear analysis techniques become applicable. To establish the virtual-digital system dynamics in (1), we employ the extended dynamic mode decomposition (EDMD) method to approximate the Koopman operator $\mathcal{K}$ defined in (14). EDMD provides a data-driven framework to estimate the Koopman operator through:

$$\Psi_{\mathcal{K}_2}(\chi_r(t_{i+1})) = \mathcal{K}^{\top}\Psi_{\mathcal{K}_2}(\chi_r(t_i)) + \epsilon(\chi_r(t_i)) \tag{15}$$

where $\Psi_{\mathcal{K}_2} : \mathbb{R}^{n_r+l} \to \mathbb{R}$ represents the transfer function as previously defined, and $\epsilon(\chi_r(t_i))$ denotes the operator approximation error. Given data samples $(\chi_r(t_j), \chi_r(t_{j+1})), j = 1, \ldots, N_{\mathcal{K}}$ from system evolution $\chi_r(t_{j+1}) = F(\chi_r(t_j))$, we approximate the Koopman operator $\mathcal{K}$ by minimizing:

$$E_{\mathcal{K}} = \sum_{j=1}^{N_{\mathcal{K}}} \|\epsilon(\chi_r(t_j))\|^2 = \sum_{j=1}^{N_{\mathcal{K}}} \|\Psi_{\mathcal{K}_2}(\chi_r(t_{j+1})) - K^{\top}\Psi_{\mathcal{K}_2}(\chi_r(t_j))\|^2 \tag{16}$$

where $\Psi_{\mathcal{K}_2}(\chi_r(t_j)) = [\Psi_{\mathcal{K}_2,1}(\chi_r(t_j)), \ldots, \Psi_{\mathcal{K}_2,N_{\Psi_{\mathcal{K}_2}}}(\chi_r(t_j))]^{\top}$ represents a vector of transfer functions $\Psi_{\mathcal{K}_2,i} : \mathbb{R}^{n_r+l} \to \mathbb{R}$, $i = 1, \ldots, N_{\Psi_{\mathcal{K}_2}}$.

Let $\mathbf{X}_{\mathcal{K}} = [\Psi_{\mathcal{K}_2}(\chi_r(t_1)), \ldots, \Psi_{\mathcal{K}_2}(\chi_r(t_{N_{\mathcal{K}}-1}))]$ denote transformed state data, $\mathbf{Y}_{\mathcal{K}} = [\Psi_{\mathcal{K}_2}(\chi_r(t_2)), \ldots, \Psi_{\mathcal{K}_2}(\chi_r(t_{N_{\mathcal{K}}}))]$ represent transformed output states, and $\mathbf{U}_{\mathcal{K}} = [\Psi_{\mathcal{K}_2}(U_r(t_2)), \ldots, \Psi_{\mathcal{K}_2}(U_r(t_{N_{\mathcal{K}}}))]$ contain transformed control inputs, where relation $\chi_r(t_{j+1}) = F_r(\chi_r(t_j), U_r(t_j))$ holds. The following assumption is introduced to ensure the convergence of the EDMD algorithm.

Assumption 2. The transformed state data $\mathbf{X}_{\mathcal{K}}$, output states $\mathbf{Y}_{\mathcal{K}}$, and control inputs $\mathbf{U}_{\mathcal{K}}$ are collected from the real-world system. We assume that: the data matrices $\mathbf{X}_{\mathcal{K}}$, $\mathbf{Y}_{\mathcal{K}}$, and $\mathbf{U}_{\mathcal{K}}$ are of full column rank, ensuring that they are Moore–Penrose pseudo-invertible. This ensures the convergence of the EDMD algorithm, enabling accurate estimation of the Koopman operator $\mathcal{K}$.

It is important to note that the EDMD algorithm is a data-driven method used to approximate the Koopman operator $\mathcal{K}$ based on collected state data. The accuracy of the Koopman operator estimation is significantly influenced by the quality and quantity of the data samples. To ensure the convergence of the EDMD algorithm, Assumption 2 is introduced to guarantee the full column rank of the data matrices. By satisfying this assumption, the Koopman operator $\mathcal{K}$ can be accurately estimated, enabling effective modeling of the virtual-digital system.

Using the collected data matrices $\mathbf{X}_{\mathcal{K}}$, $\mathbf{Y}_{\mathcal{K}}$ and $\mathbf{U}_{\mathcal{K}}$, the linearized system dynamics matrices $\hat{A}_r \in \mathbb{R}^{n\times n}$ and $\hat{B}_r \in \mathbb{R}^{n\times m}$ can be obtained by solving the minimization problem in (15):

$$\left[\hat{A}_r, \hat{B}_r\right] = \mathbf{Y}_{\mathcal{K}}\left[\mathbf{X}_{\mathcal{K}}, \mathbf{U}_{\mathcal{K}}\right]^{\dagger} \tag{17}$$

where † denotes the Moore–Penrose pseudo inverse operator. Based on these linearized matrices, the virtual-digital system dynamics in (1) can be approximated as:

$$F_v = \hat{A}_r \times \Theta_r, \quad G_v = \hat{B}_r \tag{18}$$

Through this process, we establish a virtual-digital model that mirrors the real-world system by leveraging the Koopman operator framework (14) and EDMD algorithm (15). This enables prediction of the virtual system's future states while maintaining consistency with the physical system. In the following subsection, we present a digital-twin-based actor-critic algorithm to achieve optimal control of this integrated virtual–real system.

Remark 2. The effectiveness of the virtual agent's dynamics estimation is intrinsically tied to both data quality and quantity. While the EDMD algorithm demonstrates inherent robustness to noise and can function with sparse datasets, substantial deviations in the virtual

agent's behavior may emerge under severe data constraints. To mitigate these challenges, several enhancement strategies can be employed: (1) Advanced data augmentation techniques including system identification and comprehensive model validation processes; (2) Sophisticated regularization methods to improve generalization; (3) Adaptive noise filtering algorithms to maintain signal fidelity. Through systematic implementation of these approaches, high-fidelity approximation of the virtual agent's dynamics can be achieved, thereby maintaining the reliability and effectiveness of the shared control framework.

Remark 3. Mismatches between the real and virtual agents' dynamics can significantly impact the performance and robustness of the shared control framework. These discrepancies may lead to suboptimal control inputs, reduced system robustness, and inefficiencies in learning. To mitigate inaccuracies in the virtual model, iterative deep Koopman operator learning from literature [52,53] can be employed to refine the virtual agent's dynamics based on real-world observations. This iterative process enhances the model's accuracy and adaptability over time, ensuring that the virtual agent's predictions align with the real system's behavior. By systematically addressing model inaccuracies, the shared control framework can maintain high performance and reliability across diverse operating conditions.

3.2. Shared confidence allocation

In this subsection, a novel shared mechanism is proposed to dynamically determine the confidence allocation between autonomous and human control inputs. While existing Maxwell's Demon Algorithm (MDA) methods [48,54] use binary switching to allocate control authority, such abrupt transitions can potentially destabilize the system. Our approach instead implements smooth confidence allocation based on the angular deviation between autonomous and human control vectors:

$$\alpha = \begin{cases} 0, & \text{if } \eta \geq \alpha_1 \\ 1, & \text{if } \eta \leq \alpha_2 \\ \frac{\eta-\alpha_1}{\alpha_2-\alpha_1}, & \text{otherwise} \end{cases} \tag{19}$$

where η represents the angle between the autonomous optimal control vector and human control vector, with thresholds α_1 and α_2 determining the transition boundaries. This continuous allocation mechanism enables smooth handover of control authority based on the alignment between autonomous and human control objectives.

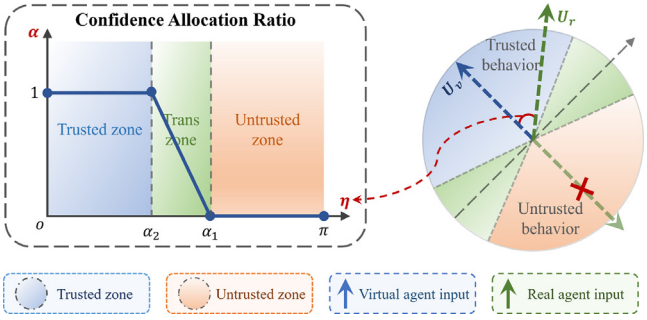

Fig. 2. The mechanism of confidence allocation.

Fig. 2 illustrates the detailed shared mechanism. In the figure, the optimal control input from the autonomy is represented by the blue slash-dot vector, while the green dotted vector represents the human control input. Let η denote the angle between these two control vectors. The final shared control input is synthesized by combining both the optimal and human control inputs based on their relative alignment. When $\eta > \alpha_1$, the confidence parameter α is set to 0, indicating full reliance on the autonomous system. Conversely, when $\eta < \alpha_2$, α becomes 1, shifting control authority entirely to the human operator. For intermediate angles where $\alpha_2 < \eta < \alpha_1$, α is determined proportionally based on η's position between the threshold values α_1 and α_2, enabling smooth transitions between human and autonomous control.

With the shared mechanism (19), the confidence to the human operator and the autonomy could be adaptively adjusted, the composite optimal value function could be constructed by the shared mechanism:

$$\mathcal{V}^* = \alpha V_r^* + (1-\alpha) V_v^* \tag{20}$$

where $\mathcal{V}^*$ is the composite optimal value function of the virtual–real system, V_r^* and V_v^* are the optimal value functions of the real-world system and the virtual-digital system defined in (24). The composite optimal control input could be obtained by the shared mechanism:

$$\mathcal{U}^* = \underset{\mathcal{U}\in\Theta_U}{\operatorname{argmin}}\ \mathcal{H}\left(\mathcal{U}, \nabla\mathcal{V}^*\right) \tag{21}$$

where $\mathcal{U}^*$ is the composite optimal control input of the virtual–real system, $\mathcal{H}$ is the composite Hamilton function of the virtual–real system which is defined as:

$$\mathcal{H} = \left(\nabla\mathcal{V}^*\right)^\top (F + G\mathcal{U}) + r_r(\Theta_r, \mathcal{U}) \tag{22}$$

where $r_r(\Theta_r, \mathcal{U})$ is the reward function of the real-world system.

Remark 4. The shared mechanism is designed to optimally allocate control authority between the human operator and autonomy. The composite optimal value function directly embodies this shared mechanism by blending the optimal value functions from both the real-world system and virtual-digital system. The composite optimal control input emerges from optimizing the composite Hamilton function associated with the composite optimal value function. Compared to existing human–AI interactive control methods [40,47], the proposed shared mechanism achieves smoother and more effective composition of both optimal value functions and control inputs, enabling more sophisticated allocation of control authority between human operator and autonomy.

