Received March 13, 2022, accepted March 23, 2022, date of publication March 30, 2022, date of current version April 8, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3163296

Reinforcement Learning-Based Composite Controller for Cable-Driven Parallel Suspension System at High Angles of Attack

WEIPING WANG, XIAOGUANG WANG, CHULUN SHEN, AND QI LIN
School of Aerospace Engineering, Xiamen University, Xiamen, Fujian 361102, China

Corresponding author: Xiaoguang Wang (xgwang@xmu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 12172315, Grant 12072304, and Grant 11702232; and in part by the Natural Science Foundation of Fujian Province of China under Grant 2021J01050.

ABSTRACT This paper investigates an intelligent method for the motion control of a cable-driven parallel suspension system (CDPSS) in wind tunnel tests, especially at high angles of attack, which is characterized by unsteady and nonlinear aerodynamics. Considering the modeling uncertainties and the complex aerodynamic interference, a composite controller that combines deep deterministic policy gradient (DDPG) and computed-torque is proposed to improve the control performance. The tasks at hand consist in the construction of the training environment based on the dynamic equations and the Markov Decision Process (MDP) design. The supplementary computed-torque control is used to enhance the learning rate of the agent. Then a well-trained agent is applied in the high angles of attack maneuvers control with different examples, including single-DOF and multi-DOF coupled motion. The simulation results show the controller could fulfill the training tasks efficiently, and it turns out to be robust and maintain strong generalization ability despite handling the unlearned tasks.

INDEX TERMS Wind tunnel test, cable-driven parallel mechanism, high angle of attack, motion control, DDPG.

I. INTRODUCTION

Cable-driven parallel mechanism (CDPM) is a special kind of robotic manipulator that employs flexible cables instead of traditional rigid links, which produces the advantages of simple structure, low inertia, relatively larger workspace, and stiffness, and superior modularity and reconfigurability. This has attracted increasing interest among researchers, and rich literature exists for CDPM in many different applications, such as cooperative cranes [1], haptic devices [2], rescue or medical rehabilitation [3], radio telescopes [4]. Particularly, due to the cables' more minor flow field interference and higher dynamic performances, CDPM provides a new method of suspension for the aircraft model in the wind tunnel tests. For example, the French National Aerospace Research Centre (ONERA) proposed an active suspension method for wind tunnel tests project (SACSO) by using cables, and it was successfully applied in a vertical wind tunnel for aerodynamic design and validation of fighters [5]. Bruckmann *et al.* [6] investigated an active cable-driven suspension system for ship maneuvers in the wind tunnel. The system is designed for 150 kg weight and a motion frequency of 0.5 Hz at an amplitude of approximately 0.5m in the low-speed wind tunnel. Lambert [7]–[8] developed a cable-driven dynamically-controlled wind tunnel traverse mechanism in 6-DOF to investigate the coupling between a blunt body and the embedding flow during a controlled maneuver. Lin and Wang [9], [10] made kinematic and dynamic analyses of the cable-driven suspension system, conducted some typical tests in the low-speed wind tunnel, and obtained static and unsteady aerodynamic forces by using the internal six-force balance.

The high precision and robust control of the cable-driven parallel suspension system (CDPSS) is critical to its applications in dynamic wind tunnel tests. In the literature, many advanced control approaches have been proposed in the Cartesian space or joint space for CDPM. Begey *et al.* [11] designed a cascade dynamic controller in a singular

The associate editor coordinating the review of this manuscript and approving it for publication was Nishant Unnikrishnan.

perturbation framework for CDPM with highly flexible cables. Stability was demonstrated and the method's efficiency was experimentally validated. A robust adaptive approach was proposed to control CDPM which is subject to dynamic uncertainties in [12]. It's designed based on function approximation technique, which can adaptively learn the uncertain terms in dynamics of both manipulator and actuators. Santos *et al.* [13] proposed a model predictive control strategy for large-dimension CDPM working at low speeds. The simulations results indicate a superior performance of the proposed controller compared with two commonly used strategies: Sliding mode control and PID+ control. Shang *et al.* [14] designed a composite controller based on the tracking error and synchronization error of the moving platform in the cable space to realize the synchronization motion between the cables, and finally increase the tracking accuracy of the moving platform. The vibration rejection control was dealt with in Cartesian space for CDPM to address the vibrations induced by the axial flexibility of massless cable in [15], and the approach was tested experimentally. However, when dealing with kinematic and dynamic uncertainties and external interferences, new approaches are still worthy of research, especially for the motion control of cable-driven parallel suspended aircraft at high angles of attack. During this kind of maneuver, the flow field around the aircraft is highly complex due to the multi-degree-of-freedom coupling, and the aerodynamic characteristics usually show strong nonlinearity and unsteadiness [16], [17]. The aerodynamic forces and moments vary with multiple parameters, including both static and dynamic motion states, such as angle of attack, sideslip angle and their variations, rates of pitch, yaw, and roll. As a result, the conventional aerodynamic model does not meet the requirements of flight simulation and control law design. The existence of modeling uncertainty and external interference makes it a great challenge for the high precision motion control of CDPSS, especially in high angle-of-attack maneuvers.

With the upsurge of artificial intelligence researches, intelligent algorithms represented by reinforcement learning (RL) have received considerable attention in the past decades. Especially, the deep reinforcement learning (DRL) technique can handle large state space by building a deep neural network to relate the value estimates and associated state-action pairs, thereby overcoming the shortcoming of conventional RL. Some recent works have started developing such approaches in various practical problems [18]–[22]. For example, the DRL technique has been shown to be successful in playing Atari and Go games [18], emerges as a powerful data-driven method for solving complex control problems. Lee *et al.* [19] studied a DRL algorithm, which is adopted to learn the efficient adaptive gain-tuning strategy, and validated that the proposed system can result in better station-keeping performance without deterioration in its control efficiency. Considering the limitations of conventional PID controllers for the dynamic positioning system, Modares *et al.* [20] developed an online AC algorithm along with a novel NN identifier to learn the optimal control solution for nonlinear systems with completely unknown dynamics and input constraints. Huang *et al.* [21] proposed an agent trained by deep deterministic policy gradient (DDPG) in unmanned aerial vehicle (UAV) control. The agent's generalization ability and robustness were proved reliable in handling unlearned flying tasks. Lin *et al.* [22] investigated a supplementary controller based on RL to improve the control performance of quadrotor UAVs by constructing an actor-critic (AC) structure and some improved technologies, e.g., Q-learning, temporal difference, and experience replay. With the proposed method, the speed and stability of training can be significantly improved.

In this paper, considering the complex, nonlinear and unsteady aerodynamic characteristics of the aircraft model at high angles of attack, an alternative approach to standard computed-torque based on reinforcement learning is studied. The main contributions of the paper are the following. 1) An reinforcement learning-based DDPG algorithm is designed. Its training environment is constructed based on the dynamic equations of the system, and the reward functions are set in a segmental form according to the errors of state parameters and their variations. 2) A novel composite controller combining DDPG and computed-torque method for the typical application of CDPM in wind tunnel tests is proposed to achieve better control performance in high angle-of-attack maneuvers characterized by multi-DOF. This approach is verified through simulations.

The structure of this paper is as follows. Section II describes the dynamic model of this cable-driven system. Section III presents the construction of a deep deterministic policy gradient improved algorithm. Section IV expounds on the training and simulation analysis. Finally, Section V summarizes the main achievements of the paper.

II. DESCRIPTION OF CDPSS AND DYNAMIC EQUATIONS

Taking the low-speed wind tunnel test as the application background, the cable-driven suspension system can be set up in the test section accordingly, consisting of a fixed frame, aircraft model, actuated cables, servo motors, motion control system, and so on. The prototype of a typical aircraft model suspended by eight cables is given in Fig. 1. The i-th cable $i = 1, \ldots, 8$ exits from the fixed base at point B_{i}, and it is connected to the aircraft model at point P_{i}. By adjusting the cable length, the position and orientation of the aircraft model could be dynamically controlled. The model's pose could be measured by visual method, such as a camera or laser, in which some cooperative targets are set on the model. Cable tension sensors are also used to monitor the tension in case of slackness.