Remark 5. In contrast to existing MDA methods [48,54], the proposed shared mechanism achieves smooth and effective allocation of control authority parameters α by intelligently evaluating the intentions of both human operator and autonomy. When the human operator's intentions align with the autonomy's objectives, the mechanism maximizes utilization of human control input. Conversely, when significant discrepancies exist between human and autonomous intentions, the mechanism smoothly transitions control authority to the autonomy system. The intermediate transition zones ensure continuous and smooth handover of control, avoiding abrupt switches that could destabilize the system.

Remark 6. Compared to the adaptive mechanisms in [22,55], the proposed shared mechanism reduces computational costs by directly adjusting the confidence parameter α based on the angle between human and autonomous control inputs. This approach is more straightforward and efficient for real-time control applications, facilitating seamless integration of human expertise with AI-driven automation. Additionally, the shared mechanism is more interpretable and explainable, as it does not rely on complex learning algorithms or extensive training data, making it better suited for safety-critical applications.

To construct the value function for the human operator and obtain the optimal shared control input, the optimal controller is approximated by a novel digital-twin-based actor-critic algorithm, which is presented in the next subsection.

3.3. Digital-twin actor-critic design

In this subsection, we present a digital-twin-based actor-critic design to achieve optimal control approximation. First, actor-critic neural networks (NNs) are employed to reconstruct both virtual and real optimal

value functions and control inputs. Through a shared mechanism, these reconstructed functions and inputs are integrated into a composite optimal value function and shared control input. A shared Bellman error is then defined to enable joint training of the optimal shared controller. The actor-critic NNs are trained by minimizing this shared error to approximate the optimal value functions and control inputs. For function approximation, we develop the following actor-critic NN structure to reconstruct the optimal value functions and control inputs:

$$V_i^*(\Theta_r) = W_{ci}^\top \varphi_{ci}(\Theta) + \varepsilon_{ci}(\Theta_r), \quad i \in \{v, r\} \tag{23}$$

$$U_i^*(\Theta_r) = -\mu_i \tanh\left\{ \frac{R^{-1} G_i^\top}{2\mu_i} \left(\nabla \varphi_{ai}^\top(\Theta_r) W_{ai} + \nabla \varepsilon_{ai}^\top \right) \right\} \tag{24}$$

where $W_{ci} \in \mathbb{R}^{n_{\varphi ci} \times 1}$ represents the critic NN weights, and ε_{ci}, ε_{ai} denote approximation errors. The composite optimal value function integrating both virtual and real components is:

$$\begin{aligned} \mathcal{V}^*(\Theta_r) &= \alpha V_r^*(\Theta_r) + (1-\alpha) V_v^*(\Theta_r) \\ &= \alpha W_{cr}^\top \varphi_{cr}(\Theta_r) + (1-\alpha) W_{cv}^\top \varphi_{cv}(\Theta_r) + \varepsilon_c(\Theta_r) \end{aligned} \tag{25}$$

Substituting this into the HJB equation yields:

$$\begin{aligned} 0 =& \Theta_r^\top Q \Theta_r + \psi(\mathcal{U}) + \left[\alpha \nabla \varphi_{cr} W_{cr}^\top + (1-\alpha) \nabla \varphi_{cv} W_{cv}^\top \right. \\ &\left. + \nabla \alpha W_{cr}^\top \varphi_{cr} - \nabla \alpha W_{cv}^\top \varphi_{cv} \right]^\top (F + G\mathcal{U}) \end{aligned} \tag{26}$$

The optimal shared control input emerges as:

$$\begin{aligned} \mathcal{U}^*(\Theta_r) = -\mu \tanh\Bigg\{ & \frac{R^{-1} G_r^\top}{2\mu} \left(\alpha \nabla \varphi_{cr}^\top W_{cr} + (1-\alpha) \nabla \varphi_{cv}^\top W_{cv} \right. \\ & \left. + \nabla \alpha W_{cr}^\top \varphi_{cr} - \nabla \alpha W_{cv}^\top \varphi_{cv} + \nabla \varepsilon_c^\top \right) \Bigg\} \end{aligned} \tag{27}$$

In practice, since the ideal weights W_{ci} and W_{ai} are unknown, we employ estimated weights to approximate the optimal value functions and control inputs:

$$\hat{V}_i(\Theta_r) = \hat{W}_{ci}^\top \varphi_{ci}(\Theta_r), \quad i \in \{v, r\} \tag{28}$$

$$\begin{aligned} \hat{\mathcal{V}}(\Theta_r) &= \alpha \hat{V}_r(\Theta_r) + (1-\alpha) \hat{V}_v(\Theta_r) \\ &= \alpha \hat{W}_{cr}^\top \varphi_{cr}(\Theta_r) + (1-\alpha) \hat{W}_{cv}^\top \varphi_{cv}(\Theta_r) \end{aligned} \tag{29}$$

where $\hat{W}_{ci} \in \mathbb{R}^{n_\varphi \times 1}$ represents the estimated critic NN weights for the optimal value function. The estimated optimal control input is consequently derived as:

$$\hat{U}_i(\Theta_r) = -\mu_i \tanh\left\{ \frac{R^{-1} G_i^\top}{2\mu_i} \nabla \varphi_{ai}^\top(\Theta_r) \hat{W}_{ai} \right\}, \ i \in \{v, r\} \tag{30}$$

By integrating the estimated optimal control inputs (30) with the composite optimal value function (29), we obtain the estimated optimal shared control input:

$$\begin{aligned} \hat{\mathcal{U}}(\Theta_r) = -\mu \tanh\Bigg\{ & \frac{R^{-1} G_i^\top}{2\mu} \Big(\alpha \nabla \varphi_{ar}^\top \hat{W}_{ar} + (1-\alpha) \nabla \varphi_{av}^\top \hat{W}_{av} \\ & + \nabla \alpha (\varphi_{ar}^\top \hat{W}_{ar} - \varphi_{av}^\top \hat{W}_{av}) \Big) \Bigg\} \end{aligned} \tag{31}$$

The derived composite optimal controller $\mathcal{U}^*$ represents the optimal shared control input that balances the contributions of the real and virtual agents. The adaptive parameter α determines the relative influence of the human operator and the AI-driven digital system on the control output. When $\alpha = 1$, the control input is entirely determined by the real agent, reflecting a human-centric control strategy. Conversely, when $\alpha = 0$, the virtual agent's control input fully dictates the control action, indicating an autonomy-centric approach. The shared control mechanism dynamically allocates control authority between the human operator and the autonomous system based on the system's state and performance metrics. This adaptive blending of human and AI contributions enables effective shared control, enhancing system performance and robustness across diverse operating conditions.

Substituting the optimal shared control input (30) into the composite HJB equation (26) yields the shared Bellman error:

$$\begin{aligned} \delta(\Theta_r, \hat{\mathcal{V}}, \hat{\mathcal{U}}) =& \nabla \hat{V}_i^\top (F_r + G_r \hat{\mathcal{U}}) + r_r(\Theta_r, \hat{\mathcal{U}}) \\ =& \Big(\alpha \nabla \varphi_{ar}^\top \hat{W}_{ar} + (1-\alpha) \nabla \varphi_{av}^\top \hat{W}_{av} \\ & + \nabla \alpha (\varphi_{ar}^\top \hat{W}_{ar} - \varphi_{av}^\top \hat{W}_{av}) \Big)^\top \\ & \times (F_r + G_r \hat{\mathcal{U}}) + \Theta_r^\top Q_r \Theta_r + \psi_r(\hat{\mathcal{U}}) \end{aligned} \tag{32}$$

where δ denotes the shared Bellman error used to train the actor-critic NNs for approximating optimal value functions and control inputs. The weights of the critic NNs are updated by minimizing this shared error to approximate the optimal shared controller. The online composite optimal value function approximation is presented in the next subsection.

3.4. Online composite optimal value function approximation

In this subsection, we present the online weights update method for actor-critic NNs based on Bellman error minimization. To enable joint training of actor-critic NNs for optimal shared control of the virtual–real system, we introduce compact representations of the virtual–real and shared Bellman errors:

$$\delta(\Theta_r, \hat{\mathcal{V}}, \hat{\mathcal{U}}) = \hat{W}_{ar}^\top \Psi_{ar} + \hat{W}_{av}^\top \Psi_{av} + r_r(\Theta_r, \hat{\mathcal{U}}) \tag{33}$$

$$\zeta_i^k(\Theta^k, \hat{V}_i^k, \hat{U}_i^k) = \hat{W}_{ai}^{k\top} \Phi_{ai}^k + r_i(\Theta^k, \hat{U}_i^k), \ i \in \{v, r\} \tag{34}$$

where Ψ_{ar}, Ψ_{av}, Φ_{ar}^k, and Φ_{av}^k are regression vectors defined by:

$$\begin{cases} \Psi_{ar} = \left(\alpha \nabla \varphi_{ar}^\top + \nabla \alpha \, \varphi_{ar}^\top \right)^\top \left(F_r + G_r \hat{\mathcal{U}} \right) \\ \Psi_{av} = \left((1-\alpha) \nabla \varphi_{av}^\top - \nabla \alpha \, \varphi_{av}^\top \right)^\top \left(F_r + G_r \hat{\mathcal{U}} \right) \\ \Phi_{ai}^k = \nabla \varphi_{ai}(\Theta^k) \left(F_i(\Theta^k) + G_i(\Theta^k) \hat{U}_i^k \right), \quad i \in \{v, r\} \end{cases} \tag{35}$$

These regression vectors enable efficient calculation of Bellman errors for both virtual and real agents, facilitating the adaptive update of actor-critic NN weights.

To facilitate the learning process, a historical data stack $\{\delta(t), \zeta_v(t), \zeta_r(t), \{\delta^k, \zeta_v^k, \zeta_r^k\}_{k=1}^N\}$ is maintained without extrapolation, where $\{\delta^k, \zeta_v^k, \zeta_r^k\}$ represents the kth stored data collection. The actor-critic NN weights are optimized by minimizing a composite loss function:

$$\mathcal{E} = \delta^\top \delta + \sum_{i=v,r} \sum_{k=1}^{N} \zeta_i^{k\top} \zeta_i^k \tag{36}$$

For the critic NN, a concurrent gradient descent method is employed to update the weights:

$$\begin{aligned} \dot{\hat{W}}_{ci} =& -k_{ci1} \frac{\Psi_{ai}}{\left(\Psi_{ai}^\top \Psi_{ai} + 1 \right)^2} \delta \\ & - \frac{k_{ci2}}{N} \sum_{k=1}^{N} \frac{\Phi_{ai}^k}{\left(\Phi_{ai}^{k\top} \Phi_{ai}^k + 1 \right)^2} \zeta_i^k, \ i \in \{v, r\} \end{aligned} \tag{37}$$

where $k_{cij} > 0$ $(i = v, r, j = 1, 2)$ denotes the learning rates, and the regression vectors are defined as: $\Phi_{ai} = \nabla \varphi_{ci}^\top(\Theta_r)(F + G\hat{U})$, $\Phi_{ai}^k = \nabla \varphi_{ci}^\top(\Theta_r^k)(F + G\hat{U}^k)$, with Θ_r^k being the kth historical state sample.

For the actor NN, a gradient projection mechanism is utilized to update the weights:

$$\dot{\hat{W}}_{ai} = \Gamma \left\{ -k_{ai} \mathcal{F}_{ai} \left(\hat{W}_{ai} - \hat{W}_{ci} \right) \right\}, \ i \in \{v, r\} \tag{38}$$

where $k_{ai} > 0$ represents the learning rate, $\mathcal{F}_{ai} \in \mathbb{R}^{n_\varphi \times n_\varphi}$ denotes a positive definite matrix for weight updates, and Γ is a projection operator ensuring bounded weights. Through this online learning framework, the optimal value function and control input are approximated via the actor-critic architecture.

Remark 7. To address computational complexity challenges associated with high-dimensional spaces, the following computational optimization strategies could be integrated with in the proposed optimal

Algorithm 1 Digital-twin-based optimal shared control algorithm

Initialize the actor-critic NN weights $\hat{W}_{ci}, \hat{W}_{ai}$, learning parameters k_{cij}, k_{ai}, $\mathcal{F}_{ai}$ $(i = v, r,\ j = 1, 2)$, the experience replay stack $\{\delta(t), \{\zeta_v^k(t), \zeta_r^k(t)\}_{k=1}^N\}$, and the Koopman transformed data $\{Y_{\mathcal{K}}, X_{\mathcal{K}}, U_{\mathcal{K}}\}$.
while $t < T_{end}$ **do**
  Collect current state X from operating system.
  if Model of the real-world system is unknown **then**
    $\Psi_{\mathcal{K}_2} \leftarrow$ compute via (14) using $\left[\Theta_r(t)^\top, U_r(t)^\top\right]^\top$
    Store $\Psi_{\mathcal{K}_2}$ in $\{Y_{\mathcal{K}}, X_{\mathcal{K}}, U_{\mathcal{K}}\}$
    Approximate Koopman operator $\mathcal{K}$ via EDMD algorithm (15) and minimization (16)
    F_v, $G_v \leftarrow$ compute via (17)
  end if
  $\hat{V}_i$, $\hat{U}_i \leftarrow$ approximate via (28), (30)
  $\hat{\mathcal{V}}$, $\hat{\mathcal{U}} \leftarrow$ compute via (29), (31)
  $\zeta_i(\Theta_r, \hat{V}_i, \hat{U}_i)$, $\delta(\Theta_r, \hat{\mathcal{V}}, \hat{\mathcal{U}}) \leftarrow$ compute via (34), (32)
  Stack ζ_i and δ in the experience replay stack
  Update weights $\hat{W}_{ci}$, $\hat{W}_{ai}$ via (37)–(38)
  Apply $\hat{\mathcal{U}}$ to system (3)
end while

shared control algorithm: (1) Online concurrent learning: Compared to traditional off-policy and offline learning methods [56,57], online concurrent learning enables real-time training and control updates without requiring extensive historical data storage. This approach enhances computational efficiency and reduces memory requirements, making it well-suited for dynamic and high-dimensional systems; (2) ADP-related optimization techniques: To facilitate efficient training and convergence, methods such as state-following kernel neural networks [58] and sparse learning-based extrapolation [59,60] can be employed to reduce the computational burden of the actor-critic training process. By incorporating these computational optimization strategies, the actor-critic algorithm can be effectively implemented for real-time control applications, ensuring computational feasibility and scalability across diverse system configurations.

The detailed algorithm is presented in Algorithm 1, and the learning process of the optimal shared control algorithm is illustrated in Fig. 3. The proposed method employs actor-critic neural networks (NNs) to approximate the optimal value function and optimal control input. The shared Bellman error is used to train the actor-critic NNs, and the weights are updated online by minimizing this error. In the next section, we provide a theoretical analysis of the proposed optimal shared control algorithm.

4. Stability analysis of the optimal shared control algorithm

In this section, we utilize Lyapunov stability theory to prove that the closed-loop virtual–real system states and the actor-critic NN errors are ultimately uniformly bounded (UUB) under the proposed digital-twin-based optimal shared control algorithm. To facilitate the stability analysis, we first state two assumptions. Then, we analyze the Lyapunov stability of the algorithm.

Assumption 3 (*Boundedness of Parameters and Operators [26,61]*)**.** Assume that the following parameters and operators are bounded:

- $\|\hat{W}_{ci}\| \le W_{Hi}$, $\forall i \in \{v, r\}$
- $\|\Phi_{ai}(\Theta_r)\| \le \Psi_{Hij}$, $\|\nabla\Phi_{ai}(\Theta_r)\| \le \Psi_{D,Hij}$, $\forall i \in \{v, r\}$
- $\|\varphi(\Theta_r)\| \le \varphi_{Hi}$, $\|\nabla\varphi(\Theta_r)\| \le \varphi_{D,Hi}$, $\forall i \in \{v, r\}$
- $\|\varepsilon(\Theta_r)\| \le \varepsilon_{Hi}$, $\|\nabla\varepsilon(\Theta_r)\| \le \varepsilon_{D,Hi}$, $\forall i \in \{v, r\}$

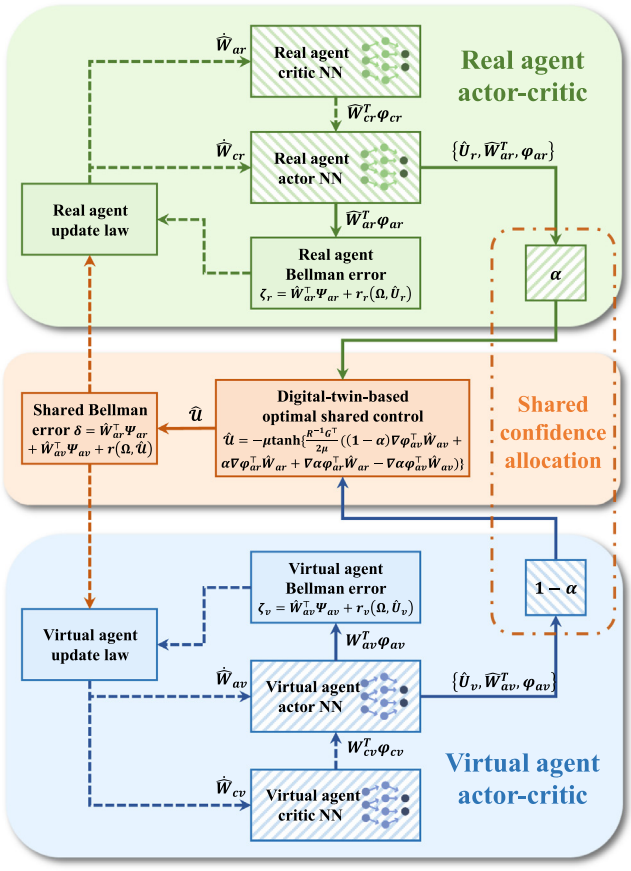

Fig. 3. The learning process of the optimal shared control algorithm.

Assumption 4 (*Persistent Excitation Condition [25,49]*)**.** Assume that the data set collected online and used for the weight update law satisfies the following excitation condition:

1. $\phi_{1,i} I_{\mathcal{K}} \le \inf_{t \ge t_0} \left\{ \frac{1}{N} \sum_{k=1}^{N} \frac{\Phi_{ai}^k \Phi_{ai}^{k\top}}{\rho_i^k} \right\},\ i \in \{v, r\}$
2. $\phi_{2,i} I_{\mathcal{K}} \le \int_t^{t+T} \frac{\Psi_{ai} \Psi_{ai}^\top}{\rho_i} \mathrm{d}\tau,\ i \in \{v, r\}$

where $I_{\mathcal{K}}$ is the identity matrix, $\rho_i^k = (\Phi_{ai}^{k\top} \Phi_{ai}^k + 1)^2$, $\rho_i = (\Psi_{ai}^\top \Psi_{ai} + 1)^2$, and at least one of the non-negative constants $\phi_{1,i}$ or $\phi_{2,i}$ is positive.

Remark 8. It is important to note that Assumptions 3 and 4 are standard within the stability analysis of actor-critic algorithms [34,49,62]. Assumption 3 ensures that the parameters and operators remain bounded, whereas Assumption 4 guarantees the persistent excitation condition for the dataset utilized in the weight update law. The excitation condition is crucial for the convergence of the actor-critic algorithm, and the boundedness of parameters and operators is a common premise in stability analysis. These two assumptions can be satisfied by selecting appropriate learning rates and initial weights for the actor-critic neural networks.

Based on the design of input (31), we derive an inequality for the control error as follows:

$$\left\| U_i^*(\Theta_r) - \hat{U}_i(\Theta_r) \right\|^2 \le \zeta_i \tilde{W}_{ai}^\top \tilde{W}_{ai} + \Pi_{u_i}, \quad i \in \{v, r\} \tag{39}$$

where $\tilde{W}_{*i} = \hat{W}_{*i} - W_{*i}$ represents the estimation error of the NN weights, Ψ is an upper bound associated with φ_H, $\varphi_{D,H}$, Ψ_{Hi}, and $\Psi_{D,Hi}$, and Π_u is an upper bound related to $\varepsilon_{D,H}$.

The Bellman error δ_i is defined as:

$$\delta = -\Psi_{a1}^{\top}\tilde{W}_{c1} - \Psi_{a2}^{\top}\tilde{W}_{c2} + \Delta(\Theta_r) + \xi_H, \tag{40}$$

$$\zeta_i^k = -(\Phi_{ai}^k)^{\top}\tilde{W}_{ci} + \frac{1}{4}\tilde{W}_{ai}G_{\Phi_{ai}}^k\tilde{W}_{ai} + \zeta_i^k(\Theta_r), \tag{41}$$

where $i \in \{v, r\}$, $G_{\Phi_{ai}}^k = G_{\Phi_{ai}}(\Theta_r^k)$, and Δ_i, $\zeta_i^k : \mathbb{R}^n \to \mathbb{R}$ are uniformly bounded over χ. Additionally, $\overline{\|\Delta_i\|}$ and $\overline{\|\zeta_i^k\|}$ decrease as $\overline{\|\nabla\epsilon_i\|}$ and $\overline{\|\nabla W_i\|}$ decrease.

The stability analysis of the closed-loop system state and the NN weight estimation errors is provided in the following theoretical result.

Theorem 1. *Consider the system dynamics in (3) and the proposed optimal shared control algorithm. Assume that Assumptions 1, 3, and 4 are satisfied. Let the actor-critic neural networks (NNs) be updated using the adaptive update laws (37) and (38), and let the control input be estimated by (31). Then, the closed-loop system states Θ_r and the weight errors*

$$\left[\tilde{W}_{cr}^{\top}, \tilde{W}_{ar}^{\top}, \tilde{W}_{cv}^{\top}, \tilde{W}_{av}^{\top}\right]^{\top}$$

are ultimately uniformly bounded (UUB) provided that:

$$\|\Xi\| \geq (\Gamma_{\text{res}}/\underline{\lambda}_{\mathcal{M}})^{1/2} \tag{42}$$

where $\Xi = \left[\Theta_r^{\top}, \tilde{W}_{cr}^{\top}, \tilde{W}_{ar}^{\top}, \tilde{W}_{cv}^{\top}, \tilde{W}_{av}^{\top}\right]^{\top}$, Γ_{res} is a positive constant, and $\underline{\lambda}_{\mathcal{M}}$ is the minimum eigenvalue of the matrix $\mathcal{M}$.