Considering the limited dimension of wind tunnel test section, CDPSS is a small-scale mechanism, the ratio between end-effector and cables' masses is quite large, thus it is reasonable to assume the cable to be massless and straight. The impact of cable elasticity on the pose of aircraft model is studied in [23], and it could be further minimized by using the high elastic modulus material, such as steel cable. Therefore,

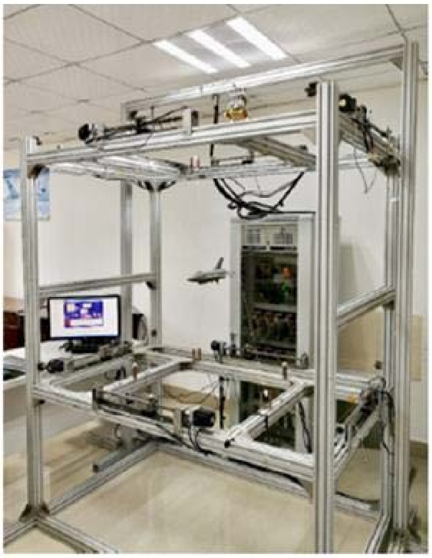


FIGURE 1. The prototype of CDPSS.

the cable is assumed to be rigid in this paper, and the errors induced by this assumption could be taken as the parameter uncertainties, and will be dealt with by the new approach proposed in Section III. Moreover, the diameter of the fixed pulley is generally small enough for CDPSS, and could be neglected to facilitate the establishment of the kinematic model. Then, the fixed pulley is regarded as a point hinge with fixed coordinates.

The kinematic relationship diagram is shown in Fig. 2, the static coordinate system $o_w x_w y_w z_w$ is fixed with the frame, and the moving coordinate system $O'X'Y'Z'$ is fixed on the aircraft model. Let $\boldsymbol{P}_i = \overrightarrow{o_w P_i}$, $\boldsymbol{B}_i = \overrightarrow{o_w B_i}$. Therefore, the vector $\boldsymbol{P}_i(X_{P_i}, Y_{P_i}, Z_{P_i})^{\mathrm{T}}$ $(i = 1, 2, \ldots, 8)$ in the static coordinate system satisfies,

$$\boldsymbol{P}_i = \boldsymbol{X}_{O'} + \boldsymbol{R}\boldsymbol{x}_{P_i} \tag{1}$$

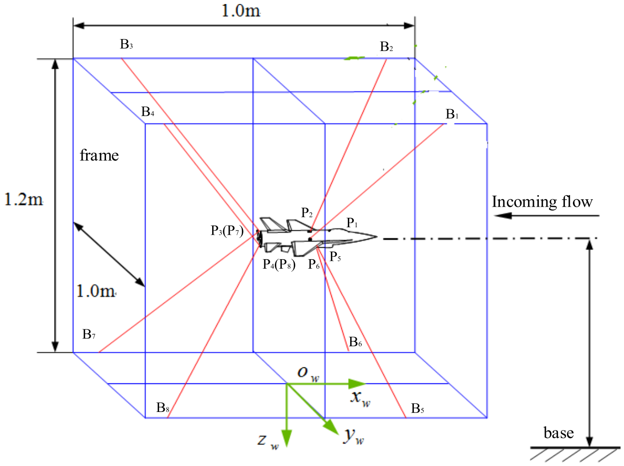


FIGURE 2. The sketch of geometric structure.

where $\boldsymbol{X}_{O'}(X_{O'}, Y_{O'}, Z_{O'})^{\mathrm{T}}$ is the coordinate of origin O' in the static coordinate system, $\boldsymbol{R}$ is the rotational transformation matrix.

Let $\boldsymbol{X} = (X_{O'}, Y_{O'}, Z_{O'}, \phi, \theta, \psi)^T$ be denoted as the pose of the aircraft model. According to the differential kinematics of parallel structure, the kinematic model of cable length vector is,

$$\dot{\boldsymbol{L}} = \boldsymbol{J}\dot{\boldsymbol{X}}_{\omega} \tag{2}$$

where $\boldsymbol{L}$ is the cable length vector, $\boldsymbol{J}$ is the Jacobian matrix of the system, $\dot{\boldsymbol{X}}_{\omega}$ is the velocity vector.

Based on the Newton-Euler method, the motion of the aircraft model and actuators are modeled separately. These dynamic equations are shown as follows,

$$\begin{cases} M\ddot{\boldsymbol{X}}_{O'} = \boldsymbol{f}_e + \sum_{i=1}^{8}(-T_i\boldsymbol{u}_i) + M\boldsymbol{g} \\ A_G\dot{\boldsymbol{\omega}} = \boldsymbol{\tau}_e + \sum_{i=1}^{8}\boldsymbol{r}_i \times (-T_i\boldsymbol{u}_i) - \boldsymbol{\omega} \times (A_G\boldsymbol{\omega}) \end{cases} \tag{3}$$

$$\boldsymbol{M}_0\ddot{\boldsymbol{\theta}}_m + \boldsymbol{C}_0\dot{\boldsymbol{\theta}}_m + \mu T = \boldsymbol{\tau} \tag{4}$$

where M is the mass of aircraft model, $\boldsymbol{f}_e$ and $\boldsymbol{\tau}_e$ are the aerodynamic forces and moments acting on the aircraft model, T_i is the i-th cable tension, A_G is the model's inertia matrix expressed in the static coordinate system, $\dot{\boldsymbol{\omega}}$ is the vector of angular acceleration. $\boldsymbol{M}_0$ is the inertia matrix of actuators, $\boldsymbol{C}_0$ is the viscous friction coefficients matrix for actuators, μ is the drive coefficient of ball screw, $\boldsymbol{\tau}$ is the input torque, $\boldsymbol{T}$ is the cable tension vector. $\boldsymbol{\theta}_m$ is the rotational angle vector of actuators, $\dot{\boldsymbol{L}} = \mu\dot{\boldsymbol{\theta}}_m$.

Generally, the cable tensions are effective only when they are positive,

$$\boldsymbol{T}_{\max} > \boldsymbol{T}_i > \boldsymbol{T}_{\min}, \quad i = 1, \ldots, 8 \tag{5}$$

where $\boldsymbol{T}_{\min}$ and $\boldsymbol{T}_{\max}$ are respectively the lower and upper bounds of cable tensions.

Use equations (3), (4), and (5) to derive

$$\boldsymbol{A}\ddot{\boldsymbol{X}} - \boldsymbol{B}\dot{\boldsymbol{X}} + \boldsymbol{D} = \boldsymbol{C}\boldsymbol{\tau} + \boldsymbol{d} \tag{6}$$

$$\boldsymbol{A} = \boldsymbol{M}(\boldsymbol{X}) - \mu^{-2}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{M}_0\boldsymbol{J}\boldsymbol{G}$$

in which, $\boldsymbol{B} = \mu^{-2}(\boldsymbol{J}^T\boldsymbol{M}_0\dot{\boldsymbol{J}}\boldsymbol{G} + \boldsymbol{J}^T\boldsymbol{M}_0\boldsymbol{J}\dot{\boldsymbol{G}} + \boldsymbol{J}^T\boldsymbol{C}_0\boldsymbol{J}\boldsymbol{G})$

$$\boldsymbol{C} = -\mu^{-1}\boldsymbol{J}^{\mathrm{T}}, \boldsymbol{D} = \boldsymbol{N} - \boldsymbol{w}_g, \boldsymbol{d} = \left[\boldsymbol{f}_e^{\mathrm{T}} \quad \boldsymbol{\tau}_e^{\mathrm{T}}\right]^{\mathrm{T}}$$

In high angle-of-attack maneuvers, the model of aerodynamic forces and moments is a nonlinear function of $\boldsymbol{X}$ and $\dot{\boldsymbol{X}}$, and is developed in the following form,

$$\begin{aligned} \boldsymbol{d} &= qS\left[C_L \quad C_D \quad C_Y \quad bC_R \quad \bar{c}C_M \quad bC_N\right]^{\mathrm{T}} \\ C_D &= C_{D0} + C_D^{\alpha}\alpha \\ C_Y &= C_Y^{\beta}\beta \\ C_M &= C_{M0} + C_M^{\alpha}\alpha + \frac{\bar{c}}{2V}\left(C_M^{Q}\bar{Q} + C_M^{\dot{\alpha}}\dot{\alpha}\right) \\ C_R &= C_R^{\beta}\beta + \frac{b}{2V}\left(C_R^{\bar{P}}\bar{P} + C_R^{\bar{R}}\bar{R}\right) \\ C_N &= C_N^{\beta}\beta + \frac{b}{2V}\left(C_N^{\bar{P}}\bar{P} + C_N^{\bar{R}}\bar{R}\right) \end{aligned} \tag{7}$$

where b is the wing semispan, $\bar{c}$ is mean aerodynamic wing chord, q is the dynamic pressure, S is the reference wing area, V is the flow speed. $C_L, C_D, C_Y, C_M, C_M, C_M$ are respectively the aerodynamic coefficients of lift, drag, lateral forces, and roll, pitch, yaw moments. $C_L^\alpha, C_D^\alpha, C_Y^\beta, C_M^\alpha, C_M^{\bar{Q}}, C_M^{\dot{\alpha}}, C_R^\beta, C_R^{\bar{P}}, C_R^{\bar{R}}, C_N^\beta, C_N^{\bar{P}}, C_N^{\bar{R}}$ are the aerodynamic derivatives, and could be identified in light of the typical low amplitude forced oscillations and rotary tests data [24]. $\bar{P}$, $\bar{Q}$, $\bar{R}$ are the roll, pitch, and yaw rates. Angle of attack α and sideslip angle β are defined by the following relations,

$$\tan\alpha = \tan\theta\cos\phi - \sin\phi\tan\psi/\cos\theta \tag{8}$$

$$\sin\beta = \sin\theta\sin\phi\cos\psi + \sin\psi\cos\phi \tag{9}$$

III. CONTROL ALGORITHM

As to the CDPSS, there are multiple sources of uncertainties, such as the nonlinear and unsteady aerodynamic forces caused by the boundary layer separation and flow hysteresis characteristics at high angles of attack, the variations of tangency points of driving cables and guide pulleys, the cable elasticity and the transmission efficiency of the ball screw. Considering the above factors and the self-learning and self-adaption characteristics of DRL, this intelligent method is used to design the control law of CDPSS.