Proof. Utilizing Lyapunov stability theory, we select the Lyapunov function as

$$V(\Xi) = \mathcal{V}^*(\Theta) + \frac{1}{2}\tilde{W}_{cv}^{\top}\tilde{W}_{cv} + \frac{1}{2}\tilde{W}_{av}^{\top}\tilde{W}_{av} + \frac{1}{2}\tilde{W}_{cr}^{\top}\tilde{W}_{cr} + \frac{1}{2}\tilde{W}_{ar}^{\top}\tilde{W}_{ar} \tag{43}$$

With the optimal value functions (24) and optimal control inputs (23), and taking the time derivative of V along the system trajectories, we obtain (see Table 1):

$$\dot{V} = \nabla\mathcal{V}^* \cdot \left(F_r(\Theta_r) + G_r(\Theta_r)\mathcal{U}^*\right) + \tilde{W}_{cv}^{\top}\dot{\hat{W}}_{cv} + \tilde{W}_{av}^{\top}\dot{\hat{W}}_{av} + \tilde{W}_{cr}^{\top}\dot{\hat{W}}_{cr} + \tilde{W}_{ar}^{\top}\dot{\hat{W}}_{ar} \tag{44}$$

Substituting the $(\nabla\mathcal{V}^*)^{\top} F(\Theta)$ term from (40) and (41) into (44), and employing the Bellman errors from (40) and (41), the derivative can be rewritten as:

$$\begin{aligned}
\dot{V} = {} & \left\{-\Theta_r^{\top}Q_r\Theta_r - \psi_r(U_r)\right. \\
& - \left\{k_{av}\tilde{W}_{av}^{\top}\mathcal{F}_{av}\left(\hat{W}_{av} - \hat{W}_{cv}\right)\right\} - \left\{k_{ar}\tilde{W}_{ar}^{\top}\mathcal{F}_{ar}\left(\hat{W}_{ar} - \hat{W}_{cr}\right)\right\} \\
& + \left\{k_{cv1}\tilde{W}_{cv}^{\top}\frac{\Phi_{av}}{\rho_v}\left(-\Phi_{av}^{\top}\tilde{W}_{cv} + \frac{1}{4}\tilde{W}_{av}^{\top}G_{\Phi_{av}}\tilde{W}_{av} + \Delta_v + \xi_{Hv}\right)\right\} \\
& + \left\{k_{cr1}\tilde{W}_{cr}^{\top}\frac{\Phi_{ar}}{\rho_r}\left(-\Phi_{ar}^{\top}\tilde{W}_{cr} + \frac{1}{4}\tilde{W}_{ar}^{\top}G_{\Phi_{ar}}\tilde{W}_{ar} + \Delta_r + \xi_{Hr}\right)\right\} \\
& + \left(\tilde{W}_{cv}^{\top}\frac{k_{cv2}}{N}\sum_{k=1}^{N}\frac{\Phi_{av}^k}{\rho_v^k}\left(\frac{1}{4}\tilde{W}_{av}^{\top}G_{\Phi_{av}}^k\tilde{W}_{av} - (\Phi_{av}^k)^{\top}\tilde{W}_{av} + \Delta_v^k\right)\right) \\
& \left. + \left(\tilde{W}_{cr}^{\top}\frac{k_{cr2}}{N}\sum_{k=1}^{N}\frac{\Phi_{ar}^k}{\rho_r^k}\left(\frac{1}{4}\tilde{W}_{ar}^{\top}G_{\Phi_{ar}}^k\tilde{W}_{ar} - (\Phi_{ar}^k)^{\top}\tilde{W}_{ar} + \Delta_r^k\right)\right)\right\}
\end{aligned}$$

Substitute inequality (39), then employ Young's inequality [25] and Assumptions 1–4. The derivative can then be rewritten as:

$$\dot{V} \leq -\Xi^{\top}\mathcal{M}\Xi + \Gamma_{\text{res}} \tag{45}$$

where $\mathcal{M}$ is a positive definite matrix defined as:

$$\mathcal{M} = \begin{bmatrix} m_1 & 0 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 & 0 \\ 0 & m_3 & m_4 & 0 & 0 \\ 0 & 0 & 0 & m_5 & 0 \\ 0 & 0 & 0 & m_6 & m_7 \end{bmatrix}$$

where m_i are functions of the system parameters and the basis functions. The functions are defined as:

- $m_1 = \underline{\lambda}_{Q_r}$
- $m_2 = \frac{1}{2}k_{cr1}\Psi_r\Psi_r^{\top} + \frac{1}{2}k_{cr2}\phi_{1,r}I_{\mathcal{K}}$
- $m_3 = \frac{1}{2}k_{cr1}\Psi_r\Psi_r^{\top} + \frac{1}{2}k_{cr1}\phi_{1,r}I_{\mathcal{K}}$
- $m_4 = \mathcal{F}_{a1}I_{\mathcal{K}} - \bar{\lambda}_{R_{aa}}\Psi_u I_{\mathcal{K}}$
- $m_5 = \frac{1}{2}k_{cv1}\Psi_v\Psi_v^{\top} + \frac{1}{2}k_{cv2}\phi_{1,v}I_{\mathcal{K}}$
- $m_6 = \frac{1}{2}k_{cv1}\Psi_v\Psi_v^{\top} + \frac{1}{2}k_{cv2}\phi_{1,v}I_{\mathcal{K}}$
- $m_7 = \mathcal{F}_{av}I_{\mathcal{K}} - \bar{\lambda}_{R_{hh}}\Psi_u I_{\mathcal{K}}$

and Γ_{res} is a residual defined as:

$$\Gamma_{\text{res}} = \sum_{i=v,r}\left\{\bar{\lambda}_{R_{ii}}\Pi_{ui} + \Psi_j\tilde{W}_{aj}^{\top}\tilde{W}_{aj} + \frac{1}{2}k_{ci1}\left(\frac{1}{4}\tilde{W}_{ai}^{\top}G_{\Phi_{ai}}\tilde{W}_{ai} + \Delta_i\right)^2 + \frac{1}{2}k_{ci2}\left(\frac{1}{4}\tilde{W}_{ai}^{\top}G_{\Phi_{ai}}\tilde{W}_{ai} + \Delta^k\right)^2\right\}$$

When a suitable positive definite matrix $\mathcal{M}$ is chosen, the closed-loop system state Θ and the network weight errors $\left[\tilde{W}_{cr}^{\top}, \tilde{W}_{ar}^{\top}, \tilde{W}_{cv}^{\top}, \tilde{W}_{av}^{\top}\right]^{\top}$ are ultimately uniformly bounded (UUB) provided that:

$$\|\Xi\| \geq (\Gamma_{\text{res}}/\underline{\lambda}_{\mathcal{M}})^{1/2} \tag{46}$$

The proof is completed. □

5. Simulation verifications

5.1. Example 1: Representative nonlinear system control

Simulation setup

In this example, the nonlinear system is designed in 2-dimensional space. The dynamic model of this system is selected a representative nonlinear dynamics system from [46,49]:

$$F_r(\Theta_r) = \begin{bmatrix} -\Theta_r(1) + \Theta_r(2) \\ -\frac{1}{2}\Theta_r(1) - \frac{1}{2}\Theta_r(2)\left(1 - \left(\cos\left(2\Theta_r(1)\right) + 2\right)^2\right) \end{bmatrix},$$

$$G_r(\Theta_r) = \begin{bmatrix} \sin\left(2\Theta_r(1)\right) + 2 & 0 \\ 0 & \cos\left(2\Theta_r(1)\right) + 2 \end{bmatrix}$$

where $\Theta_r(1)$ and $\Theta_r(2)$ are the state variables, and $F_r(\Theta_r)$ and $G_r(\Theta_r)$ are the system dynamics matrices. The control objective is to stabilize the system states at the origin. Simulations are conducted using MATLAB R2023b Simulink on a PC equipped with an Intel Core i3-12100F CPU (3.3 GHz) and 24 GB of RAM. The ODEs are solved using the fourth-order Runge–Kutta method with a fixed step size of $T = 0.001$ s. The

Table 1
The parameters of examples.

Parameters	Initial conditions	Controller design	Learning parameters
Example 1 :	$\Theta_r(0) = [2, 3]^T$, $\Theta_v(0) = [2.5, 2.5]^T$	$R_r = 10I_2$, $R_v = 20I_2$	$k_{cr1} = 0.5$, $k_{cv1} = 0.5$
Representive nonlinear system	$\hat{W}_{cr} = \hat{W}_{ar} = 0.1 \times \mathbf{1}_6 + rand(12)$ $\hat{W}_{cv} = \hat{W}_{av} = 0.1 \times \mathbf{1}_6 + rand(12)$	$Q_r = 2I_2$, $Q_v = 1I_2$ $\mu_r = 0.5$, $\mu_v = 1.5$	$k_{cr2} = 0.1$, $k_{cv2} = 0.1$ $k_{ar} = k_{av} = 1, \mathcal{F}_r = \mathcal{F}_v = I_6$
Example 2 :	$\Theta_i(0) = [-2.198, 0.753, -0.48, 0.103,$ $-0.008, -0.057, -0.0013, 0.027,$ $0.534, -0.045, 0.032, -0.058]$	$R_i = 50diag([3, 1, 0.5, 1])$ $Q_i = diag([10, 10, 10, 1,$ $1, 1, 1, 1, 1, 1, 1, 0])$	$k_{cr1} = 0.02$, $k_{cv1} = 0.01$ $k_{cr2} = 0.01$, $k_{cv2} = 0.01$
Unmanned aerial vehicle system	$\hat{W}_{ij} = \mathbf{1}_{12} + rand(12), (i = c, a, j = c, v)$	$\mu_r = 0.5$, $\mu_v = 1.5$	$k_{ar} = 0.05$, $k_{av} = 0.05$ $\mathcal{F}_r = 0.1I_6$, $\mathcal{F}_v = 0.1I_6$

simulation runs for a total duration of $t_{\text{end}} = 5$ s. The actor–critic neural networks for both virtual and real systems are structured as follows:

$$\varphi_{ij} = [\Theta_i(1)^2, \Theta_i(1)\times\Theta_i(2), \Theta_i(2)^2, \Theta_i(1)^2\times\Theta_i(2), \Theta_i(1)\times\Theta_i(2)^2, \Theta_i(1)^2\times\Theta_i(2)^2],\ \forall i \in \{c,a\}, \forall j \in \{r,v\}$$

The corresponding gradient of the actor-critic NNs are given as:

$$\nabla\varphi_{ij} = \begin{bmatrix} \Theta_i(1), \Theta_i(2), 0, 2\Theta_i(1)\times\Theta_i(2), \Theta_i(2)^2, 2\Theta_i(1)\times\Theta_i(2)^2 \\ 0, \Theta_i(1), 2\Theta_i(2), \Theta_i(1)^2, 2\Theta_i(1)\times\Theta_i(2), 2\Theta_i(1)^2\times\Theta_i(2) \end{bmatrix}^\top$$

where $i = c, a$ and $j = r, v$. Note that the actor-critic neural networks are updated online using the adaptive update laws (37) and (38). The selection of the NN basis functions and their corresponding gradients plays a crucial role in the convergence and stability of the actor-critic algorithm, as well as the overall performance of the shared control system. In our design, we have chosen the above NN basis functions based on the system dynamics and control objectives by considering the system states and the system dynamics matrices, inspired by the literature on actor-critic algorithms [46,61]. The detailed parameters for the controller design are shown in Table 1.