A. DDPG ALGORITHM

The learning process of RL is similar to the human brain's. The process can be described as Trial-and-Error; during each episode, a reward instructs the agent to achieve the ultimate objective. The basic principle of RL is shown in Fig. 3, i.e., the agent continuously executes actions in the environment; after each episode, the agent receives a reward from the environment; through this learning process, the agent will take the action that ranks the highest reward value.

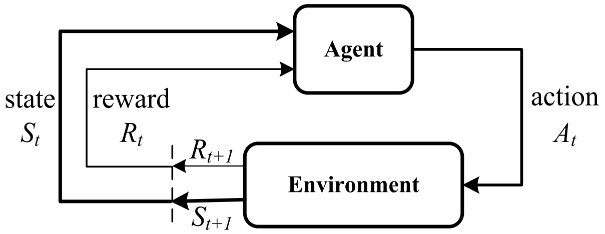


FIGURE 3. Sketch of reinforcement learning.

The process of RL can be described mathematically as Markov Decision Process (MDP). MDP is made up of five elements, i.e., $M = \langle \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{P}, \boldsymbol{R}, \boldsymbol{\gamma} \rangle$, where $\boldsymbol{S}$ is the state set of the environment, $\boldsymbol{A}$ is the agent action, P is the state transition probability, R is the reward function which shows the preference degree of the environment for different action the agent took; $\gamma$is the discount factor, $\gamma \in [0, 1]$, when γ takes a relatively large value, it means the agent pays more attention to the future earnings than current rewards.

The Deep Deterministic Policy Gradient (DDPG) algorithm mainly comprises environment, experience replay set, Actor network module, and value evaluation network module. The environment is the interaction space and exploration space of the agent. The agent obtains interaction samples during the interaction with the environment and stores the samples in the experience playback set for the training process of the agent. The network architecture of the algorithm is shown in Fig. 4.

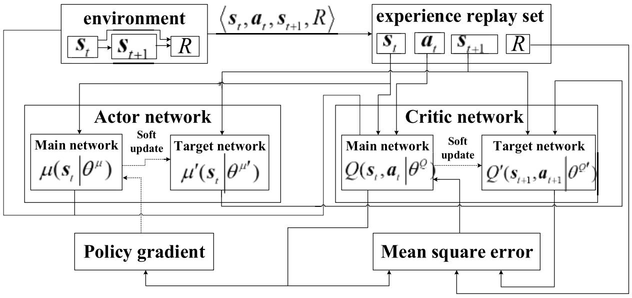


FIGURE 4. Algorithm block diagram of DDPG.

There are four neural networks in the DDPG algorithm, and the main functions of each neural network are listed below:

1. Actor Current Network: Updating policy parameter; executing current action $\boldsymbol{a}_t$ according to the current state $\boldsymbol{s}_t$; and $\boldsymbol{a}_t$ will be used to generate $\boldsymbol{s}_{t+1}$ and R by interacting with the environment.

2. Actor Target Network: Choosing optimal action $\boldsymbol{a}_{t+1}$ based on $\boldsymbol{s}_{t+1}$, copying network parameter $\theta^{\mu'}$ from θ^μ regularly.

3. Critic Current Network: Updating value network parameter θ^Q; calculating current value $Q\left(\boldsymbol{s}, \boldsymbol{a} \middle| \theta^Q\right)$.

4. Critic Target Network: Calculating $Q'(\boldsymbol{s}_t, \boldsymbol{a}_t \left| \theta^{Q'} \right.)$ in target Q value.

The loss function of Critic Current Network can be expressed as follows,

$$L\left(\theta^Q\right) = \frac{1}{N}\sum_{i=1}^{N}\left[y_i - Q\left(\boldsymbol{s}_i, \boldsymbol{a}_i \middle| \theta^Q\right)\right]^2 \tag{10}$$

The loss gradient of Actor Current Network can be expressed as follow,

$$\nabla_{\theta^\mu} J\left(\theta^\mu\right) = \frac{1}{N}\sum_{i=1}^{N}\left[\nabla_{\theta^\mu}\mu\left(\boldsymbol{s}_i\right)\nabla_a Q\left(\boldsymbol{s}_i, \boldsymbol{a}_i\right)|_{\boldsymbol{a}_i=\mu(\boldsymbol{s}_i)}\right] \tag{11}$$

The two Target Networks are updated by the following formulas,

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\,\theta^{Q'} \tag{12}$$

$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\,\theta^{\mu'} \tag{13}$$

where $\boldsymbol{\tau}$ is the update coefficient, it represents the degree of network replication of each episode; usually, it takes a small value such as 0.1.

B. SYSTEM MODEL DESIGN

To train the agent to achieve motion control of the aircraft model, MDP should be designed, and several key parameters need to be determined, such as the environmental state, executable actions, and rewards generated from the interactions with the environment.

In this paper, the dynamic equations of CDPSS are taken as the training environment of the agent. The process of solving the equations is considered as the construction of the training environment. By solving the differential equations (6), the attitude angles of the aircraft model and their angular velocities could be obtained and denoted as $\boldsymbol{X}$ and $\dot{\boldsymbol{X}}$. The errors between these state parameters and the desired ones are taken as observation of the agent, i.e., the three angle errors and three angular velocity errors. All the new parameters are grouped together to be the environment state set,

$$observation = [\Delta\phi,\ \Delta\theta,\ \Delta\psi,\ \Delta\dot{\phi},\ \Delta\dot{\theta},\ \Delta\dot{\psi}]^{\mathrm{T}} \quad (14)$$

where ϕ, θ, ψ respectively denote the roll, pitch, and yaw angles. Each of the eight cables is controlled by a motor, then the torques of motors are taken as environment's output and are the agent action set,

$$action = [\boldsymbol{\tau}_{1t}, \boldsymbol{\tau}_{2t}, \ldots, \boldsymbol{\tau}_{8t}]^{\mathrm{T}} \quad (15)$$

After constructing the state and the action sets, an environmental reward function is set to enable the agent to reach the expected target. The reward function consists of two parts, one based on the angle errors and the other based on the angular velocity errors. Therefore, the total reward function can be expressed as,

$$r_t = r_a + r_v \quad (16)$$

where r_a represents the part of angle errors, and it can be divided into three parts, i.e.,

$$r_{a1} = \begin{cases} 25 & |\Delta\phi_t| \le 0.1 \\ 15 & 0.1 < |\Delta\phi_t| \le 2 \\ 10 & 2 < |\Delta\phi_t| \le 5 \\ -25 & 5 < |\Delta\phi_t| \end{cases} \quad (17)$$

In the formula, $|\Delta\phi_t| = |\phi_t - \phi_d|$, ϕ_d is the desired roll angle, and ϕ_t is the observation value, r_{a2} and r_{a3} are respectively the reward function of $\Delta\theta_t$ and $\Delta\psi_t$, and the two reward functions have the similar form with r_{a1}. r_v represents the part of angular velocity errors, and can also be expressed as

$$r_v = r_{v1} + r_{v2} + r_{v3} \quad (18)$$

where $r_{v1} = \begin{cases} 10 & \left|\Delta\dot{\phi}_t\right| \le 0.1 \\ 5 & 0.1 < \left|\Delta\dot{\phi}_t\right| \le 3 \\ -10 & 3 < \left|\Delta\dot{\phi}_t\right| \end{cases}$, r_{v2} and r_{v3} are respectively the reward functions of $\Delta\dot{\theta}_t$ and $\Delta\dot{\psi}_t$, and have the similar form withr_{v1}. In the formula, $\left|\Delta\dot{\phi}_t\right| = \left|\dot{\phi}_t - \dot{\phi}_d\right|$, $\dot{\phi}_d$ is the desired angular velocity, and $\dot{\phi}_t$ is the observation value.

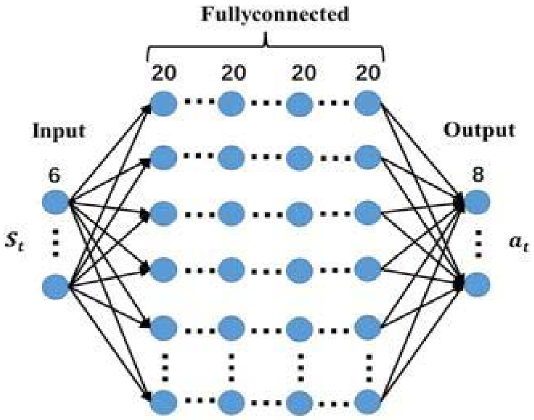


FIGURE 5. Actor neural network.

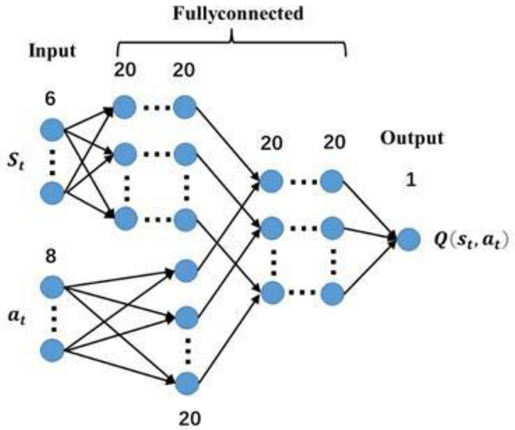


FIGURE 6. Critic neural network.

The neural network construction is shown in Fig. 5. The actor neural network's input layer contains six neurons corresponding to the six-dimensional environment. There is an interlaced connection between four fully connected layers and three convolution layers in the hidden layer, with *Relu* being its activation function. In the output layer, eight neurons correspond to the agent's eight-dimensional action. *Tanh* is the neural network activation function.

Fig. 6 shows the critic neural network, which has fourteen neurons in the input layer, corresponding to the six-dimensional environment and the eight-dimensional agent action. There are several steps between state input and output: firstly, the state input goes through two twenty-neuron fully connected layers and then adds up with the action output generated by a twenty-neural fully connected layer with the action input, finally goes through the other two twenty-neuron fully connected layers as the state output. The ultimate output is a one-dimensional state action value corresponding to state input and action input. *Relu* is the neural network activation function.

C. DESIGN OF COMPOSITE CONTROL LAW

Considering the uncertainties and interferences exist in the modeling process, a composite control law combining computed-torque and DDPG algorithm is proposed. The controller's structure is shown in Fig. 7. The new controller is divided into the control subsystem and the learning subsystem; the control subsystem is an online control, i.e., the environmental state inputs of the actor target network are well trained. By integrating the agent action output $\boldsymbol{a}_t$ and the computed-torque controller output $\boldsymbol{\tau}_1$, a new control law is generated and applied for the aircraft model, and it can be expressed as,

$$\boldsymbol{\tau}_t = \boldsymbol{\tau}_1 + \boldsymbol{a}_t \tag{19}$$

To achieve the high-precision pose control of the aircraft model, direct pose feedback is adopted by using monocular vision to measure the motion state. Based on the dynamic equations, a typical computed-torque control law is designed. It ensures the stability of the aircraft model when moving at a specific trajectory and it could rectify the deviation between the actual pose and the expected pose of the aircraft model, as well as the deviation of rotation angle and the rotation angular velocity of the motor.

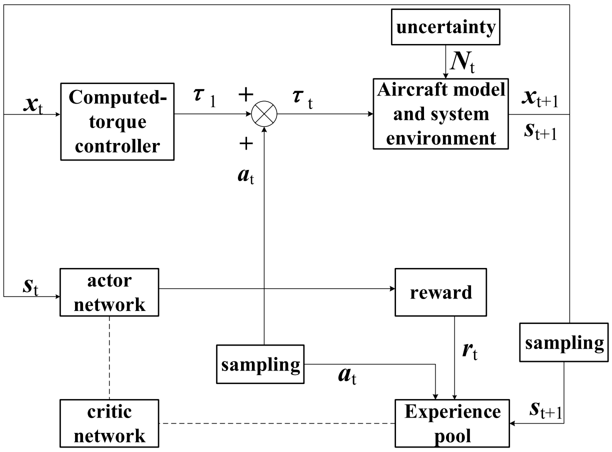


FIGURE 7. Structure diagram of computed-torque+DDPG controller.

Besides the proportional and derivative terms, nonlinear and other terms should be compensated; thus, the control law is set as,

$$\boldsymbol{\tau}_1 = \boldsymbol{C}^+ \left[\boldsymbol{A}\left(\ddot{\boldsymbol{X}}_d + \boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{K}_d\dot{\boldsymbol{e}}\right) + \boldsymbol{B}\dot{\boldsymbol{X}} + \boldsymbol{D} - \boldsymbol{d}\right] + \mu\boldsymbol{Q} \tag{20}$$

where $\boldsymbol{C}^+$ is the Moore-Penrose generalized inverse matrix of $\boldsymbol{C}$, $\boldsymbol{e} = \boldsymbol{X} - \boldsymbol{X}_d$, $\dot{\boldsymbol{e}} = \dot{\boldsymbol{X}} - \dot{\boldsymbol{X}}_d$, $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are respectively the proportional and derivative gains of the control system. $\boldsymbol{Q} = \left(\boldsymbol{I} - \boldsymbol{J}^{\mathrm{T}+}\boldsymbol{J}^{\mathrm{T}}\right)\boldsymbol{w}$, which is the internal cable tension, and could be adjusted through the factor $\boldsymbol{w}$ to avoid the cable's slackness.

1) OFF-LINE AGENT TRAINING PROCESS

When the agent training process starts, initialize the critic policy network randomly and assign the parameters to the corresponding networks. Then start an episode; in this experiment, each episode has its fixed task and task time, and the agent control period is defined as a time step. Each time step, $\boldsymbol{s}_t$ is input to actor policy network and then output $\boldsymbol{a}_t$. $\boldsymbol{\tau}_t$ is the combination of $\boldsymbol{a}_t$ and computed-torque controller signal, then the $\boldsymbol{s}_{t+1}$ is gained by integrating, also, the well-designed reward r_{t+1} function gains the immediate reward. Finally, storing $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_{t+1}, \boldsymbol{s}_{t+1})$ in the experience pool as a trajectory sample and the flight strategy can be updated by applying the DDPG algorithm.

According to the above description, DDPG off-line training process is summarized as follow, and the flow chart is shown in Fig. 8,

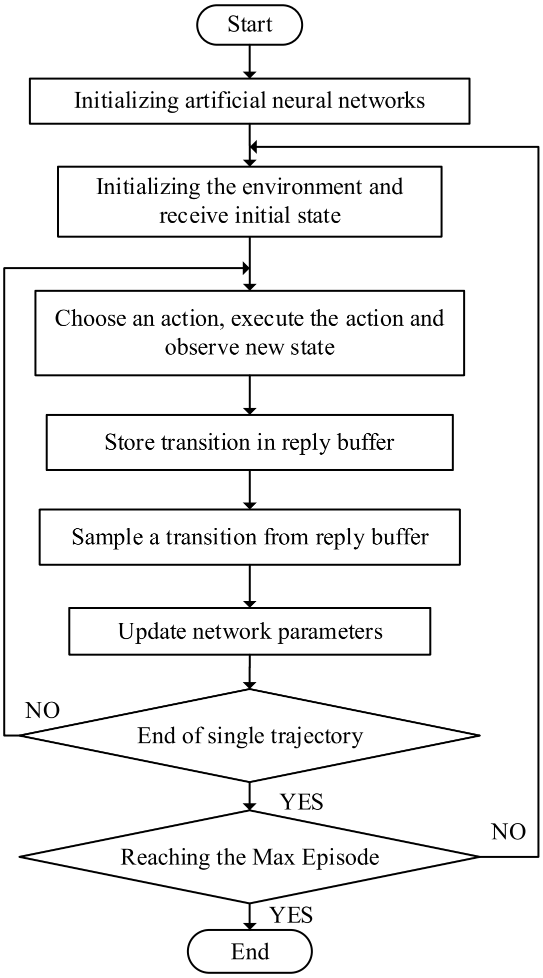

Start
Initializing artificial neural networks
Initializing the environment and receive initial state
Choose an action, execute the action and observe new state
Store transition in reply buffer
Sample a transition from reply buffer
Update network parameters
NO
End of single trajectory
YES
NO
Reaching the Max Episode
YES
End

FIGURE 8. The flow chart of off-line agent training process.