Simulation results

The results of the proposed optimal shared control algorithm are shown in Figs. 4–8. The weights of the critic and actor NNs in the virtual and real agents are shown in Fig. 4, where the weights of the critic and actor NNs are updated online by the proposed adaptive update law. The state trajectory of the virtual and real systems are shown in Fig. 5, in which the state trajectory of the virtual and real agents are stable. The detailed state stability of the virtual and real

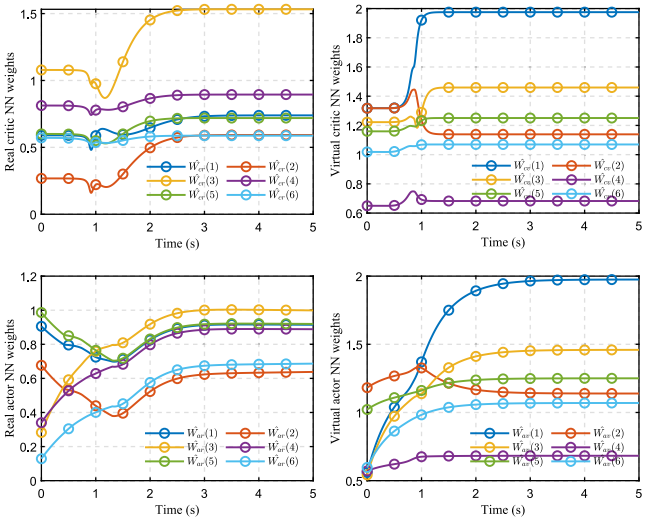

Fig. 4. The weights of the critic and actor NNs in the virtual and real agents.

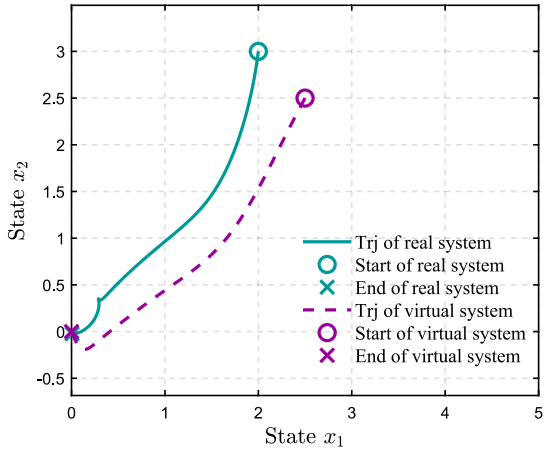

Fig. 5. State trajectory of exp 1.

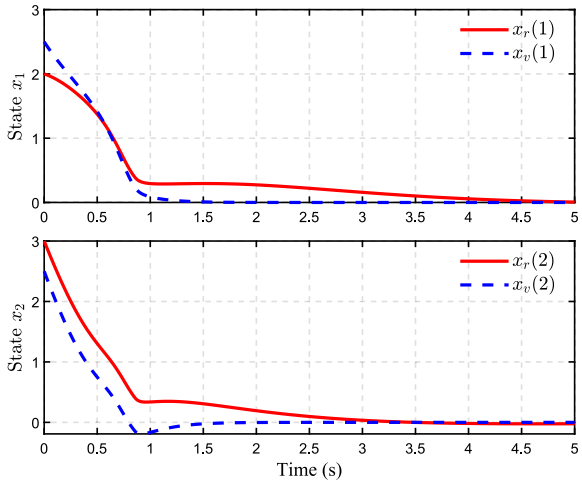

Fig. 6. State stability of exp 1.

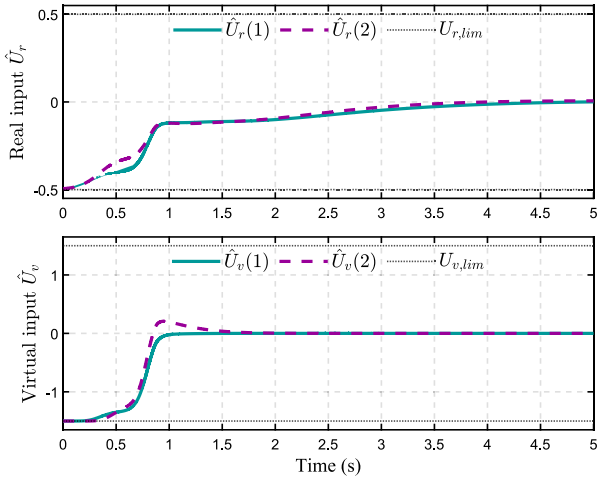

Fig. 7. Control input of exp 1.

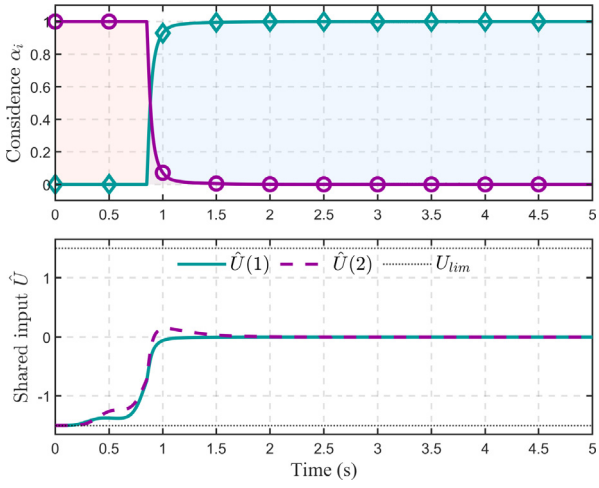

Fig. 8. Confidence allocation of exp 1.

agents are shown in Fig. 6, where all the virtual and real system states are bounded and stable. Fig. 7 shows the control input of the virtual and real agents, where the control inputs of the virtual and real agents are bounded. The confidence allocation of the virtual and real agents is shown in Fig. 8. During 0 ∼ 1 s, the confidence of the virtual agent is higher than the real agent, which means that the virtual agent is more confident in the control input. After 1 s, the confidence of the real agent is higher than the virtual agent, which means that the real agent is more confident in the control input. The results show that the proposed optimal shared control algorithm is able to approximate the optimal value function and control input by the actor-critic NNs.

5.2. Example 2: Unmanned aerial vehicle control

Simulation setup

In this example, we consider the control of an unmanned aerial vehicle (UAV) in the 3-dimensional space. The human-UAV cooperation system is illustrated in Fig. 9, which consists of an expert operator and a UAV. The frame of the UAV is defined as the body frame with the origin at the center of mass of the UAV. The body frame is defined as x_b pointing forward, y_b pointing to the right, and z_b pointing up. The expert operator is responsible for observing UAV states and sending control inputs to the UAV in the north-east-down (NED) frame. Consider the UAV in the earth-fixed frame, the position state of the UAV in the earth-fixed frame is composed of x, y, and z, $\dot{x}, \dot{y}$, and $\dot{z}$ are the corresponding velocities, ϕ, θ, ψ, $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ denote the angular and corresponding angular velocity. Define the control input as $\dot{x}_d$, $\dot{y}_d$, $\dot{z}_d$, and $\dot{\psi}_d$, which are the desired velocities. Assuming that the angles are small enough to apply the small angle approximation, the UAV dynamics in the NED frame is given by

$$\begin{cases} \ddot{x} = -\dfrac{k_t \dot{x}}{m} - g\theta \\ \ddot{y} = -\dfrac{k_t \dot{y}}{m} + g\phi \\ \ddot{z} = -\dfrac{k_t \dot{z}}{m} + h_{z_1}\left(\dot{z}_d - \dot{z}\right) \\ \ddot{\phi} = -\dfrac{l h_{\phi_1}}{I_{xx}}\dot{\phi} - \dfrac{l h_{\phi_2}}{I_{xx}}\phi + \dfrac{\pi l h_{\phi_2} h_{y_1}}{4\, g I_{xx}}\left(\dot{y} - \dot{y}_d\right) \\ \ddot{\theta} = -\dfrac{l h_{\theta_1}}{I_{yy}}\dot{\theta} - \dfrac{l h_{\theta_2}}{I_{yy}}\theta + \dfrac{\pi l h_{\theta_2} h_{x_1}}{4\, g I_{yy}}\left(\dot{x}_d - \dot{x}\right) \\ \ddot{\psi} = \dfrac{l h_{\psi_1}}{I_{zz}}\left(\dot{\psi}_d - \dot{\psi}\right) \end{cases} \tag{47}$$

where l is the distance from the center of mass to the rotor, m is the mass of the UAV, g is the gravitational acceleration, k_t is coefficient of aerodynamic drag, I_{xx}, I_{yy}, and I_{zz} are the inertia of the UAV about the x, y, and z axes, respectively. Define the state vector as $\Theta_r = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]^\top$, and the control input as $\mathcal{U} = [\dot{x}_d, \dot{y}_d, \dot{z}_d, \dot{\psi}_d]^\top$. The actor-critic NNs for the virtual and real systems are designed as the following form:

$$\begin{aligned} \varphi_{ij} =& [x \times \dot{\theta}, \dot{x} \times \dot{\theta}, \psi \times \dot{\theta}, \dot{\theta}^2, y \times \dot{\phi}, z \times \dot{\phi}, \phi \times \dot{\phi}, \dot{\phi}^2, \\ & z \times \dot{z}, \dot{z}^2, \psi \times \dot{\psi}, \psi^2], \quad \forall i \in \{c, a\},\ \forall j \in \{r, v\} \end{aligned} \tag{48}$$

The detailed parameters of the UAV control system are listed in Table 2. The simulation environment is implemented in MATLAB R2023b Simulink on a PC equipped with an Intel Core i3-12100F CPU (3.3 GHz) and 24 GB of RAM. The ODEs are solved using the fourth-order Runge–Kutta method with a fixed step size of $T = 0.001$ s. The simulation runs for a total duration of $t_{\text{end}} = 5$ s. In this example, to demonstrate the effectiveness of the proposed optimal shared control algorithm under different confidence setups, three cases with different setting of the confidence parameters are considered, and the parameters of three cases are shown in Table 3.