1. Randomly initialize critic network $Q(\boldsymbol{s}, \boldsymbol{a} \left|\theta^Q\right.)$ and actor network $\mu(\boldsymbol{s} \left|\theta^\mu\right.)$ with weight θ^Q and θ^μ.

2. Initializing target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$.

3. Initialize replay buffer D.

4. For episode=1 to Max Episode do

5. Initialize noise distribution $\boldsymbol{N}_t(0, \sigma^2)$.

6. Receive initial observation state $\boldsymbol{s}_1$.

7. For t=1 to Max Step do.

8. Chose an action $\boldsymbol{a}_t = \mu(\boldsymbol{s}_t|\theta^{\mu}) + \boldsymbol{n}_t$, where $\boldsymbol{n}_t$ is exploration noise.

9. The signals of $\boldsymbol{a}_t$ and computed-torque controller are integrated into the control system input $\boldsymbol{\tau}_t$

10. Execute action $\boldsymbol{a}_t$, observe new state $\boldsymbol{s}_{t+1}$, and calculate r_{t+1} by formula (19).

11. Store transition $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_{t+1}, \boldsymbol{s}_{t+1})$ in D.

12. Randomly sample a transition $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_{t+1}, \boldsymbol{s}_{t+1})$ in D.

13. Set $y_i = r_i + \gamma Q'(\boldsymbol{s}_{i+1}, \mu(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$.

14. Update policy critic network parameter by Equation (10).

15. Update policy actor network parameter by Equation (11).

16. Update target network parameters $\theta^{Q'}$ and $\theta^{\mu'}$ by Equation (12), Equation (13).

17. End of single trajectory, end of loop.

18. End of Max episode trajectory, end of loop.

IV. ANALYSIS OF TRAINING AND SIMULATION

2) AGENT TRAINING

During the agent training process, the initial state of the aircraft model is shown in Table 1, and the correlated training parameters are shown in Table 2. This training task assumes that the aircraft model undergoes a sinusoidal oscillation, e.g., $\theta = 6° \sin(t)$, the yaw angle and roll angle are expected to maintain a stable state at around 0°. The control period is set as 0.1s, and a single experiment lasts 15s.

TABLE 1. **The initial state of the aircraft mode.**

Symbol	State	Range
θ_0	Initial pitch angle	0°
ϕ_0	Initial roll angle	[-5,5]°
ψ_0	Initial yaw angle	[-5,5]°
$\dot{\theta}_0$	Initial pitch angular velocity	0°/s
$\dot{\phi}_0$	Initial roll angular velocity	[-5,5]°/s
$\dot{\psi}_0$	Initial yaw angular velocity	[-5,5]°/s

TABLE 2. **Training parameters.**

Symbol	Training parameters	value
a_Q	Critic network learning rate	0.001
a_μ	Actor network learning rate	0.0001
M	Experience pool size	10000
γ	Discount factor of reward	0.9
MaxStep	Maximum timestep of single experiment	150

The reward value curve reflects the performance variation of the agent to some extent. The agent's average reward value variation of every ten steps is shown in Fig. 9. At the beginning of the training process, the reward values are negative, which means the agent does not meet the requirement of the experiment; with the increasing of the step times, the value turns positive and continues to rise; finally, it stables near a fixed value. When applying the DDPG algorithm only, the agent reward value turns positive at about the seventieth time and finally converges at about the third hundredth time; while applying the DDPG+computed-torque algorithm, the agent reward value turns positive at about the tenth time, and for a long period after the fortieth time, the reward value is stable. From the analysis above, it is evident that computed-torque auxiliary controller speeds up the training process and improves the steadiness of the training.

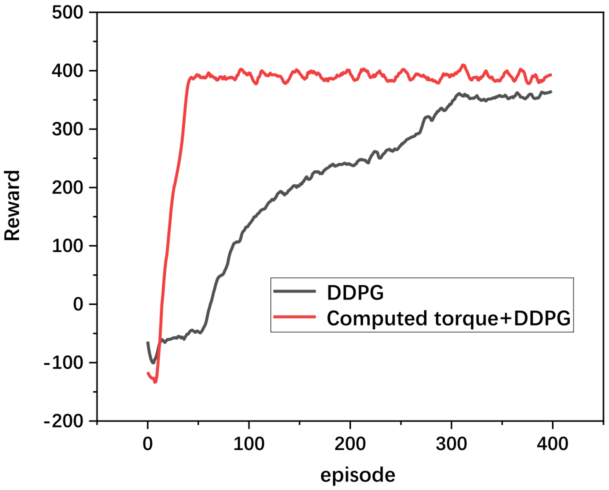


FIGURE 9. **Reward value curve.**

As shown in Figs. 10–12, after the agent training, the computed-torque+DDPG controller manages to track the

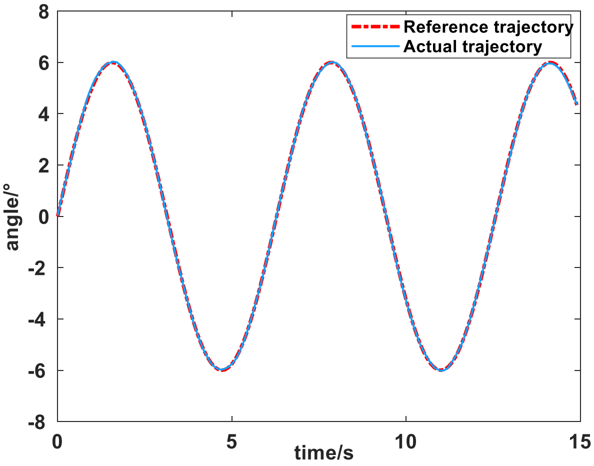


FIGURE 10. **Pitch angle trajectory tracking.**

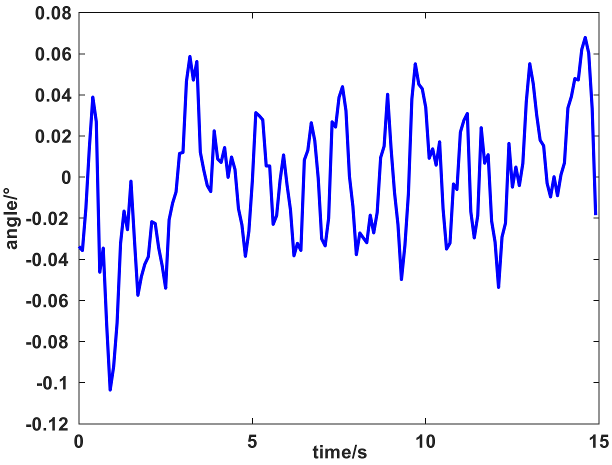


FIGURE 11. Tracking error of pitch angle trajectory.

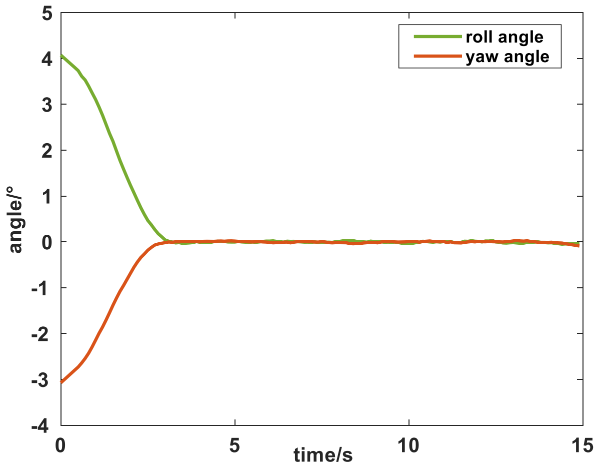


FIGURE 12. Curves of pitch angle and roll angle.

expected trajectory of pitch angle rapidly and accurately; the tracking error is within 0.1°. In roll and yaw direction, it can enter a stable state in 4s.

A. VARIABLE TASKS CONTROL

In the agent training process, the pitch angle tracking trajectory is $6^{\circ} \times \sin(t)$, but now instead, using the unlearned high angles of attack trajectories as flying tasks. According to the analysis CDPSS workspace, i.e., the movement range of aircraft model, several typical tasks are taken as examples:

Task 1: Oscillatory maneuver at high angles of attack in pitch direction with a central pitch angle 20°, an amplitude of 20°, and an angular frequency of 0.6,

$$\theta = 20^{\circ} + 20^{\circ} \times \sin(0.6t)$$

and the initial roll and yaw angles are limited to $-10^{\circ} - 10^{\circ}$.