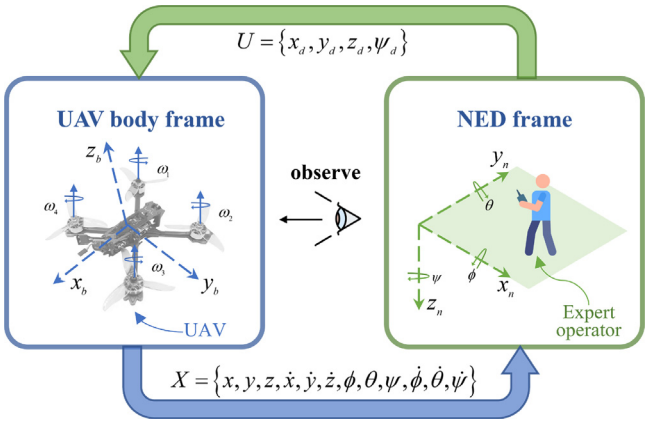

Fig. 9. The configuration of the human-UAV cooperation system.

Table 2
Parameters of the UAV control system and the update law.

Parameters	Value
UAV model	$I_{xx} = 0.00226$ kg m^2, $m = 0.5799$ kg $I_{yy} = 0.00282$ kg m^2, $g = 9.81$ m/s^2 $I_{zz} = 0.0021$ kg m^2, $k_t = 0.01$ (s kg)$^{-1}$
Autopilot gains	$h_{x_1} = -5.25, h_{y_1} = -5.25, h_{z_1} = 3.0$ $h_{\phi_2} = 3.50,\ h_{\theta_2} = 3.50,\ h_{\psi_2} = 0.35$ $h_{\phi_1} = 0.40,\ h_{\theta_1} = 0.40,\ h_{\psi_1} = 0.10$

Table 3
Parameters of three cases.

Parameters	Case 1	Case 2	Case 3
θ_1	0°	45°	90°
θ_2	180°	135°	90°

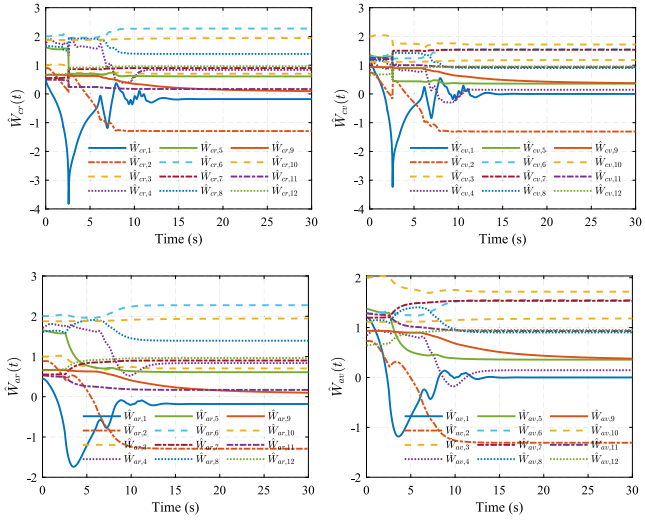

Fig. 10. The weights of the critic and actor NNs in the virtual and real agents.

Simulation results

The detailed results of the UAV control system are shown in Figs. 10–19. The weights of the critic and actor NNs in the virtual and real agents are shown in Fig. 10, where the weights of the critic and actor NNs are updated online by the proposed adaptive update law, and the weights of the critic and actor NNs are bounded. Fig. 11 shows the comparison results of the UAV trajectory with three different parameters. The evolution of the UAV states are shown in Figs. 12–17, in which the UAV states $x, y, z, \phi, \theta, \psi$ are stable. The Bellman errors

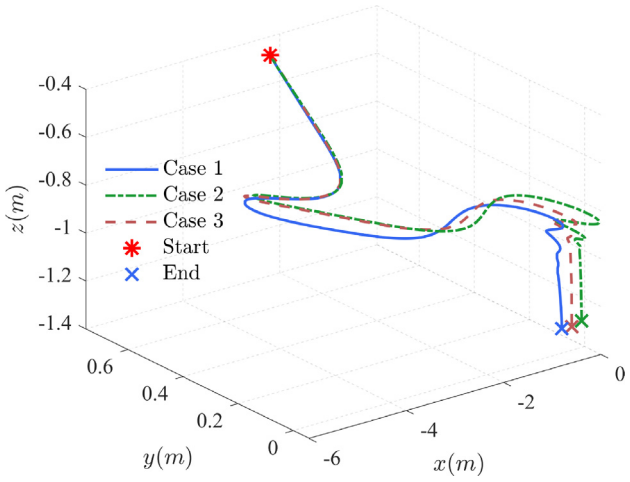

Fig. 11. Comparison results of the UAV trajectory.

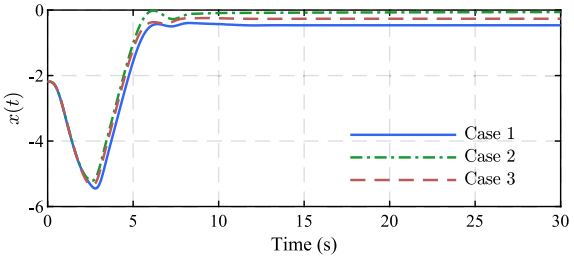

Fig. 12. Position x.

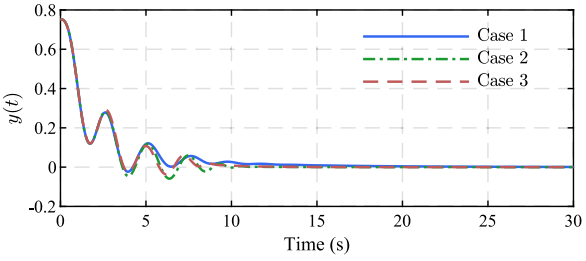

Fig. 13. Position y.

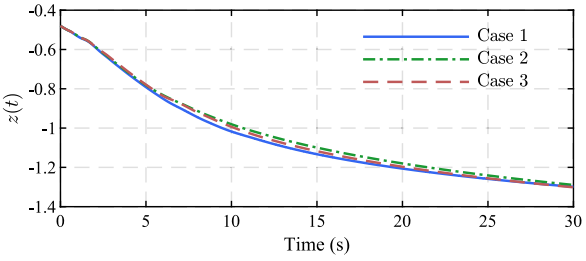

Fig. 14. Position z.

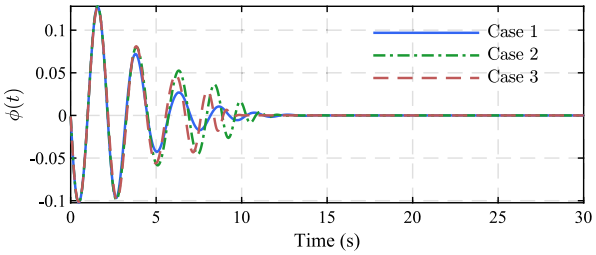

Fig. 15. Attitude ϕ.

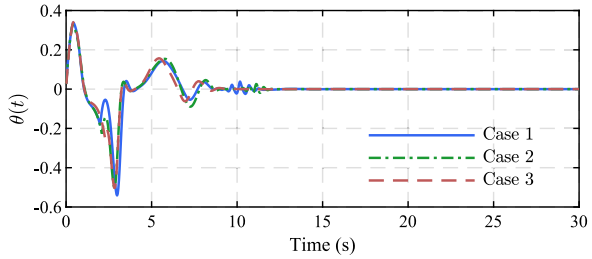

Fig. 16. Attitude θ.

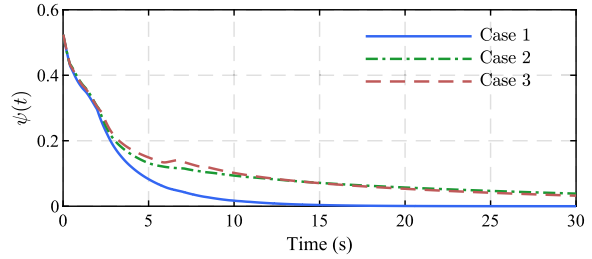

Fig. 17. Attitude ψ.

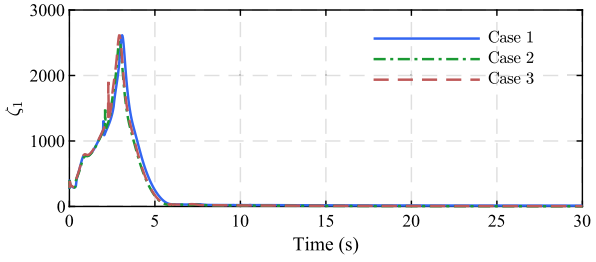

Fig. 18. Bellman error of the virtual agent.

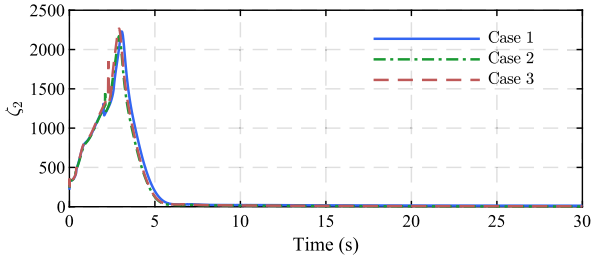

Fig. 19. Bellman error of the real agent.

of the UAV control system are shown in Figs. 18 and 19. The results show that the proposed optimal shared control algorithm is able to approximate the optimal value function and control input by the actor-critic NNs. The shared Bellman error is utilized to train the actor-critic NNs and approximate the optimal value function and control input. All the states of the UAV control system are stable and bounded in three different cases, which means that the proposed optimal shared control algorithm is able to achieve the optimal shared control input in the UAV control system. The results demonstrate the effectiveness of the proposed optimal shared control algorithm in the UAV control system under different confidence setups.

For real-world applications, the proposed algorithm can be utilized to enhance the performance of human-UAV cooperation systems. In emergencies where the expert operator is unavailable, the proposed

algorithm can automatically switch to the virtual agent to ensure the continuous operation of the UAV control system. In scenarios such as obstacle avoidance and path planning, the proposed algorithm can integrate the expert operator's supervision with the AI agent's optimization to achieve the optimal shared control input, enabling the UAV to avoid obstacles and plan paths effectively. In mission planning tasks, the human operator can provide high-level commands to the UAV, while the AI agent can act as the low-level controller.

6. Conclusion

This paper proposes a human–AI interactive optimized shared control algorithm implemented through a digital twin framework. The proposed algorithm obtains optimal shared control input within a digital-twin framework, relying on a real agent and a virtual agent to approximate the optimal value function and control input through actor-critic neural networks. The optimal shared control input is a composite of the virtual and real optimal inputs, incorporating a confidence allocation mechanism to distribute confidence between the virtual and real agents. The Lyapunov stability theory is employed to analyze the stability of the closed-loop system. Finally, examples of a nonlinear system and a UAV control system are presented to validate the effectiveness of the proposed algorithm. Future work will concentrate on the practical application within human-UAV control systems.

CRediT authorship contribution statement

Junkai Tan: Writing – original draft, Visualization, Software, Methodology, Conceptualization. **Shuangsi Xue:** Writing – original draft, Methodology, Funding acquisition, Conceptualization. **Hui Cao:** Project administration, Methodology, Conceptualization, Funding acquisition. **Shuzhi Sam Ge:** Supervision, Resources, Conceptualization, Methodology, Writing - Original Draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.Shuzhi Sam Ge is an editorial board member for this journal and was not involved in the editorial review or the decision to publish this article.

Acknowledgment

This research is supported by China Postdoctoral Science Foundation (Project ID: 2024M762602), the National Natural Science Foundation of China under Grant No. 62306232, and Natural Science Basic Research Program of Shaanxi Province under Grant No. 2023-JC-QN-0662.

Data availability

No data was used for the research described in the article.

References

[1] B. Liu, P. Rocco, A.M. Zanchettin, F. Zhao, G. Jiang, X. Mei, A real-time hierarchical control method for safe human–robot coexistence, Robot. Comput.-Integr. Manuf. 86 (2024) 102666, http://dx.doi.org/10.1016/j.rcim.2023.102666.

[2] H. Hu, D. Isele, S. Bae, J.F. Fisac, Active uncertainty reduction for safe and efficient interaction planning: A shielding-aware dual control approach, Int. J. Robot. Res. (2023) 02783649231215371, http://dx.doi.org/10.1177/02783649231215371.

[3] P. Hang, C. Lv, C. Huang, Y. Xing, Z. Hu, Cooperative decision making of connected automated vehicles at multi-lane merging zone: A coalitional game approach, IEEE Trans. Intell. Transp. Syst. 23 (4) (2022) 3829–3841, http://dx.doi.org/10.1109/TITS.2021.3069463.

[4] G.-P. Liu, Control strategies for digital twin systems, IEEE/CAA J. Autom. Sin. 11 (1) (2024) 170–180, http://dx.doi.org/10.1109/JAS.2023.123834.

[5] D. An, Y. Chen, Non-intrusive soil carbon content quantification methods using machine learning algorithms: A comparison of microwave and millimeter wave radar sensors, J. Autom. Intell. 2 (3) (2023) 152–166, http://dx.doi.org/10.1016/j.jai.2023.09.001.

[6] F. Tao, H. Zhang, A. Liu, A.Y.C. Nee, Digital twin in industry: State-of-the-art, IEEE Trans. Ind. Inform. 15 (4) (2019) 2405–2415, http://dx.doi.org/10.1109/TII.2018.2873186.

[7] C. Ruah, O. Simeone, B.M. Al-Hashimi, A Bayesian framework for digital twin-based control, monitoring, and data collection in wireless systems, IEEE J. Sel. Areas Commun. 41 (10) (2023) 3146–3160, http://dx.doi.org/10.1109/JSAC.2023.3310093.

[8] R. He, G. Chen, C. Dong, S. Sun, X. Shen, Data-driven digital twin technology for optimized control in process systems, ISA Trans. 95 (2019) 221–234, http://dx.doi.org/10.1016/j.isatra.2019.05.011.

[9] J. Liu, X. Liu, J.r. Vatn, S. Yin, A generic framework for qualifications of digital twins in maintenance, J. Autom. Intell. 2 (4) (2023) 196–203, http://dx.doi.org/10.1016/j.jai.2023.07.002.

[10] J. Leng, H. Zhang, D. Yan, Q. Liu, X. Chen, D. Zhang, Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop, J. Ambient Intell. Humaniz. Comput. 10 (3) (2019) 1155–1166, http://dx.doi.org/10.1007/s12652-018-0881-5.

[11] A.-J. Wang, H. Li, Z. He, Y. Tao, H. Wang, M. Yang, D. Savic, G.T. Daigger, N. Ren, Digital twins for wastewater treatment: A technical review, Engineering 36 (2024) 21–35, http://dx.doi.org/10.1016/j.eng.2024.04.012.

[12] Z. Lei, H. Zhou, X. Dai, W. Hu, G.-P. Liu, Digital twin based monitoring and control for DC-DC converters, Nature Commun. 14 (1) (2023) 5604, http://dx.doi.org/10.1038/s41467-023-41248-z.

[13] F. Mehmood, L. Hadjidemetriou, P.M. Papadopoulos, M.M. Polycarpou, Synergistic frameworks for sensor fault isolation and accommodation in grid-side converters, J. Autom. Intell. (2024) http://dx.doi.org/10.1016/j.jai.2024.09.001.

[14] C. Gehrmann, M. Gunnarsson, A digital twin based industrial automation and control system security architecture, IEEE Trans. Ind. Inform. 16 (1) (2020) 669–680, http://dx.doi.org/10.1109/TII.2019.2938885.

[15] Y. Liu, H. Xu, D. Liu, L. Wang, A digital twin-based sim-to-real transfer for deep reinforcement learning-enabled industrial robot grasping, Robot. Comput.-Integr. Manuf. 78 (2022) 102365, http://dx.doi.org/10.1016/j.rcim.2022.102365.

[16] A. Perrusquía, Human-behavior learning: A new complementary learning perspective for optimal decision making controllers, Neurocomputing 489 (2022) 157–166, http://dx.doi.org/10.1016/j.neucom.2022.03.036.

[17] Q. Zhang, Y. Kang, Y.-B. Zhao, P. Li, S. You, Traded control of human–machine systems for sequential decision-making based on reinforcement learning, IEEE Trans. Artif. Intell. 3 (4) (2022) 553–566, http://dx.doi.org/10.1109/TAI.2021.3127857.

[18] J. Tan, S. Xue, H. Cao, H. Li, Nash equilibrium solution based on safety-guarding reinforcement learning in nonzero-sum game, in: 2023 International Conference on Advanced Robotics and Mechatronics, ICARM, 2023, pp. 630–635, http://dx.doi.org/10.1109/ICARM58088.2023.10218910.

[19] M. Li, J. Qin, J. Li, Q. Liu, Y. Shi, Y. Kang, Game-based approximate optimal motion planning for safe human-swarm interaction, IEEE Trans. Cybern. (2023) 1–12, http://dx.doi.org/10.1109/TCYB.2023.3340659.

[20] K. Tong, M. Li, J. Qin, Q. Ma, J. Zhang, Q. Liu, Differential game-based control for nonlinear human–robot interaction system with unknown desired trajectory, IEEE Trans. Cybern. (2024) 1–11, http://dx.doi.org/10.1109/TCYB.2024.3402353.

[21] J. Tan, S. Xue, H. Cao, H. Li, Safe human-machine cooperative game with level-k rationality modeled human impact, in: 2023 IEEE International Conference on Development and Learning, ICDL, 2023, pp. 188–193, http://dx.doi.org/10.1109/ICDL55364.2023.10364413.

[22] Z. Jin, A. Liu, W.-A. Zhang, L. Yu, C.-Y. Su, A learning based hierarchical control framework for human–robot collaboration, IEEE Trans. Autom. Sci. Eng. 20 (1) (2023) 506–517, http://dx.doi.org/10.1109/TASE.2022.3161993.

[23] A. Franchi, C. Secchi, M. Ryll, H.H. Bulthoff, P.R. Giordano, Shared control : Balancing autonomy and human assistance with a group of quadrotor UAVs, IEEE Robot. Autom. Mag. 19 (3) (2012) 57–68, http://dx.doi.org/10.1109/MRA.2012.2205625.

[24] M. Marcano, S. Díaz, J. Pérez, E. Irigoyen, A review of shared control for automated vehicles: Theory and applications, IEEE Trans. Hum.-Mach. Syst. 50 (6) (2020) 475–491, http://dx.doi.org/10.1109/THMS.2020.3017748.

[25] Q. Ma, P. Jin, F.L. Lewis, Guaranteed cost attitude tracking control for uncertain quadrotor unmanned aerial vehicle under safety constraints, IEEE/CAA J. Autom. Sin. 11 (6) (2024) 1447–1457, http://dx.doi.org/10.1109/JAS.2024.124317.

[26] A. Perrusquia, W. Guo, Optimal control of nonlinear systems using experience inference human-behavior learning, IEEE/CAA J. Autom. Sin. 10 (1) (2023) 90–102, http://dx.doi.org/10.1109/JAS.2023.123009.

[27] W. Chen, Q. Wei, A new optimal adaptive backstepping control approach for nonlinear systems under deception attacks via reinforcement learning, J. Autom. Intell. 3 (1) (2024) 34–39, http://dx.doi.org/10.1016/j.jai.2023.11.001.