As shown in Fig. 13, when the aircraft model moves with a single degree of freedom, the three different controllers manage to track the expected high angles of attack trajectory in 5s in the pitch motion, while computed-torque +DDPG outperforms the other two controllers. The results in Fig. 14 show the tracking error variation corresponding to the three controllers. In which the tracking error of the computed-torque+DDPG control algorithm is within 0.07°, and the maximum error for the single DDPG and computed-torque control algorithm are respectively 0.1° and 0.25°.

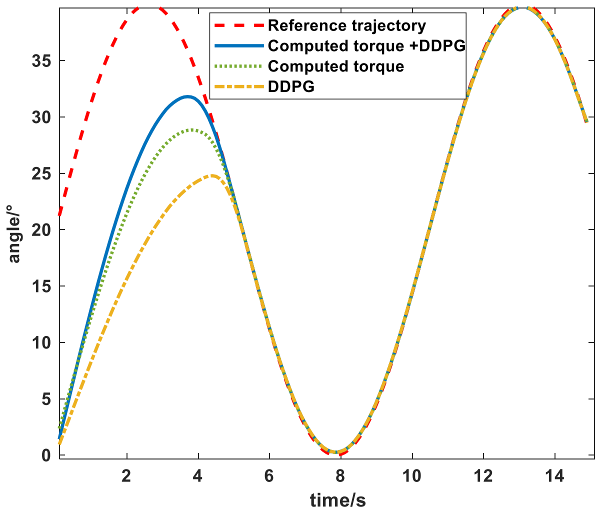


FIGURE 13. Pitch angle tracking history.

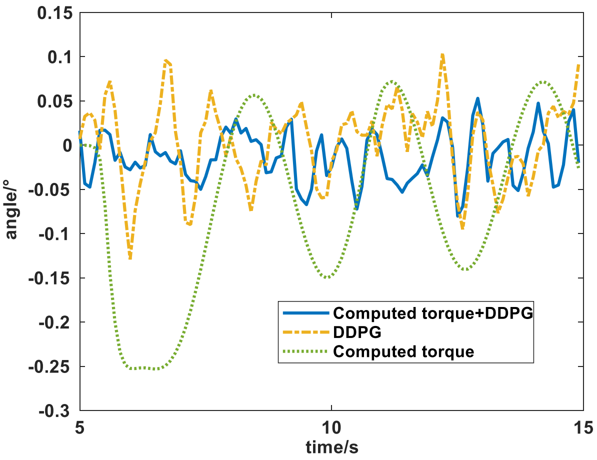


FIGURE 14. Pitch angle tracking error.

Similar results in the roll and yaw motion are shown in Fig. 15 and Fig. 16,, it can be seen that the pre-trained agent manages to maintain stability at around 0° within 5s, and the computed-torque+DDPG control algorithm achieves faster convergence than the other two.

Task 2: Three-DOF coupled oscillatory maneuver at high angles of attack in the pitch direction with a central pitch angle 20°, an amplitude of 30°, and an angular frequency of 0.5; and in the roll direction with a phase $\pi/6$, an amplitude of 5°, and an angular frequency of 0.6; in the yaw direction with an amplitude of 6° and an angular frequency of 0.6.

$$\theta = 20^{\circ} + 30^{\circ} \times \cos(0.5t),$$

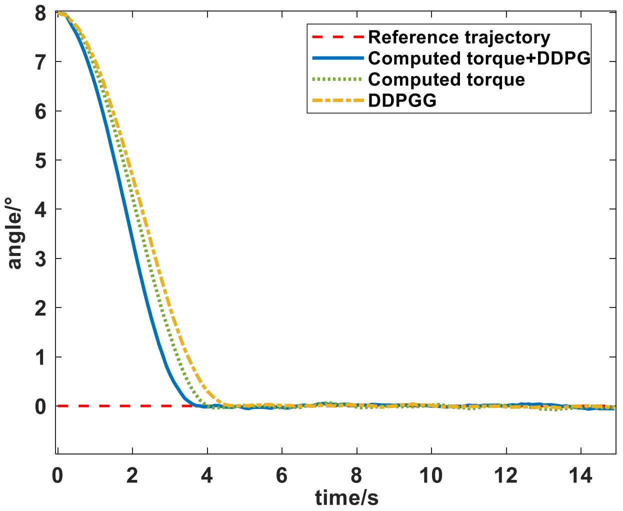


FIGURE 15. Curve of roll angle tracking.

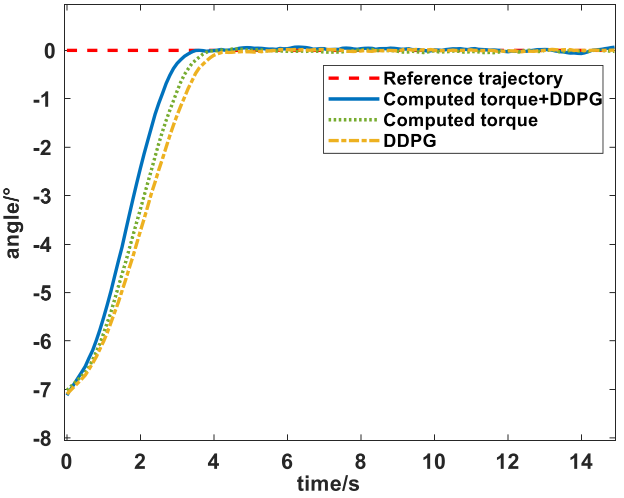


FIGURE 16. Curve of yaw angle tracking.

and the trajectory function of roll direction is,

$$\phi = 5^{\circ} \times \sin\left(0.6t + \pi/6\right),$$

and the trajectory function of yaw direction is,

$$\psi = 6^{\circ} \times \cos\left(0.6t\right)$$

Initial yaw angle is limited to −10° − 10°. The simulation results are shown below.

As shown in Fig. 17 and Fig. 19, when the aircraft model moves with two-DOF in the pitch and roll directions, three different controllers are used to make a comparison. As for pitch motion, computed-torque+DDPG controller manages to track the corresponding trajectory at around 3.2s, and 3.6s for computed-torque controller, 3.9s for DDPG controller; as for roll motion, computed-torque+DDPG controller spends 3.9s managing to track the trajectory, and 4.1s for computed-torque controller, 5.2s for DDPG controller. As can be seen from Fig. 18 and Fig. 20, as for pitch motion, the steady stable errors of the computed-torque+DDPG controller is within 0.08°, and the max tracking error is about 0.1° for the DDPG controller, and 0.23° for computed-torque controller; as for roll direction, the tracking error of computed-torque+DDPG controller is within 0.07°, and the max tracking error is about 0.1° for DDPG controller, and 0.15° for computed-torque controller.

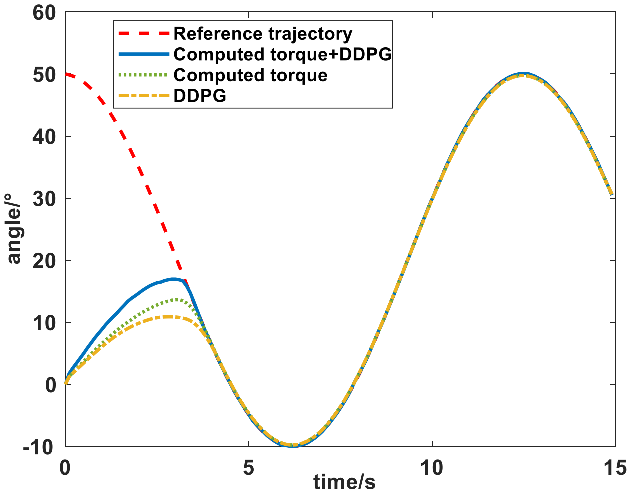


FIGURE 17. Pitch angle tracking history.

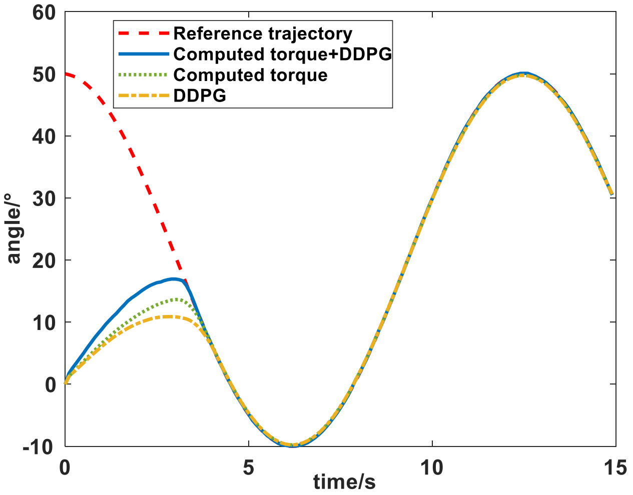


FIGURE 18. Pitch angle tracking error.