[28] H.-N. Wu, M. Wang, Human-in-the-loop behavior modeling via an integral concurrent adaptive inverse reinforcement learning, IEEE Trans. Neural Netw. Learn. Syst. (2023) 1–12, http://dx.doi.org/10.1109/TNNLS.2023.3259581.
[29] W. Jin, T.D. Murphey, Z. Lu, S. Mou, Learning from human directional corrections, IEEE Trans. Robot. 39 (1) (2023) 625–644, http://dx.doi.org/10.1109/TRO.2022.3190221.
[30] W. Xue, B. Lian, Y. Kartal, J. Fan, T. Chai, F.L. Lewis, Model-free inverse H-infinity control for imitation learning, IEEE Trans. Autom. Sci. Eng. (2024) 1–12, http://dx.doi.org/10.1109/TASE.2024.3427657.
[31] S. Xue, B. Luo, D. Liu, Y. Gao, Event-triggered ADP for tracking control of partially unknown constrained uncertain systems, IEEE Trans. Cybern. 52 (9) (2022) 9001–9012, http://dx.doi.org/10.1109/TCYB.2021.3054626.
[32] K. Xia, C. Sacco, M. Kirkpatrick, C. Saidy, L. Nguyen, A. Kircaliali, R. Harik, A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence, Digital Twin towards Smart Manufacturing and Industry 4.0, J. Manuf. Syst. 58 (2021) 210–230, http://dx.doi.org/10.1016/j.jmsy.2020.06.012.
[33] D. Wang, H. Ma, J. Qiao, Multilayer adaptive critic design with digital twin for data-driven optimal tracking control and industrial applications, Eng. Appl. Artif. Intell. 133 (2024) 108228, http://dx.doi.org/10.1016/j.engappai.2024.108228.
[34] J. Tan, S. Xue, H. Li, H. Cao, D. Li, Safe stabilization control for interconnected virtual-real systems via model-based reinforcement learning, in: 2024 14th Asian Control Conference, ASCC, 2024, pp. 605–610.
[35] F. Tao, B. Xiao, Q. Qi, J. Cheng, P. Ji, Digital twin modeling, J. Manuf. Syst. 64 (2022) 372–389, http://dx.doi.org/10.1016/j.jmsy.2022.06.015.
[36] F. Tassi, E. De Momi, A. Ajoudani, An adaptive compliance hierarchical quadratic programming controller for ergonomic human–robot collaboration, Robot. Comput.-Integr. Manuf. 78 (2022) 102381, http://dx.doi.org/10.1016/j.rcim.2022.102381.
[37] H. Modares, I. Ranatunga, F.L. Lewis, D.O. Popa, Optimized assistive human–robot interaction using reinforcement learning, IEEE Trans. Cybern. 46 (3) (2016) 655–667, http://dx.doi.org/10.1109/TCYB.2015.2412554.
[38] Y. Li, G. Carboni, F. Gonzalez, D. Campolo, E. Burdet, Differential game theory for versatile physical human–robot interaction, Nat. Mach. Intell. 1 (1) (2019) 36–43, http://dx.doi.org/10.1038/s42256-018-0010-3.
[39] W. Guo, S. Zhao, H. Cao, B. Yi, X. Song, Koopman operator-based driver-vehicle dynamic model for shared control systems, Appl. Math. Model. 114 (2023) 423–446, http://dx.doi.org/10.1016/j.apm.2022.10.014.
[40] M. Li, J. Qin, Q. Ma, Y. Shi, W.X. Zheng, Master-slave safe cooperative tracking via game and learning based shared control, IEEE Trans. Autom. Control (2024) 1–8, http://dx.doi.org/10.1109/TAC.2024.3462254.
[41] M. Li, J. Qin, Z. Wang, Q. Liu, Y. Shi, Y. Wang, Optimal motion planning under Dynamic Risk Region for safe human–robot cooperation, IEEE/ASME Trans. Mechatronics (2024) 1–11, http://dx.doi.org/10.1109/TMECH.2024.3408810.
[42] H.-N. Wu, M. Wang, Learning human behavior in shared control: Adaptive inverse differential game approach, IEEE Trans. Cybern. (2023) 1–11, http://dx.doi.org/10.1109/TCYB.2023.3244559.
[43] A. Broad, T. Murphey, B. Argall, Learning models for shared control of human-machine systems with unknown dynamics, in: Robotics: Science and Systems XIII, Robotics: Science and Systems Foundation, 2017, http://dx.doi.org/10.15607/RSS.2017.XIII.037.
[44] S. Islam, P.X. Liu, A.E. Saddik, R. Ashour, J. Dias, L.D. Seneviratne, Artificial and virtual impedance interaction force reflection-based bilateral shared control for miniature unmanned aerial vehicle, IEEE Trans. Ind. Electron. 66 (1) (2019) 329–337, http://dx.doi.org/10.1109/TIE.2018.2793178.
[45] E. Eraslan, Y. Yildiz, A.M. Annaswamy, Shared control between pilots and autopilots: An illustration of a cyberphysical human system, IEEE Control Syst. 40 (6) (2020) 77–97, http://dx.doi.org/10.1109/MCS.2020.3019721.
[46] K. Wang, C. Mu, Z. Ni, D. Liu, Safe reinforcement learning and adaptive optimal control with applications to obstacle avoidance problem, IEEE Trans. Autom. Sci. Eng. (2023) 1–14, http://dx.doi.org/10.1109/TASE.2023.3299275.
[47] B. Barros Carlos, A. Franchi, G. Oriolo, Towards safe human-quadrotor interaction: Mixed-initiative control via real-time NMPC, IEEE Robot. Autom. Lett. 6 (4) (2021) 7611–7618, http://dx.doi.org/10.1109/LRA.2021.3096502.
[48] A. Broad, I. Abraham, T. Murphey, B. Argall, Data-driven koopman operators for model-based shared control of human–machine systems, Int. J. Robot. Res. 39 (9) (2020) 1178–1195, http://dx.doi.org/10.1177/0278364920921935.
[49] M.H. Cohen, C. Belta, Safe exploration in model-based reinforcement learning using control barrier functions, Automatica 147 (2023) 110684, http://dx.doi.org/10.1016/j.automatica.2022.110684.
[50] H. Yang, Q. Hu, H. Dong, X. Zhao, D. Li, Optimized data-driven prescribed performance attitude control for actuator saturated spacecraft, IEEE/ASME Trans. Mechatronics 28 (3) (2023) 1616–1626, http://dx.doi.org/10.1109/TMECH.2022.3230993.
[51] L. Chen, F. Hao, Optimal tracking control for unknown nonlinear systems with uncertain input saturation: A dynamic event-triggered ADP algorithm, Neurocomputing 564 (2024) 126964, http://dx.doi.org/10.1016/j.neucom.2023.126964.
[52] W. Hao, B. Huang, W. Pan, D. Wu, S. Mou, Deep Koopman learning of nonlinear time-varying systems, Automatica 159 (2024) 111372, http://dx.doi.org/10.1016/j.automatica.2023.111372.
[53] Z. Liang, W. Hao, S. Mou, A data-driven approach for inverse optimal control, in: 2023 62nd IEEE Conference on Decision and Control, CDC, IEEE, Singapore, Singapore, 2023, pp. 3632–3637, http://dx.doi.org/10.1109/CDC49753.2023.10383220.
[54] E. Tzorakoleftherakis, T.D. Murphey, Controllers as filters: Noise-driven swing-up control based on maxwell's demon, in: 2015 54th IEEE Conference on Decision and Control, CDC, 2015, pp. 4368–4374, http://dx.doi.org/10.1109/CDC.2015.7402901.
[55] X. Xing, W. Li, S. Yuan, Y. Li, Fuzzy logic-based arbitration for shared control in continuous human–robot collaboration, IEEE Trans. Fuzzy Syst. 32 (7) (2024) 3979–3991, http://dx.doi.org/10.1109/TFUZZ.2024.3386822.
[56] Y. Yang, Z. Ding, R. Wang, H. Modares, D.C. Wunsch, Data-driven human-robot interaction without velocity measurement using off-policy reinforcement learning, IEEE/CAA J. Autom. Sin. 9 (1) (2022) 47–63, http://dx.doi.org/10.1109/JAS.2021.1004258.
[57] V.S. Donge, B. Lian, F.L. Lewis, A. Davoudi, Data-efficient reinforcement learning for complex nonlinear systems, IEEE Trans. Cybern. (2023) 1–12, http://dx.doi.org/10.1109/TCYB.2023.3324601.
[58] P. Deptula, H.-Y. Chen, R.A. Licitra, J.A. Rosenfeld, W.E. Dixon, Approximate optimal motion planning to avoid unknown Moving Avoidance Regions, IEEE Trans. Robot. 36 (2) (2020) 414–430, http://dx.doi.org/10.1109/TRO.2019.2955321.
[59] M.L. Greene, P. Deptula, S. Nivison, W.E. Dixon, Approximate optimal trajectory tracking with sparse bellman error extrapolation, IEEE Trans. Autom. Control (2022) 1–8, http://dx.doi.org/10.1109/TAC.2022.3194040.
[60] M.L. Greene, P. Deptula, S. Nivison, W.E. Dixon, Sparse learning-based approximate dynamic programming with barrier constraints, IEEE Control Syst. Lett. 4 (3) (2020) 743–748, http://dx.doi.org/10.1109/LCSYS.2020.2977927.
[61] C. Mu, K. Wang, T. Qiu, Dynamic event-triggering neural learning control for partially unknown nonlinear systems, IEEE Trans. Cybern. 52 (4) (2022) 2200–2213, http://dx.doi.org/10.1109/TCYB.2020.3004493.
[62] L. Chen, F. Hao, Optimal tracking control for unknown nonlinear systems with uncertain input saturation: A dynamic event-triggered ADP algorithm, Neurocomputing 564 (2024) 126964, http://dx.doi.org/10.1016/j.neucom.2023.126964.

Junkai Tan received the B.E. degree in electrical engineering at the School of Electrical Engineering in Xi'an Jiaotong University, Xi'an, China. He is currently working toward the M.E. degree in electrical engineering at the School of Electrical Engineering, Xi'an Jiaotong University. His current research interest includes human-automation interaction and adaptive dynamic programming.

Shuangsi Xue received the B.E. degree in electrical engineering and automation from Hunan University, Changsha, China, in 2014, and the M.E. and Ph.D. degrees in electrical engineering from Xian Jiaotong University, Xian, China, in 2018 and 2023, respectively. He is currently an Assistant Professor at the School of Electrical Engineering, Xian Jiaotong University. His current research interest includes adaptive control and data-driven control of networked systems.

Hui Cao received the B.E., M.E., and Ph.D. degrees in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2000, 2004, and 2009, respectively. He is a Professor at the School of Electrical Engineering, Xi'an Jiaotong University. He was a Postdoctoral Research Fellow at the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, from 2014 to 2015. He has authored or coauthored over 30 scientific and technical papers in recent years. His current research interest includes knowledge representation and discovery. Dr. Cao was a recipient of the Second Prize of National Technical Invention Award.

Shuzhi Sam Ge is a Professor with the Department of Electrical and Computer Engineering, Member of NUS AI Institute (NAII) and Institute of Functional Intelligent Materials (I-FIM) , The National University of Singapore, Singapore, and Founding Honorary Director of Institute for Future (IFF), Qingdao University, Qingdao, China. He received the Ph.D. degree from the Imperial College London, London, U.K., in 1993, and the B.Sc. degree from the Beijing University of Aeronautics and Astronautics, China, in 1986. He serves as President, 2024-2026, President Elect, 2022-2024, Asian Control Association; Member of IEEE Control

Systems Award Committee, 2024-2025; IFAC executive officers as a Council Member, 2023-2026, and Chair of International Standing Committee, International Conference on Social Robotics. He serves as the Founding Editor-in-Chief, International Journal of Social Robotics, Springer Nature, 2008-present, book editor for Automation and Control Engineering of Taylor & Francis/CRC Press, and Senior Editor of IEEE SMC- Systems, and has served/been serving as an Associate Editor for a number of flagship journals including IEEE TAC, IEEE TCST, IEEE TNN, Automatica, and CAAI Transactions on Intelligence Technology. At IEEE Control Systems Society, he served as Vice President for Technical Activities, 2009-2010, Vice President of Membership Activities, 2011-2012, Member of Board of Governors of IEEE Control Systems Society, 2007-2009. He is the recipients of Singapore AI Grand Challenge Award 2023, Distinguished Member Award, IEEE Control Systems Society (CSS), 2013, National Technology Award, Singapore, 1999. He is a Clarivate Analytics high-cited scientist in 2016-2024. He is a Fellow of SAEng, IEEE, IFAC, IET, and CAA, and ACA. His current research interests include robotics, intelligent systems, artificial intelligence, and smart materials.