The yaw tracking history and error curves are shown in Fig. 21 and Fig. 22, the pre-trained agent maintains stable at around 0° within 5s, and the computed-torque+DDPG control algorithm is the first to reach a stable state.

On the basis of the two tasks' simulation results, whether in single or multiple degrees of freedom movement, the computed-torque+DDPG control algorithm is superior to computed-torque or DDPG alone; in other words, it exhibits a shorter tracking period required, faster adaption for a new task and smaller error.

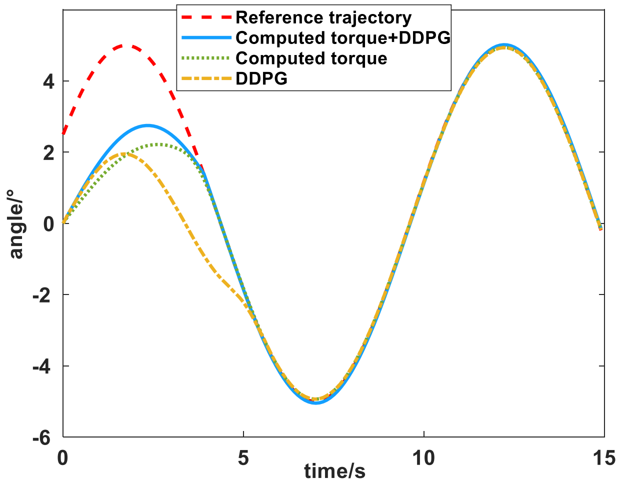


FIGURE 19. Roll angle tracking history.

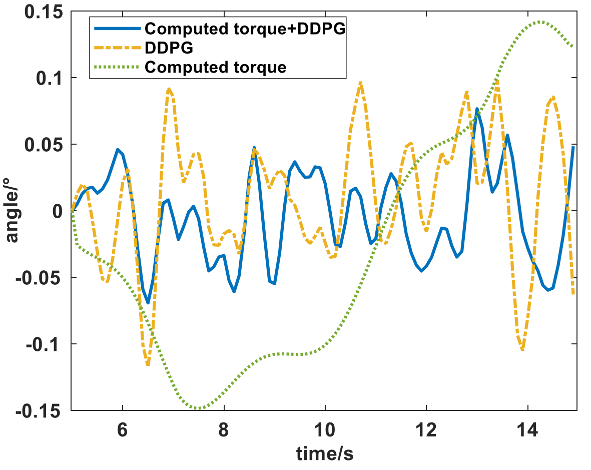


FIGURE 20. Roll angle tracking error.

1) CONTROLLER ROBUSTNESS DISCUSSION

In order to check the control robustness, simulations have been conducted for several values of parameters. The reference trajectory is set as a motion with an initial angle of 0° and an amplitude of 15° in the pitch direction. Case 1: model weight is increased by 10%. Case 2: model weight is reduced by 10%, and aerodynamic force is reduced by 10%.

As shown in Fig. 23, under the conditions of two-parameter changes, the designed controller combining computed-torque and DDPG can recover the stable tracking state in about 6s.

It indicates that the proposed composite controller can still maintain certain rapidity and stability in the control process despite the model uncertainties.

As to the CDPSS, when the suspended aircraft model maneuvers at high angles of attack, the aerodynamic forces show nonlinear and unsteady characteristics, and it is difficult to model it accurately. In such a situation, the performance of the controller robustness is analyzed.

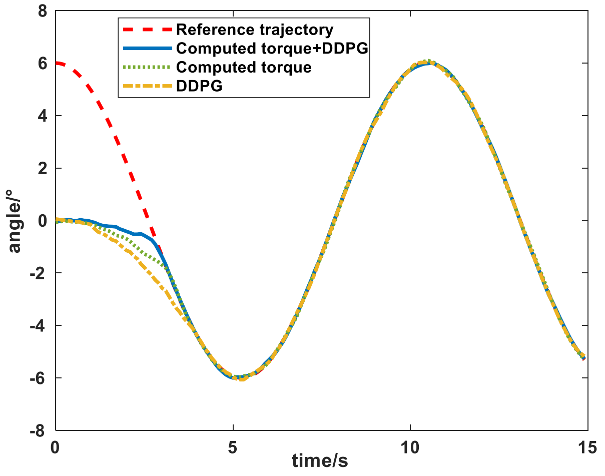


FIGURE 21. Yaw angle tracking history

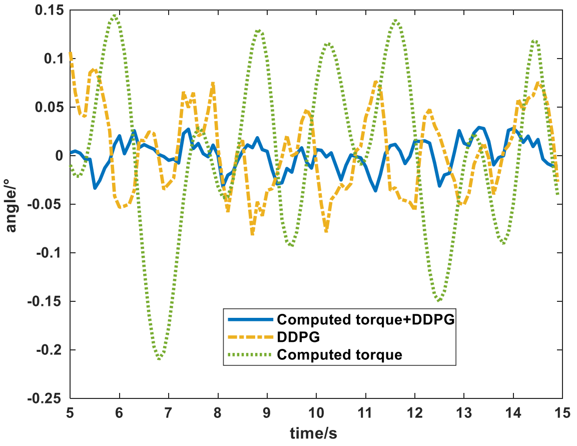


FIGURE 22. Yaw angle tracking error.

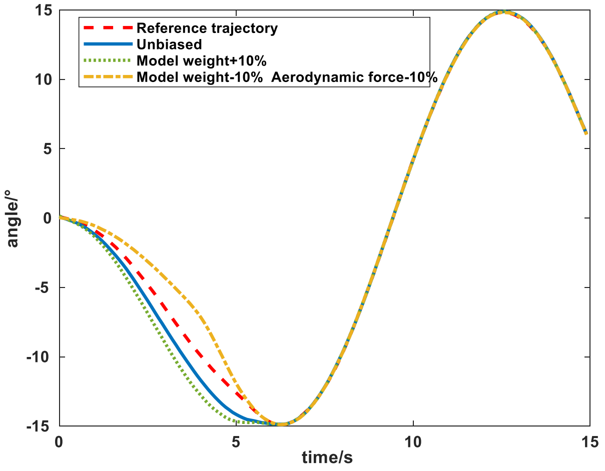


FIGURE 23. Control results under different model uncertainties.

When the controller is in a stable state, the aerodynamic disturbance with the function of $\sin(t)[4.5 \times 10^{-4}; -1.8 \times 10^{-3}; 0; 0; -1.2 \times 10^{-2}; 0]^{\mathrm{T}}$ is added during 1.5s-3.5s. The

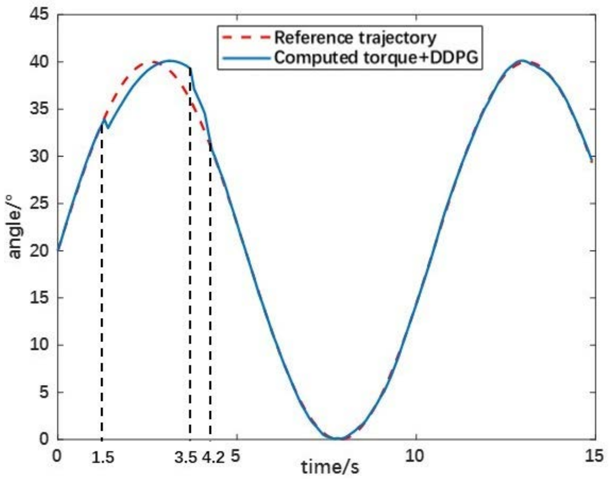


FIGURE 24. Pitch angle tracking history.

simulation results are given in Fig. 24. The computed-torque+DDPG controller restores stability after about a 0.7s oscillation. It shows that the designed controller can adapt to a certain degree of external interference.

V. CONCLUSION

This paper addresses control of fully constrained cable-driven parallel suspension system in high angle-of-attack maneuvers. Nonlinear and unsteady aerodynamic forces, and other kinematic and dynamic parameter uncertainties inevitably have negative impacts on the motion accuracy and performance. Thus, to cope with these uncertainties and bounded interferences, a reinforcement learning-based composite control algorithm is proposed to achieve suitable tracking performance. This novel composite controller consists of two major parts: 1) an intelligent DDPG controller, in which the MDP is designed based on the system dynamic equations; and 2) a computed-torque controller, which is based on a rigid model of the system, and is used to enhance the DDPG training speed and guarantee stability. After the off-line agent training, several tasks are simulated, and the results show the composite controller outperforms the traditional computed-torque controller and single DDPG in the view of convergence speed and precision. The impacts of different uncertainties and aerodynamic interferences were evaluated through the robustness analysis, which further validates the effectiveness and feasibility of the designed controller. However, the constructed training environment is based on the theoretical dynamic equations in this paper, and the proposed controller is only examined through simulation studies. Therefore, future works aim at better training performance of DDPG and experimental validation for this approach in the dynamic high angle-of-attack wind tunnel tests.

REFERENCES

[1] J. Albus, R. Bostelman, and N. Dagalakis, "The NIST robocrane," *J. Feild Robot.*, vol. 10, no. 5, pp. 709–724, 1993, doi: 10.1002/rob.4620100509.

[2] Y. Hirata and M. Sato, "3-dimensional interface device for virtual work space," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 2, Jul. 1992, pp. 889–896, doi: 10.1109/IROS.1992.594498.

[3] Y. Mao and S. K. Agrawal, "Design of a cable-driven arm exoskeleton (CAREX) for NEURAL rehabilitation," *IEEE Trans. Robot.*, vol. 28, no. 4, pp. 922–931, Aug. 2012, doi: 10.1109/TRO.2012.2189496.

[4] G. Meunier, B. Boulet, and M. Nahon, "Control of an overactuated cable-driven parallel mechanism for a radio telescope application," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 5, pp. 1043–1054, Sep. 2009, doi: 10.1109/TCST.2008.2004812.

[5] P. Lafourcade, M. Llibre, and C. Reboulet, "Design of a parallel wire-driven manipulator for wind tunnels," in *Proc. Workshop Fundam. Issues Future Res. Directions Parallel Mech. Manipulators*, Quebec City, QC, Canada, 2002, pp. 187–194, doi: 10.2514/1.J056782.

[6] T. Bruckmann, M. Hiller, and D. Schramm, "An active suspension system for simulation of ship maneuvers in wind tunnels," *New Trends in Mechanism Science*. Amsterdam, The Netherlands: Springer 2010, pp. 537–544, doi: 10.1007/978-90-481-9689-0_62.

[7] T. J. Lambert, B. Vukasinovic, and A. Glezer, "A six degrees of freedom dynamic wire-driven traverse," *Aerospace*, vol. 3, no. 2, p. 11, Oct. 2009, doi: 10.3390/aerospace3020011.

[8] T. J. Lambert, B. Vukasinovic, and A. Glezer, "Aerodynamic flow control of axisymmetric bluff body by coupled wake interactions," *AIAA J.*, vol. 56, no. 8, pp. 2992–3007, 2018, doi: 10.2514/1.J056782.

[9] Y. W. Xiao, Q. Lin, Y. Q. Zheng, and B. Liang, "Model aerodynamic tests with a wire-driven parallel suspension system in low-speed wind tunnel," *Chin. J. Aeronaut.*, vol. 23, pp. 393–400, Aug. 2010, doi: 10.1016/S1000-9361(09)60233-8.

[10] X. G. Wang and Q. Lin, "Progress in wire-driven parallel suspension technologies in wind tunnel tests," *Acta Aeronautica et Astronautica Sinica*, vol. 39, no. 10, pp. 6–25, 2018, doi: 10.7527/S1000-6893.2018.22064.

[11] J. Begey, L. Cuvillon, M. Lesellier, M. Gouttefarde, and J. Gangloff, "Dynamic control of parallel robots driven by flexible cables and actuated by position-controlled winches," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 286–293, Feb. 2019, doi: 10.1109/TRO.2018.2875415.

[12] A. Izadbakhsh, H. J. Asl, and T. Narikiyo, "Robust adaptive control of over-constrained actuated cable-driven parallel robots," in *Cable-Driven Parallel Robots*. Cham, Switzerland: Springer, 2019, pp. 209–220.

[13] J. C. Santos, A. Chemori, and M. Gouttefarde, "Model predictive control of large-dimension cable-driven parallel robots," in *Cable-Driven Parallel Robots*, T. Bruckmann and A. Pott, Eds. Cham, Switzerland: Springer, 2019, pp. 221–232.

[14] W. Shang, B. Zhang, B. Zhang, F. Zhang, and S. Cong, "Synchronization control in the cable space for cable-driven parallel robots," *IEEE Trans. Ind. Electron.*, vol. 66, no. 6, pp. 4544–4554, Jun. 2019, doi: 10.1109/TIE.2018.2864512.

[15] H. Jamshidifar, S. Khosravani, B. Fidan, and A. Khajepour, "Vibration decoupled modeling and robust control of redundant cable-driven parallel robots," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 2, pp. 690–701, Apr. 2018, doi: 10.1109/TMECH.2018.2793578.

[16] S. Hu, J. Zhu, and W. Yang, "Progressive identification of lateral nonlinear unsteady aerodynamics from wind tunnel test data," *Sci. China Inf. Sci.*, vol. 62, no. 10, pp. 1–3, Oct. 2019, doi: 10.1007/s11432-017-9458-9.

[17] P. C. Murphy, V. Klein, and N. T. Frink, "Nonlinear unsteady aerodynamic modeling using wind-tunnel and computational data," *J. Aircr.*, vol. 54, no. 2, pp. 659–683, Mar. 2017, doi: 10.2514/1.C033881.

[18] D. Silver, A. Huang, C. J. Maddison, A. Guez, and L. Sifre, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016, doi: 10.1038/nature16961.

[19] D. Lee, S. J. Lee, and S. C. Yim, "Reinforcement learning-based adaptive PID controller for DPS," *Ocean Eng.*, vol. 216, Nov. 2020, Art. no. 108053, doi: 10.1016/j.oceaneng.2020.108053.

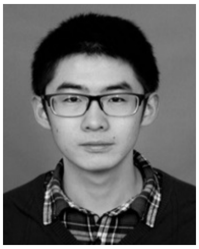
[20] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1513–1525, Oct. 2013, doi: 10.1109/TNNLS.2013.2276571.

[21] X. Huang, J. R. Liu, and C. H. Jia, "Deep deterministic policy gradient for UAV control," *Acta Aeronautica et Astronautica Sinica*, vol. 42, Feb. 2021, Art. no. 524688, doi: 10.7527/S1000-6893.2020.24688.

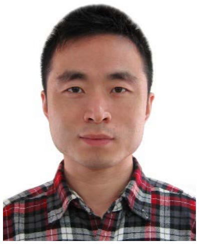
[22] X. Lin, Y. Yu, and C. Sun, "Supplementary reinforcement learning controller designed for quadrotor UAVs," *IEEE Access*, vol. 7, pp. 26422–26431, 2019, doi: 10.1109/ACCESS.2019.2901295.

[23] X. G. Wang, M. Shaoyu, and P. Miaojiao, ‘‘Influence of elastic deformation on pose precision of moving platform for wire-driven parallel robot,’’ *Chin. J. Comput. Mech.*, vol. 33, no. 3, pp. 306–312, 2016, doi: 10.7511/jslx201603005.

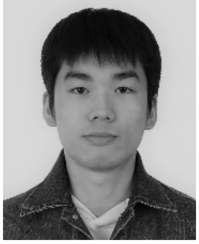
[24] P. Chen, ‘‘Comprehensive analysis of aerodynamic characteristics for standard dynamic model,’’ Chin. Aerodyn. Res. Inst., Shenyang, China, Internal Rep., 2015.



WEIPING WANG received the B.S. degree in materials science from Jilin University, China, in 2015. He is currently pursuing the M.S. degree with the School of Aerospace Engineering, Xiamen University, Xiamen, Fujian, China. His current research interests include nonlinear control for cable-driven parallel robots and reinforcement learning.



XIAOGUANG WANG received the Ph.D. degree in computer applied technologies from the Academy of Opto-Electroncs, Chinese Academy of Sciences, Beijing, China, in 2011. He is currently an Associate Professor with the School of Aerospace Engineering, Xiamen University, Xiamen, Fujian, China. He is the author of more than 20 peer-reviewed journal articles and five inventions. His current research interests include dynamics modeling and linear/nonlinear control, cable-driven parallel robots, and advanced wind tunnel tests technologies.



CHULUN SHEN received the B.S. degree in measurement and control technology and instrument from Nanchang University, Nanchang, China, in 2017. He is currently pursuing the M.S. degree with the School of Aerospace Engineering, Xiamen University, Xiamen, Fujian, China. His current research interests include nonlinear control for cable-driven parallel robots and reinforcement learning.

QI LIN received the B.S., M.S., and Ph.D. degrees in propulsion engineering from the Nanjing University of Aeronautics and Aerospace, Nanjing, China, in 1982, 1984, and 1988, respectively. She is currently a Professor with the School of Aerospace Engineering, Xiamen University, Xiamen, Fujian, China. She is the author of more than 50 peer-reviewed journal articles and more than ten inventions. Her current research interests include cable-driven parallel robots, advanced wind tunnel tests technologies, and flow control.